

Actas de las XXIX Jornadas sobre la Enseñanza Universitaria de la Informática

GRANADA, DEL 5 AL 7 DE JULIO DE 2023

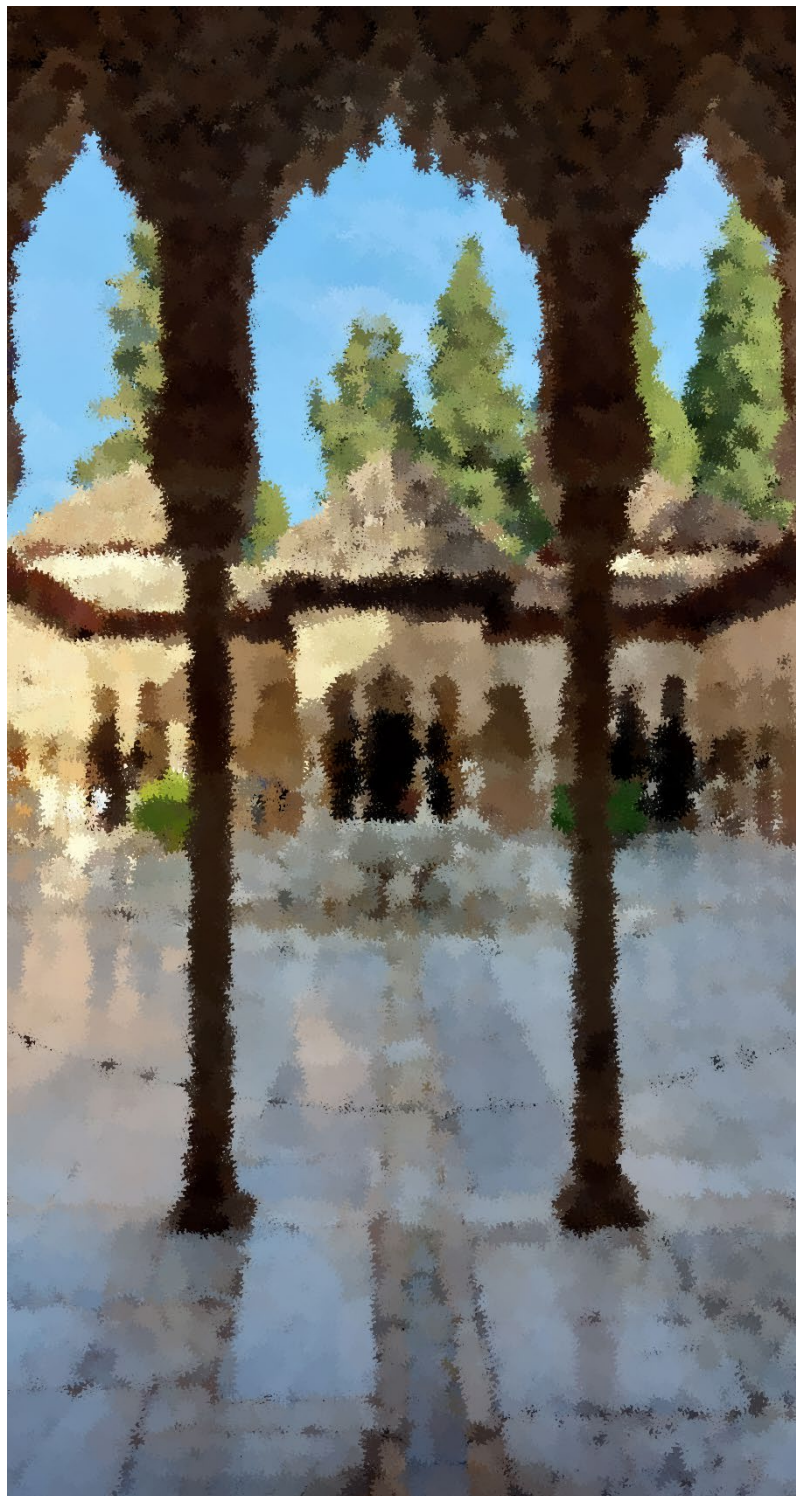
ISSN: 2531-0607

Editores:

José Antonio Cruz Lemus
Nuria Medina Medina
María José Rodríguez Fórtiz

Revisores de formato:

Agustín Cernuda del Río
Elena Jurado Málaga



Actas de las XXIX Jornadas sobre la Enseñanza Universitaria de la Informática

Editores:

José Antonio Cruz Lemus
Nuria Medina Medina
María José Rodríguez Fórtiz

Revisores de formato:

Agustín Cernuda del Río
Elena Jurado Málaga

Con apoyo del Plan Propio de Investigación y Transferencia de la Universidad de Granada

ISSN: 2531 – 0607



Este texto está sujeto a una licencia **Reconocimiento – NoComercial – CompartirIgual (by-nc-sa)**: No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Introducción a los conceptos del pensamiento computacional en educación infantil y primaria con programación tangible

Sergio Trilles, Aida Monfort-Muriach, Diego Lacomba, Carlos Granell

Departament de Llenguatges i Sistemes Informàtics

Universitat Jaume I

Castelló de la Plana

strilles, amuriach, al375896, carlos.granell@uji.es

Resumen

El programa Sucre tiene como objetivo el fomento del pensamiento computacional y la programación en cada etapa educativa. Tras una reestructuración del programa, la iniciativa Sucre4Kids se reorienta a la educación infantil y primaria (entre 5 y 10 años). Sucre4Kids presenta los conceptos básicos de la programación con la menor fricción posible, prescindiendo de dispositivos habituales de programación y utilizando en su lugar elementos tangibles para la programación. La estrategia propuesta de programación tangible se basa en la utilización de tarjetas con pictogramas, donde cada una de ellas lleva una etiqueta de comunicación de campo cercano (NFC) que codifica las instrucciones de programación. El alumnado dispone las tarjetas en secuencia, representando el orden lógico de instrucciones. La lectura de cada tarjeta produce un código distinto, según el ensamblaje de sensores y actuadores, de forma reactiva, es decir, la ejecución es inmediata tan pronto se lee una tarjeta. Sucre4Kids aprovecha el desarrollo realizado dentro del programa Sucre, adaptando un microcontrolador con un lector NFC y una pequeña pantalla informativa. El prototipo diseñado incluye varios modos de juego, con los que se trabaja diferentes conceptos del pensamiento computacional.

Abstract

The Sucre programme aims to promote computational thinking and programming at each educational stage. After restructuring the programme, the Sucre4Kids initiative is reoriented to early childhood and primary education (between 5 and 10 years). Sucre4Kids introduces the basics of programming with as little friction as possible, dispensing with usual programming devices and instead using tangible elements for programming. The proposed tangible programming strategy is based on the use of cards with pictograms and each card carries a near field communication (NFC) tag that

encodes the programming instructions. Students arrange the cards in sequence, representing the logical order of instructions. Depending on which sensors and actuators are wired, the reading of each card produces reactive code, that is, the execution is immediate as soon as a card is read. Sucre4Kids takes advantage of the development carried out within the Sucre programme, adapting a microcontroller with an NFC reader and a small display. The designed prototype includes several game modes in order to work with different computational thinking concepts.

Palabras clave

Pensamiento computacional, programación tangible, educación infantil y primaria, dispositivos electrónicos

1. Motivación

En pasadas ediciones de las Jornadas sobre la Enseñanza Universitaria de la Informática (JENUi) hemos descrito la evolución tecnológica de la iniciativa *Sucre4Stem* y su puesta en marcha en las aulas [2, 6]. *Sucre4Stem* es un kit para el ensamblado de sensores y actuadores complementado con una herramienta de programación visual basada en bloques [3] para incentivar la adquisición de la competencia del pensamiento computacional en alumnado de educación secundaria y de bachillerato. Se podría pensar que a esas edades los estudiantes ya han tenido un contacto relativamente prolongado con las tecnologías digitales y la programación. Lo primero es cierto, pero lo segundo no tanto. Muchos de los estudiantes que acuden a nuestros talleres e intervenciones nos cuentan que es su primer contacto con los fundamentos de la programación.

Esta situación nos hizo reflexionar hasta que punto podríamos adaptar el desarrollo de *Sucre4Stem* a edades más tempranas, remontándonos incluso hasta el último curso de infantil (5 años). Tras realizar una reestructuración del programa Sucre (www.

programasucres.com), la iniciativa *Sucres4Kids* surge para acercar conceptos de computación a los más pequeños, prescindiendo de los dispositivos tradicionales de programación como ordenadores, móviles, o tabletas, y utilizando únicamente elementos tangibles (sin pantalla) para la programación. Ante este reto, nuestra premisa fue la de reutilizar al máximo el desarrollo tecnológico alcanzado y adaptar la experiencia previa con *Sucres4Stem* [5, 4] al nuevo contexto de aprendizaje.

Desde el primer momento tuvimos claro que un único producto no podía cubrir las necesidades de aprendizaje de un rango tan amplio de edades. Un juguete para un niño de 5 años, por ejemplo, no interesa a uno de 10. Aunque el objetivo de la iniciativa *Sucres4Kids* es cubrir desde los 5 a los 10 años, la adaptación que presentamos aquí se dirige a un público de 5 a 7 años, es decir, desde el último curso de infantil hasta segundo de primaria. Este grupo objetivo comparte necesidades similares de aprendizaje y se diferencia notablemente de las del grupo de 8 a 10 años. Por lo tanto, en este artículo presentamos un prototipo experimental para que el grupo objetivo se familiarice y experimente conceptos fundamentales de “programación”, como son los condicionales, la secuencia o bucles.

2. *Sucres4Kids*

Salta a la vista que conceptos de programación no se pueden transmitir de la misma forma a alumnado de secundaria que de infantil y primaria. Lógicamente, introducir conceptos técnicos requiere de una comunicación eficaz y adaptada a la audiencia. En este sentido, el trabajo de Mercer-Mapstone y Kuchel [1] nos inspiró para adaptar *Sucres4Kids* al grupo objetivo. Los autores identifican una serie de competencias clave para divulgar apropiadamente conceptos científicos a un público no científico. Entre ellas, destacan la habilidad de comprender los conceptos previos sobre la materia que posee el público objetivo, diferenciar claramente los aspectos esenciales de los que no los son, emplear un lenguaje y modo de comunicación adecuado, y tener en cuenta el contexto social, cultural y físico en el que la actividad se lleva a cabo. Estas competencias son imprescindibles para divulgar el concepto de declaraciones condicionales a los pequeños. El trabajo diario y el entorno de un aula de infantil/primaria condiciona notablemente el formato de la intervención: sin pantallas, disposición en grupo en mesas compartidas, trabajo en fichas para el desarrollo de la lectoescritura para reconocer la representación de fonemas en letras y la síntesis de las letras en palabras, y el aprendizaje sensorial y tangible para fomentar el control progresivo de la movilidad fina en el uso de utensilios rutinarios como lápices y tijeras.

Dado el conocimiento previo del alumnado y las peculiaridades del contexto de actuación, el reto consistía en identificar las partes de *Sucres4Stem* que podrían adaptarse al desarrollo de *Sucres4Kids* y qué aspectos deberíamos diseñar desde cero. Ambas iniciativas comparten el maletín (*SucresKit*), compuesto por sensores, actuadores y un microcontrolador. Esta es la parte hardware, absolutamente necesaria para que el alumnado ensamble los sensores y actuadores al microcontrolador (*SucresCore*). Como se describe en la Sección 3, mantuvimos intactos muchos de los componentes del maletín pero añadimos cambios en el modo en que el alumnado interactúa con esos componentes (p.e., uso de pictogramas para emparejar sensores al pin correspondiente) y reducimos la complejidad del ensamblaje a un sensor y actuador a la vez. El cambio más significativo se aplica a la parte de programación. En lugar de utilizar la herramienta *SucresCode* para la programación visual como en *Sucres4Stem*, diseñamos para *Sucres4Kids* un conjunto de tarjetas NFC para la programación tangible. El trabajo con tarjetas (o fichas) resulta familiar para estas edades. Como se detalla en la Sección 4, cada tarjeta viene impresa con pictogramas y se asocia a los diferentes conceptos de la programación. El alumnado coloca la secuencia de cartas en orden sobre la mesa para especificar la lógica del programa. Una vez establecido el algoritmo, el alumnado pasa cada una de las cartas de forma secuencial por el lector del *SucresCore* para su lectura. Finalmente, utilizando la carta especial “Ejecutar”, se ejecuta el comportamiento programado.

3. *SucresKit*

La pieza central del maletín *SucresKit* es el *SucresCore* [5]. *SucresCore* permite conectarse a IP/Internet mediante Wi-Fi e incluye una batería para operar de forma autónoma. La mayor diferencia entre las versiones de *SucresCore* en *Sucres4Stem* y *Sucres4Kids* radica en la adaptación hardware necesaria para habilitar la programación tangible mediante tarjetas (Sección 4). Como *Sucres4Kids* utiliza tarjetas de comunicación de campo cercano NFC para definir las secuencias de programación, se requiere de un lector NFC integrado. Hemos añadido también un pequeño *display* en la parte superior del *SucresCore* para mostrar breves instrucciones a medida que se “pasa” cada tarjeta por el lector. Como resultado, hemos rediseñado la tapa superior de la carcasa del *SucresCore* para incorporar estos nuevos elementos (Figura 1), mientras que el resto de *SucresCore* es común en ambas iniciativas.

Otro cambio significativo es la simbología para identificar el tipo de conexión (digital/análogica) para sensores y actuadores. Como consideramos que esta distinción es irrelevante para estas edades [1], cada



Figura 1: *SucreCore* en *Sucres4Stem* (izquierda) y *Sucres4Kids* (derecha).

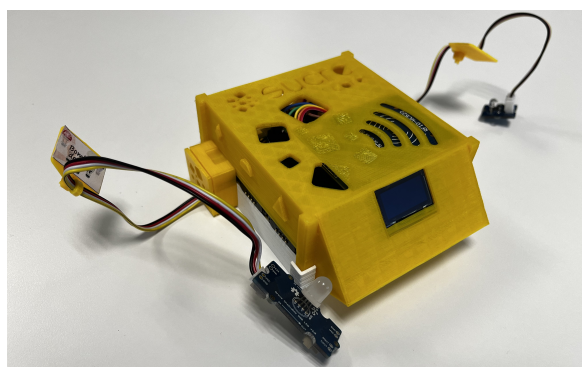


Figura 2: Formas geométricas para emparejar un sensor/actuador con *SucresCore* en *Sucres4Kids*.

puerto de conexión se identifica con una forma geométrica (p.e., rombo, semicírculo). La conexión de los sensores y actuadores se realiza de forma guiada a lo largo de la programación, indicándose en la pantalla la forma geométrica donde se debe conectar cada uno.

4. SucresCards: modos de juego

La programación en *Sucres4Kids* se basa íntegramente en tarjetas tangibles denominadas *SucresCards*. El aspecto visual de las *SucresCards* está diseñado para el grupo objetivo: uso de pictogramas, formato de gran tamaño, flexibles, modo puzzle –la combinación de dos tarjetas válidas forma la palabra “sucres”–, aparte de otras pistas visuales para emparejarlas como los marcos exteriores o las líneas discontinuas horizontales (en el modo básico). Cada tarjeta tiene integrada una etiqueta NFC en la parte trasera que codifica el tipo de elemento de programación según el pictograma que representa.

Actualmente se cuenta con tres modos de juegos diferentes. Cada uno de ellos se adapta a diferentes niveles educativos y conceptos del pensamiento computacional. En el primer modo, el básico, se trabaja el concepto de condicionales. Este va dirigido al nivel educativo de infantil y primer ciclo de primaria (1º y 2º curso). El modo básico ofrece dos tipos de car-



Figura 3: Ejemplo de programación del modo básico.

tas diferenciadas por el color del borde: las cartas de condiciones (*IF*) de color azul y las cartas de acciones (*THEN/ELSE*) de color verde. Lógicamente, cada una de estas tarjetas representa a un sensor (condición) o a un actuador (acción). Además, cada tarjeta se divide horizontalmente en dos condiciones diferentes (*IF*) para sensores, o en dos acciones (*THEN*) para actuadores. Es decir, si se cumple la condición superior de una carta de condición, se ejecutará la acción superior de la carta de acción emparejada. En cambio, si se cumple la condición inferior se ejecutará la acción inferior. Las condiciones de una carta son disjuntas, nunca se evaluarán a verdadero a la vez. La Figura 3 muestra una traza de una secuencia condición-acción, combinando varias tarjetas, e indicando la pareja activa después de cada paso. Las tarjetas se sitúan en secuencia (parte superior de la Figura 3) y su orden de izquierda a derecha representa la lógica del programa. El alumno pasa las dos primeras (#1 y #2) por el lector y la ejecución del nuevo comportamiento que según los pictogramas consiste en que si se pulsa el botón se emite un pitido y nada en caso contrario, se lleva a cabo cuando pasa la tarjeta “Ejecutar”. Al pasar la siguiente tarjeta (#3), el sensor de intensidad de luz se convierte en la condición (sensor) activa en vez del botón (#1), resultando en una nueva pareja activa de sensor-actuador.

El segundo modo va dirigido a segundo y tercer ciclo de educación primaria (3º a 6º curso). El funcionamiento de las tarjetas se asemeja a la programación tradicional por conjunto de instrucciones. Aunque el objetivo va dirigido también a las sentencias condicionales, se han definido diferentes tarjetas por cada una de las sentencias *IF*, *THEN* o *ELSE*, todas con un marco de color rojo. Al igual que en modo básico, las cartas de condiciones vienen con marco azul y las de acción con marco verde, asociadas a los sensores y actuadores respectivamente. Lógicamente, las cartas de condiciones se deben combinar tras la tarjeta *IF*. También se permite definir condiciones más complejas con operadores *AND* y *OR* (existe una *SucresCard* para cada uno de los operadores). Las cartas de acción se combinan tras las cartas *THEN* o *ELSE*. Finalmente este modo permite definir diferentes bloques de *IF* pudiéndose ejecutar en paralelo. La Figura 4 ilustra la gramática de uso de las cartas en modo avanzado y la Figura 5 muestra un ejemplo de programación que comprueba la intensidad de luz, y si esta es alta, el led multicolor

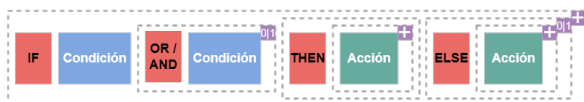


Figura 4: Representación visual de la gramática del modo avanzado.



Figura 5: Ejemplo de programación modo avanzado.



Figura 6: Ejemplo de programación del modo música.

se ilumina color verde o en rojo en caso contrario.

El tercer modo de programación incide sobre la música para trabajar los conceptos de secuencia y bucles. Se han definido un conjunto de cartas para simular las notas musicales y poder generar melodías con el uso del actuador zumbador. En total se han definido 48 *SucreCards* con las diferentes notas (Do, Re, Mi, Fa, Sol, La, Si y do) y tipos (negra, blanca y corchea) que permiten generar una melodía a medida que el *SucreCore* las lee y almacena de forma secuencial. También se han añadido tarjetas de bucle y de X número de repeticiones (siendo X de 2 a 10), para repetir conjuntos de notas que estén entre ellas (a modo de estribillo). La Figura 6 muestra un ejemplo de programación con el modo música.

A lo largo de los diferentes modos existen varias *SucreCards* comunes: a) “Ejecutar”, reproduce el nuevo comportamiento programado; b) “Borrar última carta”, borra la última tarjeta leída; c) “Borrar todo”, borra toda la programación guardada por el *SucreCore*; y d) “Borrado bloque/bucle”, elimina el último bloque condicional guardado en el modo avanzado, o el último bucle en el modo música. También se proporcionan tarjetas para indicar cada uno de los modos.

5. Conclusiones

En este trabajo hemos presentado la adaptación del programa *Sucre* al alumnado de infantil y primaria, para introducirles en los principios básicos de la programación de la mano de las declaraciones tangibles

de condicionales, secuencia y bucles. La iniciativa *Sucre4Kids* está siendo evaluada en las aulas a través de cursos de formación a profesorado de educación infantil y primaria. Con estas primeras experiencias obtendremos la retroalimentación necesaria para facilitar la aplicación de *Sucre4Kids* en las aulas. Estamos convencidos que iniciativas como *Sucre4Kids* son fundamentales para inculcar la competencia del pensamiento computacional y la curiosidad por la ciencia y la tecnología desde edades tempranas, para así fortalecer estas habilidades a lo largo de la educación secundaria. Nuestros planes de futuro incluyen ampliar los modos de juego donde se trabajen otros conceptos del pensamiento computacional, como variables o funciones.

Referencias

- [1] Lucy Mercer-Mapstone y Louise Kuchel. Core skills for effective science communication: A teaching resource for undergraduate science education. *International Journal of Science Education, Part B*, 7(2):181–201, 2017.
- [2] Sergio Trilles y Carlos Granell. SUCRE4Kids: El fomento del pensamiento computacional a través de la interacción social y tangible. En *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática*, volumen 3, pp. 303–310, 2018.
- [3] Sergio Trilles y Carlos Granell. Advancing pre-university students’ computational thinking skills through an educational project based on tangible elements and virtual block-based programming. *Computer Applications in Engineering Education*, 28(6):1490–1502, 2020.
- [4] Sergio Trilles, Aida Monfort-Muriach, Enrique Cueto-Rubio, Carlos Granell, y Pablo Juan. *Sucre4Stem: Internet of things in classrooms*. En *2022 XV Technologies Applied to Electronics Teaching Conference*, pp. 1–4. IEEE, 2022.
- [5] Sergio Trilles, Aida Monfort-Muriach, Águeda Gómez-Cambronero, y Carlos Granell. *Sucre4Stem: collaborative projects based on IoT devices for students in secondary and pre-university education*. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 17(2):150–159, 2022.
- [6] Sergio Trilles, David Tortosa, y Carlos Granell. La evolución del proyecto *Sucre4Kids* mediante el paradigma del Internet de las Cosas. En *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática*, volumen 5, pp. 53–60, 2020.