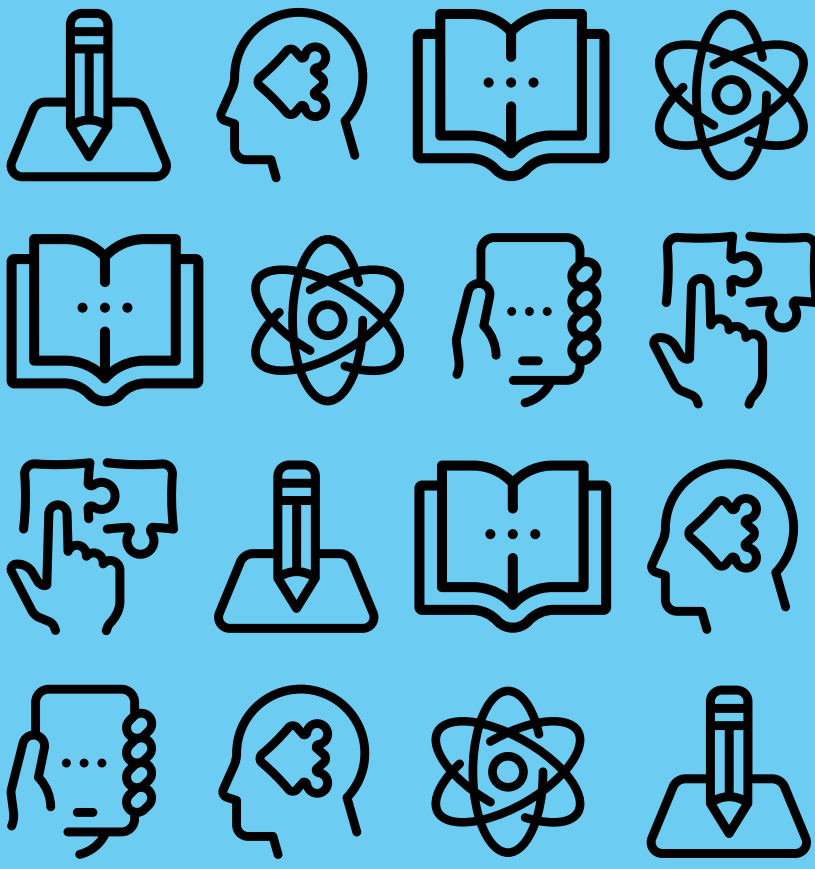
Universitat Oberta de Catalunya

Telefónica



PROGRAMAR JUGAR APRENDER

Cátedra Telefónica-UOC en Diseño y Creación Multimedia

Programar, jugar, aprender es una publicación de la **Cátedra Telefónica-UOC en Diseño y Creación Multimedia** que recoge una selección de artículos escritos por **Pau Waelder** y publicados en el blog **Design Matters** entre 2016 y 2018.

La Cátedra Telefónica-UOC en Diseño y Creación Multimedia se crea en 2015 con el objetivo de investigar, reflexionar y debatir sobre el diseño, la tecnología y la educación y su impacto en la creación digital y multimedia.

Nace con carácter innovador, entendiendo la relación entre el diseño y la tecnología como elemento transformador de la educación y la creación digital, con voluntad de impacto social y retorno del conocimiento.

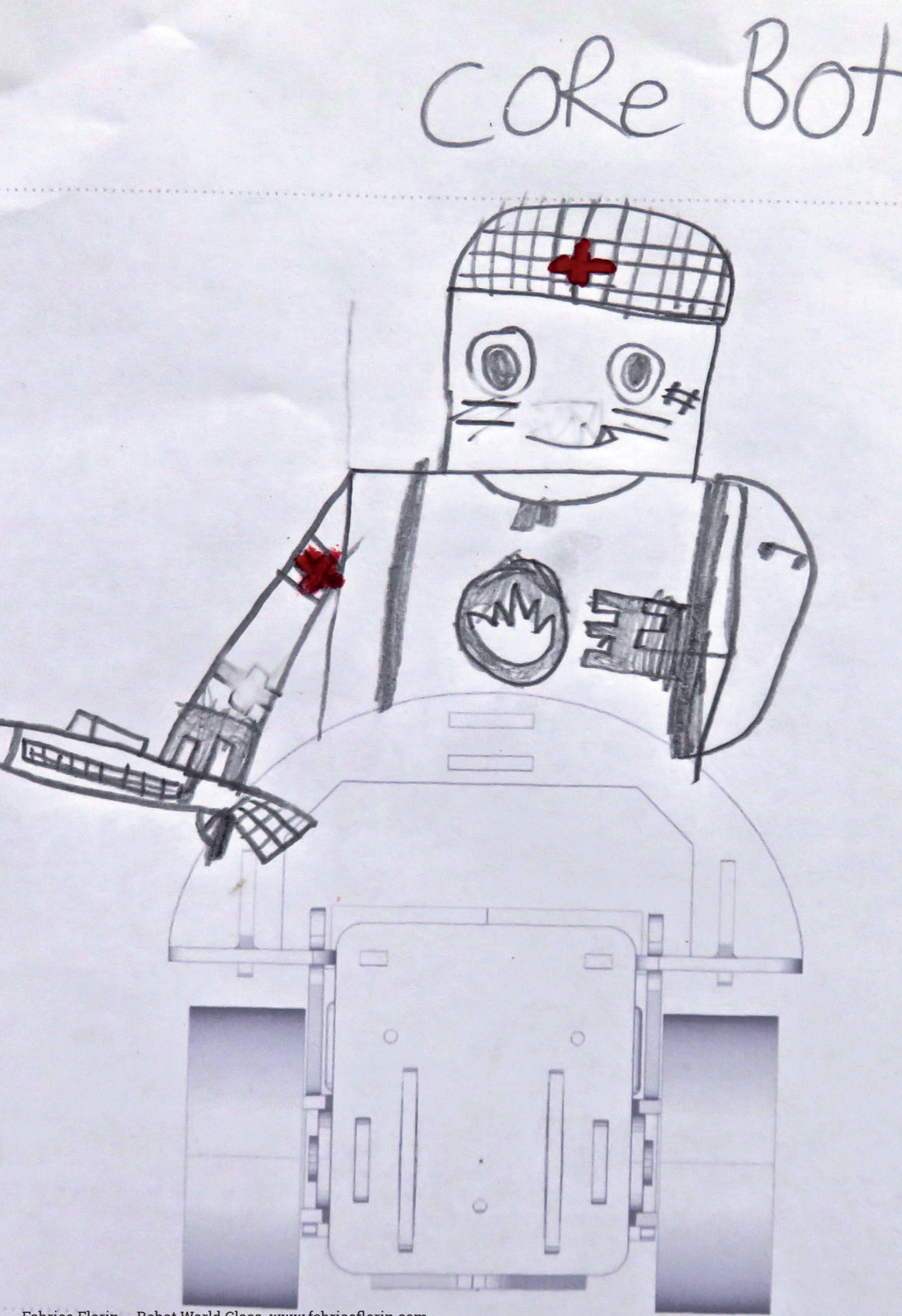
En particular, la cátedra potencia tanto actividades, conferencias y talleres como trabajos académicos solventes, de carácter docente así como de investigación, relacionados con:

- la interrelación entre tecnología y diseño y sus usos innovadores la creación multimedia y la educación,
- las metodologías y herramientas para la educación en línea en los ámbitos de diseño y creación multimedia.

Equipo de trabajo

Enric Mor (Dirección)
Irma Vilà (Coordinación)
Pau Waelder (Edición de Contenidos)

Portada: iconos de Gregor Cresnar de The Noun Project – thenounproject.com


CoRe Bot

DE STEM A STEAM

La separación entre las ciencias y las humanidades, pese a asumirse como algo natural, se da en realidad a finales del siglo XIX, coincidiendo con el inicio del Romanticismo. Esta es una separación que se concibe como algo irrevocable y genera lo que en 1959 el científico **Charles Pierce Snow** describiría como "las dos culturas" en su conferencia (y posterior libro, publicado en 1964) **The Two Cultures and the Scientific Revolution**. Snow señala que las ciencias y las humanidades establecen culturas paralelas, cada una con su lenguaje específico, lo cual hace prácticamente imposible la comunicación y el intercambio de conocimientos entre los expertos de dichos ámbitos. Esta separación entre la cultura humanística y la cultura científica se ha hecho particularmente palpable en el contexto de las universidades, donde no ha existido prácticamente ningún intercambio entre las facultades y los investigadores de las diferentes disciplinas. Con todo, ya en la segunda edición de su influyente libro, C.P. Snow sugiere la posibilidad de una "tercera cultura" en la que, a imagen del Renacimiento, se conciban las humanidades y las ciencias como parte de un conocimiento común.

Los diferentes métodos de investigación y los productos que ésta genera en cada disciplina han sido, en parte, lo que ha ensanchado la división entre ciencias y artes, puesto que a menudo los científicos han despreciado los métodos de los humanistas, mientras que estos últimos han criticado la arrogancia de quienes han pretendido resumir la complejidad del mundo a unas pocas fórmulas matemáticas. Ciertamente, la separación entre las disciplinas, la incomprensión y los prejuicios no hacen sino limitar las posibilidades de ampliar el conocimiento humano y lograr nuevas metas, además de mermar las capacidades de las personas que, sin necesidad de convertirse en expertos, pueden nutrirse de conocimientos de otros ámbitos para desarrollar su propia investigación o trayectoria profesional. Los investigadores **Robert Root-Bernstein** (Doctor en Historia de la Ciencia por la Universidad de Princeton) y **Michele Root-Bernstein** (Doctora en Historia por la Universidad de Princeton) han dedicado especial atención a las interacciones entre arte y ciencia, compaginando su

trabajo científico con diversos proyectos artísticos. En 1999 publican **Sparks of Genius, The 13 Thinking Tools of the World's Most Creative People, un influyente libro en el que plantean la importancia de la actividad artística como parte de una metodología de investigación científica** y defienden una educación que abarque y combine todas las disciplinas. Los autores parten del concepto de "pensamiento creativo" (creative thinking), indicando que las personas que trabajan en diferentes disciplinas emplean una serie de herramientas de pensamiento similares, de manera que éstas permiten establecer conexiones entre las ciencias, artes, humanidades y tecnologías. **Un aspecto que destacan es la importancia de la intuición, a menudo menospreciada en las ciencias y valorada en las artes.** Recopilando las experiencias de numerosos científicos, los Root-Bernstein afirman que éstos no piensan de manera más lógica, sino que la intuición tiene un papel esencial en sus descubrimientos: "nuestros sentimientos –nuestras intuiciones– no son impedimentos para el pensamiento racional," afirman, "sino que forman parte de su origen y fundamento." También critican la extendida concepción de la existencia de una forma de pensamiento diferente en artistas y científicos, indicando que lo que se ven como rasgos esenciales (por ejemplo, los estímulos visuales en el caso de los artistas) son sólo elementos individuales de un conjunto de herramientas de pensamiento mucho más amplio.

En Sparks of Genius, Robert y Michele Root-Bernstein dedican un capítulo a proponer **un modelo de educación basado en la síntesis de las diferentes disciplinas, que "sólo requiere cambiar cómo educamos", basándose en ocho objetivos básicos:**

1. **Enseñar procesos universales de invención** además de los productos del conocimiento de las diferentes disciplinas: los estudiantes no sólo deben estudiar las novelas, poemas, experimentos, teorías, canciones o pinturas sino imitarlas, para aprender de los procesos que llevan a su invención.

2. **Enseñar las habilidades intuitivas e imaginativas para desarrollar procesos de invención:** por ello, es preciso que los estudiantes reciban estimulación visual y de los sentidos y aprendan a trabajar con estas emociones y sensaciones corporales.

3. **Implementar una educación multidisciplinaria que sitúe las artes al mismo nivel que las ciencias:** desde el jardín de infancia, cada estudiante debería conocer tanto las artes como la ciencia, humanidades y matemáticas. Las artes no son sólo para la

expresión personal o el entretenimiento, sino que son la base de la que se nutre la imaginación, que también facilita el desarrollo de las ciencias, matemáticas y la tecnología.

4. **Emplear un mismo lenguaje descriptivo para la innovación:** no tiene sentido enseñar artes y ciencias a partir de un currículo que fragmenta el conocimiento y crea especialistas que no pueden comunicarse más allá de sus propias disciplinas.

5. **Enfatizar la transdisciplinaridad:** dar menor importancia a etiquetas como "arte", "música" o "ciencia", que colocan el conocimiento en cajas aisladas y en lugar de eso mostrar cómo los mismos contenidos pueden emplearse en diferentes disciplinas. El objetivo es que cualquiera pueda pensar simultáneamente como un artista y como un científico.

6. **Usar como ejemplo las experiencias de aquellos/as que han sabido superar las barreras entre disciplinas:** al dar un rostro humano a estos principios, los estudiantes ven que ellos también pueden crear su propia visión del futuro.

7. **Presentar las ideas en diferentes formas:** de esta manera se demuestra que para una misma idea se pueden aplicar diferentes herramientas de pensamiento y dar lugar a diferentes productos.

8. **Forjar una educación pionera,** cuyo objetivo sea formar a personas imaginativas con un amplio espectro de intereses: las personas creativas son pioneras y por tanto deben tener mentes flexibles, que no se limiten a una única disciplina.

Estas propuestas dan lugar al concepto de un nuevo modelo de educación que integre en el mismo nivel artes, humanidades, ciencia y tecnología. Actualmente, se refiere a este modelo con las siglas **STEAM** (Science, Technology, Engineering, Arts, Mathematics) puesto que añade a las disciplinas científicas (ciencia, tecnología, ingeniería, matemáticas) las humanidades (artes, diseño). Este término parte de una iniciativa de la **Rhode Island School of Design (RISD)**, que promueve el cambio de modelo educativo en instituciones, corporaciones y comunidades en Estados Unidos. Los objetivos de STEAM son básicamente integrar el arte y diseño en la formación científica, apoyar la integración del arte y diseño en la educación escolar (donde suele ser menospreciada) e influir en las empresas para que contraten a artistas y diseñadores para proyectos de innovación.

Este planteamiento coincide con el defendido por Roger F. Malina, científico y astrónomo, quien considera fundamental la interacción entre la ciencia y el arte en el contexto de una sociedad expuesta a un desarrollo tecnológico acelerado para el que no se ha creado aún un marco cultural. En la relación entre ciencia y arte, Malina propone dos casos posibles: un "caso débil", en el que la presencia de un artista en residencia en los laboratorios de investigación científica supone la introducción de una nueva visión, más creativa, que aporta nuevos puntos de vista; y un "caso fuerte", que daría lugar a un nuevo tipo de investigador, que introduce ideas y técnicas propias del arte y el diseño, lo cual lleva en última instancia a una ciencia diferente, mejor conectada con las necesidades de la sociedad. Malina ve en la red de trabajo (network) el medio de eludir la oposición entre arte y ciencia, creando un sistema pluridisciplinar. Este método de trabajo en el que se encuentran puntos de vista diversos para afrontar los diferentes enfoques de un problema es el que dará lugar a una ciencia y una tecnología "diferentes", lo que Malina ve como signo de progreso.

REFERENCIAS

Charles Pierce Snow (1964) The Two Cultures and the Scientific Revolution. https://es.wikipedia.org/wiki/Las_dos_culturas

Robert Root-Bernstein y Michele Root-Bernstein (1999) Sparks of Genius: The Thirteen Thinking Tools of the World's Most Creative People. Houghton-Mifflin
http://www.goodreads.com/book/show/627421.Sparks_of_Genius

STEM to STEAM. http://stemtosteam.org/

Rhode Island School of Design (RISD). http://www.risd.edu/

Roger Frank Malina (2004). Leonardo Timeshift, 1959, 1969, 2004, 2029
http://90.146.8.18/en/archives/festival_archive/festival_catalogs/festival_artikel.asp?iProjectID=12927

Nintendo Labo – labo.nintendo.com

Root – rootrobotics.com

APRENDER A PROGRAMAR JUGANDO

Los principios de la iniciativa STEAM se llevan a la práctica en una amplia variedad de proyectos destinados a facilitar a los más jóvenes a iniciarse en la programación por medio de diversos juguetes, robots y entornos de programación que buscar eludir el aspecto más rígido y tedioso de trabajar con código. Estos proyectos plantean otras maneras de acercarse a la programación y extraer competencias que van más allá del uso de los ordenadores. En este capítulo veremos algunos ejemplos destacados de aproximaciones creativas a las nuevas tecnologías.

SCRATCH

Scratch es un entorno y lenguaje de programación creado por **Mitchel Resnick** junto con el grupo de investigación Lifelong Kindergarten en el Media Lab de MIT en 2003 con el objetivo de ayudar a niños y adolescentes a partir de 8 años a aprender a programar. Gracias a su sencillez de uso y su enfoque en los aspectos educativos, **Scratch es ampliamente usado por estudiantes, profesores y padres para crear animaciones, juegos y programas sencillos que permiten iniciarse en el ámbito de la programación.** El software es gratuito, de código abierto y cuenta con una amplia comunidad de usuarios en la que se comparten los proyectos generados con Scratch, que pueden ser fácilmente reutilizados para crear proyectos nuevos. Creada en 2007, **ScratchEd** es la comunidad específica para educadores, que cuenta con más de 7.500 miembros que comparten sus historias, recursos y experiencias con Scratch. En 2013 se lanzó Scratch 2, que puede emplearse como una aplicación directamente en el navegador o bien descargarse como un programa independiente para Windows, OSX y Linux.

Se suele asumir que los niños y adolescentes tienen una relación natural con las nuevas tecnologías, como indica el término **"nativos digitales."** Pero Mitchel Resnick es escéptico acerca de este término y lo que implica, puesto que **en general la relación que tienen los jóvenes con la tecnología es la de usuarios, no creadores.** "No hay duda que los jóvenes tienen facilidad para consultar información en la web, chatear, enviar mensajes y jugar a videojuegos," afirma Resnick, "pero eso no implica que puedan expresarse de forma fluida con las nuevas tecnologías. Los jóvenes tienen mucha experiencia usando entornos interactivos, pero no tanta creando o expresando sus ideas a través de estas tecnologías. Dicho de otro modo, pueden leer pero no escribir con las nuevas tecnologías." Para no verse limitados a ser simples usuarios, el investigador considera que es fundamental aprender a programar, a fin de ser capaces de crear sus propios programas. La importancia de integrar el código de programación en la enseñanza básica ha sido un tema que ha generado un creciente interés en los últimos años: en países como Estonia se ha integrado en las escuelas, mientras que en otros países se han creado numerosas "escuelas de programación" en la que personas de todas las edades aprender a crear sus propios programas. El problema, en este punto, es la aridez del código de programación, que resulta muy poco atractivo a la mayoría de los usuarios y en general no les permite ver claramente su estructura lógica, ya que se muestra como una sucesión de instrucciones. En Scratch, de forma similar a **Processing**, el código se muestra en un entorno gráfico en el que el usuario trabaja con bloques que puede colocar, reordenar y vincular estableciendo el funcionamiento de un sencillo programa o una animación.

Programar en Scratch es sencillo e intuitivo, parecido a completar un puzzle con unas pocas piezas, y además la propia visualización del código permite entender el proceso que se está generando. Este puede visualizarse en una ventana integrada en el propio entorno de programación. De esta manera, resulta muy útil para niños y jóvenes puesto que evita la frustración que produce no saber qué parte del código corresponde a cada acción que se produce en la visualización final y a la vez muestra claramente las relaciones entre los diferentes elementos, lo cual permite pensar más allá de las funciones existentes e inventar nuevas combinaciones. La amplia variedad de ejemplos que se comparten en la web de Scratch demuestra la flexibilidad y el potencial creativo de este software. Además, siguiendo el principio de compartir y recombinar (en informática se denomina "scratching" a la práctica de coger partes del código de un programa para usarlo en otro programa diferente), cada proyecto que se publica en Scratch permite visualizar el código que lo ha generado, de manera que resulta sencillo

emplear una solución desarrollada por otra persona para incorporarla a un programa propio.

Si bien personas de cualquier edad pueden usar Scratch, la plataforma está pensada para niños de entre 8 y 16 años, por lo que se rodea de una comunidad controlada para que los jóvenes usuarios puedan aprender a programar, emplear código de otros programas y compartir ideas y preguntas en foros en un ambiente constructivo y respetuoso. El programa, además, permite crear código que posteriormente controle objetos físicos e interactúe con una cámara o un micrófono, lo cual facilita crear juegos y juguetes que van más allá de la pantalla y pueden compartirse con amigos. Por otra parte, **los jóvenes aprenden a través del uso de Scratch a entender conceptos matemáticos y computacionales, tales como coordenadas, variables y números aleatorios**. Dado que estos conceptos se integran en funciones concretas del juego o animación que están creando, no resultan abstractos o aburridos sino que forman parte de algo dotado de sentido. El propio proceso de crear algo en Scratch les lleva a aprender el proceso básico de diseño, que parte de una idea, convertida luego en un prototipo que es puesto en práctica, corregido cuando presenta fallos, y mejorado a medida que se adquieren conocimientos más avanzados. Este proceso también implica mostrar el prototipo a los demás, escuchar sus opiniones, revisar y rediseñar el proyecto hasta lograr el objetivo deseado. Al hacer esto, el joven programador aprende también a pensar de forma creativa, comunicarse con claridad, analizar su proyecto, trabajar en colaboración con otros y aprender de forma continuada. De esta manera, aunque los usuarios de Scratch no se conviertan necesariamente en programadores profesionales, tendrán una mejor comprensión de cómo funcionan los ordenadores y habrán adquirido otras competencias de gran utilidad en cualquier ámbito.

Para empezar a usar Scratch, basta con acceder al editor en el propio navegador web. La página muestra una sencilla interfaz con una ventana de visualización (a la izquierda), una columna con los principales bloques de código, ordenados en categorías y diferenciados con colores, y un área en la que se colocan estos bloques para crear el programa. En unos pocos minutos se puede crear una animación con el objeto suministrado (un dibujo de un gato) simplemente colocando diversas instrucciones y observando como el orden de las mismas determina la manera en que se desarrolla la animación. Además, es posible introducir eventos, con los que la animación puede controlarse por medio de determinadas teclas o el clic del ratón, y también añadir otros elementos como por ejemplo bucles que repiten continuamente (o un número determinado de veces)

todas las acciones que se sitúan dentro de dicho bucle. En este caso, puede verse gráficamente cómo se aplica esta función, puesto que la pieza que forma el bucle es una pinza que agrupa en su interior todas las acciones a las que se aplica.

Programar con Scratch es por tanto una tarea sencilla y amena, que tiene la virtud de ser un programa que no sólo facilita la introducción al código de programación, sino que incita a descubrir todas sus posibilidades y poner a trabajar la propia creatividad.

SWIFT PLAYGROUNDS

Apple creó en 2014 el lenguaje de programación **Swift**, destinado a crear apps para iOS, Mac, Apple TV y Apple Watch. La empresa californiana lanzó este recurso para facilitar a todo tipo de desarrolladores la creación de apps que enriquecen sus sistemas operativos y los dispositivos que dependen de ellos. Por ello, Swift es un lenguaje de código abierto, rápido y eficiente, pensado para ser empleado por cualquier programador en poco tiempo, que cuenta con un entorno en el que es posible visualizar en tiempo real los resultados del código a medida que éste se ejecuta. Apple ha puesto Swift a disposición de desarrolladores, profesores y estudiantes con una licencia de código abierto y binarios para OSX y Linux que permiten compilar el código para iOS, OSX, watchOS, tvOS y Linux. A fin de facilitar su uso, Apple ofrece además de forma gratuita el manual **The Swift Programming Language** en iBooks, la aplicación **Xcode** para MacOS que permite crear software para MacOS e iOS y una comunidad de desarrolladores en la que se puede obtener información avanzada, novedades y otros recursos como vídeos, guías, y código de ejemplo. De esta manera, Swift es una puerta abierta para que cualquiera desarrolle un programa compatible con los productos de Apple.

El eslabón que faltaba para completar la accesibilidad de Swift a todo tipo de creadores era un entorno dirigido a personas sin conocimientos previos de programación, que les facilite adentrarse en los principios del código de programación de manera amena y sencilla. Con este fin, Apple crea en 2016 **Swift Playgrounds**, una app gratuita para iPad. Playgrounds se presenta como un juego en el que el usuario controla a un personaje en una serie de escenarios estructurados como un tablero tridimensional dividido en casillas. Los escenarios, de una gran belleza, recuerdan en cierto modo al universo del popular juego **Monument Valley** (desarrollado por ustwo) y tienen por objetivo plantear un entorno visualmente atractivo pero también apacible, en el que el usuario

Swift Playgrounds – www.apple.com/swift/playgrounds/

observará los resultados del código que ha escrito. El personaje que habita en este mundo, un simpático cíclope llamado **Byte**, es controlado por el usuario por medio de los comandos que escribe en Swift. Siguiendo la dinámica de muchos videojuegos populares entre los más jóvenes (y el gran público), el personaje debe moverse por los escenarios para coleccionar gemas. El usuario debe entonces escribir el código, en definitiva una serie de instrucciones, que permitirá a Byte desplazarse por el mundo imaginario y atrapar una gema. Las acciones del personaje, pautadas por la estructura de casillas que domina en cada escenario, responde a las líneas de código, al menos en los comandos iniciales que se centran en movimientos básicos y permiten al usuario entender cómo cada instrucción corresponde a una acción en el entorno virtual.

Por supuesto, un aspecto importante aquí es que **el usuario debe escribir correctamente el código para que este se pueda traducir en acciones concretas**, lo cual también enseña cómo un código mal escrito conduce a errores al ser ejecutado. Para facilitar la correcta introducción del código y evitar frustraciones al usuario novel (en especial si es un niño), el programa sugiere comandos específicos que se introducen en el código al hacer clic en ellos. Con ello, también, se aprovechan las posibilidades de la pantalla táctil del iPad y se evita la desagradable experiencia de teclear durante mucho tiempo en un teclado virtual (algo que la mayoría de los usuarios evitan comprando teclados inalámbricos o integrados en la funda de la tablet). La interfaz también permite explorar el mundo virtual rotando el escenario con los dedos o haciendo zoom con los gestos habituales. Esto facilita explorar el entorno y decidir qué instrucciones se deben dar a Byte. Finalmente, la app incluye un teclado virtual QuickType para escribir código, que permite añadir comandos con unos pocos toques o bien arrastrando elementos en la pantalla.

Swift Playgrounds está pensado específicamente para el iPad tanto por la manera en que emplea la pantalla como por su planteamiento como un juego al que se puede jugar en las ocasiones en que habitualmente se emplea una tablet, momentos de relax o de espera, en un medio de transporte o en el sofá de casa. En este sentido, **Apple afirma que la app plantea una forma de aprendizaje diferente, en la que el usuario aprende a escribir código a medida que resuelve retos y progresa en las fases del juego.** La app ofrece en primer lugar una serie de lecciones (la primera de las cuales es "Los fundamentos de Swift") con las que los neófitos pueden aprender a programar, y unos retos para los usuarios más avanzados.

Más allá del juego, Swift Playgrounds permite crear proyectos nuevos con código a partir de plantillas en las que se muestra cómo utilizar

los recursos del iPad, tales como la interacción con la pantalla táctil, el acelerómetro y el giroscopio. El usuario puede añadir además sus propias imágenes y sonidos. El código creado de esta manera puede visualizarse en el iPad o también se puede compartir por email, Messages o AirDrop, así como crear un vídeo que se puede compartir en cualquier web o red social. Otro usuario de Swift Playgrounds puede luego crear una versión diferente modificando el código. Finalmente, es posible llevar el código de Swift Playgrounds a Xcode para trabajar a un nivel más profesional y desarrollar apps para iOS y Mac OS.

Con esta app, Apple enlaza la formación en programación para todo tipo de usuarios (en línea con su slogan "Todo el mundo puede programar") con la difusión de un lenguaje pensado para sus propios dispositivos. Al hacer que Swift sea el primer lenguaje de programación que aprenden muchos jóvenes y neófitos, no sólo contribuye a que un mayor número de personas adquieran la competencia de saber programar, sino que los dirige de forma natural hacia su propio ecosistema. Swift Playgrounds se suma así a otras iniciativas que van desde **Processing** a **Scratch** y juguetes como **Cubetto** en la formación de nuevas generaciones de programadores.

GOOGLE BLOCKS

A mediados de 2016, **Google** anunció un nuevo proyecto destinado a facilitar el aprendizaje de la programación entre los más jóvenes que se sumaba a otras iniciativas que lleva apoyando durante los últimos años, tales como **Blockly, Scratch Blocks**, **CS First** o **Made w/ Code**. Reconociendo la importancia de desarrollar las competencias de los niños en las nuevas tecnologías, y particularmente en el conocimiento de los lenguajes de programación, Google pone en marcha **Project Bloks,** un proyecto de investigación que desarrolla en colaboración con **Paulo Blikstein** de la Stanford University e **IDEO**, con el objetivo de crear una plataforma de hardware de código abierto y distribución libre con la que investigadores, desarrolladores y diseñadores puedan crear maneras de aprender a programar con elementos físicos. El sistema de programación tangible cuenta con un prototipo que la empresa muestra con detalle en la web del proyecto, que sirve también como documentación del desarrollo del mismo.

Bloks se basa en la facilidad que tienen los niños para manejar elementos físicos y establecer relaciones entre ellos. El sistema está compuesto por una serie de bloques que pueden conectarse entre sí, estableciendo una

serie de instrucciones que se envían por medio de la red wifi a un objeto inteligente u otro dispositivo, como por ejemplo una tablet. Pensado como un sistema modular y adaptable, Bloks presenta tres tipos de componentes principales:

- **Placa principal (Brain Board):** se encarga de procesar las instrucciones que recibe de los otros componentes y las envía al objeto inteligente por medio de una red wifi o por Bluetooth.
- **Placa base (Base Board):** cada uno de los "bloques" que se conectan entre sí y envían las instrucciones a la placa principal.
- **Disco (Puck):** controlador que se coloca sobre una placa base para determinar una instrucción. Pueden ser de diferentes tipos, según estén programados para proporcionar una instrucción concreta o facilitar determinada interacción (por ejemplo, encender/apagar, mover hacia la izquierda/derecha, reproducir música, etc.).

Las placas pueden colocarse en diferentes configuraciones (tanto de posición como de número de elementos) y disponer de distintos tipos de controladores para generar experiencias muy diversas. El equipo de investigadores prevé la posibilidad de insertar estas placas en un envoltorio creado expresamente para hacer del sistema un dispositivo independiente, como por ejemplo un juguete con diversos controles que permita crear música y reproducirla usando un altavoz inalámbrico.

Google piensa por tanto en la posibilidad de desarrollar proyectos para educación pero también en nuevos productos que despierten en los niños tanto el conocimiento de los conceptos básicos de la programación como el interés por trabajar con código y pensar de forma creativa con nuevas tecnologías.

Completado en 2018, Bloks resulta una interesante aportación a las iniciativas de educación en programación para niños, no tanto por el sistema en sí sino por el hecho de ser una plataforma de distribución libre que cuenta con los recursos de Google en su fase de investigación y permite a educadores, investigadores y diseñadores crear numerosas aplicaciones y productos destinados a difundir la programación creativa.

A entornos de programación como **Scratch**, **Swift Playgrounds** y **Bloks** se suman algunos juguetes que, partiendo de este mismo planteamiento, hacen de la programación un juego e introducen a los más jóvenes en la lógica del código por medio de interfaces sencillas e intuitivas. Como hemos comentado anteriormente, saber programar (o al menos comprender cómo funciona el código de programación)

Project Bloks – projectbloks.withgoogle.com

es una competencia cada vez más necesaria, y por ello se multiplican las iniciativas destinadas a enseñar programación creativa a todos los niveles y con diferentes enfoques. A ello se suman las habilidades que desarrollan los niños mientras aprenden a programar, y que como hemos comentado incluyen la capacidad de trabajar en colaboración con otros, pensar de manera creativa y resolver problemas aplicando diferentes soluciones. Estos son algunos de los juguetes que enseñan a los niños a entender el código de programación.

Dash and Dot

Los robots **Dash y Dot**, creados por la empresa **Wonder Workshop**, son unos simpáticos juguetes pensados no para el simple entretenimiento sino como un completo proyecto educativo para niños a partir de 8 años. Dash es un robot móvil, que cuenta con ruedas y sensores con los que puede moverse en todas direcciones y percibir su entorno. Dot es un robot fijo dotado con un micrófono, luces y un acelerómetro con el que puede emitir respuestas si es agitado o desplazado. Ambos robots se comunican entre sí, de manera que Dot es el "cerebro" y Dash el "músculo" en su interacción con el entorno. Los niños pueden programar los robots empleando el entorno de programación **Google Blockly**, que funciona de forma similar a **Scratch**.

Codeybot

El robot **Codeybot** es la propuesta de **Makeblock**, una startup situada en Shenzhen (China), para enseñar a los niños a programar jugando. Con un diseño sencillo, este juguete se centra en su capacidad para moverse en todas direcciones mientras se mantiene erguido gracias a un sistema de equilibrio interno y a expresarse a través de una pantalla de LEDs en la que se muestran diversas expresiones faciales a modo de emoticonos. Según anuncia la empresa, el robot también puede bailar, reproducir música, grabar y reproducir voz y finalmente "disparar rayos láser" con un accesorio adicional, de manera que dos robots pueden interacturar en un "modo de batalla". Una app para iPhone y iPad permite al niño programar el robot con el entorno **mBlockly**, basado en Google Blockly, que facilita crear sencillos programas arrastrando bloques de instrucciones. Además de esta app, Codeybot también incluye una app de control remoto que permite manejar el robot apretando una serie de botones. Lanzado en una campaña de crowdfunding, el robot se anuncia como "tecnología increíble a un precio asequible" y se comercializa a través de su sitio web.

Goldie Blox

Tras licenciarse en Princeton, la ingeniera **Debbie Sterling** decidió dedicarse a crear un juguete que animase a las niñas a desarrollar sus habilidades como futuras ingenieras y no limitarse al papel de princesas de cuento. Así nació Goldie Blox, un proyecto basado en una línea de juguetes de "chicas inventoras" que rompen con estereotipos e invitan a las niñas a desarrollar sus habilidades mientras crean historias con las muñecas y un surtido de juegos de construcción. Si bien este juguete no se centra específicamente en la programación con código, trabaja con las mismas competencias de una manera lúdica.

Cubetto

La startup **Primo Toys** lanzó **Cubetto** en una campaña de crowdfunding y actualmente comercializa este robot programable que tiene la particularidad de no necesitar ninguna pantalla. Cubetto se basa en tres elementos: el robot en sí, un cubo de madera que puede moverse en todas direcciones, un panel en el que los niños programan las acciones de Cubetto con fichas de colores y un "mapa del mundo" en el que el robot se desplaza y que facilita crear historias basadas en los movimientos del juguete. Sus creadores han buscado una estética amigable que se aleja de lo tecnológico (muy presente en otros robots como Codeybot) y evita que los niños tengan que emplear tablets o smartphones. Los bloques de colores representan diferentes acciones que puede ejecutar el robot (moverse en las cuatro direcciones cardinales), así como otros operadores que permiten completar un programa básico. El panel de madera en el que se colocan las fichas describe una secuencia que se envía al robot al apretar un botón. De esta manera, los niños pueden aprender a crear conjuntos de instrucciones y entender cómo su orden afecta a las acciones del robot. Aprobado por **Montessori**, este juguete introduce la posibilidad de aprender código sin pantallas y se dirige a niños de a partir de 3 años. Si bien esto implica unas ciertas limitaciones en sus posibilidades, no deja de ser una interesante manera de iniciar a los niños en la programación.

Kano

Dirigido a un público en edad escolar (de 6 a 12 años), Kano es un kit con el que los niños pueden construir su propio ordenador y aprender a programar con él. Compuesto por un teclado inalámbrico y una placa Raspberry Pi, el kit está pensado para animar al usuario a conocer el funcionamiento interno de la máquina en lugar de verla como una "caja negra" cuyas opciones han sido previamente delimitadas por

el fabricante (la caja que contiene el procesador es transparente y es preciso montarla). Con todo, no es preciso que los niños creen su propio sistema operativo: Kano incluye un software pre-instalado por medio del cual el usuario aprende a manejar los lenguajes de programación Python y Javascript jugando a **Minecraft** y **Pong** o bien se adentra en el entorno Linux con los juegos **Terminal Quest** y **Snake**.

Coding Awbie

Parte de una serie de juegos desarrollados por la empresa **Osmo**, este juguete se usa en combinación con un iPad para enseñar a programar jugando. En este caso, un conjunto de piezas de plástico que pueden conectarse entre sí y cuentan con elementos móviles (a la manera de Project Bloks) se disponen frente a la pantalla del iPad, colocado en un soporte especial que redirige la cámara de la tablet hacia abajo y permite a un programa creado por Osmo identificar las piezas que se han colocado e interpretarlas como un conjunto de instrucciones, por medio de las cuales se controla a un personaje que tiene que cumplir una serie de misiones en un juego. Esta forma de visualizar los resultados del código es muy similar a la que plantea Swift Playgrounds, si bien en este caso no es preciso escribir los comandos sino que basta con colocar las piezas una junto a otra y apretar un botón en una pieza que "ejecuta" el código. El juguete está pensado para niños y niñas de 5 a 12 años, y por ello simplifica al máximo la interacción, dejando oculto el código.

Root

Desarrollado por la empresa **Scansorial** a partir de una campaña de crowdfunding, **Root** es un robot diseñado para que cualquier persona, niño o adulto, aprenda a programar de una manera lúdica e intuitiva. Con un aspecto similar al de un aspirador **Roomba** (los creadores trabajaban anteriormente para **iRobot**), pero de tamaño más reducido, este robot se suma a la creciente gama de juguetes pensados para enseñar a programar a los más pequeños (y no tan pequeños). Una de sus características más notables es la inclusión de un soporte para un rotulador, gracias al cual el robot puede dibujar sobre una hoja de papel o también en una pizarra magnética, puesto que puede desplazarse por cualquier superficie vertical de metal gracias a unos imanes en su parte inferior. De esta manera, el robot no sólo puede moverse según las instrucciones que reciba, sino que también puede crear dibujos e interactuar con ellos.

Como otros juguetes dedicados a la programación creativa, Root se controla por medio de una app para tablet en la que el usuario introduce

las instrucciones que debe seguir el robot en forma de código de programación. Aquí introduce la particularidad de contar con tres niveles de programación, empezando con una interfaz sencilla en la que cada instrucción es un icono que se coloca en una sucesión para establecer las acciones que llevará a cabo la máquina. En el segundo nivel, la interfaz de programación permite introducir comandos más complejos y variables, siempre empleando bloques de texto, similares a los del entorno **Google Blockly**. El tercer nivel presenta ya una interfaz de texto en la que el usuario maneja lenguajes profesionales como **Python**, **JavaScript** y **Swift**. Según sus creadores, esto hace de Root un robot útil para cualquier edad, no sólo para usarlo unas semanas sino para aprender código durante años.

Pese a su reducido tamaño y peso, Root contiene numerosos sensores que le permiten interactuar de diversas maneras con su entorno, ya sea por simple contacto físico (al chocar con un objeto) o bien "leyendo" las marcas y colores en la superficie sobre la que se desplaza. Así, el robot puede desplazarse siguiendo una línea, o bien realizar una acción diferente en función de los colores que vea. También puede reproducir música y emitir diferentes señales lumínicas combinando los LEDs alojados en su parte superior, dibujar empleando diferentes tipos de rotuladores y borrar lo que ha dibujado (si se usa en una pizarra magnética). Más allá de sus propios sensores (que detectan superficies magnéticas, la posición del aparato y la luz ambiente), Root puede programarse para reaccionar a los inputs detectados por los sensores de un smartphone o tablet.

El robot es por tanto fácilmente expandible y puede adaptarse a instrucciones de diferentes niveles de complejidad. Además, es posible compartir los programas creados con la app, lo cual genera una comunidad de programadores que pueden aportar nuevas funciones. Root se puede convertir así en algo más que un juguete: tiene el potencial de convertirse en un dispositivo de referencia para introducir la programación y la robótica en un amplio espectro de usuarios y ser un elemento clave para proyectos educativos de todo tipo.

Estos y otros productos actualmente comercializados por empresas de varios países ejemplifican una nueva generación de juguetes que van más allá del mero entretenimiento o bien de fidelizar a los niños con una determinada franquicia. Con una clara intención educativa, llevan a los niños a adentrarse en el ámbito de la programación creativa jugando, de manera que su relación con la tecnología se hace más fluida y natural.

ThingMaker – Mattel.com

JUGUETES PARA UNA CREATIVIDAD ACOTADA

Los ejemplos anteriores nos muestran las posibilidades de emplear las nuevas tecnologías en un contexto lúdico con fines educativos. Algunos de los juguetes pueden ser más o menos efectivos en su intención de enseñar a los más pequeños a programar, pero sin duda ofrecen herramientas y soluciones para este fin. A la vez, responden a un creciente interés por integrar las competencias vinculadas a la programación en la formación de niños y niñas en un contexto de creatividad que responde a los principios de STEAM.

Sin embargo, no todas las iniciativas que se acercan a esta corriente o a las ideas del movimiento maker tienen por objetivo desarrollar realmente la creatividad de los jóvenes usuarios. **En ocasiones, bajo la apariencia de una propuesta "hazlo-tú-mismo" se esconde una muy delimitada customización del producto bajo unos parámetros determinados por el fabricante.** Esto da como resultado un juguete que parece estimular la creación individual, pero en definitiva sólo hace del usuario parte del proceso de fabricación, de forma similar a como los clientes de IKEA montan sus propios muebles. A continuación veremos dos juguetes creados por grandes empresas que ejemplifican esta versión acotada de la creatividad que se ofrece al usuario.

ThingMaker

La multinacional Mattel viene desarrollando desde los años 60 un juguete que se ha popularizado como ThingMaker y también Creepy

Crawlers. El primer modelo, que se empezó a vender en 1964, consistía en una placa de metal que se calentaba con una resistencia eléctrica, una serie de moldes de metal, una cubeta de plástico y varios botes de una sustancia química de colores. El juego consistía en escoger uno de los moldes (con forma de flores o insectos, de ahí el nombre Creepy Crawlers) y verter en su interior una combinación del líquido de colores. El molde se colocaba en la placa de metal, que se calentaba hasta casi 200ºC. El calor solidificaba la sustancia, convirtiéndola en un plástico blando pero resistente. Una vez completado el proceso, el molde se tiraba en la cubeta llena de agua para enfriarlo y sacar la figura que se había creado. Mattel siguió desarrollando esta idea con otros métodos y materiales como plastilina y láminas de plástico. También era posible comprar diferentes juegos de moldes (con forma de soldados, dragones, etc.) para crear nuevos juguetes.

En 2016, Mattel anunció el lanzamiento de un nuevo ThingMaker, esta vez como una impresora 3D de bajo coste (300 dólares) pensada para que los niños puedan imprimir sus propios juguetes. Siguiendo con el principio de las versiones anteriores de este juguete, se trata de una máquina que produce pequeñas figuras dentro de un rango establecido por el fabricante. Empleando una app para tablet, los niños pueden crear combinaciones de piezas de un amplio catálogo y enviar dicha combinación a la impresora. De esta manera, se genera una gran variedad de pequeños muñecos en diferentes colores personalizados por los propios consumidores, que no necesitan conocer el funcionamiento de la máquina (descrita por Mattel como "family friendly") ni tampoco la manera de modelar objetos en 3D.

Este lanzamiento se anuncia en un momento de gran popularidad de la impresión 3D, que parece finalmente entrar en los hogares. Cabe tener en cuenta que, si bien la fabricación por adición se viene desarrollando desde los años 80, no es hasta principios de la década de 2000 cuando se empieza a popularizar la impresión 3D con la venta de impresoras 3D que permiten construir objetos a partir de modelos tridimensionales creados por ordenador. El uso de las impresoras 3D se ha ido ampliando a lo largo de la última década, particularmente en ámbitos como el diseño industrial, ingeniería, arquitectura y numerosos sectores de la industria, para crear prototipos y moldes de manera más rápida y precisa. Junto a este uso en grandes empresas, se ha dado una acelerada difusión de la impresión 3D entre artistas, diseñadores y aficionados de todo tipo, pasando de los centros de investigación e universidades a FabLabs y estudios de profesionales independientes. En marzo de 2009, **Greg J. Smith** hacía balance de los

usos creativos de la impresión 3D en un artículo publicado en Rhizome bajo el título **"Means of Production: Fabbing and Digital Art"** y afirmaba que las impresoras 3D eran ya más baratas de lo que eran las impresoras láser en 1985. Entre las iniciativas que contribuyen a lo que el economista **Jeremy Rifkin** considera como la **"tercera revolución industrial"** destaca ya en 2004 **RepRap**, un proyecto open source que facilita toda la información necesaria para que una persona con ciertos conocimientos de electrónica pueda construir su propia impresora 3D y fabricar objetos de pequeñas dimensiones. **Zach Smith**, uno de los impulsores de RepRap, fundó junto con dos socios en 2009 la empresa **MakerBot Industries**, que se dedica a desarrollar sus propias máquinas y complementos y ha logrado situarse como una de las empresas más destacadas en el mercado.

Con todo, la impresión 3D se ha mantenido hasta ahora principalmente en manos de especialistas y algunos aficionados. Una notable excepción es la que ejemplifica el **Fab Lab** de **Ars Electrónica**, creado en 2009, en el que cualquier visitante puede emplear la impresora 3D y crear sus propios objetos. Según señala **Alina Sauter**, directora del Fab Lab, pese a que existen numerosos espacios como el de Ars Electrónica, la mayoría están limitan su acceso a un grupo reducido de individuos con una formación especializada en el uso de esta tecnología. A esta dificultad de añade el hecho de tener que trabajar previamente con un modelo simulado en 3D en el ordenador, algo que generalmente implica una mayor complejidad que el trabajo con fotografías o diseño gráfico para la mayoría de usuarios. Recursos como **Thingiverse** facilitan trabajar con modelos ya creados que se pueden descargar e imprimir, pero también requieren ciertos conocimientos.

ThingMaker parece anunciar una nueva fase en la que la impresión 3D va a entrar en los hogares, bajo la habitual fórmula de limitar las opciones disponibles a cambio de disponer de un producto de uso sencillo y un ecosistema propio que asegura al fabricante la fidelidad del consumidor y el uso de la máquina dentro de unos parámetros previamente establecidos. **No obstante, esta "tercera revolución industrial" ha quedado en suspenso, al menos en lo que respecta a Mattel, que ha ido retrasando la fecha de lanzamiento de su impresora 3D y finalmente parece haberla descartado**. El juguete de Mattel, fiel a la línea planteada desde los años 60, no aprovecha el potencial educativo del producto sino que lo limita a la mera fabricación de modelos preexistentes y no estimula la creatividad de los niños, que con otro sistema podrían convertir cualquier cosa que se les ocurra en un (diminuto) objeto tridimensional.

Nintendo Labo

Nintendo lanzó a principios de 2018 **Nintendo Labo**, un kit que amplía las maneras en que se puede jugar con su popular consola portátil **Nintendo Switch**. Diseñada en base a tres elementos (la pantalla con el procesador y dos mandos que se enganchan a ambos lados), Switch se ha concebido ya como un dispositivo modular, que permite emplear los mandos de diferentes maneras, por ejemplo como controladores individuales que permiten interactuar con el videojuego por medio de movimientos del brazo o la mano, algo que ya introdujo la empresa hace más de una década con la consola **Wii**. Al igual que otros sistemas de entretenimiento, Switch cuenta con diversos accesorios de plástico (por ejemplo, un volante) para facilitar diversas formas de juego. La novedad que introduce Labo es que los accesorios debe construirlos el propio usuario a partir de una serie de planchas de cartón en las que las piezas de cada elemento vienen listas para montar, sin necesidad de tijeras o pegamento.

Nintendo lanza este nuevo producto con dos kits, uno de los cuales (Variety Kit) incluye 5 juegos diferentes, mientras el otro (Robot Kit) permite construir un artefacto que un niño puede llevar para controlar con sus movimientos un robot en un videojuego. Además de las piezas de cartón, cada kit incluye software específico diseñado para los juegos que se pueden crear con dichas piezas, lo cual en parte justifica el precio de venta (entre 56 y 65€) y además completa la experiencia del usuario, ofreciendo unos contenidos interactivos pensados para cada uno de los elementos que se pueden construir. Así, por ejemplo, el Variety Kit incluye todos los elementos para construir dos coches de carreras, una caña de pescar, una casa, una moto y un piano, empleando piezas de cartón, cuerdas, elásticos, esponjas y adhesivos reflectantes. Es preciso montar cada elemento siguiendo unas precisas instrucciones que se ofrecen en la pantalla de la consola, una vez instalado el software correspondiente. Una vez ensambladas las piezas, la pantalla y los controladores se insertan en los huecos asignados para hacer posible la interacción por medio del artefacto de cartón. En este punto, el software de Nintendo Labo aprovecha los sensores de los controladores (que incluyen una cámara capaz de detectar la forma, posición y movimiento de cualquier objeto) para lograr que, por ejemplo, al apretar una tecla del piano de cartón suene una nota. Los cartones vienen impresos en un sólo color, lo cual abarata los costes para la empresa y a la vez sirve para ofrecer la posibilidad de personalizar los juguetes pintando los cartones o decorándolos con pegatinas (para esto último, Nintendo también comercializa un kit de personalización).

La apuesta de Nintendo claramente busca atraer a niños y padres con

la promesa de algo que es más que mero entretenimiento, un juguete educativo a la manera de los que se han ido desarrollando estos últimos años para enseñar a los niños a programar. **El uso de cartón, la estética hazlo-tú-mismo y el propio nombre del producto llevan a pensar en creatividad, experimentación e innovación, si bien no deja de tratarse de un juguete diseñado para una función muy concreta y que no deja espacio a la improvisación:** el niño crea su juguete, sin duda, y al hacerlo aprende cómo funciona, pero también lo hace siguiendo unas instrucciones precisas para finalmente jugar a un único juego con el artefacto que ha ensamblado (no inventado o creado). Incluso la personalización de la estructura de cartón puede verse limitada a los elementos que comercializa la empresa. En este sentido, Nintendo Lab es muy parecido a la impresora 3D ThingMaker. En ambos casos, se ofrece una posibilidad de "crear" que en realidad se ve limitada a producir o ensamblar a partir de unos modelos concretos. Los juguetes que se crean con Nintendo Lab, además, resultan más frágiles puesto que están hechos con cartón e hilos, lo cual les augura una vida útil reducida en manos de los más pequeños y si bien se trata de materiales que se pueden reemplazar a bajo coste, es probable que el resultado final sea una línea de juguetes que al cabo de poco tiempo se tirarán para comprar otros nuevos.

Un aspecto positivo de este producto es que muestra las maneras en que se pueden emplear los sensores de los dispositivos, pudiendo inspirar así otros usos. Con todo, el resultado final de la interacción viene siempre mediado por lo que se muestra en la pantalla de la consola, lo cual está controlado por el software propiedad de Nintendo. Incluso si la empresa facilita una herramienta de creación de nuevos juegos, como parece sugerir en su sitio web, la creación y experimentación vuelve a limitarse al circuito cerrado creado por Nintendo. **Labo no es, por tanto, un laboratorio si no más bien una cadena de producción en la que el propio consumidor es quien ocupa el último eslabón, ensamblando el producto para poder disfrutarlo.**

Dominic Wilcox. Inventors Project – dominicwilcox.com

PEQUEÑOS INVENTORES

En 2015, el diseñador **Dominic Wilcox** llevó a cabo el proyecto **Inventors!,** que consistió en pedir a 450 niños de la ciudad de Sunderland (Reino Unido) que le presenten sus inventos para mejorar la vida cotidiana y posteriormente hacer realidad algunos de ellos con la ayuda de artesanos y fabricantes locales. El proyecto tenía por objetivo hacer que los niños sean conscientes del poder de su imaginación y puedan desarrollar ideas innovadoras. **Wilcox llevó a cabo diecinueve talleres de formación para niños entre 4 y 12 años, en los que les mostraba sus propios diseños e invenciones y les animaba a pensar en sus propias ideas y describirlas en un papel.** Las ideas podían ser funcionales o alocadas, siempre partiendo de un problema que hayan detectado en su entorno y que pueda resolverse por medio de un invento.

Los talleres dieron como resultado más de 600 dibujos con inventos de todo tipo, entre los cuales Wilcox escogió 60 que consideró tenían un gran potencial. Tras mantener reuniones con fabricantes y artesanos locales interesados en participar en el proyecto, se acordaron los diseños que se harían realidad. Los jóvenes inventores participaron en este proceso, hablando con los fabricantes para explicarles detalladamente sus ideas. Los prototipos se realizaron a lo largo de 4 semanas y fueron presentados en un local vacío de una céntrica calle de Sunderland entre el 16 y el 30 de enero de 2016.

Entre los prototipos que expusieron se encontraban: un patinete familiar, un gancho para extraer las patatas Pringles de su envase, una lámpara-persiana veneciana, un tenedor que enfría la comida, un cepillo de dientes con pasta incorporada y unas gafas para mirar hacia atrás. Los inventos de los niños demuestran que una mente abierta a descubrir y probar cosas nuevas puede generar interesantes ideas. Algunas, en su candidez, resultan elocuentes denuncias del absurdo mundo de los adultos, como

el "evitador de guerras", una plataforma que eleva una casa familiar varios metros por encima del suelo y proyecta una cúpula transparente a prueba de balas. El proyecto tiene tanto éxito, que poco después de la primera exposición se crea la plataforma web **Little Inventors,** que invita a los más jóvenes a enviar sus ideas y desarrolla algunos de los prototipos con la ayuda de una comunidad de makers.

Inventors! surge de la colaboración entre Wilcox y el colectivo STEAM.co, durante la cual el diseñador se da cuenta del potencial de innovación real que hay en las invenciones de los niños, y de hecho es una clara aplicación de los principios de la iniciativa STEAM y las propuestas descritas por Robert y Michele Root-Bernstein en su libro Sparks of Genius. Los niños tienen la ventaja de no haber sido aún formados en la división entre las artes y las ciencias ni en los rigurosos métodos del proceso científico y el desarrollo tecnológico. Por ello, son libres de crear cualquier cosa sin ataduras ni prejuicios, aplicando de forma intuitiva sus propias soluciones a problemas que han detectado en su entorno y aspectos del mismo que pueden mejorarse. **El hecho de pensar fuera de todo método o disciplina les permite en ocasiones plantear ideas que reflejan una perspectiva inusual y pueden por ello generar soluciones innovadoras.** Al mismo tiempo, al tomar las propuestas más factibles y llevarlas a la práctica con la colaboración de fabricantes expertos se produce un doble efecto positivo: por una parte, los niños ven que sus ideas se toman en serio y participan del proceso de producción, aprendiendo así cómo se lleva una idea a la práctica y qué retos plantea; por otra parte, los profesionales que intervienen en la manufactura del diseño se ven enfrentados a ideas nuevas y que en principio podrían rechazarse como absurdas, pero plantean nuevos retos y les llevan a buscar soluciones que posiblemente no se habrían planteado de otro modo. Unos y otros se ven así involucrados en un juego. Un juego en el que todos ganan.

REFERENCIAS

APRENDER A PROGRAMAR JUGANDO

Scratch
https://scratch.mit.edu/

ScratchEd
http://scratched.gse.harvard.edu/

Processing
http://processing.org/

Swift
http://www.apple.com/swift/

The Swift Programming Language
https://itunes.apple.com/us/book-series/swift-programming-series/

Xcode
https://itunes.apple.com/us/app/xcode/

Comunidad de desarrolladores de Swift
https://developer.apple.com/swift/

Swift Playgrounds
http://www.apple.com/swift/playgrounds/

Monument Valley
http://www.monumentvalleygame.com/

Apple: Todo el mundo puede programar
http://www.apple.com/education/everyone-can-code/

Blockly
https://developers.google.com/blockly/

Scratch Blocks
https://developers.googleblog.com/2016/05/scratch-and-google-introduce-scratch-blocks.html

CS First
https://www.cs-first.com/

Made w/ Code
https://www.madewithcode.com/

Project Bloks
https://g.co/projectbloks/

Paulo Blikstein
https://tltl.stanford.edu/people/paulo-blikstein/

IDEO
http://www.ideo.com/

Wonder Workshop | Dash and Dot
https://www.makewonder.com/

Makeblock | Codeybot
http://www.codeybot.com/

Debbie Sterling | Goldie Blox
http://www.goldieblox.com/

Primo Toys | Cubetto
https://www.primotoys.com/

Kano
https://kano.me/us

Osmo | Coding Awbie
https://www.playosmo.com/en/coding/

Root Robotics | Root
https://rootrobotics.com/

JUGUETES PARA UNA CREATIVIDAD ACOTADA

Mattel | ThingMaker
http://thingmaker.com/printer [eliminado]

Greg J. Smith. "Means of Production: Fabbing and Digital Art"
http://rhizome.org/editorial/2009/mar/4/means-of-production-fabbing-and-digital-art/

RepRap
http://reprap.org/wiki/WebHome

MakerBot Industries
http://www.makerbot.com/

Ars Electronica Fab Lab
http://www.aec.at/center/en/ausstellungen/new-views-of-humankind/

Thingiverse
http://www.thingiverse.com

Nintendo Labo
https://labo.nintendo.com/

Nintendo Switch
https://www.nintendo.com/switch/

PEQUEÑOS INVENTORES

Dominic Wilcox
http://dominicwilcox.com/

Inventors!
http://inventorsproject.co.uk/

Little Inventors
https://www.littleinventors.org


Universitat Oberta
de Catalunya


Telefónica