

# Sucre4Stem: proyectos colaborativos mediante dispositivos IoT para el alumnado de secundaria y pre-universitario

Sergio Trilles, Aida Monfort-Muriach, Águeda Gómez-Cambronero y Carlos Granell

**Resumen**—This paper describes a new technological evolution of the *Sucre* project, which aims to promote scientific vocations, develop computational thinking and programming skills in pre-university students. The improved version is called *Sucre4Stem* and has been designed from the Internet of Things perspective. At a technological level, we differentiate two main tools, *SucreCore* and *SucreCode*. *SucreCore* provides a new, more compact design, encapsulates an advanced microcontroller, and supports wireless connectivity with the ability to create online variables and functions. *SucreCode*, the block-based visual programming tool, has a revamped interface and allows wireless communication with *SucreCore*. At the pedagogical level, *Sucre4Stem* makes it easier to undertake new group dynamics and create novel types of collaborative projects between groups of students. In this article, we also explore how these collaborative projects can be realised by taking advantage of the different types of communications between *SucreCore* and the server-side platform using shared online variables and functions.

**Index Terms**—Computational thinking, programming promotion, pre-university ages, Internet of Things

## I. INTRODUCCIÓN

PODRÍA darse la paradoja de que, frente a una sociedad cada vez más orientada a la tecnología [1], 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, del inglés STEM). Contrariamente, 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, según el reciente informe PISA [2]. 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 [3]. Especialmente sangrante es la matriculación de mujeres en estudios en CTIM. En el último anuario de indicadores universitarios disponible en [4], 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 [4]. 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 [5], tanto en el trabajo como en la vida diaria. El pensamiento computacional incide en pensar cómo lo haría un científico computacional y/o informático [6]. 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 [7]. Reino Unido propuso en 2014 la computación como parte del currículo nacional en las escuelas de primaria y secundaria [8]. La administración Obama puso en marcha en 2016 un programa educativo [9] para llevar la computación a profesores y escolares K-12 (de primaria y secundaria). Los programas educativos en estos países y otros coinciden en la necesidad de introducir competencias digitales de las Tecnologías de la Información y Comunicación (TIC), para el fomento de la creatividad, el pensamiento lógico y la resolución de problemas complejos. El hecho de añadir la creatividad (o mejor dicho el arte) a la disciplina CTIM, ha evolucionado el concepto de CTIM a Ciencia, Tecnología, Ingeniería, Arte y Matemáticas (CTIAM, del inglés STEAM)

En contraposición al avance realizado en otros países desarrollados, el currículo nacional español [10] sigue mayoritariamente centrado en el uso de las TIC y únicamente considera la computación como un mero instrumento y no como una "disciplina" en sí, dejando de lado el pensamiento computacional. Esta percepción ha sido corroborada por informes recientes realizados por la Fundación Telefónica [11] y por la FECYT/Google España [12]. Por ejemplo, no se instruye al alumnado en cómo están hechas 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 que sea a su vez consecuente con la realidad digital en la

que viven los escolares.

En respuesta a esta problemática, el proyecto *Sucre*<sup>1</sup> (en sus siglas en inglés *Sense yoUr Context and REact*) quiere contribuir a paliar esta necesidad 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 [13] 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 (*Practica a l'UJI*), visitas a centros educativos y ferias, alcanzando en total a más de 600 estudiantes de forma directa y a más de 20.000 de forma indirecta. En un artículo reciente, Trilles y Granell [14] resumen todas las intervenciones realizadas durante esta primera etapa del proyecto *Sucre*.

Tras esta primera etapa, el proyecto *Sucre* 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 *Sucre* 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.

Este artículo es una extensión de un trabajo previo [15] (vídeo presentación accesible en [16]) presentado en las XXVI Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI 2020), el cual fue seleccionado para la sesión de mejores trabajos de las jornadas. En [15], detallamos de forma técnica la evolución de *Sucre* y su primera evaluación en el programa *Practica a l'UJI*. En este artículo, exploramos con mayor detalle un abanico de proyectos e intervenciones colaborativas que el alumnado puede realizar en las aulas con la versión mejorada del proyecto *Sucre*. Como el público objetivo son estudiantes de educación secundaria y pre-universitarios, utilizamos una marca propia del proyecto *Sucre* que denominamos *Sucre4Stem*, para diferenciarla de la otra marca *Sucre4Kids* dirigida a un público más joven, de primaria. Ambas marcas beben de la misma tecnología desarrollada en el contexto del proyecto *Sucre*. Sin embargo, somos conscientes de la complejidad cognitiva que pueden acarrear los conceptos y tecnología descrita en este trabajo así como las habilidades requeridas. Por ese motivo, adaptamos el tipo de proyectos y la forma de interactuar con la tecnología y la plataforma *Sucre* en función de la franja de edad de los estudiantes. Por tanto, en este trabajo nos centramos en un público adolescente para la adquisición

de competencias tecnológicas avanzadas mediante proyectos colaborativos basados en dispositivos IoT.

## II. ESTADO DEL ARTE

En el mercado y en el ámbito académico existen diferentes iniciativas similares a *Sucre4Stem*, 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, Kits virtuales, y Kits híbridos con programación virtual por bloques o mediante programación tangible por bloques<sup>2</sup>. El proyecto *Sucre4Stem* se enmarca en la categoría de kits híbridos con programación virtual por bloques, pero añadiendo la posibilidad de acoplar y desacoplar elementos electrónicos como sensores y actuadores. A continuación, se analizan diferentes proyectos dentro de esta misma categoría y que además también se caracterizan por tener elementos electrónicos. Dentro de esta categoría únicamente se han escogido iniciativas capaces de plantear proyectos abiertos, más similares a *Sucre4Stem*. No se incluyen en el análisis comparativo otras iniciativas de robótica educativa donde hay elementos motorizados, como por ejemplo Makeblock o LEGO MINDSTORMS EV3.

*Talkoo* [18] está basado en *Arduino*. Los módulos de sensores y actuadores se conectan fácilmente 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* [19], 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 basado en *Microsoft MakeCode*. *Micro:bit* [20] es una 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 la conexión con ellos. El microcontrolador puede ser programado mediante un lenguaje de bloques, *Java* o *Micropython* (un lenguaje similar a *Python*). *Chibi Chip* [21] 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* [22] 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

<sup>2</sup>En [17] se define programación tangible por bloques, cuando los bloques virtuales pasan a ser piezas físicas y estas se combinan para dar solución a un problema.

<sup>1</sup><http://www.sucre.uji.es/>

por bloques, pero permite programación tradicional mediante la IDE de *Arduino*.

La Tabla I compara los trabajos anteriores con *Sucre4Stem*, atendiendo a las siguientes características.

- Hardware libre, si se puede utilizar, modificar y distribuir libremente. Escala: *Sí/No*.
- Conexión Internet, si requiere conexión a Internet para su uso. Escala: *Sí/No*.
- Actualizaciones *Over-The-Air*, si puede actualizarse de forma remota. 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.
- Organización por grupos, si la iniciativa posibilita el uso de forma grupal. Escala: *Sí/No*.
- Cantidad de actuadores y sensores, se indica el número de sensores y actuadores que incluye cada iniciativa en el kit. Escala: *Númerica*.

Según la Tabla I, 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 *Sucre4Stem* 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 VI), 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 *Sucre4Stem*, ninguna solución analizada tiene en cuenta poder ser utilizada de forma grupal.

### III. ORIGEN DEL PROYECTO *Sucre*

Para identificar los nuevos cambios y mejoras de *Sucre4Stem*, en esta sección describimos el proyecto *Sucre* en su versión inicial [23]. 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 en su evolución, *Sucre* 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* [24]. Y el segundo, se enmarca en las competencias y habilidades propias del pensamiento computacional [25]. Con *Sucre*, intentamos que los estudiantes experimenten, que “aprendan haciendo” [26], pero también que “aprendan pensando” [27], fomentando que las sesiones sean abiertas, participativas y basadas en el trabajo colectivo para la comprensión y experimentación de los conceptos y tecnologías propuestas.

*Sucre* 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 (Figura 1) 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 dispositivos de E/S al núcleo. Los sensores actúan como dispositivos de entrada de datos y los actuadores como 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.



Figura 1: Maletín de la versión inicial de *Sucre4Kids*.

Por otro lado, la herramienta web de programación visual basada en *Blockly* [28] permitía la programación visual por bloques del *Arduino*. Se extendieron los bloques ya disponibles de la librería *Blockduino* [29], 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: el proyecto de “Planta inteligente” que permitía medir la humedad de la tierra mediante un sensor de humedad del suelo para, conectado a un actuador de barra de LEDs poder indicar el nivel alcanzado; el proyecto “Parking inteligente” que comprobaba si un objeto (p. ej. coche miniatura) ocupaba o no un sitio (estacionamiento) mediante un sensor de proximidad; y el

Tabla I: Relación entre los proyectos y los objetivos formativos del taller

Iniciativa	Hardware libre	Conexión Internet	Actualizaciones <i>Over-The-Air</i>	Basado en proyectos	Conectores <i>plug&amp;play</i>	Lenguaje de programa	Organización por grupos	Cantidad de sensores y actuadores
<i>Talkoo</i>	No especific.	No	No	No	Sí	<i>Arduino</i>	No	N/A
<i>Grove Zero</i>	No especific.	No	No	Sí	Sí	<i>Arduino</i>	No	5 sensores y 2 actuadores
<i>Micro:bit</i>	Sí	No	No	Sí	No	<i>Java y MicroPython</i>	No	6 sensores y 2 actuadores
<i>Chibi Chip</i>	Sí	No	No	Sí	Sí	<i>Arduino</i>	No	No incluye
<i>Circuit Playground Express</i>	Sí	No	No	No	No	<i>Arduino</i>	No	4 sensores y 3 actuadores
<i>Sucre4Stem</i>	Sí	Sí (variables/funciones en línea)	Sí	Sí	Sí	<i>Arduino</i>	Sí	8 sensores y 3 actuadores

Figura 2: *SucreCore* impreso con tecnología 3D para los microcontroladores *Particle Argon*.

proyecto “Luz inteligente” que activaba un LED cuanto menor era la luz ambiental captada con un sensor lumínico.

Durante los cuatros años de vida, se han realizado diferentes actividades que nos han permitido afinar tanto 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 (ver [13], [14]). 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 reunió a más de 3.000 alumnos/as de entre 6 y 18 años, en el Sagunto Gaming Fest y, finalmente, en el FIMP 2019 (Feria de videojuegos y tecnología) celebrada en Gijón.

#### IV. 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 versión mejorada de *Sucre* que describimos aquí, *Sucre4Stem*, viene con un nuevo desarrollo de encapsulamiento propio, llamado *SucreCore*, compuesto por un nuevo microcontrolador (*Particle Argon*) con una amplia variedad de posibilidades provenientes del paradigma IoT (Figura 2). En base a estas nuevas funcionalidades, se ha desarrollado enteramente la herramienta de programación visual, pasándose a denominar *SucreCode*.

Figura 3: Versión renovada de maletín de *Sucre4Kids*.

##### IV-A. *SucreCore*

El nuevo maletín en la Figura 3 reúne un mayor número de sensores y actuadores, con un coste total orientativo de unos 85 €. Concretamente, incluye:

- núcleo: 1x *SucreCore* (con batería);
- actuadores: 1x LED, 1x zumbador, 1x pantalla de segmentos;
- sensores simples: 1x botón, 1x sensor rotatorio, 1x sensor de proximidad, 1x sensor de luz y 1x sensor de ruido;
- sensores avanzados: 1x sensor de temperatura y humedad y 1x sensor humedad del suelo; y
- varios: 1x cable microUSB-USB y fichas de documentación.

La pieza central del maletín es el *SucreCore*, por lo que merece una descripción detallada por las nuevas funcionalidades que ofrece. El microcontrolador que se esconde dentro de la carcasa impresa mediante tecnología 3D de *SucreCore* es llamado *Particle Argon* [30]. *Particle Argon* ofrece unas 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*. Se caracteriza por su conectividad IP/Internet mediante Wi-Fi. Además ofrece la conectividad *Near field communication*

(NFC) y pueden funcionar con batería al contar con un conector *JSP*.

El dotar a la solución *SucreCore* de conectividad a Internet y poder adoptar diferentes modos de conectividad (*SucreCore* a servidor, servidor a *SucreCore* y entre *SucreCore*), 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 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* y un servidor remoto; o la 4) posibilidad de creación de operaciones/funciones ejecutables de forma remota o desde otro *SucreCore*.

Al igual que en la versión previa de *Sucre*, para facilitar la conexión de los sensores y actuadores sobre el *pin-out* y garantizar la compatibilidad de sensores y actuadores, *Sucre4Stem* 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, se ha dotado a cada *SucreCore* de una batería recargable de 1800 mA y accesible mediante un interruptor de encendido/apagado. El consumo energético en funcionamiento es de 31.7 mA/hora, con el que se obtiene una autonomía de aproximada de 60 horas. La batería puede recargarse con un cable micro-USB.

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 2). 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.

#### IV-B. *SucreCode*

La herramienta web también ha sido rediseñada desde cero para alinearse con las nuevas funcionalidades que ofrece *SucreCore*. La nueva aplicación web de programación *SucreCode*, 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*), *SucreCode* ha sido rediseñado para añadir nuevas funcionalidades, entre las que destacan: a) lanzamiento de actualizaciones OTA, b) generación de bloques de variables/funciones en línea, 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 la interacción y comunicación

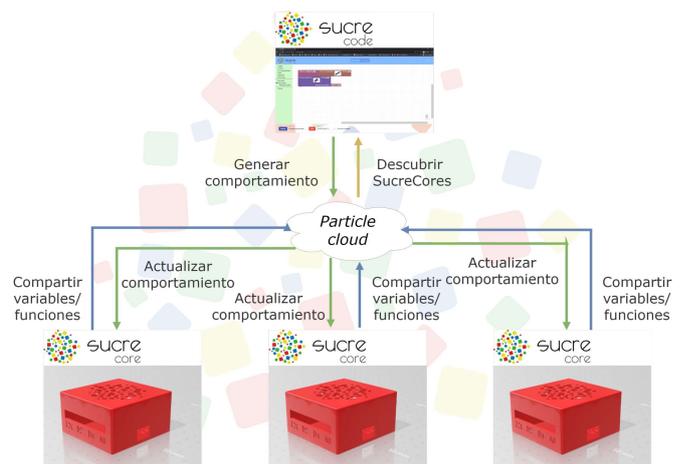


Figura 4: Flujos de comunicación entre la herramienta *SucreCode* y los *SucreCore*.

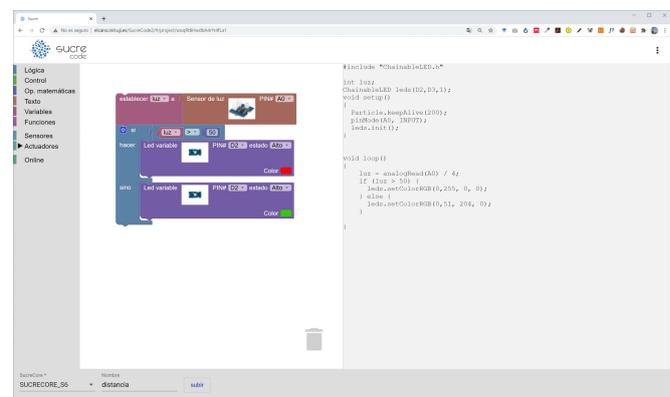


Figura 5: Vista de programación en *SucreCode*.

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. Nuevos 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 (dependiendo del bloque). Aunque el objetivo es poder consultar estas variables y lanzar estas funciones desde otros *SucreCore*, también se puede acceder externamente a *SucreCode* mediante la API que ofrece el

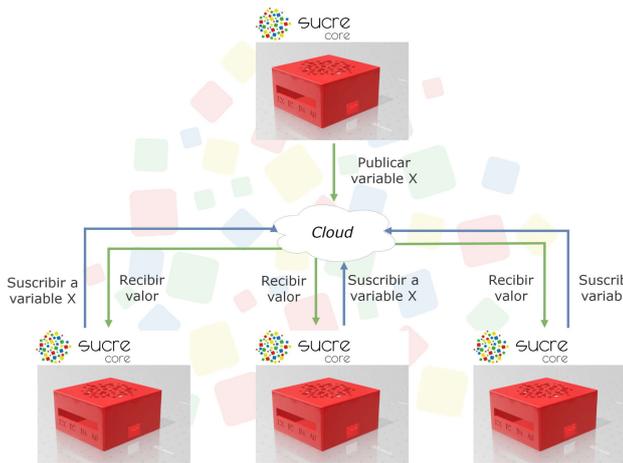


Figura 6: Descripción gráfica de las variables en línea.

cloud de Particle.

Técnicamente, las variables (y funciones) en línea propuestas se basan en las funcionalidades propias del ecosistema de Particle. Más concretamente la variable en línea se basa en el paradigma de publicar/suscribir [31] (Figura 6). Cuando desde un *SucreCore* se quiere declarar una variable en línea se utiliza el bloque "Publicar". Este bloque debe definir un nombre, que será utilizado para suscribirse a la variable por otro *SucreCore*. Al bloque publicar se acopla por la parte de la derecha la variable a compartir. Desde el punto de vista del suscriptor, se utiliza el bloque "Suscribir". Este al igual que el anterior, deberá indicar el nombre de la variable a la que quiera suscribirse y acoplar por la derecha la variable donde guarda el valor recibido. Cuando el *SucreCore* con el rol de publicar ejecute el bloque "publicar", la variable asociada al bloque "suscribir" será actualizada automáticamente. Desde el punto de vista del estudiante, sin embargo, es un recurso interesante de aprendizaje en comparación con las variables tradicionales, ya que ofrecen canales de comunicación compartidos entre *SucreCore*, es decir, entre grupos de estudiantes. En un plano práctico, un grupo de estudiantes puede *escribir* en una variable compartida y otro grupo puede *leer* el valor de dicha variable.

Otras mejoras de *SucreCode* proporcionan la posibilidad de guardar los proyectos (código) y poder retomar el trabajo posteriormente. Con ello se obtiene una colección de proyectos (Figura 7), 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 bien de forma malintencionada, o bien involuntariamente, se podría cambiar el comportamiento de los *SucreCore* de otros grupos generando confusión. 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 ta-

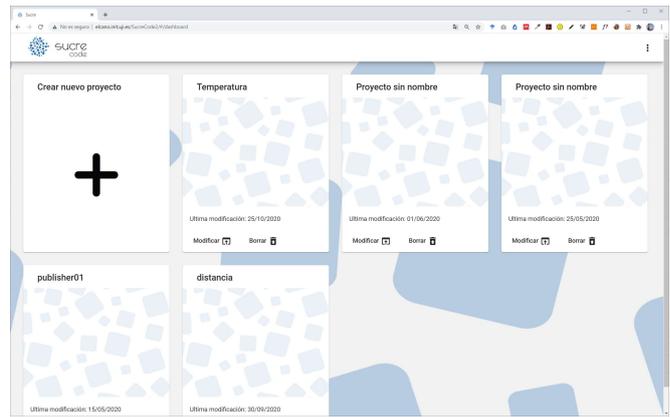


Figura 7: Vista de proyectos en *SucreCode*.

bletas, utilizados mayoritariamente en las aulas y evitar así la dependencia de un ordenador de sobremesa que añade mayor complejidad logística para el desarrollo de las intervenciones. Unido a la no necesidad de cables para alimentación y actualización, la versión mejorada del proyecto en conjunto gana en simplicidad, versatilidad y autonomía.

## V. EXPERIMENTACIÓN

Las funcionalidades que incorporan las nuevas versiones de *SucreCode* y *SucreCore* amplían significativamente el número y tipo de proyectos que se pueden realizar. En la versión inicial partíamos de tres proyectos base [14]: la planta inteligente, la luz inteligente y el parking inteligente. Esta limitación era impuesta básicamente por el reducido número de sensores y actuadores. Con la incorporación de nuevos tipos de sensores y actuadores en el nuevo maletín, aumenta en consecuencia la posibilidad de crear proyectos variopintos.

Aunque la cantidad y diversidad de sensores es importante, no es el aspecto crucial que caracteriza a *Sucre4Stem*. En la versión anterior, los tres proyectos base se desarrollaban de forma individual, es decir, por un grupo de estudiantes. La integración de *Sucre4Stem* en el ecosistema de IoT, abre la posibilidad a proyectos colaborativos entre grupos. Evidentemente, *Sucre4Stem* permite aún la creación de proyectos individuales, pero la razón de abrazar el paradigma IoT es justo promover la conectividad y colaboración entre grupos de estudiantes. Por lo tanto, entendemos por proyectos colaborativos aquellos proyectos cuya lógica puede desglosarse en partes más simples, y cada grupo de estudiantes se encarga del desarrollo de una. Hay que destacar que no se trata de partes independientes, sino que existen dependencias entre ellas para que el proyecto colaborativo en su conjunto opere correctamente. Por lo tanto, esas dependencias requieren de intercambios de datos entre las partes (es decir, cada *SucreCode* por grupo) que contribuyen al mismo proyecto colaborativo.

La concepción de proyectos colaborativos no incide únicamente en la colaboración y cooperación. Aunque son sumamente importantes, creemos que los proyectos colaborativos sustentados mediante *Sucre4Stem* puede favorecer una exploración y pensamiento más profundo en los estudiantes permitiéndoles que lleven a cabo y prueben sus propios modelos y estrategias para solucionar "variantes reales de versiones de proyectos para adultos" [32]. Dada la penetración de la digitalización en la sociedad actual, Guzdial y

colegas [32] argumentan que el impacto de la computación se produce tanto dentro como fuera de la escuela y, por lo tanto, tal como se enseña actualmente la computación en las escuelas tiene mucho menos impacto en el aprendizaje que el esperado. Nuestro objetivo a largo plazo es abrazar esta premisa, mediante proyectos colaborativos con *Sucre4Stem*, para que los estudiantes aprendan a través de proyectos que recrean situaciones reales, como con las que diariamente interactúan fuera de la escuela.

Volviendo al aspecto técnico de colaboración que propociona la interacción entre grupos de estudiantes que realizan un proyecto colaborativo, la conectividad entre *SucreCore* es una característica distintiva de *Sucre4Stem* ya que permite el intercambio de observaciones de sensores entre los *SucreCore*. No es imprescindible que todos los *SucreCore* se encuentren físicamente en la misma aula. Al igual que ocurre en el mundo real, los dispositivos IoT se encuentran a menudo geográficamente separados, lo cual no impide que se comuniquen entre ellos porque forman parte de una misma red. Los *SucreCore* que colaboran en un mismo proyecto pueden estar geográficamente separados (en casa y en el aula, en aulas distintas en el mismo instituto, en aulas de institutos distintos).

Tal como comentamos anteriormente, la conectividad se logra con variables y funciones en línea, las cuales incrementan la posibilidad de crear nuevas dinámicas de aprendizaje que incluyan la compartición de datos entre grupos y fórmulas de gamificación en el aula (colaboración, competición, etc) para la realización de un proyecto colaborativo donde cada grupo colabora o compite de forma sana. Con ello, tratamos de incentivar la colaboración mediada por la tecnología, algo plenamente instaurado en nuestra vida cotidiana, pero haciendo un uso más meditado y responsable dentro del aula.

A continuación describimos una validación preliminar de *Sucre4Stem* con los proyectos diseñados para la anterior versión. Luego, exploramos nuevas ideas para el desarrollo de proyectos colaborativos que exploten la conectividad inherente de *Sucre4Stem*.

#### V-A. Validación preliminar con proyectos individuales

En la última edición del programa *Practica a l'UJI* (enero del 2020) se realizó una primera validación de *Sucre4Stem* y de la tecnología subyacente. El objetivo era realizar los mismos proyectos que se realizaban con la versión anterior, para así comprobar si se producían mejoras organizativas (tareas de preparación del maletín, problemas técnicos, etc.) durante el desarrollo de la intervención y recoger las primeras impresiones de los participantes.

La última edición del *Practica a l'UJI* se caracterizó por tener unas tasas menores de institutos inscritos, reduciendo en casi un 40% la asistencia comparado con ediciones anteriores. En nuestro curso participaron alrededor de 60 estudiantes de bachillerato repartidos en tres sesiones. Una de estas sesiones fue bastante numerosa con un total de 25 alumnos/as, lo cual nos permitió testear la nueva versión utilizando 6 *SucreCore* conectados a la vez.

La estructura de las sesiones fue idéntica que en ediciones anteriores [13]: una parte de explicación del profesor, otra de proyecto guiado y una parte final de proyecto libre. En las anteriores ediciones, únicamente se destinó de media

16% (17 min.) al proyecto libre sobre el tiempo total de una sesión (105 min.). En la última edición con *Sucre4Stem*, todas las sesiones superaron los 30 minutos para la última parte. Prácticamente se ha duplicado el tiempo destinado a experimentación libre con *Sucre4Stem*. 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, y a la eliminación de la fase de instalación previa que realizaban los docentes en las aulas y la ejecución y verificación del software necesario. En ediciones anteriores, por ejemplo, se requería la instalación y configuración del IDE de *Arduino* y de un servidor web para permitir la conexión entre IDE y la herramienta de programación web, incurriendo en una pérdida de tiempo nada despreciable en tareas de configuración ajenas al objetivo de la intervención.

Una observación derivada de las mejoras anteriores fue la variedad de proyectos realizados por los estudiantes en la parte del proyecto libre de cada sesión. Debido al mayor tiempo disponible, unido a que *Sucre4Stem* incluye una mayor variedad de sensores y actuadores, percibimos que los estudiantes valoraban más positivamente el tiempo empleado en el proyecto libre que en ediciones anteriores, que a menudo quedaba en una parte meramente testimonial de la sesión.

#### V-B. Ejemplos de proyectos colaborativos

A continuación describimos tres ideas preliminares de proyectos colaborativos, poniendo en valor la conectividad que ofrece *Sucre4Stem*. Somos conscientes que la concepción extendida de proyectos colaborativos, entendida no únicamente como proyectos al que contribuyen varios grupos de forma colaborativa sino también que recrean situaciones reales, es una tarea aún en desarrollo. Además, la definición de un proyecto colaborativo debe alinearse con una aproximación pedagógica que tenga en cuenta las competencias y resultados de aprendizaje que los estudiantes adquirirían. En este artículo no describimos los proyectos colaborativos en el marco expuesto anteriormente. Sin embargo, a pesar de esta limitación, nuestra intención es anticipar ideas de proyectos colaborativos que son técnicamente viables con *Sucre4Stem* y, consecuentemente, abrir un diálogo con la comunidad sobre la conveniencia de cambiar el modo en que los docentes e investigadores enseñan y preparan contenidos relacionados con la iniciación a la programación y el pensamiento computacional. En lo que sigue, cada uno de los ejemplos, se caracteriza por explotar un modelo de comunicación diferente según las necesidades del proyecto: de servidor a *SucreCore*, de *SucreCore* a servidor y entre distintos *SucreCore*.

Se entiende como servidor a una aplicación web que sea capaz de interactuar con las variables y funciones en línea. En el primer modelo de comunicación (servidor a *SucreCore*), la aplicación web genera variables que pueden ser consumidas por un *SucreCore* o expone funciones que pueden ser invocadas por un *SucreCore*. En el modelo *SucreCore* a servidor, la aplicación web (servidor) se encarga de consumir los datos generados por los sensores a través de las variables en línea. El tercer modelo facilita la comunicación entre dos *SucreCore* tanto para compartir variables en línea, como para definir o invocar funciones en línea.

El primer ejemplo es una evolución de un proyecto muy simple, encender y apagar un LED, que realizábamos con

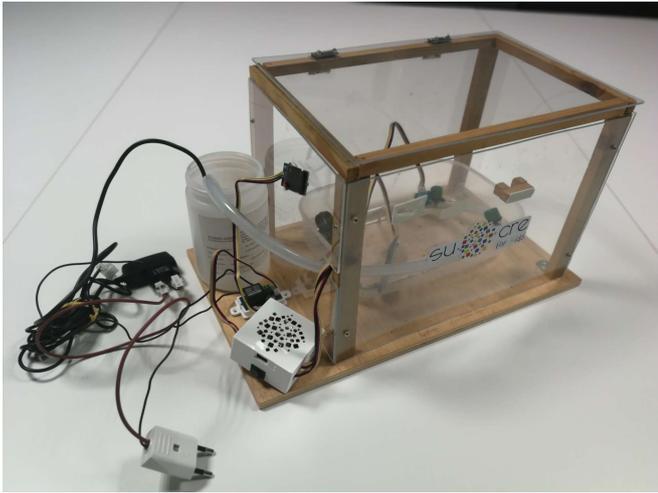


Figura 8: Ejemplo de montaje del proyecto del invernadero conectado.

la anterior versión a modo de ejemplo introductorio (“¡Hola Mundo!”). Siguiendo una comunicación desde el servidor al *SucreCore*, encenderíamos o apagaríamos un LED a distancia. Este ejemplo requería de una llamada a las funciones en línea incorporadas en *Sucre4Stem*. De esta forma, se trabajaría el concepto de función para encapsular los bloques de programación para encender o apagar un LED. Esta función en línea sería invocada desde la parte servidora pero el resultado se visualizaría en el propio montaje, esto es, en un *SucreCore*.

El segundo ejemplo consiste en un invernadero inteligente conectado (Figura 8) siguiendo un modelo de comunicación de *SucreCore* a servidor. El proyecto original fue concebido como trabajo curricular del alumnado de 4º curso de ESO en un instituto de secundaria en Orpesa (Castelló). El objetivo era monitorizar las diferentes variables de interés durante la germinación de diferentes tipos de semillas. El invernadero original consistía en un sensor de humedad del suelo y un relé que reaccionaba sobre una bomba de agua en base a la cantidad de humedad existente en la tierra de una planta. Adicionalmente se agregaron otros sensores para monitorizar variables como la luz o temperatura y humedad ambiental. Las observaciones de estas variables se mostraban en un pequeño display. El alumnado anotaba de forma manual los valores tres veces al día, para su posterior análisis de la evolución del proceso de germinación en diferentes plantas. La nueva versión que se plantea eliminaría la tediosa tarea de anotar manualmente las observaciones de los diferentes sensores mediante la definición de variables en línea que enviarían las observaciones de forma automática a un servidor. En este modelo, el *SucreCore* crea un canal de escritura con el servidor en la nube y actúa como un productor de datos, es decir, usa una variable en línea para subir las observaciones al servidor. De esta forma es posible generar tablas y gráficos de la evolución de los diferentes fenómenos a monitorizar, como ejemplo de una tarea de análisis que pueden realizar los estudiantes.

El último ejemplo de proyecto colaborativo es la evolución del proyecto base de aparcamiento inteligente. El aparcamiento inteligente colaborativo utilizaría un modelo de comunicación más complejo (de *SucreCore* a *SucreCore*). En este caso, participarán tres grupos, y cada uno de ellos

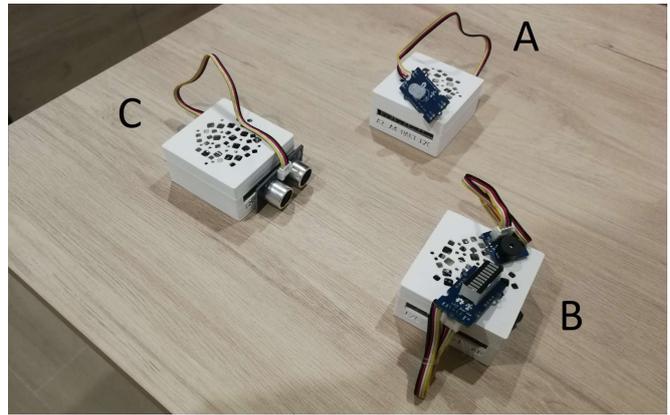


Figura 9: Montaje del proyecto aparcamiento inteligente colaborativo donde participan tres unidades de *SucreCore*.

realizará un montaje para una parte diferente del proyecto colaborativo (ver Figura 9). El montaje A en la Figura 9 consistiría únicamente en un LED, que representaría el testigo luminoso de plaza ocupada o libre. El segundo montaje etiquetado como B representaría un coche, estaría formado por un zumbador y/o una barra de LEDs para indicar la distancia entre un obstáculo (pared) y la parte trasera del vehículo. Finalmente, el tercer montaje C representaría ese obstáculo (pared) y estaría dotado de un sensor de proximidad. El montaje C definiría una variable en línea con el valor del sensor de proximidad, creando un canal de comunicación entre el *SucreCode* asociado y el servidor. Por otra parte, los otros dos montajes accederían a dicha variable para obtener el valor de la distancia entre la pared y la parte trasera del vehículo, creando por lo tanto sendos canales de comunicación ente el servidor y los *SucreCore*. El montaje B (coche) actualizaría la barra de LEDs encendiendo o apagando LEDs en función de la distancia y/o aumentando la frecuencia de pitido del zumbador. El montaje A (testigo) mostraría si la plaza de aparcamiento estaría ocupada encendiendo el LED en color rojo si el coche se acercara lo suficiente a la pared.

## VI. CONCLUSIONES

Este artículo detalla la evolución tecnológica del proyecto *Sucre* en el fomento de la tecnología y la programación en edades preuniversitarias. Las mejoras presentadas vienen marcadas principalmente por la incorporación del paradigma de IoT que 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 y organizativas, la nueva versión añade mayor conectividad en la creación de proyectos colaborativos facilitando así las dinámicas grupales de aprendizaje.

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 canales de comunicación 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 de conectividad de *SucreCore* para enviarles el código prescindiendo de conexiones físicas (cables). Además, *SucreCode* añade el soporte de autenticación por usuario/contraseña y la posibilidad de guardar/recuperar proyectos. Un aspecto interesante para explotar experimentalmente en futuras intervenciones es la posibilidad de crear variables y funciones en línea (compartidas), que permitirían generar nuevas dinámicas de aprendizaje. En este artículo hemos esbozado tanto las ventajas técnicas como las oportunidades de aprendizaje que proporcionan los proyectos colaborativos basados en las nuevas funcionalidades de *Sucre4Stem*. Sin embargo, somos conscientes de que nos quedan otros aspectos y métodos de aprendizaje interactivos por explorar. Por ejemplo, proyectos entre grupos con fórmulas de gamificación en el aula, que enfatizan estrategias de colaboración o competición entre grupos en función de los objetivos de aprendizaje buscados. En definitiva, la simbiosis entre *Sucre4Stem* y el paradigma IoT aporta versatilidad en cuanto a prácticas de aprendizaje en el aula para facilitar la comprensión profunda de la tecnología en proyectos colaborativos que replican la realidad.

*Sucre4Stem* se ha probado en la edición 2020 del programa *Practica a l'UJI* con cerca de 60 estudiantes. Aunque el proyecto está aún en fase de desarrollo, las sesiones se realizaron sin problemas técnicos, debido a la independencia y versatilidad de *SucreCore* para soportar actualizaciones OTA, con las que se evitaron dependencias con el software del ordenador utilizado para programar. Todo ello repercutió en una mayor fluidez y agilidad de las sesiones y el tiempo ganado permitió que los estudiantes aprovecharan mucho más la última parte de cada sesión centrada en el desarrollo libre con *Sucre4Stem*.

Para que los proyectos colaborativos sean una realidad, la hoja de ruta de *Sucre4Stem* contempla la implementación de variables/funciones en línea, la posibilidad de representar visualmente dichas variables en *SucreCode* para que el alumnado pueda operar/programar con ellas, y la ejecución remota de funciones en línea.

Otra funcionalidad deseable es la posibilidad de integrar *Sucre4Stem* dentro de la plataforma IFTTT ("If This, Then That" - <https://ifttt.com/>). La implementación de este tipo de reglas sencillas abre nuevas posibilidades en el desarrollo de proyectos colaborativos que recreen más fielmente la realidad, especialmente para lanzar aplicaciones externas. Por ejemplo, si un sensor (o conjunto de ellos) genera un nuevo valor, entonces automáticamente se lleva a cabo la acción asociada, que podría ser actualizar una hoja de cálculo de *Google Drive* con el valor observado.

Y todo ello, por supuesto, con validaciones progresivas en el aula y participación activa en eventos y ferias de divulgación científica, puesto que los estudiantes son los principales actores que nos pueden indicar a ciencia cierta si *Sucre4Stem* facilita el aprendizaje de conceptos, agudiza su pensamiento computacional y, eventualmente, fomenta en la medida de lo posible su vocación científica.

#### AGRADECIMIENTOS

El proyecto está financiado por la Fundación Española para la Ciencia y la Tecnología (FECYT-MICCIN) (FCT-2019-15198) y el

Plan de Promoción de la Investigación en Universitat Jaume I (UJI-B2020-31). Sergio Trilles ha sido financiado por el programa Juan de la Cierva - Incorporación (IJC2018-035017-1), del Ministerio de Ciencia e Innovación de España, y Águeda Gómez-Cambronero por el programa para la contratación de personal investigador predoctoral de la Generalitat Valenciana (ACIF/2019/278).

#### REFERENCIAS

- [1] 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.
- [2] 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.
- [3] Ministerio de Ciencia, Innovación y Universidades, «Datos y cifras del Sistema Universitario Español Publicación 2018-2019», <https://bit.ly/3aZXCj>, acceso 04-05-2020.
- [4] —, «Anuario de indicadores universitarios 2018», <https://bit.ly/2VZXjtj>, acceso 04-05-2020.
- [5] V. G. Cerf, «Computer science in the curriculum», *Communications of the ACM*, vol. 59, n. 3, págs. 7–7, feb 2016.
- [6] A. V. Aho, «Computation and computational thinking», *The Computer Journal*, vol. 55, n. 7, pág. 832–835, 2012. [Online]. Available: <http://dx.doi.org/10.1093/comjnl/bxs074>
- [7] A. Balanskat y K. Engelhardt, «Computing our future Computer programming and coding Priorities, school curricula and initiatives across Europe», <http://creativecommons.org/licenses/by-sa/3.0/>, Creative Commons, Tech. Rep., 2015.
- [8] GOV.UK, «National curriculum in England: computing programmes of study», <https://bit.ly/2uzobyD>, acceso 01-05-2020.
- [9] whitehouse.gov, «Computer Science For All», <https://bit.ly/2SgEc5J>, acceso 01-05-2020.
- [10] Ministerio de Educación y Formación Profesional, «El currículo en Primaria, ESO y Bachillerato», <https://bit.ly/31JASTJ>, acceso 01-05-2020.
- [11] F. Telefónica, *Prepara tu Escuela para la Sociedad Digital: Claves para sumarse al cambio*. Fundación Telefónica, 2016.
- [12] FECYT, «Educación de las Ciencias de la Computación en España», <https://bit.ly/2UFzcsQ>, acceso 01-05-2020.
- [13] S. Trilles y C. 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*, vol. 3, págs. 303–310, 2018.
- [14] —, «Advancing preuniversity students' computational thinking skills through an educational project based on tangible elements and virtual block-based programming», *Computer Applications in Engineering Education*, 2020. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.22319>
- [15] S. Trilles, D. Tortosa y C. Granell, «La evolución del proyecto sucre4kids mediante el paradigma del internet de las cosas», *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática*, vol. 5, págs. 53–60, 2020.
- [16] S. Trilles, «La evolución del proyecto sucre4kids mediante el paradigma del internet de las cosas», <https://www.youtube.com/watch?v=Dc8CgtQiaU4>, 2020, acceso 22-10-2020.
- [17] J. Yu y R. Roque, «A survey of computational kits for young children», en *Proceedings of the 17th ACM conference on interaction design and children*, 2018, págs. 289–299.
- [18] E.-S. Katterfeldt, D. Cuartielles, D. Spikol y N. Ehrenberg, «Talkoo: A new paradigm for physical computing at school», en *Proceedings of the 15th International Conference on Interaction Design and Children*, 2016, págs. 512–517.
- [19] Seeed Grove Zero, «Blocks / Javascript editor», <https://bit.ly/3b2JKXt>, acceso 01-05-2020.
- [20] Micro:bit, «Micro:bit Educational Foundation», , acceso 01-05-2020.
- [21] Chibitronics, «Chibitronics webpage», <https://chibitronics.com/>, acceso 01-05-2020.
- [22] Circuit Playground Express, «Circuit Playground Express webpage», <https://bit.ly/3c2FDfj>, acceso 01-05-2020.
- [23] S. Trilles, C. Granell y E. Aguilar-Moreno, «Sucre4kids: tres años de experiencia en la incentivación del pensamiento computacional en edades preuniversitarias», en *TICAI 2018 TICs para el Aprendizaje de la Ingeniería ISBN 978-84-8158-838-5*, O. M. B. André Vaz Fidalgo, Ed. IEEE, 2018.
- [24] C. Anderson, *MAKERS: La nueva revolución industrial*. Empresa Activa, 2013.
- [25] J. M. Wing, «Computational thinking», *Communications of the ACM*, vol. 49, n. 3, págs. 33–35, 2006.

- [26] C. G.-B. y. F. J. S.-P. 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.
- [27] H. Ruíz Martín, *¿Cómo aprendemos? Una aproximación científica al aprendizaje y la enseñanza*. Barcelona: Graó, 2020.
- [28] Google Developers, «Blockly», <https://developers.google.com/blockly>, acceso 23-10-2020.
- [29] BlocklyDuino, «A web-based visual programming editor for arduino.» <https://bit.ly/3fjzuxp>, acceso 01-05-2020.
- [30] Particle, «Particle Argon», <https://bit.ly/31xX4RW>, acceso 01-05-2020.
- [31] U. Hunkeler, H. L. Truong y A. Stanford-Clark, «Mqtt-s—a publish/subscribe protocol for wireless sensor networks», en *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*. IEEE, 2008, págs. 791–798.
- [32] M. Guzdial, A. Kay, C. Norris y E. Soloway, «Computational thinking should just be good thinking», *Communications of the ACM*, vol. 62, n. 11, págs. 28–30, 2019. [Online]. Available: <https://dl.acm.org/doi/10.1145/3363181>



(Portugal).

**Sergio Trilles** recibió su doctorado en Integración de la Información Geoespacial de la Universitat Jaume I en 2015. Tras diferentes becas postdoctorales a diferentes niveles, actualmente es investigador postdoctoral Juan de la Cierva - Incorporación en la Universitat Jaume I. Ha realizado diferentes estancias internacionales, como en la unidad Digital Earth and Reference Data del Centro Común de Investigación (JRC) de la Comisión Europea (Italia) o en la Escuela de Gestión de la Información (NOVA IMS) de la Universidade Nova de Lisboa



**Aida Monfort-Muriach** es Ingeniera Informática y obtuvo el Máster en Tecnologías Geoespaciales en 2015 en un programa conjunto de la Universitat Jaume I (España), Westfälische Wilhelms-Universität Münster (Alemania) y Universidade Nova de Lisboa (Portugal). Desde 2016 es personal investigador asociado al grupo Geotec (UJI, España).



**Águeda Gómez-cambronero** es graduada en Diseño y Desarrollo de Videojuegos y obtuvo el Máster en Sistemas Inteligentes, ambos por la Universitat Jaume I. Actualmente realiza su doctorado, dentro del grupo Geotec, centrando su investigación en el diseño de videojuegos serios para distintos ámbitos como la psicología, la educación o el turismo.



**Carlos Granell** es Profesor Titular en la Universitat Jaume I (España). Sus intereses de investigación versan en la aplicación multidisciplinaria de los Sistemas/Ciencia de la Información Geográfica, y en la búsqueda de sinergias entre proyectos de ciencia ciudadana, educación y difusión de la ciencia.