

La evolución del proyecto *Sucre4Kids* mediante el paradigma del Internet de las Cosas

Sergio Trilles, David Tortosa, Carlos Granell
Departament de Llenguatges i Sistemes Informàtics
Universitat Jaume I
Castelló de la Plana

strilles, dtortosa, carlos.granell@uji.es

Resumen

Esta comunicación presenta la evolución tecnológica del proyecto *Sucre4Kids*, que tiene como objetivo el fomento de las vocaciones científicas, la promoción del pensamiento computacional y la programación en estudiantes preuniversitarios. Tras cuatro años de experiencia contrastada en aulas, se ha redefinido todo el ecosistema del proyecto desde la perspectiva del Internet de las Cosas. En el plano tecnológico, *Sucre4Kids* propone dos herramientas bien diferenciadas: *SucreCore* y *SucreCode*. La primera adopta un nuevo diseño más compacto, encapsulando un microcontrolador más avanzado, con soporte para conectividad inalámbrica, y con la capacidad de crear redes en malla. *SucreCode*, la herramienta de programación visual por bloques, tiene una interfaz renovada y permite actualizaciones sin cables en un *SucreCore*. En el plano pedagógico, las nuevas características tecnológicas abren la puerta a nuevos tipos de dinámicas y proyectos colaborativos entre grupos de estudiantes. Este mayor grado de interactividad y comunicación, que representaba una limitación en la anterior versión de *Sucre4Kids*, augura un interés creciente de los estudiantes, tal como hemos observado en las primeras intervenciones en aula.

Abstract

This paper presents the technological evolution of the *Sucre4Kids* project, which aims at promoting scientific vocations, developing computational thinking and programming in pre-university students. After four years of proven classroom experience, the entire project ecosystem has been redesigned from the perspective of the Internet of Things. On the technological level, two distinct tools are proposed *SucreCore* and *SucreCode*. The first introduces a new, more compact design, encapsulating a more advanced microcontroller, and supports wireless connectivity with the ability to create mesh networks. The block-based visual programming tool, *SucreCode*, has a revamped interface and allows

wireless updates over the *SucreCore*. At the pedagogical level, new technological features undertake new dynamics and types of collaborative projects between groups of students. This greater degree of interactivity and communication, which was a limitation in the previous version of *Sucre4Kids*, augurs a growing interest of students, as we have observed in the first classroom interventions.

Palabras clave

Pensamiento computacional, promoción programación, edades preuniversitarias; Internet de las Cosas

1. Introducción

Podría darse la paradoja de que, frente a una sociedad cada vez más orientada a la tecnología [3], ésta tenga carencia de profesionales con las habilidades digitales y analíticas necesarias. Lamentablemente, ya se están dando los primeros indicios que apuntan hacia su cumplimiento. Para 2025 se estima un incremento del 13 % en la demanda de profesionales relacionados con Ciencia, Tecnología, Ingeniería y Matemáticas (CTIM). Contrariamente, los estudiantes de secundaria de hoy - que serán los profesionales de mañana - presentan destrezas desiguales en sus niveles de conocimiento en Ciencias y Matemática, ocupando España posiciones medias-bajas, según el reciente informe PISA [5]. La matrícula de nuevo ingreso en grados universitarios en la rama de Ingeniería y Arquitectura ha descendido un 15 % entre el curso académico 2010/11 y 2017/18 [8]. Especialmente sangrante es la matriculación de mujeres en estudios en CTIM. En el último anuario de indicadores universitarios disponible en [7], las mujeres representan únicamente el 12 % de las matriculaciones de primer curso (2016/17) en estudios de Informática en las universidades españolas, y el 25 % de nuevas matriculaciones en la rama de Ingeniería y Arquitectura [7]. Resulta necesario buscar incentivos

para la incorporación de capital intelectual en estudios conducentes a la obtención de grados en CTIM, así como incentivar vocaciones científicas entre los jóvenes, especialmente entre las mujeres.

Si se observa a la adolescencia de hoy, no se puede negar que los dispositivos móviles y la tecnología son una realidad cotidiana entre las nuevas generaciones. El futuro augura un protagonismo aún mayor a las tecnologías y a la programación como lengua franca [2], tanto en el trabajo como en la vida diaria. El pensamiento computacional incide en “pensar cómo lo haría un científico computacional y/o informático”. Sin embargo, no se trata de competencias exclusivamente relacionadas con la disciplina informática, sino que son transferibles al mundo científico y profesional en general (ingenieros, matemáticos, economistas, arquitectos, periodistas, etc.) y, por tanto, son esenciales para preparar a las nuevas generaciones para hacer frente a los retos inciertos de las profesiones del futuro.

Algunos países ya han adoptado políticas y medidas para la integración de las competencias computacionales en los currículos educativos [16]. Reino Unido propuso en 2014 la computación como parte del currículo nacional en las escuelas de primaria y secundaria [6]. La administración Obama puso en marcha en 2016 un programa educativo [13] para llevar la computación a profesores y escolares K-12. Los programas educativos en estos países y otros coinciden en la necesidad de introducir conceptos de computación y programación, junto con competencias digitales de las Tecnologías de la Información y Comunicación (TICs), para el fomento de la creatividad, el pensamiento lógico y la resolución de problemas complejos.

En contraposición al avance realizado en otros países desarrollados, el currículo nacional español [9] sigue mayoritariamente centrado en el uso de las TICs y únicamente considera la computación como un mero instrumento y no como una “disciplina” en sí. Esta percepción ha sido corroborada por informes recientes realizados por la Fundación Telefónica [12] y por la FECYT/Google España¹. Por ejemplo, no se instruye al alumnado en cómo están hechos las *apps* que utilizan a diario (programación, diseño, razonamiento espacial, privacidad, etc.). Esta situación revela que aún queda mucho por recorrer para que las aulas españolas tengan un currículo que integre las competencias del pensamiento computacional y la programación, capaz de fomentar la creatividad y el pensamiento computacional y analítico, y sea a su vez consecuente con la realidad digital en la que viven los escolares.

En respuesta a esta problemática, el proyecto *Sucre4Kids* (en sus siglas en inglés *Sense yoUr Context and REact for Kids*) quiere contribuir a paliar esta ne-

cesidad educativa, mostrando al alumnado un abanico de posibilidades para el desarrollo y estímulo de las competencias del pensamiento computacional y del uso creativo de la tecnología. El proyecto arrancó cuatro años atrás [15] con una versión inicial compuesta de un maletín con diferentes sensores/actuadores y un microcontrolador *Arduino UNO*, que los estudiantes programaban de forma visual (bloques) mediante una herramienta web para programar el ensamblaje de sensores y actuadores. Durante estos años, hemos aplicado el proyecto en diversas intervenciones, como por ejemplo en las últimas ediciones de un programa de fomento de los estudios en Informática de la Universitat Jaume I (*Práctica a l'UJI*), visitas a centros educativos y ferias, alcanzando en total más de 600 estudiantes de forma directa y más de 20.000 de forma indirecta.

Tras esta primera etapa, el proyecto *Sucre4Kids* ha sido replanteado y se ha redefinido todo su ecosistema, con el objetivo de adoptar y aprovechar los avances del paradigma del Internet de las Cosas (IoT, en sus siglas en inglés) y para mejorar ciertas limitaciones detectadas durante la primera versión. Esta evolución no responde únicamente a la incorporación de nueva tecnología, sino que se sustenta también en la ampliación del tipo de proyectos pedagógicos. Como veremos en las siguientes secciones, la nueva versión de *Sucre4Kids* propone dos herramientas bien diferenciadas: *SucreCore* y *SucreCode*. Estas nuevas herramientas, basadas en IoT, añaden nuevas características como la interactividad y conexión sin cables, que abren la puerta a nuevos tipos de proyectos, como la interacción entre varios grupos de estudiantes. En este trabajo se detalla de forma técnica la evolución de *Sucre4Kids* y su primera evaluación en el programa *Práctica a l'UJI*.

2. Estado del arte

En el mercado y en el ámbito académico existen diferentes iniciativas similares a *Sucre4Kids*, con el objetivo de promover el pensamiento computacional entre los jóvenes. No todas las propuestas ofrecen una parte tangible de montaje de componentes y una parte de programación, de hecho, existen muchas más iniciativas sin la parte tangible debido a su mayor coste y difícil distribución. Junnan Yu y otros [17] proponen cinco categorías para clasificar este tipo de iniciativas: Kits físicos con o sin electrónica (2), Kits virtuales (1) y Kits híbridos con programación virtual o tangible por bloques (2). El proyecto *Sucre4Kids* se enmarca en la categoría de kits híbridos con programación virtual por bloques, pero añadiendo la variación de elementos electrónicos. A continuación, se analizan diferentes proyectos dentro de esta misma categoría y que además también se caracterizan por tener elementos electrónicos. Estos serán comparados con nuestra propuesta.

¹Educación de las Ciencias de la Computación en España - FECYT. <https://bit.ly/2UFzcsQ>. Acceso 01-05-2020

Talkoo [4] está basado en *Arduino*. Los módulos de sensores y actuadores son fácilmente conectables y no requieren soldadura ni conocimientos previos en electrónica. Los componentes tienen la capacidad de “responder” a la herramienta visual y no es necesario cargar el programa. *Talkoo* en todo momento está ejecutando los módulos y muestra en tiempo real los valores de las entradas/salidas de los sensores. *Grove Zero*², basado en los conectores *Grove* originales, es un kit que propone módulos imantados para facilitar la conexión entre ellos. Siguiendo una nomenclatura por colores, el kit *Grove Zero* incluye diferentes sensores y actuadores. Utiliza su propio Entorno Integrado de Desarrollo (IDE, en sus siglas en inglés) web basada en *Microsoft MakeCode*. *Micro:bit*³ es un iniciativa de la *BBC* para el fomento de la programación en edades juveniles. *Micro:bit* se compone de una pequeña placa de desarrollo de hardware libre, al igual que *Arduino*, que permite ejecutar código en ella. Aunque tiene incorporados diferentes actuadores y sensores, como una matriz 5x5, acelerómetro o *Bluetooth*, no incluye sensores externos, ni tampoco ofrece conectores que faciliten conectar con ellos. El microcontrolador puede ser programado mediante un lenguaje de bloques, *Java* o *Micropython* (un lenguaje similar a *Python*). *Chibi Chip*⁴ es otro microcontrolador donde los pequeños pueden construir circuitos de papel interactivos y programar LEDs. También permite la codificación mediante bloques, o escribiendo líneas de código *Arduino* usando un editor. *Circuit Playground Express*⁵, similar a *Micro:bit*, es una placa que incorpora sensores (temperatura, luz o acelerómetro), botones, audio y diferentes LEDs. Tampoco incorpora conectores para permitir la conexión de otro tipo de sensores. Se basa también en *Microsoft MakeCode* para la programación por bloques, pero permite programación tradicional mediante la IDE de *Arduino*.

La Tabla 1 compara los trabajos anteriores con *Sucre4Kids*, atendiendo a las siguientes características.

- Hardware libre, si se puede utilizar, modificar y distribuir libremente. Escala: *Sí/No*.
- Basado en proyectos, si la iniciativa propone proyectos. Escala: *Sí/No*.
- Conectores *plug&play*, si la iniciativa ofrece conectores que faciliten la conexión. Escala: *Sí/No*.
- Lenguaje de programación, indica el lenguaje de programación que soporta. Escala: *Python, Arduino, Java* o *Javascript*, entre otros.

²Blocks / Javascript editor - Seeed Grove Zero. <https://bit.ly/3b2JKXt>. Acceso 01-05-2020

³Micro:bit Educational Foundation - Micro:bit. <https://bit.ly/2MnaQid>. Acceso 01-05-2020

⁴Chibitronics webpage - Chibitronics. <https://chibitronics.com>. Acceso 01-05-2020

⁵Circuit Playground Express webpage - Circuit Playground Express. <https://bit.ly/3c2FDfj>. Acceso 01-05-2020

- Organización por grupos, si la iniciativa posibilita el uso de forma grupal. Escala: *Sí/No*.

Según la Tabla 1, todas las iniciativas analizadas se clasifican como de hardware abierto, menos *Talkoo* y *Grove zero*, que no lo especifican. *Sucre4Kids*, *Grove Zero*, *Micro:bit* y *Chibi Chip* ofrecen, junto con el kit, guías y documentación para realizar proyectos. Únicamente la nueva versión de *Sucre4Kids* ofrece conectividad directa a Internet para definir variables y funciones en línea y soporta actualización Over-The-Air (OTA). Por lo que se refiere a la disponibilidad de conectores, *Talkoo*, *Grove Zero* y *Chibi Chip*, junto a nuestra propuesta (ver Sección 6), son las únicas que facilitan el montaje mediante conectores. Además de soportar programación visual, todas las soluciones ofrecen la posibilidad de programar de forma tradicional. El lenguaje más utilizado es *Arduino*, el cual es soportado por todas las plataformas analizadas, excepto *micro:bit* que ofrece *Java* y *MicroPython*. Finalmente, excepto *Sucre4Kids*, ninguna solución analizada tiene en cuenta poder ser utilizada de forma grupal.

3. Inicios de *Sucre4Kids*

Para identificar los nuevos cambios y mejoras, en esta sección describimos el proyecto *Sucre4Kids* en su versión inicial [11]. El proyecto desde su inicio tiene como objetivo el fomento de las vocaciones científicas y la promoción del pensamiento computacional en alumnado de secundaria, ciclos formativos y bachillerato. La evolución del proyecto sigue fiel a este objetivo. Desde el punto de vista didáctico, tanto en la versión inicial como su evolución, *Sucre4Kids* promueve dos conceptos para alimentar la curiosidad por la programación. El primero de ellos es familiarizar al alumnado con la tecnología relacionada con dispositivos electrónicos y con el movimiento *maker* [1]. Y el segundo, se enmarca en las competencias y habilidades propias del pensamiento computacional [14]. Con *Sucre4Kids*, intentamos que los estudiantes experimenten, que “aprendan haciendo” [10], fomentando que las sesiones sean abiertas, participativas y basadas en el trabajo colectivo.

Sucre4Kids combina el uso de dispositivos inteligentes y componentes electrónicos con la programación visual por bloques para la realización de proyectos reales. Para ello, se crearon dos productos bien diferenciados, un maletín de componentes y una herramienta de programación por bloques. Este maletín contenía componentes electrónicos que recreaban las partes funcionales de cualquier ordenador: un núcleo y dispositivos de entrada y salida (E/S). El núcleo (microcontrolador y buses) está formado por un *Arduino UNO* y un *shield*, llamado *Grove*, que facilita las conexiones de

Iniciativa	Hardware libre	Conexión Internet	Actualizaciones <i>Over-The-Air</i>	Basado en proyectos	Conectores <i>plug&play</i>	Lenguaje de programa	Organización por grupos
<i>Talkoo</i>	No especific.	No	No	No	Sí	<i>Arduino</i>	No
<i>Grove Zero</i>	No especific.	No	No	Sí	Sí	<i>Arduino</i>	No
<i>Micro:bit</i>	Sí	No	No	Sí	No	<i>Java y MicroPython</i>	No
<i>Chibi Chip</i>	Sí	No	No	Sí	Sí	<i>Arduino</i>	No
<i>Circuit Playground Express</i>	Sí	No	No	No	No	<i>Arduino</i>	No
<i>Sucre4Kids</i>	Sí	Sí (variales/funciones en línea)	Sí	Sí	Sí	<i>Arduino</i>	Sí

Cuadro 1: Comparativa entre las diferentes iniciativas analizadas. *Sucre4Kids* en última fila.

dispositivos de E/S al núcleo. Los sensores actúan de dispositivos de entrada de datos y los actuadores de salida. El maletín contaba con 4 sensores (sensor humedad del suelo y del aire, sonido, luz y proximidad) y 2 actuadores (led multicolor y barra de leds). El maletín quedaba completado con 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.

Por otro lado, la herramienta web de programación visual basada en *Blockly*⁶ permitía la programación visual por bloques del *Arduino*. Se extendieron los bloques ya disponibles de la librería *Blockduino*⁷, creando un bloque nuevo para cada uno de los sensores y actuadores incluidos en el maletín. Para la programación del *Arduino*, la herramienta web debía instalarse en el mismo ordenador que estaba instalado la IDE de *Arduino*, con todas las librerías adicionales para cada uno de los sensores/actuadores. La comunicación entre *Arduino* y ordenador se realiza mediante un cable USB y se establece una comunicación mediante un puerto COM. Tras varios años de experiencia, podemos afirmar que dicha configuración de comunicación resultó bastante problemática en general, y no solo por los fallos de comunicación entre el ordenador y el *Arduino*, sino porque perjudica la dinámica de la intervención.

Combinando los componentes electrónicos incluidos en el maletín, los estudiantes eran capaces de afrontar los retos de cada uno de los proyectos planteados: 1) proyecto “Planta inteligente”: mediante un sensor de humedad del suelo para medir la humedad de la tierra y una barra de LEDs para indicar el nivel; 2) proyecto “Parking inteligente”: mediante un sensor de proximidad, se comprueba si hay un objeto (p. ej. coche miniatura) en un sitio (estacionamiento); 3) proyecto “Luz inteligente”: mediante un sensor lumínico y activando un LED cuanto menor es la luz ambiental.

Durante los cuatros años de vida, se han realizado diferentes actividades que nos han permitido afinar tan-



Figura 1: *SucreCore* impreso con tecnología 3D para los microcontroladores *Particle Xenon* y *Boron*.

to la metodología y el público objetivo, como el proyecto en sí. Se ha participado en las ediciones 2016, 2017, 2018 y 2019 del programa “Practica a l’UJI”, con cerca de 400 estudiantes de secundaria por cada edición. También, destaca la presencia en diferentes ferias y eventos de difusión de la ciencia entre estudiantes de primaria y secundaria, como es la *FirUJICiencia* (ediciones 2017, 2018 y 2019), que en 2019 ha reunido a más de 3.000 alumnos/as de entre 6 y 18 años, en el *Sagunto Gaming Fest* o durante el *FIMP 2019* (Feria de videojuegos y tecnología en Gijón).

4. Evolución tecnológica

Con la experiencia acumulada en el aula, decidimos dar un paso adelante y adoptar un mayor potencial tecnológico, que aportará mayores funcionalidades y facilidades para mitigar limitaciones y barreras detectadas. La nueva versión de *Sucre4Kids* viene con un nuevo desarrollo de encapsulamiento propio, llamado *SucreCore*, compuesto por un nuevo microcontrolador (*Particle Xenon/Boron*) con una amplia variedad de posibilidades provenientes del paradigma IoT (Figura 1). En base a estas nuevas funcionalidades, se ha desarrollado enteramente la herramienta de programación visual, pasándose a llamar *SucreCode*.

⁶Google Developers - Blockly, <https://developers.google.com/blockly>, Acceso 01-05-2020

⁷A web-based visual programming editor for arduino - Blockly-Duino. <https://bit.ly/3fjzuxp>. Acceso 01-05-2020



Figura 2: Versión renovada de maletín de *Sucre4Kids*.

4.1. *SucreCore*

El nuevo maletín en la Figura 2 reúne un mayor número de sensores y actuadores. Concretamente, contiene: a) núcleo: 1x *SucreCore* (con batería); b) actuadores: 1x led, 1x zumbador, 1x pantalla de segmentos; c) sensores simples: 1x botón, 1x sensor rotatorio, 1x sensor de proximidad, 1x sensor de luz y 1x sensor de ruido; d) sensores avanzados: 1x sensor de temperatura y humedad y 1x sensor humedad del suelo; y e) varios: 1x USB y fichas de documentación. El precio total del maletín se encuentra alrededor de los 85 €.

La pieza central del maletín es el *SucreCore*, por lo que merece una descripción detallada por las nuevas funcionalidades que ofrece. Los microcontroladores que esconden la carcasa impresa mediante tecnología 3D de *SucreCore* pueden ser de dos versiones, *Particle Boron*⁸ y *Particle Xenon*⁹. Ambos microcontroladores tienen características similares, en cuanto a prestaciones de rendimiento a nivel de CPU (ARM Cortex-M4F de 32 bits a 64 MHz) y memoria (1 MB de memoria flash, 256 KB de RAM), y ofrecen un *pin-out* de 20 señales mixtas *GPIO* (6 x analógico, 8 x *PWM*), *UART*, *I2C* y *SPI*. La diferencia recae en la conectividad que ofrece cada uno de ellos para dotarse del protocolo IP/Internet. El primero de ellos, *Particle Boron*, ofrece conectividad 2G/3G garantizando su aplicabilidad en cualquier ambiente sin dependencia de la existencia de una red Wi-Fi. También añade conexión *Bluetooth LE* (BLE). Por otra parte, la *Particle Xenon* únicamente ofrece conectividad BLE, por lo que para dotarse de una conexión a Internet, es necesario conectarse vía *Bluetooth* a otra microcontrolador con acceso a Internet. Ambos microcontroladores ofrecen conecti-

⁸Particle Boron - Particle. <https://bit.ly/2yuxLov>. Acceso 01-05-2020

⁹Particle Xenon - Particle. <https://bit.ly/2yuxLov>. Acceso 01-05-2020

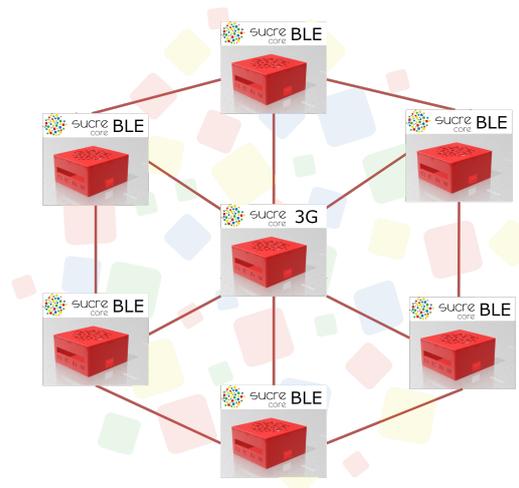


Figura 3: Configuración en malla utilizando las dos versiones de *SucreCore* con conectividad BLE y 3G.

vidad *Near field communication* (NFC) y pueden funcionar con batería al contar con un conector *JSP*.

El microcontrolador *Particle Boron* puede actuar como puerta de enlace para conectar un grupo de *Particles Xenon*. Esta característica facilita la creación de mallas (Figura 3) entre estos microcontrolador y es una solución idónea para el nuevo enfoque didáctico de la nueva versión de *Sucre4Kids*. Cada grupo de estudiantes de una clase utiliza un *SucreCore* con la versión de *Particle Xenon* y el profesorado utiliza una versión con el microcontrolador *Particle Boron*. De esta forma, el *SucreCore* del profesor ofrece conectividad a Internet a todas las unidades de los grupos de estudiantes. Se podría también utilizar la versión con conectividad 2G/3G para los grupos de estudiantes, pero se encarecía por dos motivos. El primero es el aumento de precio del microcontrolador, pasando de los 20 Euros del *Particle Xenon* a 60 Euros de la versión *Boron*. El segundo implica que cada una de las soluciones *SucreCore* con microcontrolador *Particle Boron* debe dotarse de una tarjeta SIM con datos, lo que encarecería el mantenimiento del ecosistema. Como apunte adicional, existe un microcontrolador análogo a *Particle Boron*, pero con conectividad Wi-Fi (llamado *Particle Argon*¹⁰). No se ha optado por esta solución para que el ecosistema gane en versatilidad y no dependa de puntos de acceso predefinidos, ya que en centros escolares suelen existir restricciones de acceso (p. ej. *firewall*).

El dotar a la solución *SucreCore* de conectividad a Internet y poder formar mallas entre ellos, amplía sustancialmente la cantidad de posibilidades. Estas deberán ser explotadas por la herramienta web de programación y por las nuevas dinámicas de clase. Entre ellas

¹⁰Particle Argon - Particle. <https://bit.ly/3b1S0fs>. Acceso 01-05-2020

podemos destacar las siguientes: 1) posibilidad de lanzar actualizaciones *Over-The-Air* (OTA) con el programa (*sketch*) sin tener que conectar el *SucreCore* al ordenador mediante cable; 2) posibilidad de crear variables en línea, es decir, los valores que se guarden en dichas variables estarán accesibles mediante una interfaz de programación de aplicaciones (API, en sus siglas en inglés) de forma abierta; 3) posibilidad de compartir variables entre los *SucreCore* que formen la malla; o la 4) posibilidad de creación de operaciones/funciones ejecutables de forma remota.

Al igual que en la versión previa de *Sucre4Kids*, para facilitar la conexión de los sensores y actuadores sobre el *pin-out* y garantizar la compatibilidad de sensores y actuadores, se utiliza un *shield* con conectores *Grove*. En aras de mejorar la usabilidad por parte de los estudiantes, y no depender de ninguna conexión externa, ya sea para la alimentación de electricidad o conectividad (ya se ha detallado el uso de 3G), se ha dotado a cada *SucreCode* de una batería recargable accesible mediante un interruptor de encendido/apagado.

Todos los componentes se han encapsulado mediante una caja de diseño propio e impresa con tecnología 3D (PLA). El resultado final es mucho más compacto y mejora la facilidad de uso por parte del alumnado (Figura 1). Unas aperturas laterales dejan visibles los conectores del *shield* para poder conectar los sensores/actuadores. Cada conector ha sido también etiquetado en la propia caja impresa. En la parte superior de la caja se ha optado por un diseño de rejilla (utilizando el logotipo del proyecto), para poder visualizar el interior de la caja y poder obtener retroalimentación del LED incrustado en el microcontrolador, útil para los procesos de actualización y conexión a Internet.

4.2. *SucreCode*

La herramienta web también ha sido rediseñada desde cero para alinearse a las nuevas funcionalidades de *SucreCore*. La nueva aplicación web, que mantiene todas las funcionalidades de la previa, ha sido desarrollada en *JavaScript* utilizando el *framework Angular*. Aunque continúa utilizando la misma librería para la gestión de los bloques de programación (*Blockly*), el proyecto ha sido rediseñado para añadir nuevas funcionalidades. Entre ellas destacan: a) lanzar actualizaciones OTA, b) generación de bloques de variables/funciones en línea y en malla, la c) gestión de cuentas de usuarios, d) posibilidad de poder guardar proyectos y guardado automático y e) sistema de administración de los dispositivos *SucreCore*.

Los microcontroladores *Particle* ofrecen un *cloud* propio, con el que es posible interactuar mediante una API. Esta API requiere que cada uno de los controladores se identifiquen de forma unívoca, para asegurar

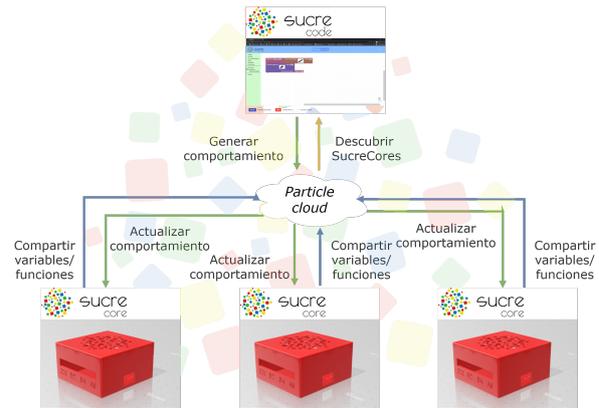


Figura 4: Flujos de comunicación entre la herramienta *SucreCode* y los *SucreCore*.

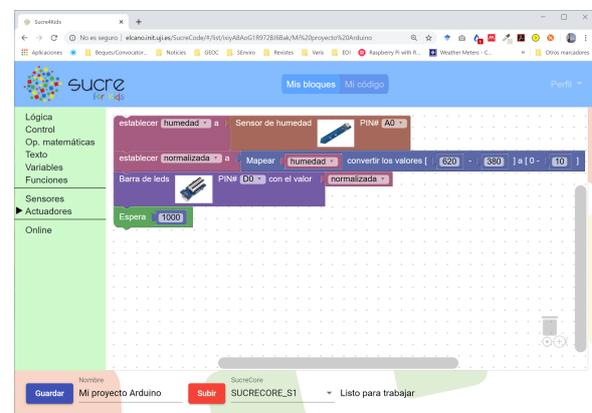


Figura 5: Vista de programación en *SucreCode*.

la interacción y comunicación con la parte cliente *SucreCode*. La conectividad a Internet de *SucreCore* posibilita las actualizaciones OTA para cambiar su comportamiento (Figura 4, línea verde). Una vez generado un nuevo programa con los bloques visuales, este se envía al *SucreCore* para su actualización sin la necesidad de conexión por cable. *SucreCode* empaqueta el código resultante de los bloques visuales con las librerías de cada uno de los sensores/actuadores utilizados (Figura 4). Este paquete se envía junto al identificador del microcontrolador a la operación de actualización (API) del servicio *cloud* de *Particle* para que compile y lance la actualización sobre el *SucreCore* correspondiente. En la interfaz web del *SucreCode* se ha añadido una barra inferior en la vista de creación de programas (Figura 5), que facilita la actualización con un clic. En esta barra también se puede seleccionar el *SucreCore* a actualizar, guardar el proyecto manualmente, añadir un nombre al proyecto y se muestran mensajes de éxito o error en los procesos de actualización y guardado.

El *cloud* de *Particle* también soporta la generación de las variables y funciones en línea y en malla. Nue-

vos bloques en la herramienta web permiten operar con este tipo de variables y funciones (Figura 4, línea azul). Estos bloques no se anidan como el resto, sino que se añaden de forma externa. Así pues, actúan sobre variables y funciones previamente definidas, que pasan a estar en línea o en malla (dependiendo del bloque). Aunque el objetivo es poder consultar estas variables y lanzar estas funciones desde otros *SucreCores*, también se puede realizar externamente a *SucreCode* mediante la API que ofrece el *cloud* de *Particle*.

Otras mejoras de *SucreCode* son la posibilidad de guardar los proyectos y poder reprendre el trabajo. Con ello se obtiene una colección de proyectos, que en un futuro se podrían compartir entre usuarios de la aplicación. Además, se ha dotado a la plataforma de gestión de usuarios para el acceso mediante credenciales. Cada usuario tiene asociados sus propios *SucreCore* que únicamente dicho usuario puede operar (actualizarlos). Esto es de vital importancia cuando se trabaja en grupos, ya que de forma malintencionada se podría cambiar el comportamiento de los *SucreCore* de otros grupos. Finalmente, también se ha añadido la funcionalidad para gestionar y vincular los *SucreCore* a los usuarios mediante un usuario con privilegios (administrador). Todas estas funcionalidades han sido añadidas mediante diferentes vistas en la aplicación Angular, utilizando *Firebase* para la gestión de usuarios y persistencia de los proyectos.

Finalmente cabe destacar que *SucreCode* sigue los principios *mobile-first* utilizando librerías HTML5 *responsive*. El objetivo es que pueda ser ejecutada en dispositivos como tabletas, utilizados mayoritariamente en las aulas y huir del uso de un ordenador que añade mayor complejidad logística para el desarrollo de sesiones. Unido a la no necesidad de cables para alimentación y actualización, el proyecto gana en simplicidad, versatilidad y autonomía.

5. Primera evaluación

Aunque la herramienta web todavía está en proceso de desarrollo (ver Sección 6), durante la última edición del *Practica a l'UJI* (enero del 2020) se realizó una primera validación de la nueva versión de *Sucre4Kids*. El objetivo era realizar los mismos proyectos que se realizaban con la versión anterior, para así observar si se producían mejoras durante la intervención.

La última edición del *Práctica a l'UJI* se caracterizó por tener unas tasas menores de institutos inscritos, lo que redujo en casi un 40 % menos la asistencia. En nuestro curso participaron alrededor de 60 estudiantes de bachillerato, repartidos en tres sesiones. Una sesión fue la más numerosa que se recuerda de las tres ediciones anteriores, con un total de 25 alumnos/as, lo cual puso a prueba la nueva versión utilizando 6 *SucreCore*

conectados en malla sobre la unidad del docente.

La estructura de las sesiones fue idéntica que en ediciones anteriores [15]: una parte de explicación del profesor, otra de proyecto guiado y una parte final de proyecto libre. En las anteriores ediciones con la versión inicial de *Sucre4Kids* únicamente se destinó de media 16 % (17 min.) a la parte del proyecto libre sobre el tiempo de la sesión (105 min.). En la última edición con la nueva versión, todas las sesiones han superado los 30 minutos durante la última parte. Prácticamente se ha duplicado el tiempo destinado a experimentación libre con *Sucre4Kids*. Gran parte de esta mejora se debe a la ausencia de problemas de configuración y conectividad entre la IDE de *Arduino* y el puerto COM.

Otro aspecto destacable ha sido la variedad de proyectos realizados en la parte del proyecto libre por los estudiantes. Debido al mayor tiempo disponible, unido a que la nueva versión de *Sucre4Kids* contiene más sensores y actuadores, hemos percibido que los estudiantes han valorado mucho más positivamente esta última parte de las sesiones que en ediciones anteriores que tenía un carácter testimonial.

6. Conclusiones

La actual comunicación detalla la evolución del proyecto *Sucre4Kids* tras cuatro años fomentando la tecnología y la programación en edad preuniversitarias. Esta evolución marcada por el paradigma de IoT soluciona algunos de los problemas técnicos que se generaban en la anterior versión durante el transcurso de las sesiones, como los problemas de comunicación del puerto COM o el tiempo dedicado a la instalación y puesta a punto de todo el software necesario para poder programar el *Arduino*. Aparte de las mejoras prácticas, la nueva versión añade diferentes funcionalidades muy interesantes para mejorar las dinámicas grupales, ampliando los tipos de proyectos a realizar.

Desde la perspectiva tecnológica, la evolución responde a dos herramientas bien diferenciadas: *SucreCore* y *SucreCode*. La primera propone un nuevo diseño más compacto, encapsulando un microcontrolador más avanzado y con soporte de conectividad inalámbrica, y con la capacidad de crear redes en malla con otros *SucreCore*, facilitando su actualización mediante actualizaciones OTA y su autonomía energética al contar con una batería recargable. Características que hacen que *SucreCore* sea simple, versátil y autónomo. La segunda, la herramienta web *SucreCode*, aprovecha las características del *SucreCore* para favorecer la programación visual y poder cambiar su comportamiento sin cables. Añade el soporte de autenticación por usuario/contraseña y la posibilidad de guardar proyectos. Sin embargo, un aspecto interesante para explotar en futuras intervenciones es la posibilidad de crear

variables y funciones en línea (compartidas), que permitirían generar nuevas dinámicas de aprendizaje. Por ejemplo, proyectos entre grupos con fórmulas de gamificación en el aula (de colaboración entre grupos, de competición entre grupos, de estrategia, etc.).

La nueva versión de *Sucre4Kids* se ha probado en la edición 2020 del programa *Practica a l'UJI* con cerca de 60 estudiantes. Aunque el proyecto está en una fase de desarrollo, las sesiones se realizaron sin problemas técnicos. Con esta versión se ganó en agilidad durante el transcurso de las sesiones, debido a la independencia y versatilidad de *SucreCore* soportando actualizaciones OTA, con las que se evita dependencias con el software del ordenador utilizado para programar. El tiempo ganado permitió que los estudiantes aprovecharan mucho más la última parte de cada sesión.

La hoja de ruta de la nueva versión de *Sucre4Kids* contempla la siguiente serie de objetivos.

- Definición y puesta de marca de proyectos colaborativos entre grupos con funcionalidades de variables/funciones en línea/malla.
- Creación de una nueva funcionalidad en la herramienta *SucreCode* para visualizar las variables en línea/malla y representarlas para que el alumnado pueda operar con ellas. También posibilitar la ejecución de las funciones en línea de forma remota.
- En la parte de *SucreCore* se está desarrollando una versión con conectores magnéticos, para facilitar la conexión de los sensores y actuadores. La finalidad es ampliar el rango de estudiantes de *Sucre4Kids* con los de últimos cursos de primaria.
- En la misma línea, también se trabaja en diseñar una versión de *SucreCore* sin la necesidad de pantallas para su programación.
- Finalmente, queremos proseguir con validaciones en el aula. Para ello continuaremos realizando las actividades de los años anteriores, e incrementaremos con la asistencia a nuevos eventos y ferias.

Agradecimientos

El proyecto está financiado por la Universitat Jaume I - PINV 2017 (UJI-A2017-14). Sergio Trilles ha sido financiando por el programa PINV2018 (Acción 3.2) de la UJI (POSDOC-B/2018/12), y Carlos Granell por el programa Ramón y Cajal (RYC-2014-16913).

Referencias

- [1] Chris Anderson. *MAKERS: La nueva revolución industrial*. Empresa Activa, 2013.
- [2] Vinton G. Cerf. Computer science in the curriculum. *Communications of the ACM*, 59(3):7–7, feb 2016.
- [3] Comisión Europea. Comunicación de la comisión al parlamento europeo, [...] hacia una economía de los datos próspera. <https://bit.ly/2OJA0ck>, 2014. Acceso 01-05-2020.
- [4] Daniel Spikol y Nils Ehrenberg Eva-Sophie Katterfeldt, David Cuartielles. Talkoo: A new paradigm for physical computing at school. In *Proceedings of the The 15th International Conference on Interaction Design and Children*, pages 512–517, 2016.
- [5] Expansión. Fracaso en el 'Informe Pisa': por qué los españoles no accederán a las profesiones del futuro. <https://bit.ly/2UGUKiA>. Acceso 01-05-2020.
- [6] GOV.UK. National curriculum in England: computing programmes of study. <https://bit.ly/2uzobyD>. Acceso 01-05-2020.
- [7] Ministerio de Ciencia, Innovación y Universidades. Anuario de indicadores universitarios 2018. <https://bit.ly/2VZXjTj>. Acceso 04-05-2020.
- [8] Ministerio de Ciencia, Innovación y Universidades. Datos y cifras del Sistema Universitario Español Publicación 2018-2019. <https://bit.ly/3aZXC1j>. Acceso 04-05-2020.
- [9] Ministerio de Educación y Formación Profesional. El currículo en Primaria, ESO y Bachillerato. <https://bit.ly/31JASTJ>. Acceso 01-05-2020.
- [10] C. Garcia-Berdone y F. J. Sanchez-Pacheco P. Juan Sotorrio-Ruiz, D. Trujillo-Aguilera. Aproximación a la técnica “aprender haciendo” para la docencia en microprocesadores. *XXV Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas (CUIEET)*, 2017.
- [11] Carlos Granell y Estefania Aguilar-Moreno Sergio Trilles. Sucre4kids: tres años de experiencia en la incentiación del pensamiento computacional en edades preuniversitarias. In Oscar Martínez Bonastre André Vaz Fidalgo, editor, *TICAI 2018 TICs para el Aprendizaje de la Ingeniería ISBN 978-84-8158-838-5*. IEEE, 2018.
- [12] Fundación Telefónica. *Prepara tu Escuela para la Sociedad Digital: Claves para sumarse al cambio*. Fundación Telefónica, 2016.
- [13] whitehouse.gov. Computer Science For All. <https://bit.ly/2SgEc5J>. Acceso 01-05-2020.
- [14] Jeannette M Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.
- [15] Sergio Trilles y Carlos Granell. Sucre4kids: El fomento del pensamiento computacional a través de la interacción social y tangible. *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática*, 3:303–310, 2018.
- [16] Anja Balanskat y Katja Engelhardt. Computing our future Computer programming and coding Priorities, school curricula and initiatives across Europe. Technical report, Creative Commons, 2015.
- [17] Junnan Yu y Ricarose Roque. A survey of computational kits for young children. In *Proceedings of the 17th ACM conference on interaction design and children*, pages 289–299, 2018.