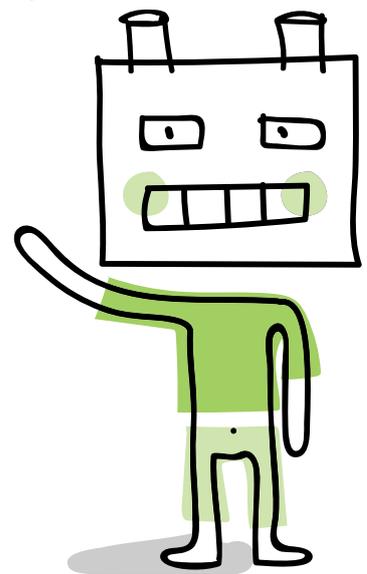


SET VEINTI UNO

GUÍA DE PROGRAMACIÓN DE LAS COSAS



Guía de Programación de las Cosas

¿En qué consiste la propuesta pedagógica?	5
¿Qué necesito para llevar a cabo cualquier proyecto?	9
Conocimientos previos	9
Requisitos técnicos mínimos	12
Instalación de entornos de programación	14
Materiales para el desarrollo de los proyectos	14
Definición de un proyecto	15
Temporalización	15
Estructura	15
• Perfil alumno	15
• Perfil docente	15
Entorno web	24
Recursos iconográficos	25
Implementación en el aula	26
Lenguajes de programación en la propuesta pedagógica	28
Bitbloq	28
Scratch	29
Tinkercad	31
Slic3r	32
OpenSCAD	32
FreeCAD	34
Python	34
Arduino	37
Otras cuestiones	38
De naturaleza legal	38
Consultas y preguntas más frecuentes	38
Anexo	41

¿En qué consiste la propuesta pedagógica?



Programación de las Cosas reúne contenidos de programación, robótica, diseño e impresión 3D dirigidos a alumnos de 10 a 16 años. La propuesta pedagógica se articula en proyectos asociados a tres temáticas genéricas diferentes, especialmente motivadoras, para dichas edades:

- **Ocio y tiempo libre** (10-12 años)
- **En busca de las estrellas** (12-14 años)
- **Convierte en Smart** (14-16 años)

Se trata de un catálogo original de proyectos con enfoque interdisciplinar (ver tabla representativa en el **Anexo**), cuya modularidad ofrece experimentar desde aprendizajes en tres dimensiones a desafíos de destreza manual.

La elaboración del plan editorial de *Programación de las Cosas* presta especial atención tanto a trabajar en cada proyecto distintas competencias y habilidades que se requieren en el proceso de enseñanza y aprendizaje del siglo XXI (la creatividad, la resolución de problemas y las habilidades para el trabajo en equipo) como a trabajar de forma transversal distintas áreas STEAM, y no solo que los alumnos aprendan a resolver los retos propuestos, sino que, posteriormente, sean ellos mismos los que creen sus propios proyectos.

Se trata de contenidos cuyo diseño gráfico y extensión huyen de un formato “muy de líneas de código”, que se estima poco motivador, procurando también incidir más en la profundidad de la propuesta pedagógica y asentar, en los alumnos, los pilares de una nueva forma de pensar frente a problemas y sistemas físicos del mundo real.

Con estos proyectos, se ha puesto especial énfasis en insertar referencias a la realidad, a películas y a personajes sugerentes para las edades de los alumnos, primando un tono divertido, innovador y realista (con ejemplos de aspectos vinculados a la diversidad, etc.).

La siguiente tabla relaciona los proyectos dirigidos a distintas franjas de edad con el alcance previsto para cada proyecto:

TIPOLOGÍA DE PROYECTOS			
Rango de edad	10-12	12-14	14-16
4 paradas	3	0	0
6 paradas	5	3	1
8 paradas	3	2	5
10 paradas	0	2	2

Así mismo, la propuesta de proyectos versa sobre distintas áreas de trabajo: programación, robótica y diseño e impresión 3D, trabajándose en todas ellas diferentes entornos de programación.

Proyecto	Edad	N.º paradas	TEMÁTICA				SW
			Programación	Robótica	Diseño 3D	Impresión 3D	Entornos de programación
¿Existieron los dragones?	10-12 años	4	●				Scratch
¿Cuál sería mi mascota ideal?	10-12 años	4	●				Scratch
¿Mejor solo que acompañado?	10-12 años	6	●				Scratch
¿Me ayudas a salvar el mundo?	10-12 años	8	●				Scratch
¿Por qué no ir a más velocidad?	10-12 años	4	●	●			Bitbloq
¿Podemos ser directores de cine?	10-12 años	6	●	●			Bitbloq
¿A qué jugamos ahora?	10-12 años	6	●	●			Bitbloq
¿Podemos crear una presa hidroeléctrica reversible?	10-12 años	6	●	●			Bitbloq
¿Cuál fue el primer deporte de la historia?	10-12 años	8	●	●	●	●	Bitbloq/ Tinkercad/ Slic3r
¿Somos capaces de distinguir el sonido de una nota?	10-12 años	6			●		Tinkercad
¿Podemos cruzar la calle, agente Robi?	10-12 años	8			●	●	Tinkercad/ Slic3r
¿Hay día y noche en el espacio?	12-14 años	6	●	●			Bitbloq
¿Cómo nos comunicamos?	12-14 años	6	●				Scratch
¿Se puede ensuciar el espacio?	12-14 años	8	●				Scratch
¿Queremos vivir en el espacio exterior?	12-14 años	10	●				Python
¿Será imposible colonizar otros planetas?	12-14 años	10	●	●			Bitbloq

Proyecto	Edad	N.º paradas	TEMÁTICA				SW
			Programación	Robótica	Diseño 3D	Impresión 3D	Entornos de programación
¿Hay distintas formas de explorar otros planetas?	12-14 años	8	●	●		●	Bitbloq/ OpenSCAD/ Slic3r
¿Nos atrevemos a diseñar un casco espacial?	12-14 años	6			●		OpenSCAD
¿Por qué comprarlo si puedo crearlo?	14-16 años	8			●		OpenSCAD
¿Es Internet la respuesta?	14-16 años	8	●				Python
¿Puede la tecnología ayudar al cuidado de las plantas?	14-16 años	6	●	●			Bitbloq
¿Podemos adelantarnos al futuro?	14-16 años	8	●	●			Arduino
¿Puedo vestirme con ropa inteligente?	14-16 años	8	●	●			Arduino
¿Podemos crear un mundo al alcance de todos?	14-16 años	10	●	●		●	Bitbloq/ FreeCAD/ Slic3r
¿Podemos crear objetos que ayuden a mejorar nuestro día a día?	14-16 años	8			●	●	FreeCAD/ Slic3r
¿Innovamos a la hora de levantarnos?	14-16 años	10			●	●	FreeCAD/ Slic3r

En la propuesta pedagógica de *Programación de las Cosas*, las actividades previstas no solo potencian el aprendizaje colaborativo (incluso con las familias y el resto del centro escolar) para identificar y contrastar diferentes soluciones ante un mismo problema, sino que buscan promover procesos de aprendizaje caracterizados por:

- **Profundidad frente a extensión**

El aprendizaje significativo es aquel que se comprende y se integra en el conocimiento previo. Para ello, se necesita involucrar al alumno en el proceso de aprendizaje. Por este motivo, consideramos que es fundamental trabajar los contenidos relevantes y profundizar en ellos para lograr su comprensión, en lugar de trabajar muchos contenidos sin asegurarnos de que se comprendan todos.

- **Aprender haciendo**

La experimentación y aplicación práctica de los contenidos es el nivel más alto al que debemos llegar para alcanzar el aprendizaje significativo. Esta experimentación permite a los alumnos investigar y reflexionar sobre el proceso, así como monitorizar y evaluar el logro de sus objetivos. Además, proponiendo situaciones cercanas a los alumnos, nos aseguramos de obtener un mayor interés por la experimentación.

- **Aprendizaje no abstracto**

Los alumnos deben mostrar interés por el problema a resolver, de lo contrario nunca aprenderán a resolverlo. Para ello, planteamos problemas surgidos en situaciones concretas que pueden ubicar en su día a día o que, al menos, tienen un reflejo en alguna parte del mundo que conocen.

- **Teoría a través de la práctica**

La teoría se aprende a través de la práctica, es decir, a través de retos concretos y cercanos, y no abstractos, como hemos comentado en el punto anterior. La teoría se va incluyendo en pequeñas píldoras, a medida que vayan surgiendo, como necesidad para resolver el reto propuesto. Por tanto, los retos están diseñados de tal forma que respondan a toda la teoría que queramos enseñar.

- **Aprendizaje por proyectos guiados y de libre elección**

A lo largo del curso se debe enfocar un proyecto global que los alumnos deben solucionar y en el cual apliquen los contenidos trabajados, fomentando el trabajo colaborativo y la asignación de roles. Así mismo, se asentarán contenidos por medio de problemas de extensión breve que pueden trabajar por parejas o grupos de tres alumnos.

- **Responsabilidad para hacer productivo el tiempo de trabajo**

Nuestra misión es conseguir que el tiempo en el aula sea lo más productivo posible, ya que disponemos de tiempos limitados en los que se puede desarrollar trabajo en grupo. Se plantea una forma de trabajar que promueve responsabilidad y autonomía por parte del alumno, en la que el alumno debe ser consciente del poco tiempo disponible y gestionar el tiempo de las actividades para poder terminar las actividades planteadas en cada jornada. Para que esto sea posible, el docente debe explicar esta forma de trabajar para concienciar a los alumnos.

- **Aprender a aprender**

Además de enseñar contenidos y competencias concretas a través de cada uno de los proyectos, se proponen estrategias de aprendizaje que ayudarán a los alumnos a enfrentarse a problemas de otros ámbitos que consideren similares o, al menos, similares en algunas partes del problema, y poder emplear estrategias generales de aprendizaje para resolver cualquier problema con características parecidas o un patrón de solución similar.

- **Aprendizaje autorregulado**

La evaluación busca reflexionar sobre el proceso seguido y comprobar la superación de dificultades durante el proyecto, así como las capacidades de resistencia al fracaso del alumno. Además, pretende reflexionar sobre la consecución de los objetivos y la preferencia en los roles, ayudando a los alumnos a determinar sus preferencias laborales futuras.

A través de cada proyecto, se persigue fomentar la consolidación de comunidades de conocimiento para distintos lenguajes de programación, con el fin de que se retroalimenten unas a otras.

¿Qué necesito para llevar a cabo cualquier proyecto?

Conocimientos previos

Es importante que, antes de comenzar cualquiera de los proyectos de *Programación de las Cosas*, tanto alumnos como profesores consulten tutoriales específicos sobre las diferentes herramientas y lenguajes de programación con los que se trabajará en ese proyecto.

Tanto en el entorno web setveintiuno.com como en <http://diwo.bq.com/cursos/> podrán consultarse tutoriales recomendados para cada uno de los diferentes lenguajes de programación.



En el caso de aquellos que contemos con experiencia previa en trabajar con lenguajes de programación, tendremos la posibilidad de consultar en setveintiuno.com cada uno de los archivos fuente utilizados en dichos entornos de programación.

En el caso de los proyectos que exigen la utilización de FreeCAD, se recomienda que los docentes se hayan familiarizado de forma extensa en el manejo de esta herramienta.

Así mismo, a la hora de analizar cada proyecto de *Programación de las Cosas*, es aconsejable tener en cuenta las siguientes orientaciones generales:

Programación

Un error típico es creer que la base para programar es aprender utilizando un lenguaje concreto. Más importante que el lenguaje que utilizamos para programar es que nos aseguremos de que nuestros alumnos comprendan adecuadamente los conceptos más básicos de la programación y sean capaces de aplicarlos a los problemas y retos planteados en los proyectos. Estos conceptos son algoritmos, variables, condicionales, bucles y funciones, y son relativos a la lógica de programación en cualquier lenguaje, ya sea por bloques o por código. Para hacer frente a los problemas planteados, requieren un conjunto de habilidades concretas relativas al pensamiento computacional. Estas habilidades, como la descomposición del problema en pequeños problemas concretos o la abstracción, son necesarias para asegurarnos de que los alumnos han comprendido el problema y saben qué necesitan para afrontarlo.

Robótica

La programación de electrónica requiere simular el resultado y reflexionar sobre la forma más efectiva de resolver un problema o crear un invento. Además de las habilidades necesarias para programar, se deben poner en práctica habilidades que permitan pensar en los componentes electrónicos adecuados para la situación.

Los alumnos deberán tener claro cómo interactúan los sensores y los actuadores con la placa controladora. Por ejemplo, un sensor siempre tendrá que ir junto a una instrucción de control, es decir, un condicional o un bucle. Esto se debe a que el sensor recaba datos que se utilizan para hacer una instrucción concreta según el dato recogido, por lo que habrá que tener en cuenta algunos detalles de programación respecto a los componentes electrónicos.

Diseño e impresión 3D

Al crear diseños, debemos tener en cuenta el espacio disponible en el área de impresión (para que se puedan imprimir las figuras de dos grupos a la vez). También estaremos pendientes de cómo orientar el diseño para facilitar la impresión. Nos tendremos que preguntar una serie de cuestiones para asegurarnos de que el diseño cuenta con todo lo necesario para facilitar la impresión. Estas cuestiones nos las haremos en el siguiente orden:

1. ¿Cuál es la posición de la pieza que la dejaría con mayor superficie apoyada en la cama de impresión?
Solución: habría que rotar el diseño.
2. ¿Qué pasaría si siguiesen quedando partes en suspensión?
Solución: habría que añadir soportes.
3. ¿Qué pasaría si rotásemos la pieza, le añadiésemos soportes y aun así no funcionase?
Solución: habría que rediseñar la pieza o hacerlo de forma que permita imprimirla por partes.

PASOS PARA IMPRIMIR

Para imprimir es necesario convertir nuestro archivo .stl, generado por el programa de diseño, en un archivo que la impresora pueda reconocer e interpretar. Este paso se hace con un *software* de laminado, en el caso de *Programación de las Cosas* se usará Slic3r. El archivo que crearemos con el *software* de laminado será el G-code, que es el conjunto de instrucciones que indican a la impresora las operaciones que debe realizar para imprimir una pieza en 3D.

Para laminar una pieza tenemos que configurar dos partes. En primera instancia, las características de la impresora con la que vamos a imprimir: su volumen de impresión, tamaño de boquilla, etc. El propio *software* incluye muchos modelos para introducir estos datos. Si no es así, podemos consultar los de nuestra impresora y agregar sus características.

En segundo lugar, configuraremos las características de la impresión propiamente dicha, es decir, todo lo necesario para que la pieza tenga la calidad final que deseamos (altura de capa, relleno de la pieza, grosor del contorno, etc.). Es importante conocer el material con el que vamos a imprimir para indicar la temperatura de impresión. El fabricante de nuestra impresora puede proporcionarnos archivos de impresión para determinados materiales y calidad final, bastará con cargarlo en el *software* para crear el G-code.

OTROS ASPECTOS A TENER EN CUENTA

Imprimir más de un archivo STL

Para unir dos .stl en el programa de laminado, solo tenemos que abrir los dos archivos desde el programa, en nuestro caso Slic3r.

Filamento

Existen diferentes tipos de filamento adecuados para cada impresora 3D disponible en el mercado. Es importante tener en cuenta el grosor del filamento a la hora de comprarlo, ya que hay máquinas que precisan un filamento de 1,75 mm, y otras, de 2,85 mm. Además, los filamentos para impresoras 3D pueden ser de distinto material: PLA y ABS son los más comunes.

Dentro de la variedad de PLA también tenemos distintos materiales con los que podemos crear diferentes acabados: tipo madera o cobre, por ejemplo. Para nuestros proyectos nos servirá cualquier tipo de filamento, pero debemos fijarnos bien, antes de adquirir el filamento, en las necesidades de nuestra impresora. No todas las impresoras pueden utilizar los mismos tipos de filamentos; esto estará indicado en las especificaciones de la impresora. Debemos leer estas indicaciones antes de utilizarla.

Tiempo de impresión

El tiempo de impresión para cada pieza varía en función de la impresora. Según los parámetros de impresión que hayamos definido, el programa de laminado puede estimar un tiempo que, aunque puede sufrir alguna variación respecto al tiempo real, nos puede dar una idea aproximada del tiempo final de impresión.

Por otra parte, el tiempo de impresión 3D no está contemplado dentro del tiempo recomendado para las actividades. Por lo que es labor del docente encargarse de imprimir las piezas entre parada y parada, con el fin de que estén impresas para las posteriores jornadas. Si se desea imprimir junto con los alumnos, deberá ser fuera del tiempo dedicado a la parada o de forma paralela a las actividades. Esta tarea exigirá una planificación ardua por parte del docente, ya que hay proyectos en los que se requiere la impresión de piezas 3D para cada uno de los distintos grupos de alumnos que componen el aula.

Requisitos técnicos mínimos

Aquellas actividades del proyecto que requieran de programación deben ser realizadas en ordenadores, y no en otros dispositivos, por varias razones:

- En el caso de los proyectos de Arduino, la programación creada en los entornos de programación debe ser trasladada a una placa controladora mediante un puerto USB, pero no todas las *tablets* disponen de este tipo de conexión.
- Las *tablets* son dispositivos diseñados para entretenimiento y no para crear tecnología. El enfoque de *Programación de las Cosas* es preparar a los alumnos para ser creadores de tecnología y no meros usuarios.
- La mayoría de entornos de programación y herramientas de diseño no poseen una interfaz para un dispositivo móvil.

No obstante, es recomendable el uso de otros dispositivos (*tablets*, teléfonos móviles...) para la realización de algunas de las secciones de los proyectos.

Para poder realizar cualquier proyecto de *Programación de las Cosas* es necesario disponer de ciertos elementos de *software* que nos permitan desarrollarlo en su totalidad. En el **Anexo**, presentamos un resumen descriptivo del siguiente detalle de los distintos entornos de programación:

Bitbloq

Caso *online*

Bitbloq es una herramienta de uso *online*. Mediante una cuenta propia, el usuario puede guardar cada uno de sus proyectos en su nube personal. El usuario puede publicar *online* su proyecto en la sección Explora, donde cualquier otro usuario puede acceder a él, crear una copia o modificarlo.

Caso *offline*

Bitbloq no permite programar *offline*. Existe la opción de descargarse proyectos en el ordenador en formato *.json* y cargar proyectos desde el ordenador en este mismo formato.

Bitbloq *offline* portable

Caso *online*

Bitbloq *offline* portable no necesita de ninguna conexión a Internet para funcionar.

Caso *offline*

Bitbloq *offline* portable sí permite programar *offline*. Existe la opción de guardar proyectos en el ordenador en formato *.json* y cargar proyectos desde el ordenador en este mismo formato. Al ser portable no requiere de ninguna instalación. Solo se debe colocar la carpeta del programa en una carpeta de usuario, como Mis documentos o el propio escritorio, y ejecutar el programa directamente.

Scratch

Caso *online*

Scratch permite trabajar *online*. Mediante una cuenta propia, el usuario puede guardar cada uno de sus proyectos en su nube personal. El usuario puede compartir sus proyectos para que cualquier otro usuario pueda acceder a ellos, crear una copia o modificarlos.

Caso *offline*

Scratch también tiene versión *offline* para Windows, MAC y Linux de 32.

OpenSCAD

Caso *online*

OpenSCAD no permite trabajar *online*. Aunque hay herramientas en *fase alpha*, como <http://openjscad.org/>, su aplicación en el aula es ahora mismo muy limitada, por lo que se recomienda el uso de OpenSCAD *offline*.

Caso *offline*

Instalación del programa (<http://www.openscad.org/downloads.html>) para Windows, MAC y Linux.

Tinkercad

Caso *online*

Tinkercad sí permite trabajar *online*. Mediante una cuenta propia, el usuario puede guardar cada uno de sus proyectos en su nube personal. El usuario puede publicar *online* su proyecto en la sección Galería, de este modo cualquier otro usuario puede acceder a él, crear una copia o modificarlo.

Caso *offline*

Tinkercad no permite trabajar *offline*.

Slic3r

Caso *online*

Slic3r no tiene opción de trabajo *online*.

Caso *offline*

Slic3r sí tiene opción de trabajo *offline*.

Python

Caso *online*

Numerosas IDE permiten trabajar *online* con Python, aunque, en ocasiones, con restricciones (no poder agregar librerías u otros).

Caso *offline*

Python sí permite el trabajo *offline* con todas las opciones disponibles.

Arduino

Caso *online*

Numerosas IDE, incluida Bitbloq, permiten trabajar *online* con Arduino, aunque, en ocasiones, con restricciones (no poder agregar librerías u otros).

Caso *offline*

Arduino sí permite el trabajo *offline* con todas las opciones disponibles.

FreeCAD

Caso *online*

FreeCAD no permite trabajar *online*.

Caso *offline*

FreeCAD sí permite el trabajo *offline*.

El navegador que se recomienda para el desarrollo de los proyectos es Google Chrome. Los *plug-in* necesarios son aquellos que cada plataforma requiere. El *plug-in* de Flash salta cuando se necesite en aquellos navegadores en los que no viene por defecto. Será preciso aceptar la instalación en el caso de que se solicite. Igualmente, en las páginas de cada plataforma se encuentran los requerimientos de cada una de ellas. En el caso de Scratch, podemos verlo en <https://scratch.mit.edu/info/faq/>.

Para más información, podemos consultar la sección correspondiente a lenguajes de programación de esta guía.

Instalación de entornos de programación

Ver la sección correspondiente a lenguajes de programación.

Materiales para el desarrollo de los proyectos

Algunos de los proyectos de *Programación de las Cosas* requieren necesariamente la utilización de un kit de electrónica. El kit de electrónica recomendado para el desarrollo de estos proyectos es la opción más completa y versátil que existe en el mercado. El mismo kit se utiliza en proyectos de distintas edades, siendo útil, incluso, en ciclos universitarios y profesionales.

Ver en el **Anexo** el detalle de los componentes que integran el kit recomendado.



Además del material impreso de cada proyecto, la propuesta pedagógica de *Programación de las Cosas* incluye un entorno web setveintiuno.com en el que poder consultar algunos contenidos de ese mismo material (imágenes de programación, recursos *online*..), además de poner a disposición de los docentes del programa SET VEINTIUNO que utilicen sus materiales toda aquella información de carácter metodológico de cada proyecto.

Definición de un proyecto

Temporalización



Cada proyecto puede estar constituido por cuatro, seis, ocho o diez paradas (una parada o sesión equivale a 45 minutos). El número de paradas varía en función de las necesidades y especificidades de cada proyecto.

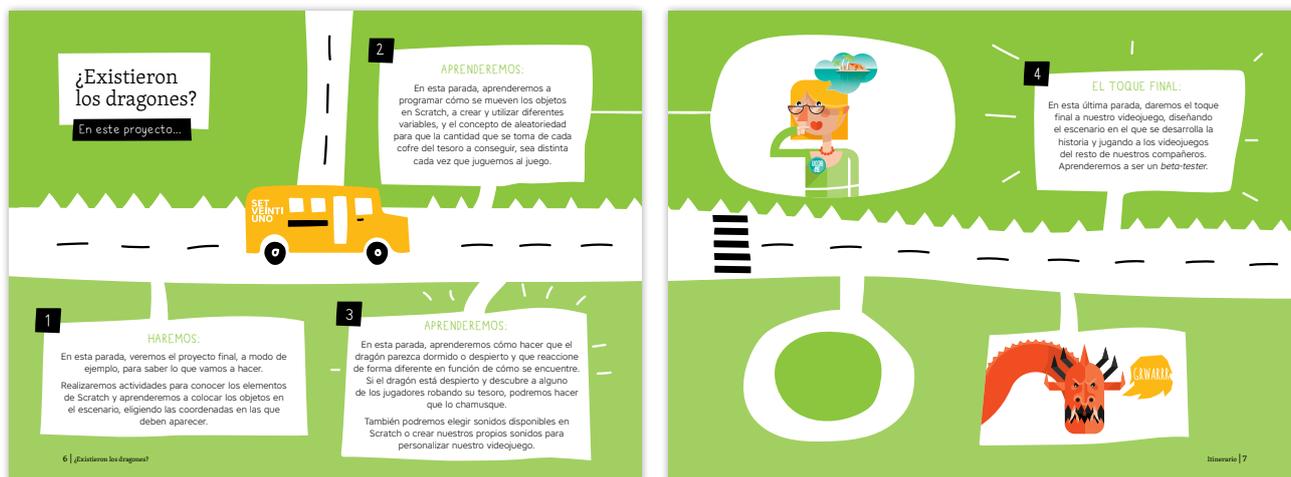
Todas las actividades de un proyecto llevan asociado un tiempo de ejecución que se podrá consultar en las Fichas del profesor. Los tiempos de las actividades son orientativos, ya que dependerán en gran parte del aula (conocimientos y habilidades de los alumnos), así como de otros factores que pueden influir en dichos tiempos.

Estructura

Itinerario

CUADERNO DEL ALUMNO

Al inicio de cada proyecto, los alumnos encontrarán un itinerario en el que se detalla el título del proyecto, el número de paradas en las que se desarrolla y la explicación de lo que se hará en cada una de ellas.



FICHAS DEL PROFESOR

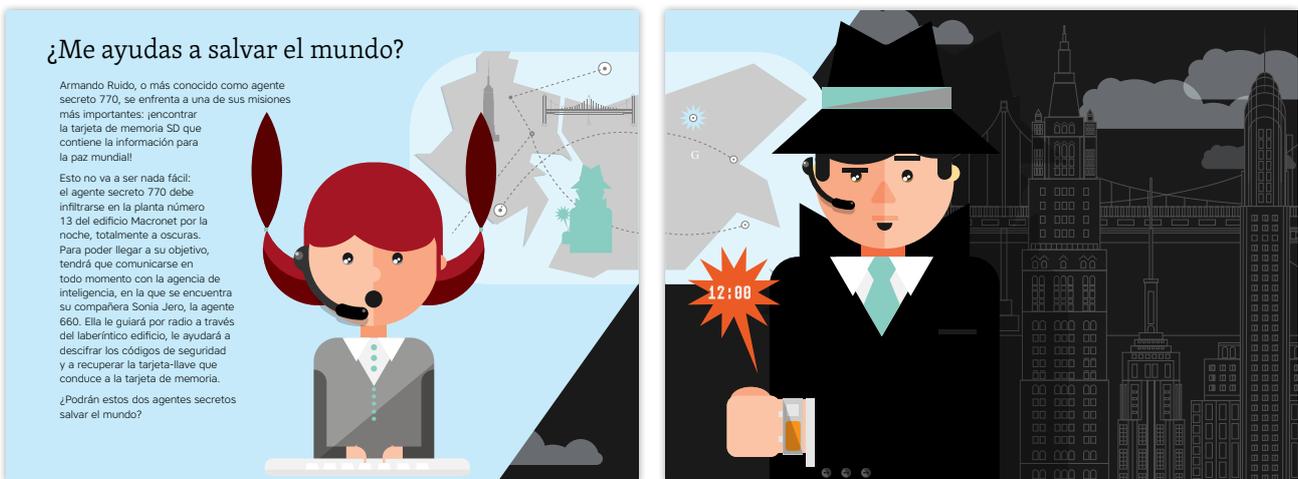
El docente no tendrá el itinerario del Cuaderno del alumno, pero podrá ver las características y el planteamiento del proyecto en la sección Metodología de la web setveintiuno.com.

Introducción

CUADERNO DEL ALUMNO

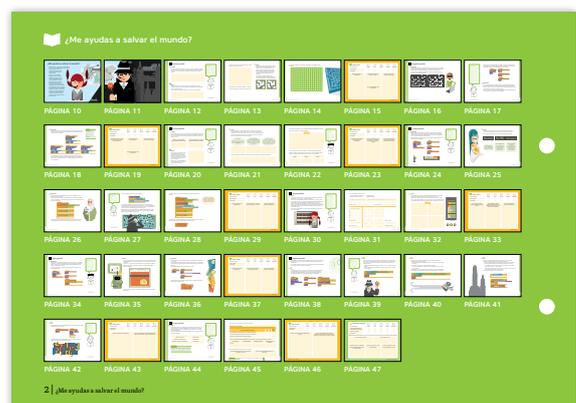
Cada proyecto comienza con una historia introductoria en la que se presenta un relato y se propone un reto a los alumnos por medio de una pregunta y una ilustración motivadoras. De esta manera, se busca despertar la curiosidad y el interés del alumno por los contenidos que se presentan. En este relato, normalmente, los personajes se ven envueltos en algún tipo de trama, y son los alumnos los que deben ayudar a estos personajes a solventar o resolver la situación.

El tiempo que debe destinarse a esta parte del proyecto no está previsto en la primera parada, sin embargo, se recomienda que su lectura ocupe un total de 5 minutos, pudiéndose plantear de diferentes formas en el aula (lectura en voz alta y puesta en común, lectura individual...).

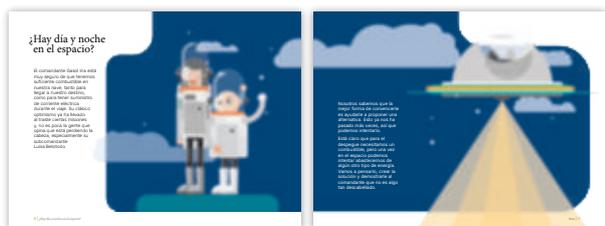


FICHAS DEL PROFESOR

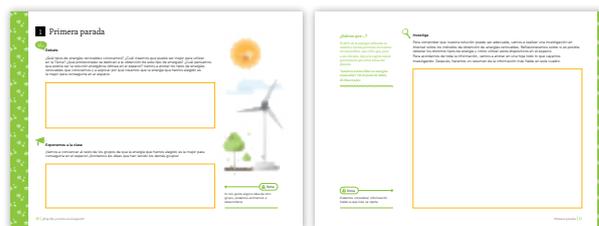
Las Fichas del profesor se inician con una sección similar a la del alumno, pero en la que no se plasma la historia introductoria, sino que se explica el desarrollo y planteamiento del proyecto, así como las dinámicas del aula, los entornos de programación en los que los alumnos van a trabajar los recursos que el docente puede utilizar en el aula para la introducción del proyecto y otras indicaciones, como puede ser la consulta de las correspondientes secciones de esta guía. En la parte de atrás de esta sección, habrá, además, una miniatura de cada una de las páginas del Cuaderno del alumno para que el docente tenga una referencia y pueda comparar y seguir con facilidad cada una de las secciones o paradas del proyecto.



Desarrollo del proyecto



Pregunta motivadora



Activación metacognitiva



Activación cognitiva



Relación de conocimientos



Aplicación en soluciones prácticas



Reflexión final

CUADERNO DEL ALUMNO

Los proyectos están divididos en paradas que, a su vez, están estructuradas en diferentes secciones o actividades. En estas, los alumnos ejercitarán diferentes habilidades y competencias. Todas estas secciones tienen un resultado tangible, como puede ser la realización de un boceto, la investigación de un tema determinado o el montaje de un desarrollo concreto. La conceptualización de cada proyecto recomienda el cumplimiento de determinados pasos en un orden concreto e inalterable, de este modo, los alumnos consolidarán los conocimientos adquiridos y crearán con éxito sus propios proyectos.

1 Primera parada

Debate

¿Qué tipos de energías renovables conocemos? ¿Cuál creemos que puede ser mejor para utilizar en la Tierra? ¿Qué profesionales se dedican a la obtención de este tipo de energías? ¿Cuál pensamos que podría ser la solución energética idónea en el espacio? Vamos a anotar los tipos de energías renovables que conocemos y a explicar por qué creemos que la energía que hemos elegido es la mejor para conseguirla en el espacio.

Exponemos a la clase

(Vamos a convencer al resto de los grupos de que la energía que hemos elegido es la mejor para conseguirla en el espacio! ¡Anotamos las ideas que han tenido los demás grupos!

Nota

Si nos gusta alguna idea de otro grupo, podemos animarnos a desarrollarla.



¿Sabías que...?

El 64 % de la energía utilizada en América Latina proviene de fuentes no renovables, una cifra que, pese a ser elevada, deja a la región mejor posicionada que otros áreas del mundo.

Investiga

Para comprobar que nuestra solución puede ser adecuada, vamos a realizar una investigación en Internet sobre los métodos de obtención de energías renovables. Reflexionaremos sobre si es posible obtener los distintos tipos de energía y cómo utilizar estos dispositivos en el espacio. Para acordarnos de toda la información, vamos a anotar en una hoja todo lo que vayamos investigando. Después, haremos un resumen de la información más fiable en este cuadro.

Nota

Podemos considerar información fiable la que más se repita.

En la primera parada, se busca el compromiso por el aprendizaje, es decir, la motivación adecuada que promueva el interés por aprender. Los alumnos llevarán a cabo principalmente actividades del tipo **Investiga, Diseña, Crea, Debate...**

3 Tercera parada

Diseña

En grupos, diseñaremos el tablero de nuestro juego y distribuiremos las casillas de cada sección. Una vez que tengamos nuestro diseño terminado, le haremos una foto y la subiremos al ePortfolio.

¿Sabías que...?

El juego Monopoly tiene su origen en un juego creado por Lizzie Magie en 1903 y patentado en 1934, llamado The Landlord's Game. El juego se hizo popular en varias ciudades de Estados Unidos en los años siguientes. Este se editó en varios formatos y versiones sin el control de su autora original.

Nota

Si nos ayuda, podemos buscar ejemplos de tableros de otros juegos en Internet.



Crea

¿Hacemos el juego más divertido? Vamos a crear con nuestro grupo casillas especiales. Podemos crear una casilla en la que cuando caigamos, perdamos turno. Para poder identificar estas casillas en el tablero, debemos dibujar sus logos.



En las paradas intermedias, se trabaja el nuevo contenido, promoviendo su comprensión al relacionarlo con su conocimiento previo. El siguiente tipo de actividades trabajadas facilita el análisis del contenido para poder descubrir errores en contenidos relacionados, asociarlo y clasificarlo.

Una vez comprendido el contenido, se proponen actividades para ser capaces de utilizar el conocimiento en otras áreas, resolver problemas y tomar decisiones que impliquen dicho contenido. En estas fases del proceso de aprendizaje se harán actividades de **Reflexiona, Experimenta y Construye**.

3 Tercera parada

Experimenta

En esta parada, vamos a programar los componentes electrónicos que harán funcionar nuestro zóotrope, pero primero debemos elegir cuáles vamos a utilizar y de qué forma.

- ¿Cómo va a funcionar el giro del zóotrope?
- ¿Vamos a poder encenderlo y apagarlo?
- ¿Sabemos cuál es la velocidad a la que tiene que girar o será mejor probar varias velocidades?

De los siguientes componentes, ¿cuáles vamos a conectar a nuestra placa controladora?



¿Sabías que...?

La primera cámara de video portátil se inventó en 1967 y estaba compuesta de dos piezas: una cámara de blanco y negro y la unidad de grabación que iba en un bolso aparte. Aunque lo podía llevar una persona, su peso hacía recomendable que mientras uno grababa, otro portase la grabadora colgada del hombro o la espalda.



14 | ¿Podemos ser directores de cine?

Experimenta

Ya tenemos claro qué componentes vamos a utilizar, así que ahora toca programarlos usando Bitbloq. Empezaremos con el botón, cuya función será encender o apagar el zóotrope.

Comenzaremos creando una variable que llamaremos encendido en la sección Variables globales, funciones y clases. Usaremos esta variable para encender el zóotrope (= 1) y apagarlo (= 0).



A continuación vamos a hacer que, al presionar el botón se encienda el zóotrope, y en caso de que ya esté encendido, se apague pulsando el botón. Para ello, copiaremos el siguiente condicional en la sección Bucle principal (Loop).



La imagen anterior indica que si se presiona el botón, ocurrirá algo, pero aún no está definido el qué. Vamos a ordenar los siguientes bloques de programación para indicar que el zóotrope se encienda o se apague al presionar el botón. Cuando seamos el orden de los tres bloques, lo construiremos en Bitbloq.



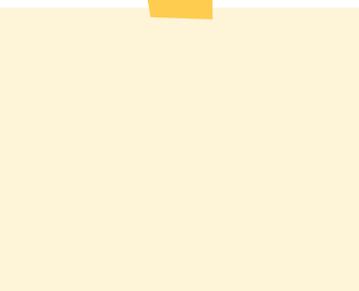
Tercera parada | 15

La última parada se reserva para establecer conjuntamente conclusiones sobre el proyecto, sobre los contenidos trabajados y su impacto, o su influencia en temas relacionados. Para ello, se trabajan actividades de **Documenta, Debate, Juega...** En esta parada final, se promueven actividades de presentación ante los compañeros en las que los alumnos trabajan aquellos aspectos clave que favorezcan la conclusión.

6 Sexta parada

Investiga

¿Qué es la velocidad de reacción? ¿se puede mejorar? ¿cómo? Investigaremos sobre estos temas y anotaremos la información en el siguiente recuadro:



¿Sabías que...?

La velocidad de reacción depende de muchas factores, como pueden ser el tipo de estímulo (auditivo, visual o táctil), sexo, edad o duración del estímulo. Además, unos estímulos activan antes que otros la velocidad de reacción, siendo la respuesta más rápida en los sonoros.



26 | ¿Somos capaces de distinguir el sonido de una nota?

Documenta

Debemos proponer ejercicios para mejorar la capacidad de reacción y ser mejores en el juego. Podemos buscar ejercicios o modificar algunos de los que encontremos, adaptándolos a nuestro juego.



Juega

Jugaremos a nuestro juego, practicando los ejercicios que hemos propuesto en la actividad anterior, para conseguir tener más velocidad de reacción e intentar obtener el mejor resultado en el juego.

Nota → Grabaremos un vídeo jugando y lo subiremos al ePortfolio.

Recuerda → Los recursos necesarios para este proyecto están disponibles en la web.

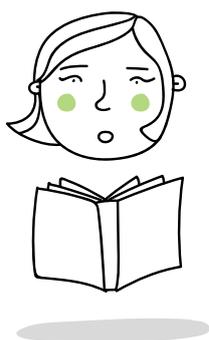
Sexta parada | 27

A lo largo de cada proyecto, los alumnos encontrarán píldoras o cápsulas de información que, además de añadir contenido extra a cada proyecto, ayudarán a que conozcan datos interesantes, así como a ampliar sus conocimientos sobre ciertos temas de cultura general. Mediante los **¿Sabías que...?** se promueve que los alumnos conecten con aspectos que enriquezcan su experiencia de aprendizaje, haciéndola más intensa y significativa.

En el caso de los proyectos de 10-12 años, las píldoras vienen presentadas por personajes del programa SET VEINTIUNO y el texto está encuadrado en un bocadillo, mientras que, en el caso de los proyectos de 12-14 o 14-16 años, se muestran en notas.

¿Sabías que...?

Un boceto es una ilustración sin detalles, que se usa para conocer los rasgos principales de una obra artística.



¿Sabías que...?

El suministro de alimentos en las misiones de la NASA empezó con productos especiales, diseñados para tener el menor volumen y tamaño posible, debido a las limitaciones de masa y volumen que podían ser lanzados fuera de la Tierra.



Todas las capturas y webs que aparezcan en el Cuaderno del alumno se podrán consultar en el área correspondiente a cada proyecto en el entorno web setveintiuno.com:



FICHAS DEL PROFESOR

Las Fichas del profesor también están organizadas por paradas, con instrucciones detalladas de cada actividad. En algunas actividades, las indicaciones simplemente son soluciones a preguntas que se les formulan a los alumnos o respuestas a actividades de investigación. En otros casos, el docente cuenta con instrucciones detalladas y un *paso a paso* de un determinado proceso.

1 Primera parada 10-11

1.1 Debate
Los alumnos deben debatir sobre el tema de las energías renovables, respondiendo las preguntas que se les plantean.



1.2 Documenta
Los alumnos deben anotar aquellas ideas que les hayan resultado interesantes durante el debate.

1.3 Investiga
Con esta actividad se pretende que los alumnos conozcan otros tipos de energías para ver cuáles pueden ser sus ventajas y desventajas, así poder valorarlas y decidir cuál creen que sería mejor para obtenerla en el espacio.

Los principales tipos de energías renovables son solar, eólica, hidráulica, mareomotriz, geotérmica y bioenergía.

La solución más idónea sería la solar, ya que un instrumento como un panel solar en el espacio conseguiría una energía entre seis y ocho veces mayor que en la Tierra. Además, en el espacio no hay problemas de interferencias meteorológicas, por lo que incrementa su eficacia.

Primera parada (Hay día y noche en el espacio) | 11

8-11

Esta actividad también nos permitirá introducir diferentes profesiones relacionadas con este ámbito, como pueden ser un ingeniero o un técnico en energías renovables.

Otras cuestiones que se podrían plantear a los alumnos son las siguientes:

- ¿Qué diferencias hay entre energías renovables y no renovables?
- ¿Por qué hasta el momento se utiliza más el petróleo, que es una fuente de energía que se agota, y no las energías renovables, que son inagotables?
- ¿La obtención de energías renovables contamina el medioambiente?

1.4 Diseña
Con esta actividad se pretende trabajar la creatividad de los alumnos. La finalidad no es que haya unos proyectos mejores y otros peores, o que un solo boceto pueda ser el correcto, sino que representen lo que se les ocurra, después de haber investigado y haber visto cómo son los dispositivos o instrumentos con los que se obtienen este tipo de energías renovables. Un ejemplo de un instrumento que los alumnos pueden dibujar para obtener la energía solar es una placa solar.

1.5 Auditoría creativa
Las evaluaciones que se realizan en cada parada van destinadas a evaluar de una manera integral la dimensión socioemocional y el proceso de aprendizaje de los alumnos. Al trabajar contenidos diferentes en cada parada, las preguntas se repiten, con el fin de que los alumnos reflexionen sobre su trabajo, su progreso y sus propias emociones.



12 | (Hay día y noche en el espacio)

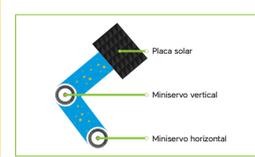
Primera parada

3 Tercera parada 18-21

3.1 Investiga
A aquellos alumnos que hayan acabado el boceto antes de tiempo, les podemos proponer que busquen un vídeo sobre el funcionamiento de estas máquinas.

Con esta actividad se pretende que los alumnos se hagan una idea de cómo se va a mover el dispositivo que van a construir. Buscando cómo funciona y qué es este tipo de máquina, se harán una imagen mental de lo que después tienen que construir.

Un ejemplo descriptivo de un dibujo que los alumnos podrían realizar sería:





Tercera parada (Hay día y noche en el espacio) | 17

18-21

El miniservo tiene que estar conectado a la placa, tal y como indica la imagen. El pin de la placa en el que lo pongan es opcional, pero debe coincidir en la programación y en la placa física.



En el tercer ejercicio, deberán ir a Bitbloq y programar el otro sensor. La programación que deberán crear es la siguiente:

Ángulo del servo vertical



La conexión del segundo miniservo debe ser la misma que en la imagen anterior.

Tercera parada (Hay día y noche en el espacio) | 19

El docente recibirá indicaciones para consultar aquellos recursos web, y otros adicionales, que favorezcan la adecuada comprensión y realización del proyecto. Así mismo, todos los recursos web (tanto necesarios para la realización del proyecto como contenido extra) y los bloques de programación estarán disponibles en el área privada del docente en el entorno web, setveintiuno.com.



El docente recibirá indicaciones a lo largo de cada proyecto para pedir a los alumnos que suban al ePortafolio evidencias del desarrollo del proyecto. Sin embargo, es el propio docente el que podrá decidir en cada actividad si desea elegir esta última opción en lugar del cuaderno de actividades, según las necesidades concretas de su aula.

Auditoría creativa

Todas las paradas terminan con una **Auditoría creativa** que se divide en dos partes: una parte inicial de autorreflexión, y otra que incluye cuestiones sobre el aprendizaje y las emociones que cada proyecto despierta tanto de modo individual como de manera colectiva, es decir, los sentimientos y emociones que cada alumno haya podido tener en relación al trabajo con su grupo o con ciertas decisiones que se hayan tenido que tomar a lo largo del proyecto.

Por un lado, se espera que los alumnos muestren cómo se han sentido: confusos porque esperaban hacer algo diferente, tristes porque creen que es muy difícil o alegres porque se ven capaces de hacerlo; y el progreso entre el primer día y el último.

Por otro lado, a través de las preguntas, se espera que los alumnos reflexionen sobre su proceso de aprendizaje y su trabajo en grupo. De esta forma, se intenta promover un aprendizaje reflexivo, clave para ser conscientes de qué se esperaba de ellos y que sepan si lo han logrado o no.

Auditoría creativa

Hoy me he sentido...

Alegre	Triste	Confuso	Enfadado	Sorprendido
<input type="checkbox"/>				

Con el proyecto

Con el grupo

Considero que lo más importante que he aprendido hoy es...

¿Tengo ganas de continuar creando mi propio videojuego?

¿Qué parte creo que me va a gustar más, el diseño, desarrollo o testeo de videojuegos?, ¿por qué?

Primera parada | 13

Auditoría creativa

Hoy me he sentido...

Alegre	Triste	Confuso	Enfadado	Sorprendido
<input type="checkbox"/>				

Con el proyecto

Con el grupo

1 ¿Cree que me ha resultado útil debatir con otro grupo sobre la información que he encontrado?, ¿por qué?

2 ¿Qué he aprendido hoy? De todo lo que he aprendido, ¿qué es lo que considero más importante?

3 ¿He tenido dificultades para encontrar la información?, ¿he comprobado que es correcta?, ¿cómo lo he hecho?

Procesa parada | 9

Auditoría creativa del proyecto

La **Auditoría creativa** que se realiza al final de cada proyecto busca la autoevaluación de los alumnos respecto a varios aspectos del proyecto.

Auditoría creativa del proyecto

Hoy me he sentido...

Alegre	Triste	Confuso	Enfadado	Sorprendido
<input type="checkbox"/>				

Con la temática del proyecto

Cuando empezamos el proyecto

Al terminar el proyecto

Con el grupo

Reflexiono sobre mis emociones (cómo me sentía al principio, cómo me siento ahora y cómo mis sentimientos han ido cambiando).

Con la creación del proyecto

Con el trabajo en grupo

De todas las cosas que he aprendido creando un código, ¿cuáles han sido las más importantes?

¿Qué me ha costado más?, ¿cómo lo he solucionado?

Si hubiese trabajado solo, ¿habría hecho algo diferente?, ¿por qué?

Auditoría creativa del proyecto | 41

Auditoría creativa del proyecto

Hoy me he sentido...

Alegre	Triste	Confuso	Enfadado	Sorprendido
<input type="checkbox"/>				

Con la temática del proyecto

Cuando empezamos el proyecto

Al terminar el proyecto

Con el grupo

1 Reflexiono sobre mis emociones (cómo me sentía al principio, cómo me siento ahora y cómo mis sentimientos han ido cambiando).

Con la creación del proyecto

Con el trabajo en grupo

2 ¿Qué he aprendido en el proyecto?, ¿qué considero que es lo más importante?

3 ¿Qué dificultades he encontrado a la hora de realizar el proyecto?, ¿he tenido alguna dificultad a la hora de ponerme de acuerdo con mi compañero?, ¿cómo lo he solucionado?

Auditoría creativa del proyecto | 29

Tanto la **Auditoría creativa** como la **Auditoría creativa del proyecto** son secciones que buscan la autorreflexión, la motivación y la regulación de sentimientos y emociones del alumno sobre cada una de las paradas, así como de todo el proyecto. En estas secciones, el alumno descubre sus dificultades y valora el progreso que ha realizado a lo largo de todo el proyecto.

Este tipo de evaluación permite al docente conocer las fortalezas y debilidades de cada alumno, y poder ajustarse a sus necesidades, además de evaluar no tanto el resultado final del proyecto, sino cada una de sus fases, es decir, el proceso.

Entorno web

La página web setveintiuno.com se divide en tres partes:

Mis cuadernos

Esta primera, a su vez, se organiza en tres secciones diferentes:

Metodología: sección en la que el docente podrá encontrar información sobre el proyecto a realizar (resumen del proyecto, estructura, objetivos a alcanzar, materias transversales que se trabajan, grupos de trabajo, materiales necesarios...).

Contenidos: sección estructurada del mismo modo que el material físico, es decir, por paradas y secciones en las que tanto alumno como profesor encontrarán los recursos necesarios para el desarrollo del proyecto. Este contenido comprende capturas de programación, páginas web, archivos fuente, el programa para la ejecución del proyecto, ejemplos de montaje...).

En este apartado, también se proporciona contenido extra o complementario para ampliar información sobre la temática del proyecto.

ePortfolio: sección en la que los alumnos subirán las evidencias de las fases del proyecto que se les indique en el cuaderno.

Todos los niveles cuentan con proyectos digitales en la web. Estos proyectos están señalizados con un icono diferente y el material, tanto para alumno como para docente, se encuentra en formato PDF en la sección de contenidos. Al igual que el resto de proyectos, disponen de las mismas secciones y los mismos recursos para su desarrollo. Estos proyectos, denominados **pack digital**, se ofrecen para que el docente pueda añadir alguno de ellos una vez finalizado el cuaderno, o bien, incluso, para sustituirlo por alguno del cuaderno, en caso de que se ajuste más a las necesidades del centro.

Acompañamiento

En esta parte, podemos encontrar contenido de formación, que nos ayudará con la implementación del programa en el aula.

Comunidad

Espacio desde el que se promueve el formato de comunidad de aprendizaje, a través de las evidencias de docentes de distintos centros.



Recursos iconográficos

Cada una de las paradas del proyecto se divide u organiza en diferentes secciones (explicadas con anterioridad en esta guía). Al inicio de cada sección se indica, tanto al alumno como al docente, mediante recursos iconográficos y un título, el tipo de actividad que se va a realizar. A continuación, podemos ver los iconos que pueden encontrarse a lo largo de un proyecto, así como una breve explicación de cuándo debe aparecer cada uno de ellos:

 Construye	Este icono se usa para indicar que, en esta actividad, los alumnos construirán físicamente algún objeto, bien con materiales reciclables o con piezas impresas en 3D (por ejemplo, un campo de fútbol). Para ello, se basarán siempre en un diseño previo.
 Crea	Este icono se usa para indicar que los alumnos deben crear, inventar o programar algo, ya sea un producto o unas reglas para un procedimiento determinado. Por ejemplo, las normas de un juego.
 Debate	Este icono indica un espacio para el debate, en el que los alumnos deben argumentar con sus compañeros su postura ante un tema y, entre todos, llegar a una conclusión sobre este.
 Diseña	Este icono se usa para aquellos momentos en los que los alumnos tienen que hacer un boceto de su invento antes de crearlo. También se utiliza para las actividades de diseño en 3D.
 Documenta	Este icono se usa cuando los alumnos tienen que dejar por escrito o grabado en audio/vídeo el proceso o resultado de alguna actividad del proyecto.
 Experimenta	Este icono indica que, en esta sección, los alumnos deben medir o probar algo para extraer una conclusión o resultado concreto. Por ejemplo, crear una programación y probar si funciona, probar varios componentes para ver cuál es el que más se adapta a sus necesidades, etc.
 Investiga	Este icono se utiliza para indicar que los alumnos deben buscar información en Internet sobre un tema, consultando diferentes recursos y entornos web, para finalmente lograr un conocimiento fiable a partir del pensamiento crítico.
 Reflexiona	Este icono se usa para que los alumnos dediquen este tiempo a pensar o reflexionar sobre una cuestión determinada, así como a analizar algún tema concreto desde una perspectiva crítica para poder extraer conclusiones razonadas.

 <p>Juega</p>	<p>Este icono se utiliza cuando los alumnos tienen que jugar o probar algún juego para poder extraer una conclusión o resultado concreto. Por ejemplo, crear una programación y probar si funciona, probar el juego o el objeto que hayan creado para ver cuál es el que más se adapta a sus necesidades e incluso implementar mejoras, etc.</p>
 <p>Nota</p>	<p>Este icono es una advertencia, explicación, comentario o noticia que ayuda a desarrollar la actividad o que añade algún aspecto importante.</p>
 <p>Recuerda</p>	<p>Este icono proporciona información necesaria para desarrollar la actividad o para la realización de próximas actividades.</p>
 <p>Consejo</p>	<p>Este icono es una opinión que orienta sobre una actuación de una determinada manera. A pesar de aconsejar sobre el modo de llevar a cabo una acción, no es necesario realizarla como se propone.</p>
 <p>Pista</p>	<p>Este icono es un indicio o señal que puede conducir o guiar hacia la averiguación o indagación de algo.</p>
	<p>Este icono es una indicación de que todas las capturas de programación y recursos web están disponibles en el entorno web setveintiuno.com.</p>
	<p>Este icono indica que los proyectos pertenecen al pack digital. Estos son proyectos que se proporcionan en la web para que el docente pueda añadirlos o sustituirlos por alguno de los proyectos del cuaderno.</p>

Implementación en el aula

Grupos de trabajo

Todos los proyectos están planteados para desarrollarse en grupo, a pesar de que haya actividades o secciones que se lleven a cabo de manera individual. En cada proyecto se explicará la distribución de los alumnos para un correcto desarrollo de este. Los grupos que se recomiendan en cada proyecto son orientativos. En los casos de robótica, se recomienda realizar el proyecto en grupos de cuatro alumnos, aunque se pueden considerar grupos de más o menos alumnos, siempre y cuando se disponga de kits de electrónica suficientes. Si formamos un grupo de menos alumnos, debemos distribuir los roles proporcionalmente al número de alumnos.

En los proyectos de impresión 3D, los grupos de trabajo dependerán del número de impresoras que haya en el centro. Los proyectos se han concebido para poder llevarse a cabo con la presencia de un solo docente. No obstante, se recomienda que, en función de las posibilidades del centro educativo, se trabaje de forma colectiva con otros docentes del centro que previamente hayan recibido la formación correspondiente.

Materiales para el desarrollo de los proyectos

Antes de cada parada y al inicio de un proyecto determinado, el docente debe revisar qué material necesita tener previamente para mitigar cualquier problema de última hora antes de poner en marcha el proyecto. Esta consideración es clave, dado que en muchos de los proyectos, o bien se solicitan materiales reciclables que no se encuentran en las aulas, o bien materiales de otra naturaleza sin los que no es posible llevar a cabo el proyecto.

Al finalizar cada parada, el docente será informado de los materiales necesarios para continuar con el proyecto. En el caso de que el proyecto sea de diseño e impresión 3D, el docente deberá estar atento a las indicaciones, al final de cada parada, sobre la necesidad de tener en cuenta los tiempos de impresión para la adecuada realización del proyecto.

Por otra parte, existen proyectos que incluyen un recortable como elemento indispensable para el desarrollo del proyecto. Los alumnos podrán encontrar este recurso en los **Anexos** de su propio cuaderno y en la sección correspondiente de su área privada en el entorno web **setveintiuno.com**. Otros recursos adicionales, como vídeos o archivos fuente necesarios para el desarrollo del proyecto, también podrán ser consultados en **setveintiuno.com**.



Lenguajes de programación en la propuesta pedagógica

A continuación, se incluye una breve descripción e indicaciones específicas de cada uno de los entornos de programación que se utilizan a lo largo de la propuesta pedagógica de *Programación de las Cosas*.

Debido a la actualización constante del *software* que utilizamos, especialmente los que son *open source*, no es posible asegurar que las indicaciones que se ofrecen en cada uno de los programas para el acceso o la instalación se mantengan en el tiempo.

Por eso, es aconsejable que, en el caso de que surjan dudas al comprobar un resultado algo diferente, consultemos algún tutorial *online* recomendado u otros manuales actualizados. Es muy importante que comprobemos que la fecha de publicación y/o que la versión que tenemos coincide con la versión con la que se trabaja en los proyectos de *Programación de las Cosas* de SET VEINTIUNO. Estos datos se podrán comprobar en la página oficial de cada lenguaje de programación.

Bitbloq

Bitbloq es un *software* que nos permite programar placas basadas en Arduino. En este caso, se programa con intuitivos y sencillos bloques, que nos permiten modificar el código que generan los bloques. Por tanto, podemos iniciarnos en la programación por bloques de placas Arduino y, cuando dominemos este tipo de programación, iniciarnos en la programación por código sin cambiar de *software*.

Dispone de una versión *online*, que permite conectarse con una cuenta desde cualquier ordenador, pudiendo trabajar, así, desde el centro educativo y desde casa, y cuenta con un entorno social para compartir proyectos. Además, dispone de una versión *offline*, para poder ser usado en situaciones en las que no dispongamos de Internet o la conexión nos dé problemas.

Permite programar varios tipos de placas Arduino y está en continuo desarrollo, por lo que la lista de placas soportadas va creciendo poco a poco.

Es importante que los alumnos prueben si la programación es correcta con el botón *Verificar* (con el símbolo ✓) antes de cargarla en la placa controladora para comprobar si funciona.

Acceso a la versión *online*

1. Acceder a la página <http://bitbloq.bq.com/#/> desde el navegador Google Chrome. Bitbloq también funciona en otros navegadores. Sin embargo, está optimizado para que sea compatible 100 % con Google Chrome, por lo que recomendamos su uso.
2. Pinchar en la opción *Registrarme*, situada en el menú superior, concretamente en la parte derecha.
3. Rellenar el nombre de usuario, la dirección de correo electrónico y la contraseña, así como la fecha de nacimiento. También podemos entrar con nuestro perfil, si tenemos, de Facebook o de Google+.
4. Leer las condiciones generales y políticas de privacidad, y marcar la casilla antes de pulsar *Registrarme*.

5. De forma opcional, marcar la casilla *Soy profesor y/o quiero recibir noticias, consejos y avisos de la aplicación* si queremos recibir noticias.
6. Entrar en la página de desarrollo, en la que podemos crear nuevos proyectos, ver los proyectos creados y nuestros proyectos compartidos.
7. Acceder a la pestaña *Explora* del menú superior, si queremos ver proyectos de la comunidad.

Instalación de la versión *offline*

1. Acceder a la página en <http://bitbloq.bq.com/#/offline>.
2. Pinchar en la opción *Descargar Bitbloq offline* para nuestro sistema operativo.
3. Tras la descarga, descomprimir la carpeta comprimida *zip* en una carpeta de usuario, como, por ejemplo, Mis documentos o el propio escritorio. Si colocamos la carpeta del programa en un directorio que no sea del usuario, puede que este no funcione correctamente al no disponer de los permisos de administrador adecuados.
4. Acceder al programa haciendo doble clic sobre el ejecutable *Bitbloq*.
5. El programa es portable, por lo que no requiere instalación. Podemos copiar la carpeta del programa a una memoria USB y pegarlo en la carpeta de usuario de cada ordenador de nuestra clase. Siempre debemos pegar y copiar la carpeta a nuestro ordenador, en vez de ejecutar directamente el programa desde el USB. Ejecutar el programa directamente desde un USB provocará que el ordenador tarde mucho más tiempo en iniciar el programa.
6. Si al intentar cargar un programa a nuestra placa controladora experimentamos problemas, debemos consultar el archivo PDF *¿No te detecta la placa?Motherboard troubleshooting.pdf* incluido en la carpeta del programa que se descargará cuando nos bajemos *Bitbloq offline*.

Formación sobre la herramienta

<http://diwo.bq.com/course/aprende-robotica-y-programacion-con-bitbloq-2/>

Formato de los programas

Encontraremos los archivos fuente de los programas de Bitbloq que se utilizan en los proyectos de *Programación de las Cosas* en la sección correspondiente de su área privada del entorno web setveintiuno.com. Estos archivos tienen la siguiente extensión: *.json*.

Scratch

Este *software*, creado por el MIT (Massachusetts Institute of Technology), tiene como objetivo programar animaciones y videojuegos. Ha ido ganando peso en el entorno educativo por su facilidad de uso y su potencial como herramienta de programación mediante sencillos bloques.

Es una herramienta gratuita que tiene versión *online* y *offline*, al igual que Bitbloq, haciendo de él un programa idóneo para aprender a programar.

Como inconveniente de este *software*, cabe destacar que es uno de los pocos programas usados que no tiene una aplicación directa en entornos profesionales o definitivos. Lo que programamos en Scratch no tiene sitio fuera de Scratch, por lo que un videojuego trabajado con esta herramienta solo podrá ser utilizado con la misma.

Acceso a la versión *online*

1. Acceder a la página del MIT en <http://scratch.mit.edu>.
2. Pinchar en la opción *Únete a Scratch*, situada en el menú superior, concretamente en la parte derecha.
3. Elegir un nombre de usuario y una contraseña. Debemos intentar que no sea la misma que la del correo que nos solicitará más tarde.
4. Rellenar la información personal que solicita.
5. Introducir una cuenta de correo electrónico. Si no tenemos correo corporativo en el centro, pediremos a los alumnos que traigan una cuenta creada por sus padres o tutores en casa.
6. Pulsar el botón *OK Vamos!* de la última pantalla de bienvenida.
7. Confirmar nuestra cuenta de Scratch desde el correo electrónico.
8. Cerrar todas las ventanas del programa, volver a abrir la página y pinchar en la opción *Ingresar*, situada en el menú superior, concretamente en la parte derecha.
9. Introducir nuestro usuario y contraseña, y acceder a la página de desarrollo de Scratch, en la que podremos crear nuestros proyectos, compartirlos y ver los proyectos compartidos en la comunidad.

Instalación de la versión *offline*

1. Acceder a la página del MIT en <http://scratch.mit.edu>.
2. Ir al *footer* de la página (a la parte inferior) y pinchar en *Editor sin Conexión*, en la parte de *Soporte*.
3. Tendremos que descargar los archivos de Adobe AIR y Editor Offline de Scratch correspondientes para nuestro sistema operativo.
4. Se descargarán dos archivos ejecutables (.exe) que debemos guardar en una memoria USB.
5. Ejecutar en cada uno de los ordenadores del aula los archivos descargados en nuestra memoria USB (Adobe AIR y Editor Offline de Scratch) para proceder a la instalación.

Formación sobre la herramienta

<http://diwo.bq.com/course/curso-de-scratch/>

Formato de los programas

Encontraremos los archivos fuente de los programas de Scratch que se utilizan en los proyectos de *Programación de las Cosas* en la sección correspondiente de su área privada del entorno web setveintiuno.com.

Estos archivos tienen la siguiente extensión: .sb2.

Tinkercad

Dentro de los programas enfocados a iniciarse en el diseño 3D, de una forma sencilla e intuitiva, encontramos Tinkercad. Se trata de un *software* muy fácil de usar en el que, sin aprendizaje previo, cualquiera puede empezar a diseñar de una manera bastante creativa y visualmente atractiva. Se trabaja mediante menús de formas prediseñadas que se pueden redimensionar y reubicar en un espacio de trabajo utilizando simplemente el ratón.

Nos permite exportar nuestros diseños para ser impresos o para ser introducidos en entornos como Minecraft, además de poder compartirlos en el entorno de trabajo virtual o almacenarlos en nuestra nube personal.

Debemos tener cuidado al diseñar con este *software*, ya que no exige incluir las medidas y proporciones de las piezas diseñadas, por lo que puede dar problemas al imprimir.

Se trata de un *software online* que requiere registrarse y, por tanto, autorización previa de padres o tutores.

Nota

Este programa tiene la interfaz en inglés, así que, al abrirlo, los alumnos verán las ilustraciones en este idioma.

Acceso a Tinkercad

1. Acceder al entorno web <https://www.tinkercad.com/>.
2. Pinchar en la opción *Sign up*, situada en el menú superior, concretamente en la parte derecha.
3. Elegir nuestro país y seleccionar la fecha de nuestro nacimiento.
4. Escribir un nombre de usuario, una contraseña y una cuenta de correo electrónico.
5. Entrar en la página de desarrollo, en la que podemos crear nuevos proyectos, ver los proyectos creados y nuestros proyectos compartidos.

Formación sobre la herramienta

<http://diwo.bq.com/course/tinkercad-curso>

Formato de los programas

Encontraremos los diseños 3D realizados con Tinkercad que se utilizan en los proyectos de *Programación de las Cosas* en la sección correspondiente del entorno web setveintiuno.com.

Como Tinkercad no permite descargar archivos editables, estos diseños están en el formato genérico STL y tienen la siguiente extensión: .stl.

Slic3r

Todo diseño en 3D que vaya a ser impreso debe pasar un proceso previo de laminado, en el que un *software* convertirá el diseño en diferentes capas que pueda crear una impresora 3D.

Slic3r se posiciona como una herramienta idónea para imprimir, pues es un *software* libre, que permite que diferentes personas y/o empresas estén siempre actualizando sus versiones para soportar más máquinas e impresoras.

Existe algún *software* de laminado gratuito, como Cura, que se podría utilizar de manera similar a este *software*.

Instalación de Slic3r

1. Acceder a la página del programa <http://slic3r.org/download>, concretamente a la pestaña de descargas (*downloads*).
2. Descargar Slic3r desde la opción 1, usando *packs* precompilados. También podemos descargarnos el código fuente más actual y compilarlo. Sin embargo, recomendamos la opción 1.
3. Desde la opción 1, hacer clic en nuestro sistema operativo para descargarnos la versión adecuada.
4. En el caso de Windows, aparece un .zip correspondiente a la versión de 64 bits y otro a la de 32 bits (nombrada con x86). Si no sabemos cómo es la arquitectura de nuestro ordenador, debemos descargarnos el de x86, ya que funciona con todos los dispositivos.
5. Descomprimir el .zip y ejecutar la aplicación Slic3r.
6. En el menú que aparece junto a la pantalla de Slic3r, podemos elegir los valores de las diferentes características para la impresión.
7. Otra opción para no elegir uno a uno los valores de impresión es precargar el perfil de nuestra impresora. En este caso, buscaremos el modelo de nuestra impresora en nuestro navegador o directamente en la web de la marca de la impresora, junto a la palabra Slic3r, para descargarnos el perfil.

Formación sobre la herramienta

<http://diwo.bq.com/slicer-tutorial>

Formato de los programas

El programa Slic3r genera archivos en formato .gcode, que contienen las instrucciones que necesita la impresora 3D para poder imprimir nuestros diseños. Estos archivos no se proporcionan en la web, puesto que no son generales, sino que dependen de cada modelo de impresora.

OpenSCAD

OpenSCAD es un *software* de diseño 3D mediante el uso de un lenguaje textual (código). Es un *software* que trabaja el diseño a través de código. Por tanto, es una herramienta magnífica para aprender a diseñar e iniciarse en algunos conceptos de programación. Tiene algunas características propias del *software*, como es la creación de módulos (que corresponden a las funciones en programación). Gracias a ello, nos permite realizar diseños más complejos o parametrizados.

Se trata de un programa *offline* que se podrá descargar y ejecutar desde el ordenador, sin necesidad de registro.

Inicialmente puede parecer que su uso, mediante código de programación, es una barrera de acceso al diseño, pero la realidad es que tiene muchas ventajas al usarlo en entornos educativos. Esto se debe a que requiere que seamos conscientes de los parámetros, o magnitudes de las figuras geométricas, y de su posición espacial, por lo que desde el inicio con este programa los alumnos deben aprender a dominar la geometría.

Este aspecto es clave para desarrollar la capacidad espacial de los usuarios y ayudarlos a visualizar mentalmente cada una de las tres dimensiones, además de que comprendan mejor cómo trabajar en 3D.

Instalación de OpenSCAD

1. Acceder a la pestaña descargas (*downloads*) de la página del programa:
<http://www.openscad.org/downloads.html>
2. Elegir el descargable para nuestro sistema operativo, de modo que podamos descargarnos la versión adecuada. En el caso de Linux, tendremos que copiar en nuestro ordenador el código que aparece.
3. En el caso de Windows, tenemos dos opciones:
 - a. exe installer. Se descarga un .exe para instalar OpenSCAD en el ordenador. Se recomienda usar el programa instalado para mejorar el rendimiento y evitar posibles problemas.
 - b. zip package. Se descarga un .zip con diversos archivos y un ejecutable que no requiere instalación.
4. Con el programa instalado, debemos pinchar sobre *Nuevo* para abrir un nuevo proyecto en el programa, o sobre *Abrir* para abrir un proyecto que tengamos guardado en nuestro ordenador.

Formación sobre la herramienta

<http://diwo.bq.com/course/curso-de-iniciacion-al-diseno-3d-con-openscad-por-obijuan/>

Formato de los programas

Encontraremos los diseños 3D realizados con OpenSCAD que se utilizan en los proyectos de *Programación de las Cosas* en la sección correspondiente del entorno web setveintiuno.com.

Estos archivos tienen la siguiente extensión: .scad.

FreeCAD

FreeCAD es un programa de diseño 3D geométrico, es decir, se diseña a partir de formas geométricas. Su mayor potencial parte de ser *software* libre, ya que tiene una gran versatilidad de uso. Esto es debido, en gran parte, a sus bancos de trabajo, que nos permiten seleccionar grupos de herramientas de diseño en función de lo que tenemos que hacer.

FreeCAD es un *software* que se muestra parejo a programas técnicos profesionales, y su uso puede satisfacer tanto a un nivel de iniciación en el diseño como a un nivel avanzado e, incluso, a nivel profesional, por lo que la línea de aprendizaje en el campo de diseño con este *software* se puede extender enormemente.

Instalación de FreeCAD

1. Acceder a la página del programa <http://www.freecadweb.org/>.
2. Pinchar en el menú lateral de la izquierda en la pestaña descargas (*downloads*).
3. Elegir el descargable para nuestro sistema operativo, de modo que podamos descargarlos la versión adecuada.
4. En el caso de Windows, aparece un .exe correspondiente a la versión de 64 bits y otro a la de 32 bits. Si no sabemos cómo es la arquitectura de nuestro ordenador, debemos descargarlos el .exe de 32 bits, ya que funciona con todos los dispositivos.
5. Leer las notas para usuarios de cada sistema operativo, si hubiese algún problema en la instalación del *software*. Estas notas se encuentran en la misma página en la que se encuentran los archivos de descarga.

Formación sobre la herramienta

<http://diwo.bq.com/course/curso-de-introduccion-a-freecad/>

Formato de los programas

Encontraremos los diseños 3D realizados con FreeCAD que se utilizan en los proyectos de *Programación de las Cosas* en la sección correspondiente de su área privada del entorno web setveintiuno.com.

Estos archivos tienen la siguiente extensión: .fcstd.

Python

Python es un lenguaje de programación por código de propósito general, y es perfecto para facilitar el primer contacto con un lenguaje de este tipo porque su sintaxis es muy sencilla.

Esto significa que, a diferencia de otros lenguajes como C, el alumno puede leer el código y entender lo que está creando en cada momento, de manera intuitiva, sin grandes problemas en configuraciones iniciales.

Además, Python es muy común en el mundo de la ciencia e ingeniería, por lo que es un lenguaje muy interesante para aprender en la medida en la que podemos mostrar su aplicación en el mundo profesional.

Se programa de forma *offline*, a través del IDLE de Python o de cualquier editor de texto, como pueden ser WordPad en Windows o Gedit en Linux. Nuestra recomendación es utilizar cualquier editor que marque la programación con diferentes colores, de modo que sea más fácil aprender y realizar búsquedas en el código.

Dependiendo del tipo de librerías que se utilice en cada proyecto, se deberá utilizar Python 2 o Python 3. Ambas versiones de Python coexisten en el tiempo para garantizar la mayor compatibilidad posible con todas las herramientas existentes en la red, siendo dos versiones vivas y actualizadas.

De cara al aprendizaje del usuario, las diferencias son mínimas y se refieren a la redacción de algunos comandos básicos como *print*, que en Python 3 requiere paréntesis, y, sobre todo, en la forma de ejecutar el programa desde una terminal en Linux.

Así, en Python 2, el programa se ejecutaría de la siguiente forma:

```
python2 nombredelprograma.py
```

Mientras que en Python 3 sustituiríamos la orden python2 por python3:

```
python3 nombredelprograma.py
```

En muchos casos, veremos que se pueden ejecutar programas Python escribiendo simplemente *python nombredelprograma.py*, sin especificar la versión de la que se trata. Esta forma abreviada corresponderá, en cada sistema operativo, con la versión de Python que venga instalada por defecto (por ejemplo, en Ubuntu 14.04 se corresponde con Python 2, mientras que en Ubuntu 16.04 sería con Python 3).

En cuanto a los proyectos de Python que se han incluido en la propuesta pedagógica de *Programación de las Cosas*, cada uno de ellos se ha creado con unos propósitos concretos, aunque buscando complementariedad entre ellos.

Conocimiento general de Python y de sus posibilidades

INTRODUCCIÓN A PYTHON Y TRABAJO EN SU TERMINAL

Creación de un programa básico que incluye la creación de listas, el manejo de archivos y el uso de funciones básicas. Se trata de un programa para utilizar sin necesidad de conexión a Internet, es decir, no interacciona con la información de la World Wide Web.

Utilización de Python para fines web

UTILIZACIÓN DE FRAMEWORKS Y CREACIÓN DE WEBS

El uso de *frameworks* en programación es habitual. Los frameworks nos ayudan a crear programas sin necesidad de crearlos desde el inicio. Estos *frameworks* proporcionan un conjunto de librerías que nos permiten realizar tareas tediosas de una forma sencilla, con pocas líneas de código. Permite a una persona, que no tenga mucha experiencia en programación, crear una web visual e interactiva, facilitando el desarrollo de competencias de representación de información.

WEB SCRAPING

La cantidad de información que hay disponible en la web hace necesario el desarrollo de competencias que permitan hacer frente a la recopilación y el análisis de información de una forma eficaz. La propuesta pedagógica plantea aprender a recopilar información de forma automática y masiva, extrayéndola de entornos web, a partir de un sencillo programa creado con Python.

Cómo instalar Python

1. Entrar en <https://www.python.org/>.
2. Ir a la sección descargas (*downloads*).
3. Elegir la versión del programa que queremos instalar. Generalmente aparecen dos pestañas, una con la última versión de Python 2 y otra con la última versión de Python 3. Al seleccionarla, se descargará la versión adecuada para nuestro sistema operativo.
4. Si queremos descargar Python para un sistema operativo diferente al que estemos usando (por ejemplo, para llevarlo a otro ordenador), podemos elegir otro instalable bajo los botones de descarga de Python 2 y Python 3.
5. En el caso de estar ejecutando el programa en Linux, lo mejor será instalar Python desde terminal. Por ejemplo, en la distribución Ubuntu, podemos abrir la terminal pulsando *Ctrl+Alt+t* e instalar Python escribiendo *sudo apt-get install python*, que instalará Python 2 o Python 3 dependiendo de nuestra distribución concreta (normalmente este paso no es necesario porque suele venir instalado por defecto). Para instalar otras versiones, tendremos que escribir *sudo apt-get install pythonX*, donde la X indica el número de versión. En el caso de no instalarlo desde terminal es muy probable que se produzcan problemas de compatibilidad de funciones de Python al programar determinadas acciones.
6. Para utilizar Python en Windows o Mac OS basta con abrir el programa descargado desde su página web.
7. En Linux usaremos Python desde la terminal escribiendo *python*, *python2* o *python3* para acceder al programa (dependiendo de la versión que necesitemos usar). Si queremos usar la IDLE de Python, debemos instalarla desde terminal (en la distribución Ubuntu se hará escribiendo *sudo apt-get install idle*). Una vez instalada, podemos acceder a ella desde la terminal escribiendo *idle* o buscándola entre los programas instalados con el buscador de nuestro sistema operativo.

Formación sobre la herramienta

<http://diwo.bq.com/course/python-curso>

Formato de los programas

Encontraremos los archivos fuente de los programas de Python que se utilizan en los proyectos de *Programación de las Cosas* en la sección correspondiente de su área privada del entorno web setveintiuno.com.

Estos archivos tienen la siguiente extensión: `.py`.

Arduino

Arduino está basado en C++, un lenguaje de programación muy utilizado actualmente. Arduino se utiliza para programar electrónica, es decir, programar una salida física que interactúa con el entorno. Por ello, aprender a programar mediante código Arduino tiene dos líneas de interés: la primera es aprender un lenguaje textual (código) similar al usado en entornos profesionales, y la segunda es aprender a utilizar código enfocado a la electrónica y la robótica.

Para programar en Arduino, necesitamos descargar su IDE. Por tanto, se trata de una programación *offline*. La interfaz de uso del IDE de Arduino es bastante simple y soporta la gran mayoría de placas, basadas en Arduino.

Para programar *online*, se puede utilizar Bitbloq. Recomendamos la utilización de Bitbloq, también en su versión *offline*, ya que podemos programar en bloques cuando dudemos del código y visualizar el código en la pestaña creada para ello. Esta es una forma más sencilla de familiarizarse con la programación por código.

Para usar Bitbloq, podemos consultar el apartado anterior en el que se explica cómo acceder a Bitbloq *online* o cómo descargar la versión *offline*. Si preferimos usar directamente el IDE de Arduino, podemos acceder a su página web.

Instalación del IDE de Arduino

1. Acceder a la página del programa <https://www.arduino.cc/>.
2. Pinchar en la pestaña descargas (*downloads*) en el menú superior.
3. En la sección de descargas, elegiremos el descargable adecuado a nuestro sistema operativo para descargarnos la versión adecuada.
4. En el caso de Windows, tenemos dos opciones:
 - a. installer. Se descarga un .exe para instalar OpenSCAD en el ordenador. Se recomienda usar el programa instalado para mejorar el rendimiento y evitar posibles problemas.
 - b. zip file. Se descarga un .zip con diversos archivos y un ejecutable que no requiere instalación. Para trabajar en este modo, debemos descomprimir la carpeta .zip.
5. En el caso de Linux, aparece un archivo correspondiente a la versión de 64 bits y otro a la de 32 bits. Si no sabemos cómo es la arquitectura de nuestro ordenador, nos descargaremos el .exe de 32 bits, ya que funciona con todos los dispositivos.

Formación sobre la herramienta

<http://diwo.bq.com/course/curso-de-programacion-de-robots-con-arduino/>

Formato de los programas

Encontraremos los archivos fuente de los programas de Arduino que se utilizan en los proyectos de *Programación de las Cosas* en la sección correspondiente de su área privada del entorno web setveintiuno.com.

Estos archivos tienen la siguiente extensión: .ino.

Otras cuestiones

De naturaleza legal

Los entornos de programación utilizados en la propuesta pedagógica *Programación de las Cosas* son, generalmente, entornos *online* en los que se requiere estar registrado con una cuenta de correo electrónico.

Cuando se trata de menores de 14 años, no es legal que tengan una cuenta personal de correo electrónico. Para estos casos, las direcciones de correo electrónico que se utilicen habrán sido proporcionadas por sus padres o tutores legales concediendo, por tanto, autorización expresa para ello.

Así mismo, se informa de que todos los contenidos que forman parte de la propuesta pedagógica de *Programación de las Cosas* están sujetos a derechos de autor. De conformidad con la normativa vigente, y a excepción de los eventuales acuerdos de licencia que pudiera celebrar Grupo Santillana, no está permitida la reproducción total o parcial de la obra sin el consentimiento de Grupo Santillana prestado por escrito.

Grupo Santillana se exime de toda responsabilidad en cuanto a la existencia o exactitud de las URL relacionadas con entornos web de terceros referenciadas en esta propuesta pedagógica, y no garantiza que el contenido de dichos entornos web sea exacto y/o apropiado.

Consultas y preguntas más frecuentes

- **¿Para qué edades es adecuado *Programación de las Cosas*?**
La propuesta pedagógica abarca proyectos para alumnos de 10 a 16 años.
- **¿Voy a poder contar con algún medio de contacto para consultar dudas de tipo pedagógico al respecto de *Programación de las Cosas*?**
Sí. Por favor, ponte en contacto con nosotros a través de **setveintiuno@santillana.com**.
- **Tengo una pregunta específica acerca de un proyecto/sección de un proyecto, ¿cómo y con quién debo contactar?**
Por favor, ponte en contacto con nosotros a través de **setveintiuno@santillana.com**.
- **¿Qué promedio de tiempo se aconseja para la preparación previa de los proyectos antes de su implementación en el aula?**
Dependiendo del alcance y naturaleza de cada proyecto, será recomendable un tiempo determinado de preparación previa a la implementación en el aula. Se recomienda consultar, con tiempo, las secciones de esta guía relativas a los distintos lenguajes de programación, el entorno web y los materiales de uso en el aula.
- **¿Cuenta este proyecto con un servicio de acompañamiento presencial para los docentes?**
Sí. Para más información, se puede consultar **setveintiuno@santillana.com**.

- ¿Existe alguna forma para conocer a otros docentes ya clientes de *Programación de las Cosas* interesados en compartir sus experiencias en el proceso de enseñanza-aprendizaje?
Sí. Por favor, consulta la información actualizada sobre la comunidad de setveintiuno.com.

Técnicas

- ¿Necesito instalarme un entorno de programación para todos los proyectos?
Sí. Se recomienda consultar, con tiempo, las secciones de esta guía relativas a los distintos lenguajes de programación, así como los materiales de uso en el aula.
- ¿Se pondrá a nuestra disposición algún medio de contacto para poder referir y solventar incidencias de tipo tecnológico a la hora de la planificación curricular de un proyecto?
Sí. Para más información, se puede consultar en setveintiuno@santillana.com.
- ¿Es posible realizar e implementar en el aula los proyectos mediante *tablets*?
No. Se recomienda consultar, con tiempo, las secciones de esta guía relativas a los distintos lenguajes de programación.
- ¿Todos los proyectos requieren la adquisición de un kit de electrónica?
No, no todos los proyectos necesitan un kit de electrónica.
- ¿Es estrictamente necesario contar con, al menos, una impresora 3D para realizar los proyectos?
No, no todos los proyectos necesitan la utilización de una impresora 3D. Se recomienda consultar, con tiempo, la sección de esta guía que describe en qué consiste la propuesta pedagógica de *Programación de las Cosas*.

Comerciales

- ¿Qué posibilidades hay de adquirir los kits de electrónica de forma independiente a los materiales *offline* y *online* para el alumno y el docente?
Para más información, se puede consultar en setveintiuno@santillana.com.
- ¿Cuál es el número de kits de electrónica recomendado para poder realizar cualquier proyecto de *Programación de las Cosas* que lo exija?
Se recomiendan seis kits de electrónica para un aula de 24 alumnos.

ANEXO



Anexo

PROYECTO	COMPETENCIAS	ÁREAS CURRICULARES	EDAD
¿Existieron los dragones?	<ul style="list-style-type: none"> • Desarrollo del pensamiento matemático • Pensamiento crítico y lógico • Creatividad • Habilidades de trabajo en equipo • Habilidades emocionales 	<ul style="list-style-type: none"> • Ciencias Naturales • Matemáticas • Ciencias Sociales 	10-12 años
¿Cuál sería tu mascota ideal?	<ul style="list-style-type: none"> • Comunicación lingüística y escucha • Creatividad • Pensamiento crítico y lógico • Competencia cultural y artística 	<ul style="list-style-type: none"> • Ciencias Naturales • Lógica • Matemáticas • Plástica 	10-12 años
¿Mejor solo que acompañado?	<ul style="list-style-type: none"> • Habilidades de trabajo en grupo • Creatividad • Responsabilidad • Comunicación lingüística y escucha • Tratamiento de la información 	<ul style="list-style-type: none"> • Historia • Inglés • Lengua 	10-12 años
¿Me ayudas a salvar el mundo?	<ul style="list-style-type: none"> • Comunicación lingüística y escucha • Pensamiento crítico y lógico • Desarrollo de hábitos de trabajo 	<ul style="list-style-type: none"> • Ciencias Sociales • Matemáticas • Plástica 	10-12 años
¿Por qué no ir a más velocidad?	<ul style="list-style-type: none"> • Colaboración • Desarrollo de pensamiento matemático • Conocimiento e interacción con el mundo físico • Competencia social y ciudadana 	<ul style="list-style-type: none"> • Física • Matemáticas • Ciencias Sociales 	10-12 años
¿Podemos ser directores de cine?	<ul style="list-style-type: none"> • Comunicación lingüística y escucha • Creatividad • Respeto y cuidado del material • Competencia social y ciudadana • Competencia cultural y artística 	<ul style="list-style-type: none"> • Plástica • Ciencias Sociales • Lengua y Literatura 	10-12 años
¿A qué jugamos ahora?	<ul style="list-style-type: none"> • Colaboración • Comunicación lingüística y escucha • Respeto y cuidado del material • Creatividad • Pensamiento crítico y lógico 	<ul style="list-style-type: none"> • Historia • Matemáticas 	10-12 años
¿Podemos crear una presa hidroeléctrica reversible?	<ul style="list-style-type: none"> • Conocimiento de conceptos físicos • Conocimiento e interacción con el mundo físico • Competencia social y ciudadana • Competencia cultural y artística 	<ul style="list-style-type: none"> • Ciencias Naturales • Tecnología 	10-12 años
¿Cuál fue el primer deporte de la historia?	<ul style="list-style-type: none"> • Creatividad • Respeto y cuidado del material • Pensamiento matemático • Colaboración • Responsabilidad 	<ul style="list-style-type: none"> • Deporte • Arquitectura • Matemáticas • Música 	10-12 años
¿Somos capaces de distinguir el sonido de una nota?	<ul style="list-style-type: none"> • Habilidades de trabajo en equipo • Creatividad • Conocimiento de conceptos físicos • Pensamiento crítico y lógico • Resolución de problemas 	<ul style="list-style-type: none"> • Música • Física • Ciencias Naturales 	10-12 años
¿Podemos cruzar la calle, agente Robi?	<ul style="list-style-type: none"> • Comunicación lingüística y escucha • Creatividad • Pensamiento crítico y lógico • Conocimiento e interacción con el mundo físico 	<ul style="list-style-type: none"> • Ciencias Sociales • Tecnología • Matemáticas (Geometría) • Plástica 	10-12 años

PROYECTO	COMPETENCIAS	ÁREAS CURRICULARES	EDAD
¿Hay día y noche en el espacio?	<ul style="list-style-type: none"> • Conocimiento de conceptos físicos • Resolución de problemas • Creatividad • Habilidades de trabajo en equipo • Tratamiento de la información 	<ul style="list-style-type: none"> • Ciencias Naturales • Matemáticas • Plástica 	12-14 años
¿Cómo nos comunicamos?	<ul style="list-style-type: none"> • Habilidades de trabajo en equipo • Creatividad • Conocimiento de conceptos físicos • Pensamiento crítico y lógico • Comunicación lingüística y escucha • Tratamiento de la información y competencia digital 	<ul style="list-style-type: none"> • Ciencias Sociales • Lengua • Astronomía 	12-14 años
¿Se puede ensuciar el espacio?	<ul style="list-style-type: none"> • Resolución de problemas • Pensamiento crítico y lógico • Conocimiento de conceptos físicos • Conocimiento e interacción con el mundo físico • Competencia social y ciudadana • Creatividad 	<ul style="list-style-type: none"> • Ciencias Naturales • Matemáticas • Tecnología • Astronomía • Inglés 	12-14 años
¿Queremos vivir en el espacio exterior?	<ul style="list-style-type: none"> • Habilidades de trabajo en equipo • Creatividad • Conocimiento de conceptos físicos • Comunicación lingüística y escucha • Desarrollo de pensamiento matemático • Resolución de problemas 	<ul style="list-style-type: none"> • Historia • Física • Matemáticas • Economía • Ciencias Naturales 	12-14 años
¿Será imposible colonizar otros planetas?	<ul style="list-style-type: none"> • Resolución de problemas • Habilidades de trabajo en equipo • Conocimiento de conceptos físicos • Pensamiento crítico y lógico • Comunicación lingüística y escucha • Tratamiento de la información y competencia digital 	<ul style="list-style-type: none"> • Ciencias Naturales • Física • Matemáticas 	12-14 años
¿Hay distintas formas de explorar otros planetas?	<ul style="list-style-type: none"> • Habilidades de trabajo en equipo • Responsabilidad • Pensamiento crítico y lógico • Comunicación lingüística y escucha • Tratamiento de la información y competencia digital 	<ul style="list-style-type: none"> • Ciencias Naturales • Historia • Plástica 	12-14 años
¿Nos atrevemos a diseñar un casco espacial?	<ul style="list-style-type: none"> • Aprender a aprender • Resolución de problemas • Habilidades de trabajo en equipo • Conocimiento e interacción con el mundo físico • Pensamiento crítico y lógico • Tratamiento de la información y competencia digital 	<ul style="list-style-type: none"> • Ciencias Naturales • Física • Matemáticas 	12-14 años
¿Por qué comprarlo si puedo crearlo?	<ul style="list-style-type: none"> • Aprender a aprender • Comunicación lingüística y escucha • Resolución de problemas • Dotes de innovación • Habilidades de trabajo en equipo • Creatividad • Autoconfianza • Pensamiento crítico y lógico • Desarrollo de pensamiento matemático • Emprendeduría, inteligencia ejecutiva • Conocimiento de conceptos físicos • Inteligencia emocional • Concentración, respeto y cuidado del material • Tratamiento de la información y competencia digital • Competencia social y ciudadana 	<ul style="list-style-type: none"> • Matemáticas • Plástica 	14-16 años

PROYECTO	COMPETENCIAS	ÁREAS CURRICULARES	EDAD
¿Es Internet la respuesta?	<ul style="list-style-type: none"> • Tratamiento de la información y la competencia digital • Resolución de problemas • Pensamiento crítico y lógico • Autonomía e iniciativa personal 	<ul style="list-style-type: none"> • Matemáticas • Lengua • Geografía 	14-16 años
¿Puede la tecnología ayudar al cuidado de las plantas?	<ul style="list-style-type: none"> • Colaboración • Competencia social y ciudadana • Creatividad • Desarrollo del pensamiento matemático • Pensamiento crítico y lógico 	<ul style="list-style-type: none"> • Biología • Matemáticas • Física 	14-16 años
¿Seremos capaces de adelantarnos al futuro?	<ul style="list-style-type: none"> • Aprender a aprender • Resolución de problemas • Habilidades de trabajo en equipo • Conocimiento de conceptos físicos • Pensamiento crítico y lógico • Comunicación lingüística y escucha • Tratamiento de la información y competencia digital 	<ul style="list-style-type: none"> • Ciencias Naturales • Física • Matemáticas 	14-16 años
¿Puedo vestirme con ropa inteligente?	<ul style="list-style-type: none"> • Emprendimiento • Dotes de innovación • Competencia cultural y artística • Competencia social y ciudadana • Resolución de problemas 	<ul style="list-style-type: none"> • Matemáticas • Economía • Plástica • Inglés 	14-16 años
¿Podemos crear un mundo al alcance de todos?	<ul style="list-style-type: none"> • Resolución de problemas • Habilidades de trabajo en equipo • Conocimiento de conceptos físicos • Pensamiento crítico y lógico • Comunicación lingüística y escucha • Tratamiento de la información y competencia digital • Competencia social y ciudadana 	<ul style="list-style-type: none"> • Física • Matemáticas • Educación en Valores 	14-16 años
¿Podemos crear objetos que ayuden a mejorar nuestro día a día?	<ul style="list-style-type: none"> • Aprender a aprender • Pensamiento crítico y lógico • Tratamiento de la información y competencia digital • Desarrollo del pensamiento matemático • Dotes de innovación 	<ul style="list-style-type: none"> • Ciencias Naturales • Física • Matemáticas 	14-16 años
¿Innovamos a la hora de levantarnos?	<ul style="list-style-type: none"> • Perseverancia • Dotes de innovación • Colaboración • Autonomía e iniciativa personal • Desarrollo del pensamiento matemático 	<ul style="list-style-type: none"> • Matemáticas • Plástica • Geografía • Economía • Lengua 	14-16 años

PROGRAMAS	CUESTIONES TECNOLÓGICAS		
	Tipo de extensión	Otras características	Requisitos técnicos mínimos
Bitbloq	.json y .ino (código Arduino)	<ul style="list-style-type: none"> • Editable: .json • Exportable: .ino (código Arduino) 	<ul style="list-style-type: none"> • Recomendable uso del navegador Google Chrome. • Linux: certificado en Ubuntu 14.04 de 32/64 bits. • Mac: Mac OS X 10.10 y superiores de 64 bits. • Windows: Windows 7, 8.1 y 10 de 32/64 bits.
Bitbloq <i>offline</i> portable	.json	<ul style="list-style-type: none"> • Editable: .json • Exportable: .ino (código Arduino) 	<ul style="list-style-type: none"> • Linux (Ubuntu 12.04 y superior Fedora 21 Debian 8).
Scratch	.sb2	<ul style="list-style-type: none"> • Editable: .sb2 • No exportables 	<ul style="list-style-type: none"> • Versión <i>online</i>. • Recomendable navegador Chrome V35, Firefox V31. Internet Explorer V9 o versiones posteriores. • Adobe Flash Player versión 10.2 o posterior instalada. • Pantallas de resolución mínima 1024x768. • Versión <i>offline</i>. • Las últimas versiones.
OpenSCAD	.scad, .stl, .off, .amf, .dxf, .svg, .csg, .png	<ul style="list-style-type: none"> • Editable: .scad • Exportables: .stl (el más común), .off, .amf, .dxf, .svg, .csg, .png (como imagen) 	<ul style="list-style-type: none"> • Linux: compatible con las principales distribuciones del mercado. • Mac: Mac OS X 10.7 o posterior. • Windows: Windows XP o superior en versiones 32/64 bits.
Tinkercad	.stl, .obj, .x3d, .vrmf, .svg	<ul style="list-style-type: none"> • Editable: no existe • Exportables: .stl (el más común), .obj, .x3d, .vrmf, .svg 	<ul style="list-style-type: none"> • Recomendable navegador Chrome V10, Firefox V4 o versiones posteriores.
Slic3r	.stl, .gcode	<ul style="list-style-type: none"> • Editable: no existe • Exportables: .stl, .obj, .amf, .xrmf, .gcode 	<ul style="list-style-type: none"> • Linux: compatible con las principales distribuciones del mercado. • Mac: Mac OS X 10.7 o posterior. • Windows: Windows 7 y versiones posteriores.
Python	.py	<ul style="list-style-type: none"> • Editable: .py • Exportables: no necesarios 	<ul style="list-style-type: none"> • Lenguaje de programación compatible con las principales versiones de Linux, Mac y Windows.
Arduino	.ino	<ul style="list-style-type: none"> • Editable: .ino • Exportables: no necesarios 	<ul style="list-style-type: none"> • Dependiendo del IDE y su naturaleza <i>offline</i> u <i>online</i>. En general compatible con las principales versiones de Linux, Mac OS X y Windows.
FreeCAD	.fcstd	<ul style="list-style-type: none"> • Editable: .fcstd • Exportables: .stl y numerosos archivos de formato 3D del mercado 	<ul style="list-style-type: none"> • Compatible con las principales versiones de Linux, Mac OS X y Windows.
Cura	.stl, .gcode	<ul style="list-style-type: none"> • Editable: no existe • Exportables: .stl, .obj, .amf, .xml, .gcode 	<ul style="list-style-type: none"> • Compatible con las principales versiones de Linux, Mac OS X y Windows.

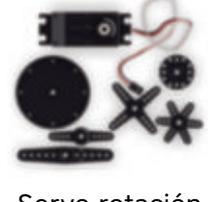
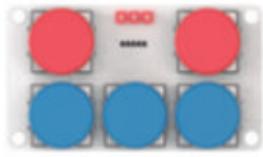
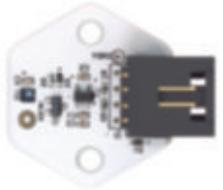
NOTAS DE AYUDA A LA LECTURA DE LA TABLA

IDE (Integrated Development Environment): entorno de programación o desarrollo. Un lenguaje de programación puede no estar sujeto a utilizarlo en un único programa, pueden existir distintas herramientas o programas distintos que permiten trabajar con él.

Archivo editable: archivo que permite acceder al código o bloques programados, o al diseño 3D creado. Su principal ventaja es que permite modificar y/o mejorar el trabajo realizado. Sería el equivalente a compartir un documento en formato Microsoft Word (.doc).

Archivo exportable: archivo que permite utilizar y compartir un programa o diseño 3D, pero que no ofrece la posibilidad de editar y/o mejorar el trabajo realizado. Sería el equivalente a compartir un archivo en formato PDF. Si bien se puede conseguir editar un PDF que no sea propio, normalmente es mucho más complejo y las opciones son mucho más limitadas que en el caso de un archivo editable del Microsoft Word.

Componentes del kit de electrónica

1  Portapilas	1  Placa controladora	2  Infrarrojo siguelíneas	1  Pulsador	2  Sensor de luz
1  Zumbador	1  Power supply	14  Extensión Wire 3p	1  Sensor ultrasónico	
1  Potenciómetro	1  LED azul	1  LED verde	2  Miniservo	2  Servo rotación continua
1  Cable USB	1  RGB LED	1  Zum bloq RTC	1  Zum bloq keypad	
1  Zum bloq humidity and temperature sensor	1  Zum bloq joystick	1  Zum bloq rotary encoder	1  Zum bloq LCD	
2  Zum bloq endstop	1  Zum bloq microphone	4  Extensión Wire 4P	2  Extensión Wire 5P	

© 2017 by Santillana Global, S. L.
Avda. de los Artesanos, 6
28760 Tres Cantos, Madrid

Printed in Spain

ISBN: 978-84-141-0822-2
CP: 839859
Depósito legal: M-3904-2017

ISBN: 978-84-141-0822-2



9 788414 108222

SET VEINTI UNO

setveintiuno.com

