Facultad de Educación

MÁSTER EN FORMACIÓN DEL PROFESORADO DE EDUCACIÓN SECUNDARIA

Análisis y propuesta de uso de la plataforma micro:bit para la enseñanza de la programación y la robótica en la Educación Secundaria Obligatoria

Analysis and proposal of micro:bit platform as a tool for teaching programming and robotics in Secondary Studies

Alumna: Carmen López de la Torre

Especialidad: Física, Química y Tecnología

Director: Ángel Cuesta García

Curso académico: 2017/2018

Fecha: 21 Junio 2018

Resumen

La programación y la robótica son dos materias que, en sintonía, están ganando un espacio significativo en las aulas. A través del trabajo de la algorítmica, y del diseño, construcción y programación de robots, el alumno adquiere nuevas habilidades tanto técnicas como sociales.

En la actualidad, estas disciplinas tienen como objetivo programar y utilizar placas programables para la realización de proyectos de robótica. Entre las más utilizadas están Arduino o Raspberry Pi. Sin embargo, es destacable el caso de Inglaterra, donde un gran número de empresas y organismos se unieron para el desarrollo de una placa, micro:bit, para ofrecer un sistema intuitivo y funcional para el trabajo de la programación y la robótica en las escuelas, obteniendo resultados muy positivos.

Este trabajo proporciona, en primer lugar, una visión del estado de la enseñanza de la programación y la robótica en las aulas. Se plantea además, un análisis y una comparativa de las placas y los lenguajes de programación más utilizados por el profesorado. A raíz de estos resultados, se describe la placa micro:bit y su potencial. Finalmente, como componente principal del trabajo, se plantea una propuesta de uso de esta placa como medio conductor para la enseñanza de la programación y la robótica en la Educación Secundaria Obligatoria.

Programación, robótica, micro:bit, secundaria

Abstract

Programming and robotics are two subjects that, together in harmony, are reaching a significant place in education. Through the usage of the algorithmic, and the design, construction and programming of robots, students acquire both technical and social abilities.

Nowadays, these disciplines aim to program and use programmable boards for the execution of robotic projects. Among the most used boards by teachers, Arduino or Raspberry Pi stand out over the others. However, the case of England is remarkable, where a big number of companies and organizations joined for the development of a programmable board, known as micro:bit, to provide an intuitive

and functional system for struggling with the programming and robotics in the schools, achieving very positive results.

This dissertation provides, in the first place, a general view of the status of the programming and robotics teaching at secondary schools. In addition, an analysis and a comparison of the most used boards and programming languages on this area are outlined. In the wake of these results, the micro:bit board and its potential are described. Finally, a proposal for the use of micro:bit as a driver for the teaching of programming and robotics in Secondary Education is provided.

Programming, robotics, micro:bit, secondary

Índice de contenidos

1. Introducción y Justificación

Cuando se habla de pensamiento computacional (PC), normalmente se piensa en programar, y más concretamente en escribir código. Sin embargo, el pensamiento computacional hace referencia a un conjunto de habilidades mucho más amplias. El PC ha tenido múltiples definiciones, entre las que podría destacarse la proporcionada por (Wing, 2010)[1], que lo define como el proceso de pensamiento involucrado en la formulación de problemas y sus soluciones, para que las soluciones sean representadas de forma que puedan ser eficientemente ejecutadas por un agente de procesamiento de información.

El pensamiento computacional debe entenderse como una herramienta más, al igual que lo son la lectura o las matemáticas. A través de esta herramienta los alumnos aprenden a enfrentarse a problemas dividiéndolos en tareas más pequeñas, más accesibles, que deberán resolver para poder llegar a la solución del problema inicial.

Con este objetivo, **la programación** ayuda a desarrollar el pensamiento computacional a través de las enseñanzas de la algorítmica y de sus fundamentos básicos. Así, el PC es algo mucho más cercano y aplicable no sólo al ámbito de la informática, sino a cualquier disciplina. El objetivo no es el de crear futuros ingenieros programadores, sino el de proporcionar a los estudiantes una herramienta más en su vida.

De igual forma, otra disciplina relacionada con el pensamiento computacional en el aula es **la robótica**. Esta materia, que va ganando aceptación según pasan los años, se trata de una materia interdisciplinar de carácter eminentemente práctico y que tiene como objetivo el diseño, construcción y programación de prototipos robóticos con fines pedagógicos. Su objetivo va más allá de que los alumnos construyan robots. Como describe (Educativa, 2011) y es citado por (Pittí Patiño, y otros, 2012), a través de la robótica se pretende trabajar con el alumnado, la adquisición de competencias básicas necesarias para la sociedad actual como son el aprendizaje colaborativo o la toma de decisiones en equipo entre otras. Además, como también describe (Pozo, 2005) en el mismo artículo,

[1] La definición hace referencia al trabajo de Jan Cuny, Larry Snyder, y Jeannette M. Wing, titulado "Demystifying Computational Thinking for Non-Computer Scientists," que no llegó a publicarse.

la robótica educativa propicia el desarrollo de habilidades productivas, creativas, digitales y comunicativas, convirtiéndose de esta forma en un motor de la innovación.

En la actualidad, la enseñanza de estas disciplinas se hace a través de algunas asignaturas concretas, troncales y optativas, en las que los alumnos aprenden a programar placas de prototipado para la realización de proyectos de robótica. Aunque en las aulas las placas/sistemas más utilizados sean Arduino, Raspberry Pi, Lego Mindstorm, etc., en Inglaterra se apostó por el uso de una placa diseñada específicamente para fomentar la enseñanza de la computación. Esta placa es micro:bit, y aunque su uso en España podría decirse que es residual, posee un gran potencial para la docencia que ha sido demostrado en su trayectoria en el país anglosajón.

Micro:bit nació como una propuesta de la BBC (*British Broadcast Coorporation*) para fomentar las ciencias de la computación en los jóvenes a partir de 12 años en Inglaterra. Para ello, unió sus fuerzas con grandes empresas como Samsung, Farnell, Cisco o Microsoft y organismos como la Universidad de Lancaster o el Ministerio de Educación inglés, para crear un sistema intuitivo y funcional para el trabajo de la programación y la robótica en las escuelas. Una vez puesto en marcha el proyecto, el *King's College* de Londres realizó un estudio sobre sus resultados (King's College London, s.f.). Su conclusión principal fue que los alumnos tenían una actitud mucho más positiva en el estudio de estas disciplinas cuando usaban la placa. En contrapartida, los profesores reportaron una falta de material curricular con el que trabajar con el alumnado.

A raíz de mi trabajo de prácticas en el Instituto IES Santa Clara (Santander, Cantabria), en el IES La Albericia (Santander, Cantabria) y la experiencia en el concurso-exhibición de robótica educativa CantabRobots (CantabRobots, s.f.), he podido trabajar con diversas plataformas de robótica y programación, y he podido comprobar qué dispositivos se están utilizando en las aulas. El uso de las placas micro:bit es muy puntual, aún con el gran potencial que tienen estas placas. Es por ello que la exploración de esta placa y de sus posibilidades en el aula resulta de especial interés para llevar a cabo una investigación más profunda.

Fundamentada la motivación en todo lo anterior, este trabajo tiene como objetivo el de proporcionar una descripción del estado actual del desarrollo de la programación y la robótica en las aulas, así como la propuesta de una serie de prácticas como herramienta de trabajo para el desarrollo de ambas disciplinas en diferentes niveles de la Educación Secundaria Obligatoria (ESO), utilizando como hilo conductor los dispositivos micro:bit anteriormente mencionados.

El trabajo está estructurado de la siguiente manera: la Sección 2 ofrece una visión sobre el estado de la enseñanza de la programación y la robótica en Europa, España y Cantabria, además de un resumen de las distintas placas y lenguajes de programación más utilizados para el desarrollo de estas competencias en el aula. Por su parte, la Sección 3 presenta una introducción a la placa micro:bit, incluyendo una descripción técnica de la misma y de sus entornos de programación. La Sección 4 presenta la propuesta de uso de micro:bit en los diferentes cursos de la ESO donde la programación y la robótica forman parte del currículum. La Sección 5 describe las conclusiones extraídas y las líneas futuras que quedan abiertas a la finalización de este trabajo. Finalmente se proporcionan las referencias necesarias y los Anexos donde se encuentran contenidos los guiones completos de las prácticas propuestas en la Sección 4.

2. Estado de la cuestión

Este capítulo tiene el objetivo de presentar la situación de la enseñanza de la programación y la robótica en Europa, España y la Comunidad Autónoma de Cantabria. En primer lugar, se proporcionará una visión general del espacio que ocupan estas disciplinas dentro de los currículums, incluyendo ejemplos concretos de cómo se lleva a cabo su enseñanza. Por otro lado se describe la corriente STEAM, en la que la programación y la robótica son un motor destacable. Finalmente se presenta una descripción de las plataformas de robótica y los lenguajes de programación más utilizados actualmente en las aulas.

2.1. Situación en Europa, España y Cantabria

En Julio del año 2014, la Vicepresidenta de la Comisión Europea y responsable de la Agenda Digital para Europa, Neelie Kroes, y la Comisionada de Educación, Cultura, Multilingüismo y Juventud, Androulla Vassiliou, enviaron una carta (Kroes & Vassiliou, 2014) a los ministros de educación de la Unión Europea. El fin de esta misiva era el de exponer el gran problema que supone el desempleo juvenil en la UE y advertir de que las previsiones para el año 2020 apuntaban a una necesidad de 900 mil profesionales en TIC (Tecnologías de la Información y la Comunicación) en el mercado laboral en Europa. Añadían también que la programación informática no sólo ayudaría en este aspecto, sino que contribuye a desarrollar habilidades trasversales como el pensamiento analítico, la resolución de problemas, el trabajo en equipo o la creatividad. Finalmente, animan a los países a implantar estas enseñanzas desde edades tempranas, y a enriquecer todo el ecosistema (docentes, familias y alumnos) para ofrecer las mejores oportunidades en las escuelas.

En el año 2016, se presentó el reporte (Bocconi, Chioccariello, Dettori, Ferrari, & Engelhardt, 2016), que trata de resumir el estado de la inclusión del pensamiento computacional en Europa. La Figura 1 presenta el resumen de este informe. En ella se muestran los países que están renovando su currículum para introducir estas disciplinas, los que lo están planteando, los que ya lo han hecho, y los que dejan en manos de las administraciones regionales esta tarea.

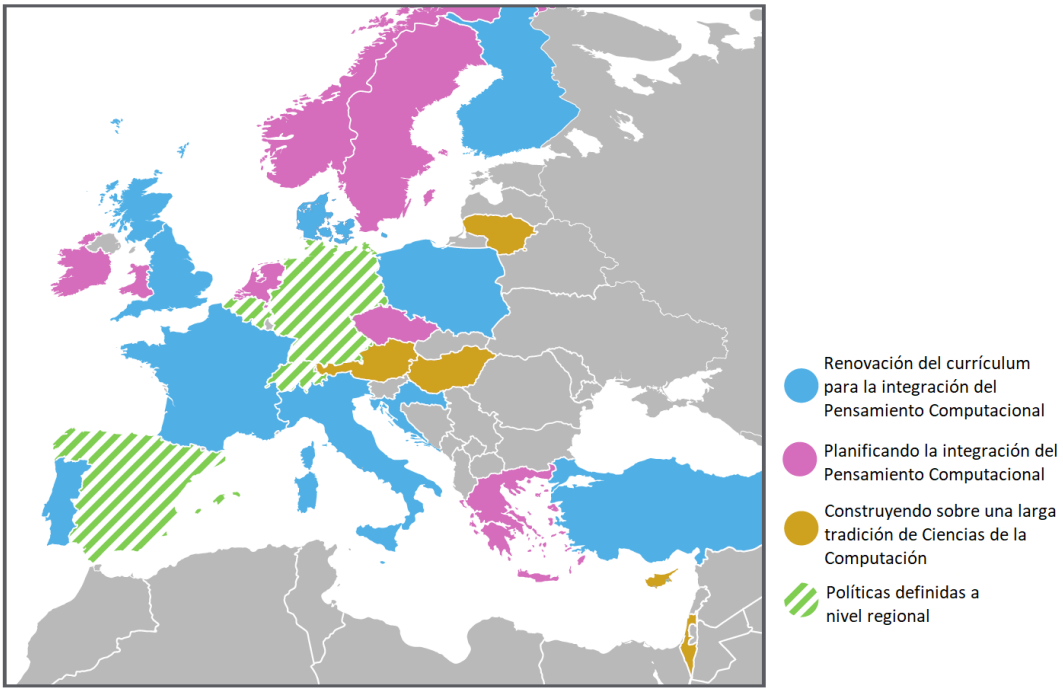


Figura 1. Resumen del estado de la inclusión del pensamiento computacional en Europa

De entre los países que llevan un largo recorrido en el desarrollo de la programación y la robótica en sus currículums, cabe destacar el trabajo de Reino Unido, Israel, Eslovaquia y Estonia.

Aunque Inglaterra esté situada en la Figura 1 como en fase de renovación de su currículum, en el año 2014 la materia TIC se sustituyó por *"Computing"* como materia obligatoria tanto en los niveles de primaria como de secundaria. La motivación de este proyecto fue el trabajo realizado por la *Royal Society* titulado "*Shut down or Restart"* (The Royal Society, 2012). En él se refleja que la enseñanza de la computación con el currículo existente era poco satisfactoria, ya que la mayoría del alumnado no conseguía habilidades más allá de la alfabetización digital básica, basada en la utilización de bases de datos o procesadores de texto. En esta nueva asignatura se introdujeron conceptos de algorítmica, programación informática, competencias digitales, diseño de habilidades de resolución de problemas, etc. De entre las iniciativas que acompañaron estos cambios, resulta destacable el proyecto micro:bit. Este proyecto, que nace de manos de la *BBC* (*British Broadcast Coorporation*) como parte del *BBC Computer Literacy Programme* (programa de alfabetización digital), tiene el objetivo de motivar a los estudiantes a ser creadores en lugar de meros consumidores.

El caso de Estonia también es remarcable por el hecho de que tiene sus orígenes a finales de los años 90, cuando se crea la *"Tiger Leap Foundation"* responsable de la estrategia nacional de enseñanza de las TIC en Estonia. En 2012, se presenta un proyecto educativo *"ProgeTiger",* financiado por el Gobierno de Estonia, que tuvo como objetivo enseñar programación en la educación primaria. Esta enseñanza se hacía de forma transversal en otras asignaturas, como música o matemáticas, utilizando "Scratch" (Scratch, s.f.) y a través de asignaturas específicas que los centros podían implantar (programación, robótica, 3D, etc.). En el caso de la enseñanza en secundaria, se trabajaba con lenguajes de programación textuales como Python o JavaScript.

En el caso de Israel, este país tiene una larga tradición en ciencias de la computación. Aunque el pensamiento computacional es una materia optativa en muchos institutos, la alfabetización digital comienza desde pequeños a través de distintas asignaturas. El currículum incluye tanto asignaturas obligatorias como optativas en esta materia, y su principal objetivo no es el de formar programadores, sino el de introducirles la lógica y la algorítmica desde pequeños. De esta forma todos construyen una base, que pueden ampliar a través de las asignaturas optativas.

Por último, en Eslovaquia la asignatura de Informática es obligatoria en todos los niveles educativos. Comenzó a impartirse en los niveles superiores de secundaria en el año 1985, en niveles inferiores de secundaria en 2005, y en primaria en 2008. La programación ha sido siempre uno de los componentes clave de esta asignatura.

Entre los países que están planeando la introducción del PC en sus currículums se encuentran República Checa, Irlanda, Noruega, Gales, Grecia, Países Bajos o Suecia. En el caso de Irlanda, por ejemplo, se ha definido una estrategia digital para centros educativos que propone un plan de acción para integrar las TIC en la enseñanza, considerando que la introducción de la programación en el currículum es importante para que cada alumno tenga la oportunidad de adquirir habilidades tales como la lógica, el pensamiento crítico, o la estrategia para resolución de problemas.

España se encontraría dentro de los países que dejan a sus regiones autonomía de decisión, en parte o en su totalidad, en la definición de los currículums. En España, la LOMCE (Ministerio de Educación y Formación Profesional de España, 2013) describe la obligatoriedad de la asignatura de Tecnología desde 2º a 4º de la ESO y la optatividad de TIC en 4º de la ESO. Desde este punto de partida, serán las comunidades las que dentro de la libre configuración puedan proponer asignaturas relacionadas con la programación o la robótica.

El Instituto Nacional de Tecnologías Educativas y de Formación del profesorado (INTEF) ha publicado a principios del presente año (2018) el informe "Programación, robótica y pensamiento computacional en el aula" (Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado, 2018) describiendo la situación en España en estos aspectos. En dicho documento, además de presentar la situación actual a nivel legislativo también presenta distintas actividades que tienen lugar en ámbitos académicos, civiles o empresariales. La Figura 2 resume la normativa e iniciativas de distintas comunidades autónomas.

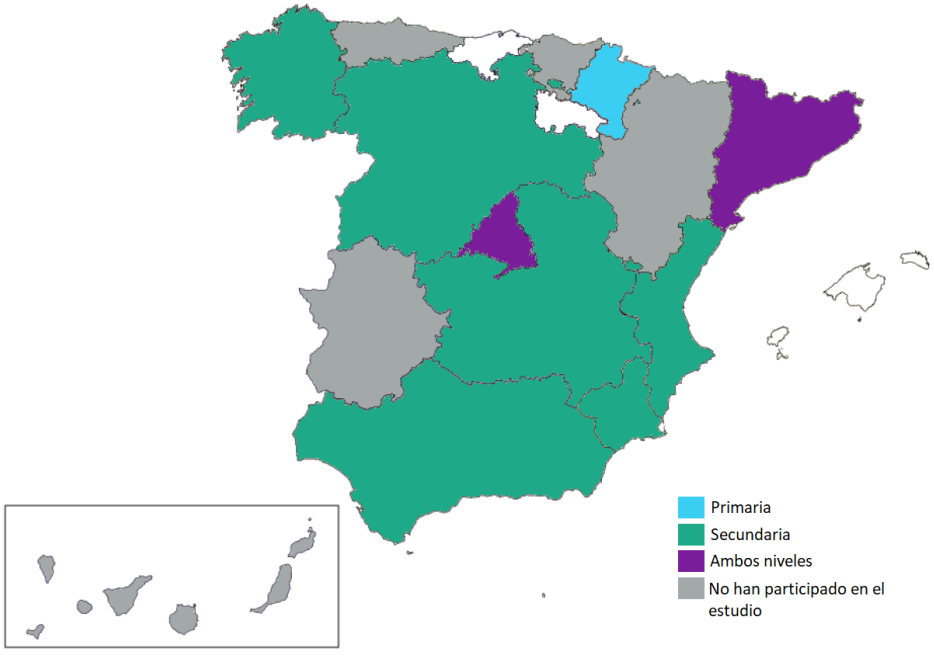


Figura 2. Comunidades Autónomas que han incluido nuevas asignaturas o contenidos sobre PC

De entre ellas, destacan los casos de Castilla León y Madrid por la completa oferta ofrecida.

Castilla y León ofrece en 3º de la ESO la asignatura de libre configuración "Control y Robótica" que abarca un conjunto de actividades pedagógicas donde se pretende trabajar el diseño y fabricación de un robot. En 4º de la ESO, también como asignatura de libre configuración, se ofrece "Programación Informática", recorriendo los tres pasos lógicos: introducción a la programación (bases del pensamiento computacional), programación en entorno gráfico y programación textual. Además de estas asignaturas, también son destacables las opciones ofrecidas para la formación del profesorado: Ingenia Primaria y Secundaria, el proyecto "Creando código", o la escuela TIC (Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado, s.f.)

En el caso de Madrid, se creó la asignatura "Tecnología, Programación y Robótica" de configuración autonómica y obligatoria para los cursos 1º, 2º y 3º de la ESO. De 5 bloques que componen la asignatura, uno de ellos se dedica a la programación y otro a robótica.

También se encuentran asignaturas de libre configuración enfocadas a la robótica en 2º de la ESO en Murcia y en 4º de la ESO en Castilla la Mancha. Por otro lado, se encuentran asignaturas enfocadas particularmente a la Programación en 1º y 2º de la ESO en Galicia, y en la Comunidad Valenciana es de obligada oferta desde 1º a 3º de la ESO.

Subrayar que en el reporte presentado por el INTEF, el caso de Cantabria contiene información incompleta, ya que si bien presenta la iniciativa CantabRobots (CantabRobots, s.f.), no se incluye la asignatura de libre configuración "Control y Robótica" ofrecida por la comunidad para el curso de 3º de la ESO. En la Tabla 1 se presenta un resumen de las asignaturas y los contenidos relacionados con la programación y la robótica que pueden encontrarse en el currículum de secundaria de Cantabria (Gobierno de Cantabria, 2015).

Tabla 1. Asignaturas y contenidos relacionados con la programación y la robótica en la ESO en Cantabria

Curso	Asignatura	Contenidos
2º ESO	Tecnología	Dentro del bloque "Tecnologías de la información y la comunicación" se presenta el contenido "lenguajes de programación con interfaz gráfica", donde se persigue que el alumno tome contacto con los conceptos y terminología básicos de la programación a través de un lenguaje de entorno gráfico.
3º ESO	Tecnología	En el bloque "Estructuras y mecanismos: máquinas y sistemas" se describen como objetivos que el alumno utilice un lenguaje de programación gráfico, el correspondiente entorno de programación y placas programables, de forma que pueda programar y automatizar sistemas con sensores y actuadores.
	Control y Robótica	La asignatura se divide en cuatro bloques: electrónica (analógica y digital), diseño de robots, conceptos básicos de programación, y el diseño e impresión 3D.
4º ESO	Tecnología	El bloque de "Tecnologías de la información y la comunicación" define el objetivo de la introducción a los lenguajes de programación. En el bloque sobre "Control y robótica" se encuentran contenidos sobre diseño y construcción de robots, el ordenador como elemento de programación y control, y lenguajes básicos de programación.
	Tecnologías de la Información y la Comunicación	No presenta bloques relacionados con la programación o la robótica.

2.2. Iniciativas STEAM

En los últimos años, la filosofía STEAM está comenzando a crear raíces en las aulas. Las iniciativas STEAM, acrónimo en inglés de cinco disciplinas académicas: ciencias, tecnología, ingeniería, arte y matemáticas (*Science, Technology, Engineering, Arts and Mathematics*), tienen el objetivo de aprovechar los puntos de intersección de estas materias para dotar de un enfoque interdisciplinar al proceso de enseñanza-aprendizaje. Además, a través de este nuevo enfoque, se pretende que aumente el interés del alumnado por el ámbito científico.

Estos proyectos suelen presentar situaciones o problemas de la vida diaria, lo que permite contextualizar los contenidos, acercándose a la metodología de enseñanza por proyectos, y apostando por la filosofía de que lo que se lee se recuerda, pero lo que se hace se aprende.

En la actualidad diversos países están impulsando iniciativas STEAM, como centros en Reino Unido (STEM Learning, s.f.) o Finlandia (University of Helsinki, s.f.). De hecho, la Comisión Europea ha lanzado un programa marco (Comisión Europea, 2014) 2014-2020 al que están destinados más de 13 millones de euros para subvencionar este tipo de iniciativas con el fin de aumentar el atractivo de las ciencias y de las carreras científicas en la juventud. En España también se ha contribuido al desarrollo de distintos proyectos que tienen como objetivo la promoción de las iniciativas STEAM como por ejemplo (Gobierno de Canarias, 2017), (Universidad de Cantabria, 2015), (Universidad de Cantabria, 2017), (ASTI Talent&Tech Foundation, s.f.).

Las enseñanzas de programación y robótica pueden ser un gran aliado para la consecución del objetivo de transversalidad entre materias. Además, a través de la programación y la robótica se puede motivar a los alumnos a que trabajen diferentes aspectos como el trabajo en equipo, ser lógicos, metódicos, ordenados, etc. de una forma atractiva.

Por todo lo anterior, la propuesta realizada en este trabajo aboga por la mentalidad STEAM, tratando de realizar actividades en las que diferentes disciplinas se combinan en una misma actividad.

2.3. Realidad de la programación y la robótica en las aulas

Esta sección ofrece una muestra de las distintas placas programables, lenguajes, y entornos de programación más utilizados en las aulas para la enseñanza de la programación y la robótica.

2.3.1. Robótica en las aulas

A continuación, se realizará una descripción de las placas programables más utilizadas en educación en la impartición de las materias de robótica y programación. Remarcar que éste es un resumen de las placas más comunes, dado que la cantidad de éstas va en aumento con los años. Por tanto, se prioriza el uso de aquellas placas que promuevan el aprendizaje de las bases de la programación, que permitan el desarrollo de la creatividad del alumno, y que ofrezcan facilidad y comodidad de uso tanto al profesorado como al alumnado.

- **Arduino**

Para comenzar, podría decirse que Arduino (Arduino Corporation, s.f.) es la placa más utilizada en la mayoría de centros en la actualidad. Arduino es una compañía que nació en 2005 como un proyecto para estudiantes en el Instituto IVREA en Italia, persiguiendo el objetivo de proporcionar una forma sencilla y económica para que principiantes y profesionales pudieran crear proyectos en los que pudieran controlar diferentes sensores o actuadores. Además de las placas básicas, también disponen de otras placas de expansión (Arduino Products, s.f.), conocidas como *shields*, que suman nuevas funcionalidades tales como conexión Ethernet, Wifi o gestión de motores.

La gran ventaja que presenta esta placa es la gran comunidad de usuarios que tiene detrás, lo que permite encontrar mucha información en la red sobre la placa, tutoriales, experiencias, etc. de forma sencilla. Sin embargo, para su programación se utiliza un entorno de desarrollo basado en lenguaje C++, que no es sencillo para programadores noveles. Aunque también existen programas que permiten su programación por bloques, esta funcionalidad no es ofrecida de forma nativa.

- **Raspberry Pi**

En el caso de Raspberry Pi (Raspberry Pi Foundation, s.f.), más que una placa de robótica educativa puede decirse que es un ordenador en miniatura. Estas placas fueron desarrolladas en Inglaterra por la Fundación Raspberry Pi para promover la enseñanza de ciencias de la computación en colegios y países en vías de desarrollo. Su acogida en el mercado fue mucho mayor de lo esperado. Al igual que Arduino, Raspberry también goza de una comunidad muy grande que la apoya, por lo que encontrar información sobre esta placa, proyectos, ayuda, etc. es muy sencillo.

En cuanto al modo de programación de estas placas, al ser prácticamente un ordenador, se pueden utilizar diferentes lenguajes de programación. Sin embargo, no está provisto de un entorno sencillo de programación en el que un estudiante novel pueda sentirse cómodo trabajando. El hecho de que sea una placa muy potente, más equiparable a un ordenador que a una placa programable de enseñanza, aumenta su precio, y sobre todo excede los objetivos generales que se quieren conseguir en las clases de secundaria. Además, su configuración es más compleja, y la curva de aprendizaje es mayor que la de otras placas.

- **Imagina-Scratch**

La placa Imagina-Scratch (Innova Didactic. Tecnologías creativas para la educación, s.f.) está enfocada a la realización de prácticas de robótica y programación para cursos de ESO y Bachillerato. La placa fue diseñada por el equipo Robolot y es compatible con entornos de programación que permiten al alumno programar por bloques y después cargar el resultado en la placa para ejecutarlo (Scratch2.0, S4A y el sistema PICAXE).

Entre sus características más destacables se encuentra el hecho de poseer una gran cantidad de sensores y módulos integrados. Dispone de LEDs, fotorresistencia, sensor de temperatura, receptor y emisor de infrarrojos, sensor de ultrasonido o *bluetooth*. De esta forma, los alumnos pueden trabajar con distintos dispositivos sin que el desembolso sea mayor por tener que comprar sensores o actuadores independientes a la placa. Es destacable también la posibilidad de controlar la placa a través de aplicaciones móviles gracias al

módulo *bluetooth* del que dispone, lo que ofrece mayores opciones para el trabajo con robots.

En contrapartida, resaltar que estas placas no permiten la enseñanza de los lenguajes textuales más utilizados y que la comunidad detrás de la misma es reducida.

- **Mindstorm Lego**

Lego Mindstorm (Lego Group, s.f.) es un kit de construcción de Lego que permite la construcción, programación y testeo de proyectos de robótica a partir de un conjunto de sensores (de luz, de temperatura, de contacto, de ultrasonidos) y motores, y un bloque programable que será el cerebro del robot. Para programar este cerebro se utiliza un entorno de programación por bloques. Sin embargo, estos bloques no presentan similitudes con las estructuras e instrucciones habituales de un lenguaje de programación, sino que son bloques que tienen un significado mucho más mecánico. Otra de las grandes desventajas de este kit es su precio, que es mucho más elevado que el resto de propuestas.

- **Microbit**

Las placas micro:bit (Micro:bit, s.f.) nacieron en el año 2015 como resultado del programa *BBC Computer Literacy* (alfabetización digital) que tenía el objetivo de fomentar la enseñanza de la computación en Inglaterra.

Esta placa tiene un tamaño pequeño pero un gran potencial en el entorno educativo. Permite al alumno fomentar su creatividad y el aprendizaje de la programación y la robótica a través de un sistema que tiene integrados una gran cantidad de sensores, y que puede ser programado en diferentes lenguajes de programación (bloques, Python, JavaScript). De esta forma, la plataforma micro:bit permite que el alumno pueda iniciarse en la programación en edades tempranas a través de la programación por bloques, de forma que en niveles superiores pueda trabajar lenguajes textuales, pudiendo así profundizar paulatinamente, aprendiendo de forma constructiva.

Tabla 2 muestra un cuadro comparativo de las características de las placas analizadas en esta sección. Como puede verse, la placa micro:bit es la que

mejores resultados ofrece. Son estas características las que inspiran este trabajo, que propone el uso de esta placa como hilo conductor a lo largo de diferentes niveles educativos, para trabajar tanto la programación como la robótica en la educación secundaria obligatoria.

Tabla 2. Resumen de las características más reseñables de las placas más utilizadas en las aulas en la actualidad

	Sensores o actuadores integrados	Conexión sensores externos	Entornos y lenguajes[2]		Complejidad	Comunidad	Precio
			Bloques	Textual			
Arduino	No	Sí	No	Basado en C++	3/5	Alta	20€ + sensores
Raspberry Pi	No	Sí	No	Admite varios	4/5	Alta	40€ + sensores
Imagina-Scratch	LEDs, fotoresistencia LDR, temperatura NTC, receptor y emisor de infrarrojos, sensor de ultrasonido o bluetooth	Sí	Sí	No	2/5	Media	40€
Lego Mindstorm	Temperatura, de contacto, luminosidad, ultrasonidos, giroscopio	No	Sí[3]	No	3/5	Media	400 €
micro:bit	Matriz de LEDs, 2 botones, luminosidad, temperatura, acelerómetro, brújula	Sí	Sí	Python JavaScript	1/5	Baja	20€

2.3.2. Programación en las aulas

Existen más de 2500 lenguajes de programación documentados, cada uno con unas características que los hace idóneos para entornos o necesidades

[2] Entornos y lenguajes de programación ofrecidos de forma nativa por la placa/sistema

[3] La programación por bloques en Lego Mindstorm no presentan similitudes con las estructuras/instrucciones habituales de un lenguaje de programación, sino que son bloques que tienen un significado mucho más mecánico

concretas. Entre tanta variedad, no puede afirmarse que exista un lenguaje por excelencia. Sin embargo, desde un punto de vista pedagógico, cuando se estudia qué lenguajes enseñar en las escuelas, cabe entender que habrá que tener ciertos aspectos en consideración: que sea sencillo, que sirva para aprender las bases de la programación, que sea atractivo para el alumnado, que lo encuentre útil, etc. En este sentido, en algunos currículums no se propone ningún lenguaje en particular, quedando en manos del docente (o del departamento) la decisión acerca de qué lenguaje enseñar. En lo que sí se hace distinción es entre lenguajes gráficos y lenguajes textuales. Por ejemplo, el currículum de la Comunidad de Cantabria propone la utilización de lenguajes gráficos en los primeros cursos, para continuar con lenguajes textuales en cursos superiores. Será esta la dinámica a seguir en esta propuesta.

La programación mediante entornos gráficos, o por bloques, consiste en la realización de programas utilizando bloques que representan estructuras e instrucciones de programación que se pueden arrastrar y enlazar como si de un puzle se tratara. Este tipo de programación permite al estudiante aprender a crear algoritmos y a estructurar programas sin tener que preocuparse de la sintaxis específica de los lenguajes de programación.

El lenguaje por bloques por excelencia en educación es Scratch (Scratch, s.f.), aunque existen otros entornos como Blockly (Google Developers, s.f.) o micro:bit (Power your imagination with code, s.f.) con igual o superior potencial. Sin embargo, se le otorga esa posición de privilegio debido a su gran expansión en el ámbito educativo, ya que está presente en más de 150 países, está traducido a más de 70 idiomas, y se han realizado multitud de estudios sobre esta plataforma (Investigación acerca de Scratch, s.f.)

Aunque en ocasiones pueda verse esta herramienta como un juego, más que como programación real, este tipo de herramientas ayudan a introducir los conceptos de la programación en edades tempranas de forma no invasiva. En este sentido (Armoni, Meerbaum-Salant, & Ben-Ari, 2015) relata la experiencia de transición de un grupo de alumnos que manifiestan que el haber utilizado Scratch en cursos anteriores les ha facilitado el aprendizaje de lenguajes textuales, necesitando menos tiempo para aprender los nuevos conceptos.

Sin embargo, Scratch por sí mismo no permite ver el resultado del programa en una placa real. Si los alumnos quisieran, por ejemplo, implementar el código en placas Arduino o Picaxe necesitarían programas adicionales para traducir de bloques a lenguaje entendible por las placas. En este sentido, el entorno que tiene micro:bit para programar por bloques permite cargar y ejecutar el programa en la placa, pudiendo ver el alumno el resultado real de su creación.

Una vez que los alumnos ya han tenido su primer contacto con la programación por bloques, se encontrarán más preparados para aprender programación textual en cursos superiores. Desde un punto de vista pedagógico, hay lenguajes que por sus particularidades son más sencillos tanto para ser enseñados como para ser aprendidos. Atendiendo a las indicaciones de Milbrandt (1993) en (Grandell, Peltomäki, Back, & Salakoski, 2006) y de Rosalía Peña (Peña, 2015), un lenguaje de programación que quiera ser utilizado en educación debe ser fácil de aprender, tener una sintaxis sencilla, utilizar palabras cercanas al lenguaje común, ser estructurado, de uso generalizado, potente en cuanto a capacidades, con mensajes de error sencillos, que promueva buenos hábitos de programación, y que sea relevante en el mercado laboral.

En este sentido, en la literatura pueden encontrarse múltiples artículos que enfocan sus miradas a Python (Koulouri, Lauria, & Madredie, 2015), (Kruglyk & Lvov, 2012), (Pekka Kasurinen, 2016), (Stevens & Serate, 2016). Python es un lenguaje de alto nivel que fue diseñado por Guido van Rossum para promover el aprendizaje de la programación (Peña, 2015) y entre sus ventajas se muestran varias de las anteriormente comentadas: sintaxis limpia y muy intuitiva, dinámicamente tipado (una misma variable puede tomar valores de distintos tipos en distintos momentos), permite ejecutar de forma rápida un programa y ver sus resultados, diseño estructurado, y además es gratuito.

Por todo lo anterior, este trabajo propone comenzar a programar de forma textual en el nivel de 4º de la ESO con Python, trabajando los conceptos básicos de la programación a través de prácticas con la placa micro:bit que dispone de un entorno de programación Python.

3. Introducción a Micro:bit

Esta sección presenta información general relativa a la placa micro:bit. Se comienza con una descripción de la misma y de sus orígenes. A continuación, se describen sus características técnicas y los pasos básicos para comenzar a utilizarla, como muestra del potencial de la misma y de su facilidad de uso. Por último, se presentan los diferentes entornos de programación que ofrece.

3.1.1. ¿Qué es Micro:bit?

El proyecto micro:bit comenzó en 2012 como parte del programa *BBC Computer Literacy* (alfabetización digital) y en Julio de 2015 la BBC ya contaba con 29 socios para participar en el diseño, fabricación y distribución de los dispositivos, entre los que se encontraban grandes empresas como Microsoft, Farnell o Cisco. La mayoría de los gastos fueron asumidos por los socios y su intención fue la de utilizar una licencia de código abierto para que estas placas pudieran utilizarse con fines educativos, formalizando para ello la creación de una compañía sin ánimo de lucro: *Microbit Education Foundation* (About the micro:bit foundation, s.f.). Esta iniciativa fue la más ambiciosa en la historia de la educación en Inglaterra en los últimos 30 años, y tenía el objetivo de inspirar la creatividad digital y desarrollar habilidades en ciencia, tecnología e ingeniería entre el alumnado (Brithish Broadcast Coorporation, s.f.).

Se diseñó para ser distribuida entre todos los alumnos de 11 y 12 años en Inglaterra, comenzando en 2015, repartiéndose de forma gratuita alrededor de 1 millón de placas en el mes de octubre de dicho año.

Después de su implantación, el *Kings College* de Londres realizó un estudio (King's College London, s.f.) para analizar su repercusión en las aulas, desde el punto de vista de profesores y alumnos. La evaluación pretendía recoger información de cómo profesores y alumnos percibían la placa y qué puede aprenderse a través de su uso. Las conclusiones fueron muy positivas, la placa fue bien recibida por los colegios. Todos los niños coincidieron en que micro:bit es una forma de motivación en el aula, y todos los profesores entrevistados sintieron que los alumnos estaban motivados y que era una forma amigable de introducir al alumnado a la programación de dispositivos. Como punto negativo,

algunos profesores destacaron que no se sentían preparados a la hora de incluir proyectos con la placa dentro del currículum, lo que denota que sería necesario una formación del profesorado y materiales curriculares específicos con los que trabajar.

Con el fin de evitar este inconveniente, la propuesta realizada en este trabajo enmarca el uso de las placas micro:bit dentro del currículum de secundaria, evitando ser un conjunto de prácticas aisladas sin conexión con los contenidos del mismo.

3.1.2. Características Técnicas

La placa micro:bit es un sistema embebido basado en arquitectura ARM que dispone de diferentes sensores, botones y una matriz de diodos LED integrados en la misma (ver Figura 3).

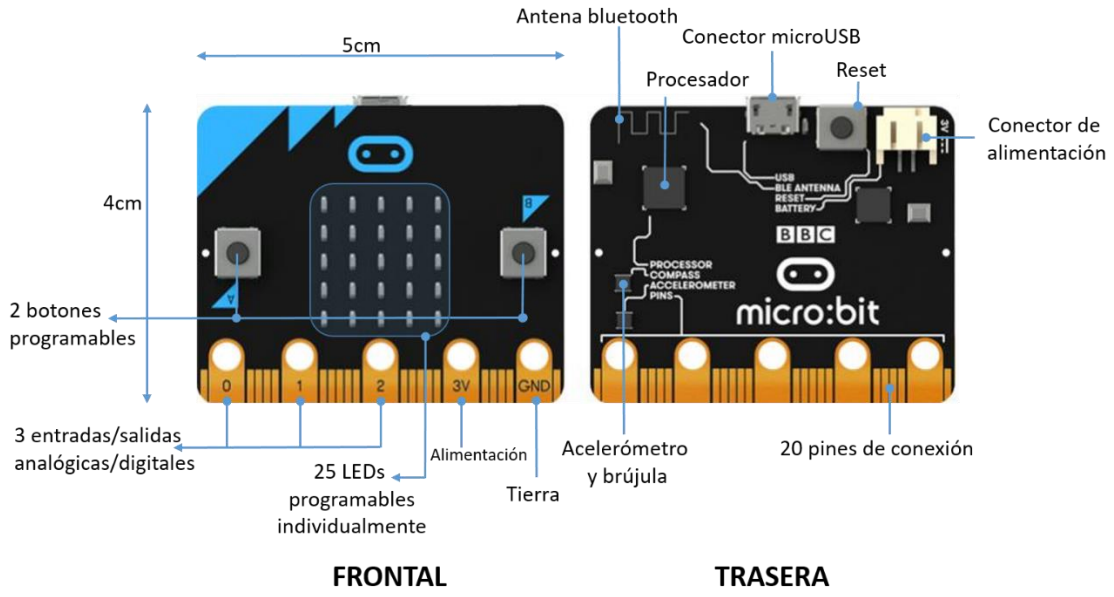


Figura 3. Descripción del lado frontal y trasero de la placa micro:bit

La Tabla 3 presenta un resumen de las características de la placa.

Tabla 3. Características de la placa micro:bit

Matriz de 25 LEDs	Los LEDs (Diodo Emisor de Luz) son programables individualmente, permitiendo al alumno jugar con la visualización de texto, imágenes, números, etc.

2 botones	En el frontal de la placa se dispone de dos botones (A y B) a los que se puede programar su comportamiento.
Pines de conexión	La placa dispone de 20 pines que son conectores externos, descritos en (BBC micro:bit pins, s.f.). A través de esos pines se pueden conectar distintos sensores o actuadores para poder ser controlados a través de la programación de la placa.
Sensor de luz	La matriz LED también funciona como un sensor de luz básico que permite detectar la luz ambiental.
Sensor de temperatura	Dispone de un sensor de temperatura que detecta la temperatura ambiente donde esté la placa en grados Celsius.
Acelerómetro	El acelerómetro mide la aceleración a la que está sometida la placa, pudiendo detectar y analizar el movimiento de la misma.
Brújula	La placa dispone de una brújula que permite detectar el campo magnético de la tierra, pudiendo entonces conocer hacia qué orientación está situada la placa.
Radio	Permite la comunicación sin cable entre placas micro:bit.
Bluetooth	Micro:bit dispone de una antena *bluetooth* que la permite tanto emitir como recibir señales *bluetooth*. De esta forma, la placa puede comunicarse con otros dispositivos que también dispongan de esta tecnología.
Conexión micro-USB	Permite conectar la placa al ordenador a través de un cable micro-USB. A través de esta interfaz se podrá alimentar y programar la placa.

3.1.3. Primeros pasos con micro:bit

La placa micro:bit tiene una curva de aprendizaje muy sencilla. La configuración se puede realizar en 4 pasos:

- Paso 1: Conectar la placa al ordenador

La placa se conecta al ordenador a través de un cable micro-USB. Una vez conectada, aparecerá en nuestro ordenador como una nueva unidad de disco conectado de nombre MICROBIT.

- Paso 2: Programar

Utilizando alguno de los editores que se describen en (Power your imagination with code, s.f.) el alumno puede diseñar y escribir el programa que quiere que sea ejecutado por la placa.

- Paso 3: Transferir el programa a la placa

El alumno deberá hacer clic en el botón "*Download*" o "Descargar" de la herramienta de programación. Lo que ocurre al hacer clic en el botón es que el programa se compila y se crea un archivo que la placa es capaz de entender y ejecutar (fichero de extensión .hex). Una vez descargado (probablemente en la carpeta de descargas), el alumno deberá copiar ese fichero en la unidad de disco MICROBIT.

- Paso 4: Ejecutar

La placa se mantendrá en reposo y el LED amarillo en la parte trasera parpadeará mientras el programa se está cargando en la placa. Una vez cargado, el programa se ejecutará automáticamente.

3.1.4. Edición de código en micro:bit

Para editar código que pueda ser ejecutado por la placa micro:bit, la plataforma proporciona dos herramientas: una que permite la programación a través de bloques o utilizando el lenguaje JavaScript, y otra que permite la programación en Python.

Para trabajar utilizando bloques o lenguaje JavaScript se proporciona la herramienta web (Microsoft, s.f.).

Como se muestra en la Figura 4 la herramienta presenta un panel donde el alumno puede crear su código utilizando de los bloques que están diferenciados por funcionalidades en la zona central de la pantalla (la caja de herramientas). A la izquierda puede verse una placa en la que es posible simular el código generado en el panel derecho.

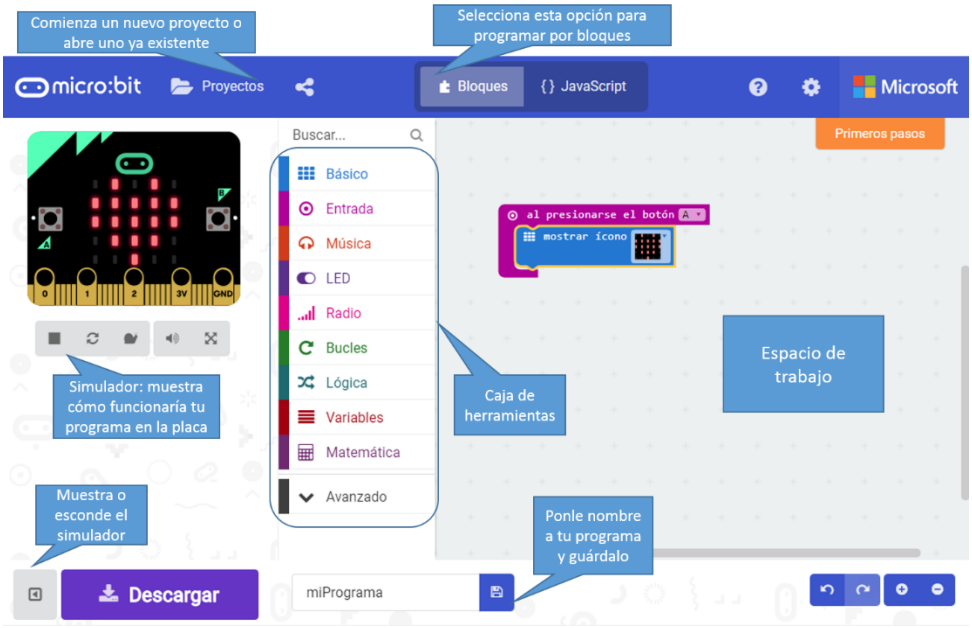


Figura 4. Captura del entorno de programación de micro:bit para trabajo con bloques o JavaScript

En la parte superior puede encontrarse un botón a través del cual es posible cambiar de modo para programar bien en bloques o bien JavaScript. Además, si el código ha sido generado con bloques, al hacer clic en ese botón los bloques se traducirán a código JavaScript y viceversa. Esto puede complementar la enseñanza en niveles iniciales, en los que el alumnado puede ver cómo el programa que están haciendo en bloques tiene una traducción más o menos directa en lenguajes de programación que aprenderán a utilizar en el futuro.

En cuanto a las posibilidades que ofrece la caja de herramientas nos encontramos la siguiente división:

- Básicos: proporciona funcionalidades básicas como mostrar números, mostrar imágenes, mostrar cadenas de texto, ciclos infinitos, pausa, etc.
- Entrada: permite gestionar eventos y datos de sensores como describir qué hacer cuando se pulsa un botón, cuando se agita la placa, leer la temperatura, aceleración, etc.
- Música: proporciona funcionalidades para crear tonos y melodías.
- Led: con bloques para controlar la matriz de leds, iluminando o apagando los diferentes leds que la componen.

- Radio: gestión de las funcionalidades *bluetooth* para enviar y recibir información.
- Loops: estructuras de programación de repetición.
- Lógica: estructuras de programación de condicionales y operadores lógicos.
- Variables: permite crear variables, cambiar su valor, etc.
- Matemáticas: con funcionalidades para realizar operaciones matemáticas.
- Avanzadas: este apartado contiene bloques para gestionar funciones, arrays, texto, imágenes, pines eléctricos, etc.

Resulta destacable que el entorno de programación es multilenguaje, por lo que puede utilizarse tanto en español como en inglés, lo cual resulta interesante para su empleo en programas bilingües.

Para la programación en Python de la placa micro:bit, puede utilizarse el entorno básico disponible en (Python Software Foundation, s.f.) o el programa mu Editor disponible en (Tollervey, s.f.). El segundo es muy similar, pero tiene la ventaja de que permite cargar en la placa módulos externos a la misma. Se recomienda el uso de ésta última herramienta, mostrada en la Figura 5.

Figura 5. Captura del entorno mu Editor para la programación en Python de la placa micro:bit

Además de estas dos herramientas online, también es posible crear código y subirlo a la placa mediante *bluetooth* a través del móvil utilizando aplicaciones ya diseñadas para Android o iOS (Mobile apps for micro:bit, s.f.).

4. Propuesta de integración de micro:bit en las aulas de la ESO

Esta sección describe la propuesta de utilización de la placa micro:bit en los niveles educativos desde 2º a 4º de la ESO para el trabajo de la programación y la robótica en las asignaturas que integren estos contenidos en el currículum. Esta propuesta se presenta en forma de prácticas cuyos guiones completos pueden encontrarse en el Anexos de este documento, y que se fundamentan en los contenidos curriculares de las diferentes asignaturas para las que se proponen.

A continuación, se describen los objetivos que persiguen las distintas prácticas, los contenidos que se trabajan, los conceptos previos necesarios, la metodología propuesta, los materiales necesarios para su ejecución, y un análisis del enfoque STEAM en aquellas prácticas en las que proceda. No se proporciona una duración estimada de las mismas debido a que su implantación en el aula no ha podido llevarse a cabo por limitaciones temporales, estando esta tarea incluida como una de las líneas futuras de este trabajo.

4.1.1. Propuesta para 2º de la ESO

- **Tecnología**

Las prácticas en esta asignatura de 2º de la ESO tienen como objetivo que el alumno se familiarice con un primer lenguaje de programación de bloques y de esta forma descubra las bases de la programación. Así, se proporciona una respuesta al currículum de esta asignatura en Cantabria, en el que se describe que el alumno deberá conocer los conceptos y terminología básicos en lenguajes de programación utilizando un entorno gráfico, siendo capaz de elaborar programas sencillos.

2ESO.P0 - Descubriendo micro:bit y la programación	
Objetivos	El principal objetivo de esta práctica es el de familiarizarse con la placa micro:bit y con el entorno de trabajo que permite programar la misma a través de bloques.

Contenidos	El alumno conocerá la placa micro:bit, comenzará a trabajar con un entorno sencillo de programación y aprenderá las distintas instrucciones que puede utilizar para programar, creando además su primer programa.
Conceptos previos	No es necesario que el alumno tenga conocimientos previos. Sin embargo, sería interesante que, antes de realizar la práctica, el profesor introdujera el tema al alumnado explicando cuáles son los objetivos de estudiar programación y robótica, y presentar qué papel tienen estas materias en la sociedad actual. Esto propiciará al alumno un contexto que motive su aprendizaje.
Metodología	Cada alumno dispondrá de un guion de prácticas, una placa y un ordenador. Las prácticas también pueden llevarse a cabo por parejas, aunque no es recomendable que el número de alumnos en el grupo sea mayor que dos. Para la ejecución de la práctica, el profesor introducirá a los alumnos los objetivos de la práctica y desarrollará la misma alternando los conceptos teóricos y prácticos mientras proyecta los pasos que va realizando en el entorno de programación. El último ejercicio lo realizarán los alumnos de forma autónoma, teniendo que aplicar los conocimientos adquiridos durante la clase.
Material	Un ordenador con acceso a Internet Una placa micro:bit + cable microUSB-USB Un guion (impreso o digital)

2ESO.P1 - Cambiar de humor apretando un botón	
Objetivos	Mostrar en la matriz de LEDs de la placa diferentes imágenes dependiendo de qué botones, de los integrados en la misma, sean apretados.
Contenidos	El alumno trabajará las estructuras condicionales y de repetición a través de la programación de un juego sencillo. Además, el alumno aprenderá a controlar los eventos relacionados con los botones de la placa.

Conceptos previos	Se deberá haber realizado anteriormente la P0 de forma que conozca cómo programar y ejecutar programas en la placa.
Metodología	El profesor realizará una introducción oral con los objetivos que se persiguen en la práctica. Sería interesante que el profesor complementara la información con ejemplos de la vida diaria en la que las estructuras de repetición y condicionales se dan habitualmente, como puede ser la decisión de encender o apagar una luz dependiendo de si es de día o de noche. El primer ejercicio puede ser realizado de forma simultánea por el profesor y por el alumno, de forma que el alumno pueda seguir la explicación del ejercicio. Se aconseja que el resto de ejercicios propuestos en la práctica sean realizados de forma individual por el alumno para trabajar de forma autónoma los conceptos aprendidos anteriormente.
Material	Un ordenador con acceso a Internet Una placa micro:bit + cable microUSB-USB Un guion (impreso o digital)

2ESO.P2 - Muévete con alegría	
Objetivos	Diseñar un sistema en el que la imagen mostrada en la matriz de LEDs de la placa cambie cada vez que la placa sea agitada.
Contenidos	Se trabajará el concepto de sensores y se ejemplificará su manejo a través del sensor acelerómetro integrado en la placa micro:bit. También se repasará la gestión de eventos.
Conceptos previos	Se deberá haber realizado anteriormente la P0 de forma que el alumno conozca la placa y cómo programar y ejecutar programas en ella.
Metodología	Se propone que el profesor comience explicando a los alumnos qué van a aprender con la práctica. Antes de comenzar, deberá explicar también qué es un acelerómetro. Esta explicación podría ser realizada por el profesor de Física si se quiere seguir una aproximación STEAM, de forma que el alumno vea la relación de conceptos y materias. A continuación, el alumno

	realizará el ejercicio como se indica en el guion de prácticas. Finalmente, se propone la realización de ejercicios en los que el alumno tenga que diseñar la solución e implementarla de forma autónoma, aplicando los conocimientos adquiridos.
Material	Un ordenador con acceso a Internet Una placa micro:bit + cable microUSB-USB Un guion (impreso o digital)
Propuesta STEAM	Se propone la participación del profesorado de Física, que se acercaría al aula para explicar lo que es un acelerómetro, qué es lo que mide y ejemplos de uso.

2ESO.P3 - Música y tecnología	
Objetivos	Crear varias piezas musicales que serán interpretadas por la placa micro:bit con ayuda de un zumbador.
Contenidos	Se profundizará en el concepto de bucles introduciendo la estructura de repetición FOR, con el fin de crear código más eficiente. También se aprenderá a trabajar con actuadores externos a la placa, como es el zumbador, aprendiendo a conectarlo correctamente y a programarlo.
Conceptos previos	Se deberá haber realizado anteriormente la P0 de forma que el alumno conozca la placa, cómo programarla y cómo ejecutar programas en ella. También es interesante que se haya hablado al alumno con anterioridad de las estructuras de repetición.
Metodología	La clase deberá comenzar con una explicación de los objetivos a conseguir. Si se sigue la propuesta STEAM de esta práctica, el profesor del área de música puede participar y describir ejemplos de instrumentos creados con tecnología, proponiéndoles crear música con su placa y el zumbador. Antes de que los alumnos realicen las prácticas de forma autónoma, el profesor deberá explicarles qué es un zumbador, cómo se utiliza y cómo se conecta a la placa.
Material	Un ordenador con acceso a Internet Una placa micro:bit + cable microUSB-USB

	Zumbador + pinzas cocodrilo Un guion (impreso o digital)
Propuesta STEAM	Esta práctica puede realizarse en colaboración con el profesorado de música. La idea a trabajar podría ser la presentación de instrumentos musicales que tengan una base tecnológica como puede ser un arpa láser, el theremin, guitarras fabricadas con impresoras 3D, etc. A partir de esto se puede motivar al alumno a crear su propio instrumento con la placa micro:bit y un zumbador.

2ESO.P4 - Jugando a piedra, papel o tijera	
Objetivos	El objetivo será programar el juego piedra, papel o tijera para que el alumno pueda jugar contra la placa micro:bit.
Contenidos	En esta práctica se reforzarán los conocimientos sobre estructuras condicionales y se aprenderá el concepto de variable en programación. También aprenderán a utilizar el bloque de obtención de números aleatorios y su importancia cuando quieres crear eventos azarosos.
Conceptos previos	El alumno deberá haber realizado la práctica 0 para conocer la placa y su programación. También deberá conocer lo que son las estructuras condicionales y haberlas programado anteriormente.
Metodología	Se propone que el profesor comience la práctica con una explicación de los objetivos de la práctica, describiendo al alumno qué conceptos se van a repasar y explicando el nuevo concepto de variable. Es importante que se dedique un tiempo a que los alumnos comprendan este concepto, ya que es una de las bases más importantes de la programación. Después de la explicación teórica, descrita brevemente en la introducción de la práctica, es importante que el profesor explique el código de bloques que se le presenta al alumno en el ejercicio, haciendo hincapié en el trabajo con la variable y en la estructura condicional utilizada. El último ejercicio será realizado

	individualmente por el alumno, y será donde pueda aplicar los conocimientos adquiridos anteriormente.
Material	Un ordenador con acceso a Internet Una placa micro:bit + cable micro USB-USB Un guion (impreso o digital)

4.1.2. Propuesta para 3º de la ESO

Las prácticas propuestas en esta sección tienen el objetivo de introducir a todo el alumnado en la robótica y la programación a través del uso de micro:bit en el curso de 3º de ESO. En la asignatura de Tecnología todos tendrán la oportunidad de conocer las bases, mientras que en la asignatura optativa de "Control y Robótica" podrán trabajar más profundamente estos conceptos.

- **Tecnología**

Las prácticas diseñadas para esta asignatura tienen el fin de cubrir los objetivos propuestos en el currículum en Cantabria. En éste se describe que el alumno deberá saber utilizar un lenguaje de programación gráfico, el correspondiente entorno de programación y placas programables, de forma que pueda programar y automatizar sistemas con sensores y actuadores.

Se recomienda la realización de la práctica P0 de 2º ESO para recordar el uso de la placa micro:bit y del entorno de programación.

3ESO.T.P1 - Cuentapasos	
Objetivos	Diseñar y crear un dispositivo contador de pasos con la placa micro:bit.
Contenidos	En esta práctica se profundizará en el concepto de variable y su manejo dentro de en un programa.
Conceptos previos	El alumno deberá haber realizado la práctica 0 para conocer la placa, cómo se programa y como ejecutar los programas en la placa.

Metodología	La clase comenzará con una explicación teórica de los objetivos que se persiguen con esta práctica, haciendo especial referencia a las variables y a su manejo. Este es un instrumento importante que el alumno debe entender claramente, ya que es uno de los conceptos más importantes en programación. Si se sigue la actividad STEAM propuesta, el profesor de educación física puede realizar la presentación de la propuesta de trabajo (explicada en el análisis STEAM). A continuación, el alumno podrá realizar el ejercicio siguiendo el guion de la práctica. Finalmente, el alumno trabajará de forma autónoma en el diseño y programación del último ejercicio propuesto, en el que tendrá que hacer uso de los conocimientos adquiridos en la primera parte.
Material	Un ordenador con acceso a Internet Una placa micro:bit + cable micro USB-USB Soporte para pilas + pilas Un guion (impreso o digital)
Propuesta STEAM	Para esta actividad se propone el trabajo conjunto con la asignatura de Educación Física. Aprovechando que se construye un cuentapasos, se propone trabajar conjuntamente la importancia de la actividad física diaria. Una vez construido el sistema se puede proponer al alumno utilizar el cuentapasos durante un día (creando un sistema de sujeción al pie) que contabilice los pasos realizados. Al final del día el alumno deberá apuntar los pasos y en la siguiente clase de Educación Física podrían analizar la información y extraer conclusiones.

3ESO.T.P2 - Brújula	
Objetivos	En esta práctica los alumnos construirán una brújula que mostrará la orientación utilizando el sensor correspondiente de la placa micro:bit y la matriz de LEDs.
Contenidos	El alumno aprenderá a utilizar el sensor brújula de la placa, además de trabajar la estructura condicional en programación y

	el manejo de variables. Por otro lado, también se trabaja con los ángulos de una circunferencia y cómo pueden traducirse para averiguar la orientación.
Conceptos previos	El alumno deberá haber realizado la práctica 0 para conocer la placa y su programación. Será interesante que ya sepa trabajar con variables. Además, la actividad será más aprovechable si el alumno tiene conocimientos de la circunferencia y los ángulos.
Metodología	Esta actividad comenzará con una explicación teórica por parte del profesor sobre los objetivos de la práctica y lo que se va a aprender. El profesor deberá hacer hincapié en la estructura condicional que se va a utilizar, de forma que recuerden este concepto que habrán visto en cursos anteriores. En esta práctica se trabajará también cómo funciona una brújula y cómo podemos saber la orientación conociendo los ángulos a partir del punto donde la brújula marque el norte. Si se sigue la propuesta STEAM definida para esta práctica, esta última parte podría trabajarla el profesor de Física y Química. En esta ocasión, el código propuesto para el ejercicio se ofrece con huecos en blanco, de forma que el alumno pueda utilizar los conocimientos adquiridos en la explicación práctica para configurar el programa.
Material	Un ordenador con acceso a Internet Una placa micro:bit + cable micro USB-USB Un guion (impreso o digital)
Propuesta STEAM	Para esta práctica se propone el trabajo conjunto con el profesorado de la asignatura de Física y Química. El análisis y el diseño de una brújula forma parte del bloque sobre el movimiento y las fuerzas, por lo que esta práctica puede ser un buen ejemplo de interdisciplinaridad de ambas materias.

3ESO.T.P3 - Detectando luz	
Objetivos	En esta práctica los alumnos medirán la luminosidad del aula a través de la placa micro:bit, diseñará una luz quitamiedos y un sismógrafo.
Contenidos	El alumno aprenderá a utilizar el sensor de luminosidad de la placa y a crear un gráfico a partir de la función *plot* de la misma. Introduciendo de esta forma al alumno el concepto de función y de parámetros de una función.
Conceptos previos	El alumno deberá haber realizado la práctica 0 para conocer la placa y su programación. Será interesante que ya sepa trabajar con variables.
Metodología	En esta práctica se propone que el profesor comience con una descripción de los objetivos que se van a desarrollar, haciendo especial hincapié en el concepto de función y de parámetros de una función que será el nuevo concepto de esta práctica. A continuación, el alumno realizará los ejercicios guiados para finalmente realizar los últimos dos ejercicios de forma autónoma. En el primero deberá diseñar y programar una luz quitamiedos a partir de los conocimientos adquiridos en la primera parte de la práctica. En el último ejercicio se le presenta un ejercicio en el que tiene que diseñar y programar un sismógrafo utilizando el sensor encargado de medir movimiento de la placa. Es decir, tiene que utilizar los conocimientos adquiridos en otro ámbito de actuación.
Material	Un ordenador con acceso a Internet Una placa micro:bit + cable micro USB-USB Un guion (impreso o digital)

3ESO.T.P4 - Alarma antirrobo	
Objetivos	En esta práctica los alumnos fabricarán un sistema de alarma antirrobo basado en detección de presencia.
Contenidos	Por un lado, el alumno aprenderá a conectar y manejar sensores digitales externos (un sensor de presencia) y repasará

	el uso del zumbador. En la programación del sistema, aprenderán a leer valores de un sensor digital, repasarán las estructuras condicionales y aprenderán a utilizar la estructura de repetición *while* dentro de un programa.
Conceptos previos	El alumno deberá haber realizado la práctica 0 para conocer la placa y su programación. También deberá conocer lo que es una variable, saber trabajar con ellas, y las estructuras condicionales.
Metodología	Se propone comenzar con una explicación de los objetivos de la práctica y de los conceptos que se van a aprender. Especialmente se deberá analizar por un lado el funcionamiento de un sensor digital y cómo interpretar la información que éstos ofrecen, y por otro lado la estructura de repetición *while.* Una vez trabajados estos conceptos, los alumnos podrán realizar la práctica guiada. Finalmente se propone al alumno la realización de un ejercicio en el que tendrá que aplicar los conocimientos adquiridos sobre el bucle de repetición *while* para, utilizando el sensor de temperatura de la placa, crear un sistema de alarma que avise cuando la temperatura de la habitación supere un cierto valor.
Material	Un ordenador con acceso a Internet Una placa micro:bit + cable micro USB-USB Zumbador + pinzas de conexión Sensor PIR + cables + pinzas de conexión Un guion (impreso o digital)

3ESO.T.P5 - Control de motores	
Objetivos	En esta práctica los alumnos aprenderán a controlar un servo motor conectado a la placa micro:bit.
Contenidos	Los conceptos a trabajar en esta práctica son la conexión y utilización de un actuador externos a la placa, en este caso un servo motor, además del repaso de los conceptos de variables, funciones y estructuras condicionales.

Conceptos previos	El alumno deberá haber realizado la práctica 0 para conocer la placa y su programación. También deberá conocer lo que es una variable, saber trabajar con ellas, y las estructuras condicionales. Será interesante el que conozca de antemano lo que es una función, de forma que pueda practicarlo. Si no se le explicaría antes de realizar los ejercicios.
Metodología	Se comenzaría con una exposición de los objetivos seguida de una explicación de los conceptos que se van a trabajar, recordando lo que son y para qué sirven las variables, las funciones y las estructuras condicionales o de decisión. Una vez terminado el recordatorio, se trabaja sobre la solución al problema propuesto y los distintos pasos que hay que realizar. También se explicará qué es un servo motor y cómo se conecta a la placa. La solución puede ser explicada por el profesor y después ejecutada de forma individual por el alumno. Finalmente, se propone al alumno el reto de diseñar y programar, de forma autónoma y utilizando lo aprendido anteriormente, una barrera que al apretar el botón A se abra o cierre dependiendo del estado anterior.
Material	Un ordenador con acceso a Internet Una placa micro:bit + cable microUSB-USB Servo motor + pinzas de conexión Un guion (impreso o digital)

- **Control y Robótica**

El currículum de la asignatura Control y Robótica en Cantabria abarca diferentes aspectos como la electrónica (analógica y digital), el diseño de robots, conceptos básicos de programación como variables, estructuras de repetición y condicionales, y el diseño e impresión 3D. En este sentido, las prácticas descritas a continuación tienen el objetivo de contribuir al desarrollo de los tres primeros bloques (pudiendo ampliar al cuarto si el docente lo considera).

3ESO.R.P0 - Construyendo una ciudad inteligente	
Objetivos	Esta práctica será una introducción a la propuesta que engloba el resto de prácticas descritas para la asignatura. La idea es la de crear una maqueta de ciudad inteligente en la que se construyan, programen e integren tres sistemas: riego sostenible, parking y alumbrado inteligentes. Además, se propondrá al alumnado que compartan ideas para llegar a un consenso y añadir un sistema adicional a la ciudad.
Contenidos	En estas prácticas se trabajarán diferentes aspectos de programación, la electrónica y la robótica que se especificarán en cada caso. Este proyecto analizará, además de estos conceptos, cómo la tecnología está intrincada en nuestras ciudades, ayudando a mejorar la calidad de vida de los ciudadanos a través de una gestión más eficiente y sostenible de sus servicios.
Conceptos previos	El alumnado deberá conocer el uso de la placa micro:bit y su entorno de programación. También será interesante que conozca los conceptos de variables, lectura y escritura de sensores y las estructuras de programación de condicionales y bucles. La asignatura marco de este proyecto es optativa, y se imparte en el mismo curso que la Tecnología de 3º ESO, por lo que sería adecuado que los alumnos hubieran terminado el bloque de lenguajes de programación antes de comenzar con este proyecto, o que se impartieran de forma simultánea.
Metodología	En la asignatura de Control y Robótica es interesante que los alumnos conozcan más de una placa de programación, por lo que este proyecto no debería abarcar todo el curso. Por ello, se propone que los alumnos sean divididos en 4 grupos y que cada grupo realice una de las 4 prácticas propuestas (3 + 1 propuesta por el alumnado). Dentro de cada grupo, los alumnos deberán trabajar en las distintas fases del proyecto (diseño, construcción, programación e integración), aunque el profesor

	repartirá los roles que permitirán un trabajo ordenado en cada grupo. Al final de todo el proyecto, se deberán integrar los distintos sistemas en una única maqueta. En esta primera práctica 0, el profesor explicará qué es una ciudad inteligente y el objetivo de este proyecto. También explicará los distintos sistemas a construir, y les propondrá que entre todos decidan un nuevo sistema para la maqueta. A la semana siguiente, los alumnos deberán traer una idea de qué sistema añadir, y la describirán brevemente en clase: qué se propone y qué se podría utilizar para construirlo sabiendo cómo funciona micro:bit. Entre todas aquellas ideas que sean factibles de realizar, se votará una y se incluirá en la maqueta.
Material	Se especifica en cada práctica del proyecto
Propuesta STEAM	Aunque no se plantea el trabajo con otros departamentos, sería interesante complementar esta actividad con una visita al centro de demostraciones de Santander, donde les explicarían cómo funciona la plataforma de ciudad inteligente de Santander.

3ESO.R.P1 - Alumbrado inteligente	
Objetivos	El alumnado realizará una maqueta de alumbrado inteligente en el que las farolas estarán controladas por un sensor de luminosidad, de forma que éstas se encenderán o apagarán dependiendo del nivel de luz ambiental.
Contenidos	En esta práctica los alumnos aprenden a trabajar con sensores analógicos y a calibrarlos para que puedan ser utilizados según sus necesidades, en este caso un LDR (resistencia dependiente de la luz). También repasarán el manejo de variables, las estructuras condicionales y la programación de actuadores externos (LEDs en este caso). Para las conexiones también se les enseñará a utilizar una placa de prototipado. Finalmente, los alumnos también trabajarán aspectos de Tecnología para la construcción física del sistema.

Conceptos previos	El alumnado deberá conocer el uso de la placa micro:bit y su entorno de programación. También deberá conocer los conceptos de variable, lectura y escritura de sensores y las estructuras de programación de condicionales y bucles.
Metodología	Los objetivos de la práctica fueron explicados durante la práctica 0. Sin embargo, el profesor deberá dedicar un tiempo al grupo para asegurarse de que están claros los objetivos, aclarar las dudas que puedan surgir, ayudarles en la organización, y asegurarse de que cada miembro del grupo conoce cuál es su papel.
Material	Ordenador con acceso a Internet Placa micro:bit + soporte para pilas + pilas Placa extensión micro:bit Placa de prototipado LDR + LED blanco + cables Materiales del aula para la construcción de la farola

3ESO.R.P2 - Riego sostenible e inteligente	
Objetivos	El alumnado realizará un sistema de riego sostenible e inteligente en el que se simula el riego de un huerto urbano utilizando un tanque de agua que se llena con agua de la lluvia. El huerto sólo deberá regarse cuando la tierra esté seca, para lo que se utilizará un sensor de humedad de tierra. También se deberá controlar la cantidad de agua del tanque, para lo que se utilizará un sensor de nivel de agua.
Contenidos	En esta práctica los alumnos aprenden a trabajar con sensores analógicos y a calibrarlos para que puedan ser utilizados según sus necesidades. En este caso un sensor de humedad de suelo y otro de nivel de agua. También repasarán el manejo de variables, las estructuras condicionales y la programación de actuadores externos, que en este caso será un servo motor. Para las conexiones también se les enseñará a utilizar una placa de prototipado. Finalmente también trabajarán conceptos

	de tecnología, ya que tendrán que construir el prototipo de maqueta.
Conceptos previos	El alumnado deberá conocer el uso de la placa micro:bit y su entorno de programación. También deberá conocer los conceptos de variable, lectura y escritura de sensores y las estructuras de programación de condicionales y bucles.
Metodología	Los objetivos de la práctica fueron explicados durante la práctica 0. Sin embargo, el profesor deberá dedicar un tiempo al grupo para asegurarse de que están claros los objetivos, aclarar las dudas que puedan surgir, ayudarles en la organización, y asegurarse de que cada miembro del grupo conoce cuál es su papel.
Material	Ordenador con conexión a Internet Placa micro:bit + soporte para pilas + pilas Placa extensión micro:bit Placa de prototipado Sensor de humedad del suelo + cables Sensor de nivel de agua + cables Servo motor + cables Materiales del aula para la construcción del sistema de riego

3ESO.R.P3 - Aparcamiento inteligente	
Objetivos	El grupo realizará la maqueta de un aparcamiento que constará de una entrada y salida con barreras. El aparcamiento detectará cuando haya un coche en la entrada o la salida. En la salida siempre se abrirá la barrera. En la entrada sólo se abrirá cuando haya plazas libres en el interior. También habrá unos LEDs que indicarán si quedan plazas libres en el interior.
Contenidos	En esta práctica los alumnos aprenden a trabajar con sensores analógicos y a calibrarlos para que puedan ser utilizados según sus necesidades. En este caso trabajarán con el sensor de ultrasonidos, teniendo que usar la librería correspondiente para su programación. También repasarán el manejo de variables,

	las estructuras condicionales y la programación de actuadores externos, que en este caso serán LEDs. Para las conexiones también se les enseñará a utilizar una placa de prototipado. Finalmente trabajarán conceptos de tecnología, ya que tendrán que construir la maqueta del aparcamiento con materiales disponibles en el aula.
Conceptos previos	El alumnado deberá conocer el uso de la placa micro:bit y su entorno de programación. También deberá conocer los conceptos de variable, lectura y escritura de sensores y las estructuras de programación de condicionales y bucles.
Metodología	Los objetivos de la práctica fueron explicados por el profesor durante la práctica 0. Sin embargo, el profesor deberá dedicar un tiempo al grupo para asegurarse de que están claros los objetivos, aclarar las dudas que puedan surgir, ayudarles en la organización, y asegurarse de que cada miembro del grupo conoce cuál es su papel.
Material	Ordenador con acceso a Internet Placa micro:bit + soporte para pilas + pilas Placa extensión micro:bit Placa de prototipado 2 sensores de ultrasonidos + cables LED rojo y LED verde + cables Materiales del aula para la construcción de la maqueta de aparcamiento.

4.1.3. Propuesta para 4º de la ESO

- **Tecnología**

Las prácticas descritas a continuación tienen el objetivo de servir a los bloques “TIC” y “Control y Robótica” del currículum de Cantabria de esta asignatura, en la que el alumno debe aprender un lenguaje de programación y desarrollar programas para controlar robots. En este sentido, las prácticas descritas a

continuación proponen el aprendizaje de un lenguaje de programación (Python, como se concluyó en la Sección 2.3.2), pudiendo aplicar estos conocimientos en el trabajo con la placa micro:bit y el desarrollo de un pequeño robot. Aunque en el currículum no se define el lenguaje a utilizar, se propone el comienzo con un lenguaje textual sencillo, Python, ya que se ha trabajado con lenguajes de programación por bloques anteriormente en varios cursos.

En esta asignatura se profundizará en las bases de la programación explicadas en cursos anteriores, y por ello las prácticas están divididas en 4 apartados: introducción, variables, estructuras condicionales y estructuras de repetición.

4ESO.P0 – Primeros pasos en Python con micro:bit	
Objetivos	Los alumnos retomarán el contacto con la placa micro:bit que habrán visto (en principio) en cursos anteriores. En este caso programarán la placa en el lenguaje textual Python, aparcando la programación por bloques. En esta práctica recordarán las posibilidades que ofrece micro:bit y se les presentará el entorno de programación de esta placa en Python. También realizarán su primer programa en este lenguaje, lo cargarán en la placa y lo ejecutarán.
Contenidos	En esta práctica el alumnado tendrá una toma de contacto con el lenguaje Python y conocerán y aprenderán a utilizar el entorno de programación a través del cual programarán en Python la placa micro:bit. Se les presentará el lenguaje Python, y reconocerán algunas de sus características como el concepto de objeto y método, o su estructura gracias al sangrado de su código.
Conceptos previos	Es interesante que el alumno conozca la placa micro:bit y que haya programado con anterioridad en algún lenguaje de programación gráfico o textual de forma que sepa qué es un programa y cuál es el objetivo de la programación.
Metodología	Esta práctica puede ser utilizada por el profesor como marco para una explicación general de los lenguajes de programación, y más específicamente sobre las razones del estudio del

	lenguaje Python en la asignatura. De esta forma se contextualiza al alumno motivándole en el trabajo de este bloque.
Material	Ordenador con acceso a Internet Placa micro:bit Guion de prácticas (en papel o digital)

4ESO.P1 – Variables en Python	
Objetivos	El grupo realizará los programas en Python de un cuentapasos, un generador de mensajes positivos, y una calculadora de sumas.
Contenidos	En esta práctica los alumnos trabajarán el concepto de variables de una forma más profunda que en cursos anteriores. Se les presentará qué es una variable, para qué sirven y los diferentes tipos de variables. Aprenderán también cómo se manejan las variables en el lenguaje Python. A través de los ejercicios descritos podrán realizar programas en los que se manejen distintas variables, además de distintas instrucciones de uso común en Python. Finalmente, también aprenderán a programar en Python el uso de algunos sensores/actuadores disponibles en la placa micro:bit como son el acelerómetro o los botones.
Conceptos previos	El alumnado deberá conocer el uso de la placa micro:bit y su entorno de programación en Python, explicado en la práctica 0. Será interesante que el alumno conozca el concepto de variable para que parta de un conocimiento previo, aunque tampoco es indispensable.
Metodología	Se propone que el profesor explique el concepto de variable acompañado de diferentes ejemplos que lo haga más cercano al estudiante. Para ello puede utilizarse la introducción del guion. La programación puede ser vista como algo muy abstracto por los alumnos, el profesor debe esforzarse en rebajar la complejidad y abstracción de estos conceptos. También deberá explicar los diferentes tipos de variables que

	puede haber dependiendo de la información que almacenen. Finalmente explicará cómo se utilizan las variables en el lenguaje Python a través de los ejercicios. Se propone que al menos el primer ejercicio sea realizado por el profesor y por el alumnado al mismo tiempo. A la vez que el profesor lo explica, lo puede ir realizando en su ordenador mientras es proyectado. De esta forma el alumno pueda realizar el ejercicio en su ordenador a la vez que sigue la explicación del profesor. Puede seguirse esta metodología para algún ejercicio o para todos, ya que la última práctica de este curso (P4) será realizada de forma autónoma por los alumnos.
Material	Ordenador con acceso a Internet Placa micro:bit Guion de prácticas (en papel o digital)

4ESO.P2 – Estructuras condicionales en Python	
Objetivos	En esta práctica los alumnos realizarán programas Python en los que manejen la matriz de LEDs, realicen una brújula, y la bola 8 de adivinación.
Contenidos	Los alumnos estudiarán las estructuras condicionales en programación. Aunque ya han visto estas estructuras en cursos anteriores, se trabajará más profundamente en ellas, y se concretará su uso en el lenguaje Python. Se estudiarán tres tipos de estructuras condicionales “if”, “if…else”, y “if…else if..else…”. A través de los ejercicios descritos realizarán programas en los que se manejen estas estructuras e instrucciones de uso común. También repasarán y aprenderán a programar en Python algunos sensores y actuadores disponibles en la placa micro:bit como son los botones, el acelerómetro o la brújula.
Conceptos previos	El alumnado deberá conocer el uso de la placa micro:bit y su entorno de programación. También deberá conocer los conceptos de variable, y sería interesante que hubieran visto en

	algún momento las estructuras condicionales. Aunque no es un requisito indispensable, de esta forma el alumno podría construir sobre conocimientos ya adquiridos.
Metodología	La práctica propone una primera introducción en la que el profesor explique las estructuras condicionales. Se aconseja que se utilice una considerable cantidad de ejemplos cercanos al alumno para que entienda estas estructuras como una toma de decisiones de un programa que también pueden ser traducidas a la vida diaria de cualquier persona. De esta forma se elimina abstracción a este concepto. Una vez explicada la teoría genérica de estas estructuras, se concretará en la utilización de las mismas en Python a través de los ejercicios propuestos. Se plantea que al menos el primer ejercicio sea realizado por el profesor y por el alumnado al mismo tiempo. A la vez que el profesor lo explica, lo puede ir realizando en su ordenador mientras es proyectado, de forma que el alumno pueda realizar el ejercicio en su ordenador a la vez que sigue la explicación. Puede seguirse esta metodología para algún ejercicio o para todos, ya que la última práctica de este curso (P4) engloba también estas estructuras, y serán realizadas de forma autónoma por el alumnado.
Material	Ordenador con acceso a Internet Placa micro:bit Guion de prácticas (en papel o digital)

4ESO.P3 – Estructuras de repetición en Python	
Objetivos	En esta práctica se propone que programen un juego de reflejos y que jueguen con la capacidad que tiene la placa de producir sonidos y música, utilizando el lenguaje Python.
Contenidos	Los alumnos estudiarán las estructuras de repetición en programación. En cursos anteriores han trabajado alguna de estas estructuras, aunque en este caso se profundizará más en

	ellas y se concretará su uso en el lenguaje Python. Estudiarán las estructuras FOR y WHILE. A través de los ejercicios descritos realizarán programas en el que se manejen estas estructuras e instrucciones de uso común. También aprenderán y repasarán la programación en Python del uso de algunos actuadores disponibles en la placa y otros externos, como los botones o los zumbadores.
Conceptos previos	El alumnado deberá conocer el uso de la placa micro:bit y su entorno de programación. También deberá conocer los conceptos de variable y estructuras condicionales, de forma que no se tenga que trabajar en exceso sobre ello teniendo que restar tiempo a las estructuras de repetición.
Metodología	Se propone al docente comenzar con una explicación teórica de lo que son las estructuras de repetición, introduciendo ejemplos de uso que sean cercanos al alumno. En la introducción de la práctica se presentan dos ejemplos, pero es importante que el profesor proporcione los suficientes para que el alumno entienda su funcionamiento antes de programar el código. A continuación, se concretarán estas estructuras en el lenguaje Python a través de la realización de los ejercicios propuestos. Se propone que la realización de los ejercicios se realice al mismo tiempo por el profesor y por el alumnado, de forma que éstos realicen la actividad a la vez que el profesor lo explica.
Material	Ordenador con acceso a Internet Placa micro:bit Zumbador + cables + cocodrilos Guion de prácticas (en papel o digital)

4ESO.P4 – Mascota robot	
Objetivos	Esta práctica tiene el objetivo de que los alumnos, por grupos, construyan un robot que realice diferentes acciones dependiendo de los estímulos que reciba a través de los sensores que tiene conectados.

	Se proponen dos robots (a escoger por los grupos), uno que utilice un sensor de ultrasonidos (para medir distancias) y otro que utilice un sensor de sonido. En función de la información captada por esos sensores los robots realizarán diferentes acciones como pueden ser producir sonidos, mostrar mensajes o imágenes en la matriz de LEDs, etc.
Contenidos	En la realización de cualquiera de las dos propuestas, se trabajarán los conceptos que han aprendido en las prácticas anteriores relacionadas con variables, estructuras condicionales y de repetición, además de las diferentes instrucciones Python que han ido utilizando. En esta práctica aprenderán a utilizar sensores externos a la placa micro:bit que necesitan de módulos (ficheros) que permiten el manejo de estos sensores de forma sencilla. Para ello se les enseñará qué son los módulos o librerías, para qué sirven y cómo utilizarlos en un programa. Aprovechando esta ocasión se les puede hablar de plataformas donde los programadores comparten código, como es GitHub, y de la importancia del respeto al trabajo de otros. Para la utilización de los sensores, además del uso de módulos, también aprenderán la necesidad de revisar las hojas de características de los dispositivos y la electrónica necesaria para su conexión (en este caso un divisor de tensión). En este último aspecto no es necesario que conozcan todos los detalles, pero sí el razonamiento que hay detrás. Como marco de estos conocimientos, estará la creación del robot: diseño, programación y construcción. Además de todo lo anterior, destacar que el alumno trabajará sus habilidades de trabajo en equipo.
Conceptos previos	Para la realización de esta práctica el alumnado deberá conocer tanto la placa micro:bit y su entorno de programación en Python, como los conceptos de variables y las estructuras condicionales

	y de repetición, tanto a nivel conceptual como su implementación en lenguaje Python.
Metodología	Para la realización de esta práctica se propone el trabajo en grupo. El profesor podrá explicar a todo el alumnado el objetivo de esta práctica y las dos opciones existentes, para que una vez formados los grupos ellos puedan escoger el robot que quieren construir. Y aprovechando que es algo que harán todos, escojan el robot que escojan, podrá explicar cómo se va a trabajar con los sensores externos. Después, propondrá la creación de grupos (número de alumnos por grupo a elección del docente) y éstos decidirán la opción de robot que quieren realizar. A continuación, el profesor explicará los roles que tiene que haber en cada grupo y, bien entre ellos o a elección del profesor, se asignarán los roles entre los alumnos del grupo. Finalmente, el profesor deberá visitar a cada grupo, comprobando que todos los alumnos saben cuál es su rol y qué tienen que hacer, ayudarles a diseñar la hoja de ruta y asegurarse de que no tienen dudas que les impida comenzar el trabajo. Para esta práctica no se proporciona el código ya que éste dependerá del diseño, aunque se proporcionan instrucciones de cómo trabajar con los sensores.
Material	Ordenador con acceso a Internet Placa micro:bit + soporte para pilas + pilas Placa de prototipado + resistencias + cables Sensores de ultrasonido Sensores de sonido Materiales del aula para la construcción del robot

5. Conclusiones y líneas futuras

La programación es una disciplina que está gozando de una gran aceptación en los últimos años. Esto es debido a que el aprendizaje de esta materia no sólo contribuye a formar al alumno en una nueva rama de la ciencia y la tecnología, sino que proporciona al alumno una herramienta más, como pueden ser la lectura o las matemáticas, para la resolución de problemas.

Es habitual que la programación vaya acompañada de la robótica, en la que los alumnos no sólo construyen robots, poniendo en práctica los conocimientos de programación que adquieren, sino que desarrollan otras capacidades como son el aprendizaje colaborativo o la creatividad.

Debido a la gran importancia que se les atribuye en la actualidad a estas disciplinas, los currículums están comenzando a tener un lugar reservado para su desarrollo. En este trabajo se ha presentado el estado de la enseñanza de la programación y la robótica en Europa, España, y la Comunidad Autónoma de Cantabria. También se ha proporcionado una visión general de las herramientas más utilizadas en las aulas para su enseñanza, proporcionando información sobre las placas programables más comunes, y los lenguajes y entornos de programación más utilizados. En esta síntesis, destaca la placa micro:bit como una herramienta que, aun teniendo un protagonismo muy puntual en las aulas españolas, posee un gran potencial para el trabajo de la programación y la robótica, tal y como ha demostrado su utilización en Inglaterra.

Con esta motivación, este trabajo propone la utilización de las placas micro:bit como hilo conductor de la enseñanza de la programación y la robótica en los niveles de Educación Secundaria Obligatoria. Para ello, proporciona una serie completa de prácticas en las que se desarrollan diferentes conceptos de la programación, y que pueden llevarse a cabo utilizando la placa micro:bit y sus diferentes entornos de programación.

Para finalizar, este trabajo deja abierta una serie de líneas futuras que podrían ofrecer mejoras a la propuesta aquí desarrollada. Por un lado, sería interesante estudiar el resultado de la implantación de esta propuesta en el aula, ya que por

limitaciones temporales no ha sido posible llevarlas a cabo en un caso real. En este estudio, sería interesante analizar tanto la temporalización de las mismas, para poder ofrecer una aproximación inicial al docente, así como su aceptación por parte del alumnado. Por otro lado, sería interesante también el realizar un estudio paralelo de los beneficios que tiene la utilización de esta placa en el aula, como el realizado en (King's College London, s.f.), pero en el entorno escolar español en el que los tiempos y la filosofía pueden ser diferentes. Por último, se propone también la evaluación de su posible aplicación en la etapa de 1º de Bachillerato, como forma de ofrecer continuidad al aprendizaje de programación y robótica a través del mismo medio conductor.

Referencias

About the micro:bit foundation. (n.d.). Retrieved Junio 17, 2018, from http://www.microbit.org/about/#mission-statement

Arduino Corporation. (n.d.). *Arduino*. Retrieved Junio 17, 2018, from https://www.arduino.cc/

Arduino Products. (n.d.). Retrieved Junio 17, 2018, from https://www.arduino.cc/en/Main/Products

Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015, Febrero). From Scratch to "Real" Programming. *ACM Transactions on Computing Education, 14*(4). doi:http://dx.doi.org/10.1145/2677087

ASTI Talent&Tech Foundation. (n.d.). *STEM Talen Girl*. Retrieved Junio 17, 2018, from https://talent-girl.com/

BBC micro:bit pins. (n.d.). Retrieved Junio 17, 2018, from http://microbit.org/guide/hardware/pins/

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education - Implications for policy and practice.* Publications Office of the European Union. doi:https://doi.org/10.2791/792158

Brithish Broadcast Coorporation. (n.d.). *The BBC micro:bit*. Retrieved Junio 17, 2018, from http://www.bbc.co.uk/programmes/articles/4hVG2Br1W1LKCmw8nSm9WnQ/the-bbc-micro-bit

CantabRobots. (n.d.). Retrieved Junio 17, 2018, from Concurso-Exhibición de Robótica Escolar en Cantabria: http://www.cantabrobots.es/

Comisión Europea. (2014). Innovative ways to make science education and scientific careers attractive to young people. *Topic identifier: SEAC-1-2014*. Bruselas. Retrieved Junio 17, 2018, from http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/seac-1-2014.html

Gobierno de Canarias. (2017). *Programa STEAM Canarias*. Retrieved Junio 17, 2018, from http://www.gobiernodecanarias.org/educacion/web/programas-redes-educativas/programas-educativos/steam/

Gobierno de Cantabria. (5 de Junio de 2015). Decreto 38/2015, de 22 de mayo, que establece el currículo de la Educación Secundaria Obligatoria y del Bachillerato en la Comunidad Autónoma de Cantabria. *(39)*. Boletín Oficial de Cantabria.

Google Developers. (n.d.). *Blocky Programming Editor*. Retrieved Junio 17, 2018, from https://developers.google.com/blockly/

Grandell, L., Peltomäki, M., Back, R.-J., & Salakoski, T. (2006). Why Complicate Things? Introducing Programming in High School Using Python. (D. Tolhurst, & S. Mann, Edits.) *Proceedings of the 8th Australasian Computing Education Conference, 52*, 71-80.

Innova Didactic. Tecnologías creativas para la educación. (n.d.). *Placa Imagina-Scratch*. Retrieved Junio 17, 2018, from http://docs.innovadidactic.com/producto/sensores/rbl0672_v2.1

Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado. (2018). *Programación, robótica y pensamiento computacional en el aula.* Ministerio de Educación, Cultura y Deporte. Retrieved Junio 17, 2018, from http://code.educalab.es/wp-content/uploads/2017/09/Pensamiento-Computacional-Fase-1-Informe-sobre-la-situaci%C3%B3n-en-Espa%C3%B1a.pdf

Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado. (n.d.). *Pensamiento computacional en Castilla y León*. Retrieved Junio 17, 2018, from Code-educaLAB: http://code.educalab.es/situacion-en-espana/castilla-y-leon/

Investigación acerca de Scratch. (n.d.). Retrieved Junio 17, 2018, from https://scratch.mit.edu/research

King's College London. (n.d.). *Evaluation of Micro-bit*. Retrieved Junio 17, 2018, from Current projects: https://www.kcl.ac.uk/sspp/departments/education/research/Research-Centres/crestem/Research/Current-Projects/Evaluation-of-Micro-bit.aspx

Koulouri, T., Lauria, S., & Madredie, R. (2015, Febrero). Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches. *ACM Transactions on Computing Education, 14*(26). doi:http://dx.doi.org/10.1145/2662412

Kroes, N., & Vassiliou, A. (2014, Julio 25). Carta a los ministros de Educación europeos. Bruselas. Retrieved Junio 16, 2018, from http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?doc_id=6597

Kruglyk, V., & Lvov, M. (2012). Choosing the first educational programming language. *Proceedings of the 8th International Conference on ICT in Education, Research and Industrial*, (pp. 188-198). Jersón. Retrieved Junio 17, 2018, from http://ceur-ws.org/Vol-848/ICTERI-2012-CEUR-WS-paper-37-p-188-198.pdf

Lego Group. (n.d.). *Lego Mindstorms*. Retrieved Junio 17, 2018, from https://www.lego.com/es-es/mindstorms

Micro:bit. (n.d.). Retrieved Junio 17, 2018, from http://microbit.org/es/

Microsoft. (n.d.). *Micro:bit Code Editor*. Retrieved Junio 2018, 17, from https://makecode.microbit.org

Ministerio de Educación y Formación Profesional de España. (2013, Diciembre 10). Ley Orgánica 8/2013, de 9 de diciembre, para la Mejora de la Calidad Educativa. Boletín Oficial del Estado (BOE).

Mobile apps for micro:bit. (n.d.). Retrieved from http://microbit.org/es/guide/mobile/

Pekka Kasurinen, J. (2016). Python as a programming language for the introductory programming courses. Thesis for the Degree of Bachelor of Science in Tecnology. Lappeenranta University of Technology, Finlandia.

Peña, R. (2015). Python como primera aproximación a la programación. *Revista de Investigación en Docencia Universitaria de la Informática, 8*(2). Retrieved Junio 17, 2018, from http://www.aenui.net/ojs/index.php?journal=revision&page=article&op=view&path%5B%5D=199

Pittí Patiño, K., Muñóz Arracera, L. E., Moreno, I., Serracín Pittí, J. R., Quintero, J., & Quiel, J. (2012). La robótica educativa, una herramienta para la enseñanza-aprendizaje de las ciencias y las tecnologías. *Education In The Knowledge Society (EKS), 13*(2), 74-90.

Power your imagination with code. (n.d.). Retrieved Junio 17, 2018, from http://microbit.org/code/

Python Software Foundation. (n.d.). *micro:bit Python editor*. Retrieved Junio 17, 2018, from http://python.microbit.org/v/1

Raspberry Pi Foundation. (n.d.). *Raspberry Pi*. Retrieved from https://www.raspberrypi.org/

Scratch. (n.d.). Retrieved Junio 17, 2018, from https://scratch.mit.edu/

STEM Learning. (n.d.). *Local STEM Ambassador Hubs*. Retrieved Junio 17, 2018, from https://www.stem.org.uk/stem-ambassadors/local-stem-ambassador-hubs

Stevens, J., & Serate, G. (2016). Python as a First Programming Language. *Science Technology Engineering & Mathematics Bilingual Online Peer Sessions (STEM BOPS).* Retrieved Junio 17, 2018, from https://www.unr.edu/Documents/engineering/stembops/justinStevens_Python.pdf

The Royal Society. (2012). *Shut down or restart? The way forward for computing in UK schools.* Londres. Retrieved Junio 16, 2018, from https://royalsociety.org/~/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf

Tollervey, N. (n.d.). *Code with Mu*. Retrieved Junio 17, 2018, from https://codewith.mu/

Universidad de Cantabria. (2015). *KIKS - Kids Inspire Kids for STEAM*. Retrieved Junio 17, 2018, from https://www.kiks.unican.es/

Universidad de Cantabria. (2017). *STEMforYouth*. Retrieved Junio 17, 2018, from https://stemforyouth.unican.es/stemforyouth/

University of Helsinki. (n.d.). *Luma Centre Finland*. Retrieved Junio 17, 2018, from https://www.luma.fi/en/

Wing, J. M. (2010). Computational Thinking: What and Why? *Link Magazine. Magazine of Carnegie Mellon University's School of Computer Science*. Retrieved Junio 17, 2018, from http://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf

Anexos

Prácticas 2º ESO – Tecnología

P0

Tecnología 2º ESO
Descubriendo micro:bit y la programación

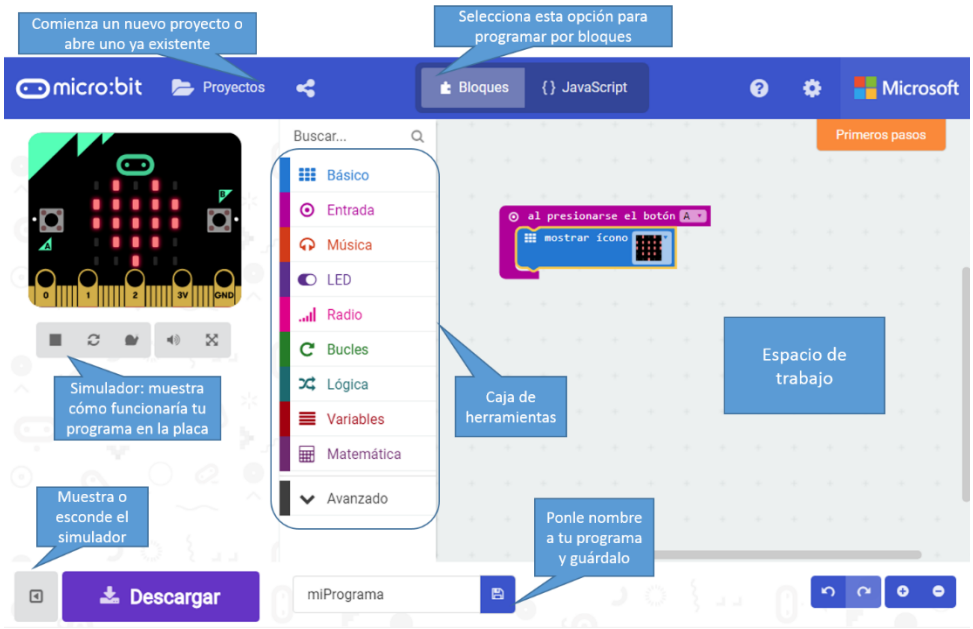
Introducción

Micro:bit es una pequeña placa programable con una gran variedad de sensores que nos permitirá aprender los conceptos de la programación a través de diferentes proyectos de robótica.

En esta práctica vamos a aprender a programar la placa micro:bit utilizando bloques, seleccionando los que nos interese para hacer nuestro programa y encajándolos como si fuera un puzle. Aprenderemos cómo utilizar el entorno de programación y al final escribiremos nuestro primer programa, lo cargaremos en la placa y lo ejecutaremos.

Desarrollo

1. Abre el navegador y accede a la siguiente página web https://makecode.microbit.org/
2. Nos encontraremos con el entorno de programación que vamos a utilizar para programar la placa.



1. **Veamos qué nos ofrece este entorno…**

- Un simulador: en la parte de la izquierda podemos ver una placa micro:bit virtual en la que podremos ejecutar el programa que diseñemos sin tener que subirlo a la placa. Esta

funcionalidad nos va a ser muy útil cuando queramos saber si el programa funciona y para corregir errores.

- Caja de herramientas: está en la parte central de la pantalla y presenta diferentes categorías. Dentro de cada categoría nos encontramos varios bloques que realizan tareas diferentes y que podemos arrastrar al espacio de trabajo para poder usarlos.
- Espacio de trabajo: en la parte derecha de la pantalla nos encontramos un espacio vacío donde crearemos nuestro programa. Para crear un programa utilizaremos los bloques de la caja de herramientas que necesitemos y los arrastraremos y encajaremos como nos convenga en este espacio.

2. ¿De qué herramientas dispongo para programar?

Dentro de la caja de herramientas tenemos diferentes bloques divididos por categorías. Fíjate en que cada categoría es de un color diferente. Vamos a analizar qué nos ofrece cada una de ellas:

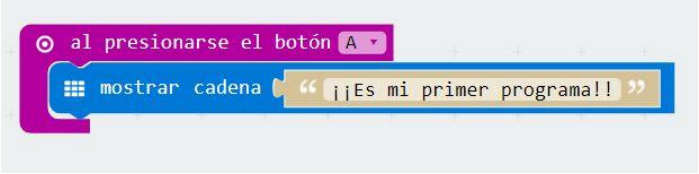
- Básico: contiene bloques de uso frecuente, como el bloque que nos va a permitir ejecutar el programa repetidamente, sólo una vez, o utilizar la matriz de LEDs para mostrar mensajes o dibujos.
- Entrada: contiene bloques que nos permitirán recoger información de los sensores de la placa o para saber cuándo se está tocando un botón o agitando la placa (entre otras cosas).
- Música: con bloques que nos van a permitir reproducir tonos para hacer música y reproducirla en la placa.
- LED: con bloques que nos permitirán encender y apagar los LEDs de forma individual. Recuerda que se empiezan a contar las columnas y las filas por 0 (la primera columna es la columna 0, y lo mismo ocurre con las filas)
- Radio: estos bloques nos permiten comunicar placas micro:bit entre ellas a través de radiofrecuencias, sin tener que unirlas por cable.
- Bucles: estos bloques nos van a permitir no tener que escribir varias veces el mismo código cuando tenemos que repetir instrucciones.
- Lógica: contiene bloques que permitirán a la placa tomar decisiones y hacer unas cosas u otras dependiendo de lo que ocurra anteriormente.
- Variables: con bloques que nos permitirán guardar información y que la placa sea capaz de recordarla y cambiarla cuando sea necesario.
- Matemática: con bloques para hacer operaciones matemáticas y obtener números aleatorios.

3. ¡Vamos a crear nuestro primer programa!

Ahora que ya conocemos un poco más sobre lo que podemos hacer con este entorno y con nuestra placa, vamos a crear nuestro primer programa.

Diseñaremos un programa en el que al apretar el botón A de la placa se muestre en la matriz de LEDs el mensaje: “¡¡Es mi primer programa!!”

Necesitaremos 2 bloques: uno que nos permitirá saber cuándo se toca el botón A (en rosa) y otro para mostrar un mensaje en la matriz de LEDs (en azul). Fíjate en el dibujo, busca los bloques en la caja de herramientas y crea tu propio código.



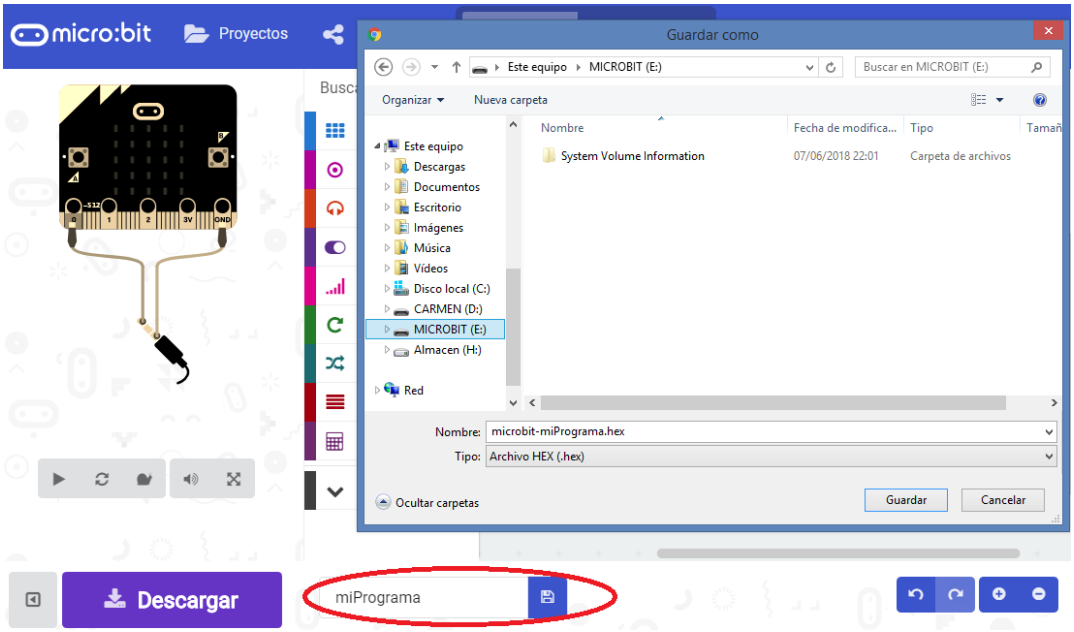
4. ¿Ves el resultado en el simulador?

Ahora vamos a probar el programa en el simulador de la izquierda de la pantalla. Cuando hagamos clic en el botón A aparecerá el mensaje en la pantalla.

5. Ejecutemos el programa en la placa.

a) Conectamos la placa con el cable microUSB-USB al ordenador

b) Ponemos nombre a nuestro programa y lo guardamos en la placa micro:bit que ha detectado el ordenador.



6. Listo!

Mientras se carga el programa en la placa parpadea un LED en la parte trasera. Una vez que deja de parpadear el programa ya estará cargado y lo podrás probar. Cuando presiones el botón A verás el mensaje en la placa.

7. Probemos otros programas….

Ahora puedes probar a mostrar otros mensajes, o a que se muestre el mensaje al pulsar el botón B en lugar del A.

P1 Tecnología 2º ESO
Cambiar de humor apretando un botón

Introducción

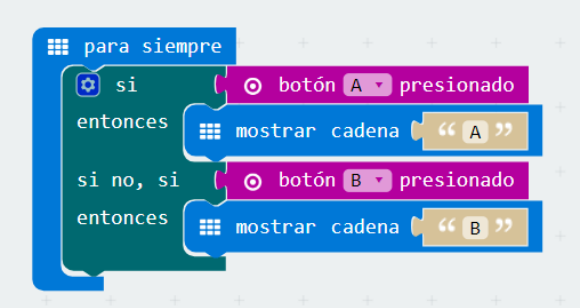
En esta práctica vamos a dotar de inteligencia a la placa y vamos a hacer que muestre mensajes diferentes dependiendo del botón que apretemos.

Vamos a aprender a realizar un programa que funcione de forma continua e infinita (hasta que desconectemos la placa), y en el que la placa hará unas acciones u otras dependiendo de una condición.

Para ello utilizaremos lo que en programación se llama un **bucle** (para que se ejecute continuamente, por siempre o hasta que le digamos que pare), y una estructura **condicional** (si "pasa esto" entonces "haz esto otro") para indicarle qué tiene que hacer en cada situación.

Desarrollo

1. Primero vamos a crear un programa que muestre en la matriz de LEDs la letra A cuando apretemos el botón A, y la letra B cuando apretemos el botón B. Busca en la caja de herramientas los bloques correspondientes para realizar el programa del dibujo. Como ves, necesitarás el bloque "para siempre" que hará que el programa se ejecute continuamente. También necesitarás el bloque "si…entonces…" que tendrás que modificar para que pueda tener en cuenta las dos opciones de nuestro programa y decirle qué tiene que hacer cuando se aprieta el botón A o el botón B.



2. Prueba el programa en el simulador, y una vez que veas que funciona como esperamos, cárgalo en la placa.

3. Ahora que ya sabes cómo hacer que la placa pueda tomar decisiones, nuestro reto será el de programar la placa micro:bit de forma que muestre una cara alegre en la matriz de LEDs cuando pulsemos el botón A y una cara triste cuando pulsemos el botón B. Busca en la caja de herramientas las instrucciones necesarias y realiza el programa.

4. Una vez hayamos conseguido que nuestra placa cambie de humor al tocar los botones, vamos a programar la placa de forma que se muestre un icono (puedes dibujarlo tú o utilizar alguno de los disponibles en la caja de herramientas) cuando apretemos el botón A, otro cuando apretemos B, y otro diferente cuando pulsemos A y B a la vez.

P2 Tecnología 2º ESO
Muévete con alegría

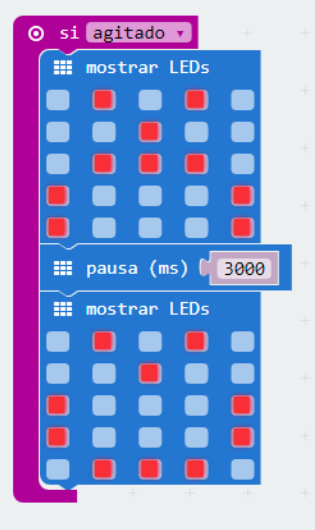
Introducción

En esta práctica vamos a aprender a utilizar alguno de los sensores que tiene la placa micro:bit. En concreto, vamos a aprender a utilizar el sensor acelerómetro.

Este sensor detecta aceleraciones, y si mueves la placa el sensor detectará que se está produciendo movimiento. De hecho, el sensor es capaz de reconocer si se está agitando la placa, si la pantalla está hacia arriba o hacia abajo, si está inclinada y si lo está hacia la izquierda o hacia la derecha, si está en caída libre, etc.

Desarrollo

1. Vamos a programar la placa de forma que cuando se detecte que la placa esté siendo agitada muestre una cara triste en la matriz de LEDs, se mantenga la cara triste 3 segundos y a continuación se muestre una cara alegre. Busca en la caja de herramientas los bloques correspondientes y configúralos para realizar el programa del dibujo.



2. Prueba el programa en el simulador, y una vez que funcione cárgalo en la placa para que veas cómo funciona.

3. Explora las diferentes acciones que puede medir el acelerómetro y programa que cuando la placa esté mirando hacia arriba (los LEDs hacia arriba) se muestre el mensaje “Arriba” y cuando esté mirando hacia abajo se muestre el mensaje “Abajo”.

4. Juega con las diferentes funcionalidades que te ofrece el acelerómetro y crea el sistema que se te ocurra.

P3 Tecnología 2º ESO
Música y tecnología

Introducción

En esta práctica vamos a aprender a crear música con nuestra placa micro:bit y a hacer más eficientes nuestros programas utilizando bucles.

La placa micro:bit nos permite crear música a través de bloques que reproducen una nota determinada durante un tiempo determinado. Ya conoces la escala música, y sabes que las notas pueden tener distinta duración, con estos conocimientos, nuestra placa y un zumbador podremos reproducir la canción que queramos.

Un zumbador es un dispositivo que nos permite reproducir sonidos a una cierta frecuencia, es decir diferentes notas musicales. El zumbador tiene dos cables: uno negro que tendremos que conectar a la tierra (GND) de nuestra placa, y un cable rojo a través del cual le daremos energía para que funcione y le mandaremos las señales que le digan qué nota tiene que interpretar, lo conectaremos con ayuda de una pinza al pin 0 de la placa.

También queremos hacer nuestros programas más eficientes y para ello utilizaremos bucles. Cuando tenemos que repetir instrucciones varias veces, por ejemplo, repetir el estribillo de una canción, tenemos dos opciones: escribir las mismas instrucciones (el estribillo) las veces que necesitemos, o crear un bucle en el que le digamos cuántas veces queremos que se repitan esas instrucciones.

Desarrollo

1. Vamos a programar nuestra placa para que cuando apretemos el botón A reproduzca la canción “Cumpleaños feliz”. Queremos que suene 4 veces, así que utilizaremos el bucle “repetir”. De esta forma le estaremos indicando que lo que metemos dentro de ese bloque queremos que lo repita 4 veces. Busca en la caja de herramientas los bloques que necesites y crea tu canción. Puedes fijarte en el dibujo para saber cuáles son las notas.

2. Ahora conecta el zumbador y después carga el programa en la placa.

Recuerda que el zumbador tiene dos cables: uno negro que tendremos que conectar a la tierra (GND) de nuestra placa, y un cable rojo a través del cual le daremos energía para que funcione y le mandaremos las señales que le digan qué nota tiene que ejecutar, que conectaremos al pin 0 con ayuda de una pinza.

3. Te proponemos un reto, que programes la siguiente canción para que suene 3 veces y adivines cuál es. Recuerda utilizar los bucles cuando sea necesario para no repetir código innecesariamente.

Notas:

Mi' Mi' Mi' Do' Mi' Sol' Sol
Do Sol Mi La Si La# La Sol
Mi' Sol' La' Fa' Sol' Mi' Do' Re' Si
Do Sol Mi La Si La# La Sol
Mi' Sol' La' Fa' Sol' Mi' Do' Re' Si

Sol' Fa#' Fa' Re' Mi'
Sol La Si La Do' Re'
Sol' Fa#' Fa' Re' Mi' Do'' Do'' Do''
Sol' Fa#' Fa' Re' Mi'
Sol La Si La Do' Re' Re#' Re' Do'

Sol' Fa#' Fa' Re' Mi'
Sol La Si La Do' Re'
Sol' Fa#' Fa' Re' Mi' Do'' Do'' Do''
Sol' Fa#' Fa' Re' Mi'
Sol La Si La Do' Re' Re#' Re' Do'

Do' Do' Do' Do' Re' Mi' Do' La Sol
Do' Do' Do' Do' Re' Mi'
Do' Do' Do' Do' Re' Mi' Do' La Sol

Mi' Mi' Mi' Do' Mi' Sol' Sol
Do Sol Mi La Si La# La Sol
Mi' Sol' La' Fa' Sol' Mi' Do' Re' Si
Do Sol Mi La Si La# La Sol
Mi' Sol' La' Fa' Sol' Mi' Do' Re' Si

Mi' Do Sol Sol# La Fa' Fa' La
La La' La' La' Sol' Fa' Mi' Do' La Sol

Mi' Do' Sol Sol# La Fa' Fa' La
La Fa' Fa' Fa' Mi' Re' Do'

Mi' Do Sol Sol# La Fa' Fa' La
La La' La' La' Sol' Fa' Mi' Do' La Sol

Mi' Do' Sol Sol# La Fa' Fa' La
La Fa' Fa' Fa' Mi' Re' Do'

P4 Tecnología 2º ESO
Jugando a piedra, papel o tijera

Introducción

En esta práctica vamos a programar el juego de "Piedra, papel o tijera" para que podamos jugar contra la placa.

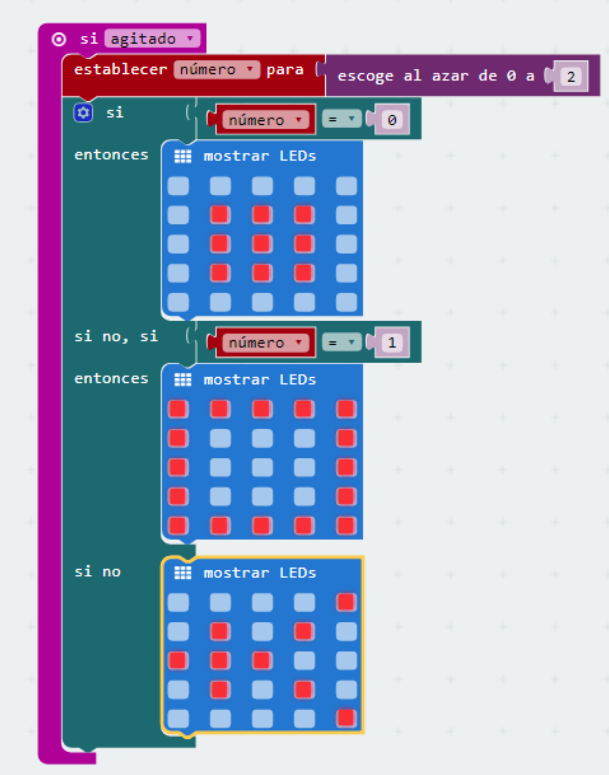
Vamos a repasar funcionalidades que ya hemos aprendido en prácticas anteriores como leer el acelerómetro o hacer que nuestra placa realice unas acciones u otras dependiendo de una condición. Pero además, vamos a aprender un concepto nuevo muy importante en programación: vamos a conocer qué es una variable y cómo nos ayudan a guardar información que podemos usar durante la vida de nuestro programa.

Puedes imaginar una variable como un contenedor de información, que nos va a permitir guardar información que luego podemos mirar, cambiar, o utilizar.

Nuestro juego va a consistir en lo siguiente: cuando agitemos la placa, en la matriz de LEDs veremos uno de los símbolos de nuestro juego (una piedra, un papel o una tijera). Para que parezca que estamos jugando con una persona, programaremos que aparezca un símbolo u otro de forma aleatoria.

Desarrollo

1. En el dibujo puedes ver el programa que vamos a realizar. El bloque que engloba todo el código indica que se ejecutará lo que hay dentro cuando se agite la placa. Cuando esto ocurra queremos que aparezca un símbolo aleatorio. Para programar esto, lo que haremos será calcular un número aleatorio y guardarlo en una variable, que en este caso hemos llamado número pero puedes llamarla como quieras. Después, una vez que tenemos el número aleatorio del 0 al 2 (porque vamos a tener sólo 3 opciones), vamos a programar las decisiones que tiene que tomar la placa. Si el número aleatorio que sale es 0 dibujaremos una piedra, si es 1 un papel, y si es 2 una tijera. Puedes programar la relación de números con símbolos como desees, esto es sólo un ejemplo.



2. Busca los bloques en la caja de herramientas y programa el juego. Después cárgalo en la placa y comprueba que funciona según lo esperado.

3. Ahora tu reto será el de programar la placa como si fuera un dado. Cada vez que agites la placa, en la matriz de LEDs deberá aparecer de forma aleatoria la cara de un dado.

P1 Tecnología 3º ESO
Cuentapasos

Introducción

En esta práctica vamos a crear un dispositivo que habrás visto en las tiendas pero que tú puedes crear fácilmente con micro:bit, un cuentapasos.

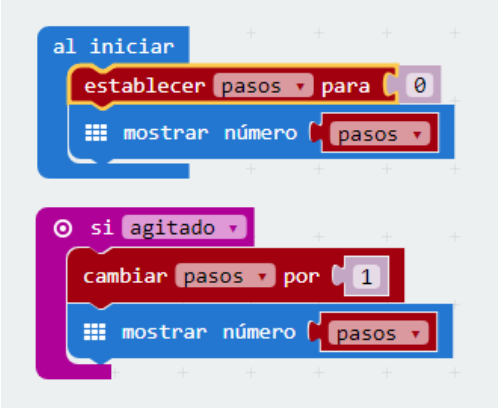
Para fabricar nuestro cuentapasos utilizaremos el sensor acelerómetro de la placa, que detecta cuándo ésta se mueve. Sabiendo que los cuentapasos funcionan detectando el movimiento que hacemos al andar, si colocamos nuestra placa en la muñeca o en el tobillo, detectará el movimiento de brazos o piernas que se hace normalmente al caminar. En nuestro caso podemos detectar estos movimientos con la función "si agitado" que detecta movimiento de agitación.

Por lo tanto, en esta práctica vamos a aprender a leer datos recogidos por un sensor en micro:bit, y a guardar información en memoria para utilizarla cuando necesitemos. Para esto último utilizamos variables, que son contenedores de información donde podemos guardar datos que podremos leer o modificar en el futuro.

Desarrollo

1. En primer lugar, cuando se encienda nuestro cuentapasos, crearemos la variable que se va a encargar de guardar el número de pasos que vayamos realizando, y la inicializamos a 0 pasos. Podemos mostrar en la matriz el valor inicial, para que el usuario pueda ver que empieza desde 0. A continuación, programamos que cada vez que se agite la placa, sumaremos 1 al valor que haya almacenado en la variable pasos. De esta forma podremos ir contabilizando los pasos que vamos haciendo. También podemos mostrar el nuevo valor en la matriz de LEDs cada vez que se modifique.

Busca los bloques correspondientes en la caja de herramientas y crea tu propio código. Después de comprobar en el simulador que funciona correctamente, súbelo a la placa y ejecútalo.



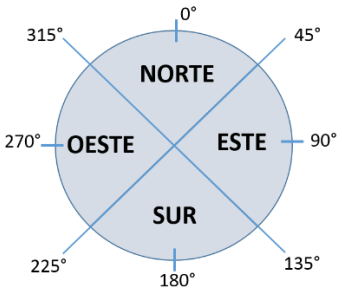
2. Ahora te proponemos el reto de programar un sistema que calcule cuándo hemos conseguido realizar 10000 pasos. Cuando se consiga el reto se deberá mostrar una cara feliz en la matriz de LEDs.

P2 Tecnología 3º ESO
Brújula

Introducción

En esta práctica vamos a construir con la placa micro:bit una brújula que nos indique la orientación. Para ello nuestro objetivo será leer los valores que nos devuelva el sensor brújula disponible en la placa, y mostrar en la matriz de LEDs la orientación hacia la que apunta la placa.

El sensor nos devolverá un ángulo (entre 0° y 360°), y en nuestro código tendremos que traducir esos ángulos a orientaciones (norte, este, sur u oeste).



El programa que tenemos que realizar tendrá que mostrar mensajes diferentes dependiendo del valor de ángulo que esté leyendo:

- Si el ángulo es menor que 45° (y mayor que 0°) estaremos en el norte.
- Si el ángulo es menor que 135° (y mayor que 45°) estaremos en el este.
- Si el ángulo es menor que 225° (y mayor que 135°) estaremos en el sur.
- Si el ángulo es menor que 315° (y mayor que 225°) estaremos en el oeste.
- Si no se da ninguno de los casos anteriores es que estamos entre 315° y 0°, por lo que estaremos también en el norte.

Como ves tendremos que programar las diferentes opciones que se pueden dar, ya que el resultado que mostraremos en la pantalla será distinto dependiendo del ángulo que nos devuelva el sensor. Por ello, necesitaremos hacer uso de una estructura condicional (si… entonces…, si no si …, entonces…) que nos ayudará a programar esta toma de decisiones de la placa.

Desarrollo

1. Queremos que la brújula funcione de forma continua, por lo que incluimos el bloque “por siempre” englobando todas nuestras instrucciones. A continuación, tendremos que leer el valor del sensor brújula y lo guardaremos en una variable para utilizarlo en el futuro. El siguiente paso será utilizar la estructura condicional “si… entonces…, si no si …, entonces…” para tratar todos los distintos casos que hemos definido en la introducción.
Busca los bloques que necesites en la caja de herramientas y configúralos para que la placa nos muestre las letras “N” cuando apunta hacia el norte, “S” cuando apunta hacia el sur, “O” cuando apunta al oeste, y “E” cuando apunta al este.

La primera vez que ejecutes tu código en la placa, ésta te pedirá que la calibres a través del siguiente mensaje en la matriz de LEDs “Draw a circle” (dibuja un círculo). La placa necesita que hagas esto para que su sistema pueda encontrar el norte geográfico de forma precisa. Si no lo haces la placa puede dar valores erróneos. Para calibrarla tendrás que dibujar un círculo moviendo la placa de forma que el punto que aparece en el centro de la matriz dibuje un círculo alrededor de la matriz.

2. Como ya sabrás, mirando las estrellas podremos orientarnos. Si somos capaces de encontrar y reconocer la estrella polar, encontraremos el norte. Modifica tu código para que cuando la placa apunte al norte se muestre la estrella polar en la matriz de LEDs.

P3 Tecnología 3º ESO
Detectando luz

Introducción

En esta práctica vamos a hacer uso de la capacidad que tiene micro:bit de detectar luz, y así poder conocer qué luminosidad hay en nuestra clase. Esto puede ser muy útil por ejemplo para saber si la luz de nuestro entorno es la apropiada para trabajar, o si por el contrario necesitamos más o menos para que nuestros ojos no tengan que forzar la vista.

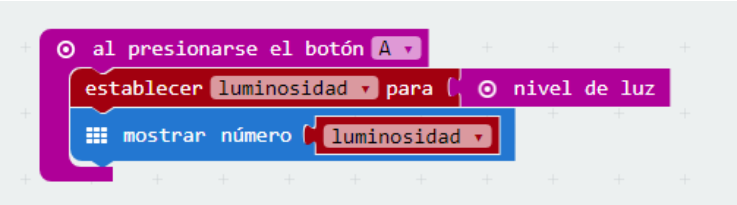
Para detectar la cantidad de luz utilizaremos la matriz de LEDs de micro:bit que puede funcionar de dos formas: como salida, es decir podemos programarla para que iluminen los LEDs que queramos, o como entrada funcionando como un sensor de luz básico, que nos permite detectar la luz ambiental.

Nuestros objetivos en esta práctica van a ser: aprender a leer datos recogidos por el sensor de luz en micro:bit, repasar el uso de variables para guardar información en memoria para utilizarla cuando necesitemos, y aprender a utilizar la función "plot" que nos permite dibujar gráficos con la placa. Al final de la práctica también podremos practicar las estructuras de decisión.

Desarrollo

1. Nuestro primer reto va a ser el de averiguar qué valores de luminosidad detecta la placa en el espacio en el que nos encontramos. Para ello, crearemos un programa que al presionar un botón guarde en una variable el valor que lea el sensor de luminosidad. Para saber qué valor se guarda en la variable mostraremos su valor utilizando la matriz de LEDs. Recuerda que el sensor de luminosidad tiene un rango que va desde 0 (oscuridad) a 255 (máxima luminosidad detectable). Es decir, cuando no reciba luz dará un valor de 0 y cuando reciba la máxima cantidad de luz que pueda medir mostrará 255.

Busca los bloques correspondientes en la caja de herramientas y crea tu propio código. Como puedes ver en el dibujo, la variable se llama "luminosidad" pero tú puedes llamarla como te parezca, siempre que la llames de la misma forma cada vez que te quieras referir a ella.

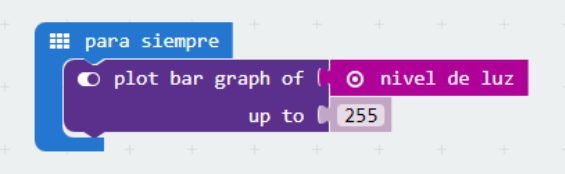


2. Prueba tu código en el simulador y cuando veas que funciona como esperabas, cárgalo en la placa y testéalo.

3. Ahora nuestro reto será mostrar en la matriz de LEDs un gráfico que nos informe del nivel de luminosidad que lea el sensor. Para ello utilizaremos el bloque "plot bar graph of" (dibuja el gráfico de barras) que nos mostrará un gráfico de nivel en la matriz de LEDs. Cuanta mayor luminosidad detecte el sensor, el gráfico mostrará un mayor nivel de llenado. A este bloque le tenemos que pasar dos parámetros (informarle de dos cosas):
    a. Datos a representar: qué información queremos que pinte

b. Límite: en la pantalla, como tenemos una cantidad de LEDs finitos, tenemos que indicarle cuál será el valor máximo que queramos que se muestre como nivel máximo en nuestro gráfico.

Busca los bloques correspondientes y realiza el mismo código que te presentamos a continuación.



4. Antes de subir el código en la placa y ejecutarlo, vamos a probar la funcionalidad que nos ofrece el simulador de micro:bit de pintarnos una gráfica con todos los valores que va recogiendo del sensor. Para ello, una vez que hayas escrito el código, haz clic en el botón “Mostrar datos simulador” que aparece debajo de la placa del simulador. Cuando muevas el símbolo de luminosidad de la placa (que simula distintos niveles de luz) podrás ver cómo cambian los valores en la gráfica.

5. Por último te animamos a que diseñes una luz quitamiedos con micro:bit. Para ello te damos unas pistas: puedes utilizar la matriz de LEDs como bombilla, y es posible que necesites una estructura de decisión para saber bajo qué condiciones de luminosidad del ambiente tiene que iluminarse la luz.

6. ¿Se te ocurre qué tipo de dispositivo podríamos crear si en lugar de detectar luz detectáramos movimiento y lo pintáramos? ¿Sabes qué es un sismógrafo? ¡Intenta programarlo con tu placa y ver los resultados en el simulador!

P4 Tecnología 3º ESO
Alarma antirrobo

Introducción

En esta práctica vamos a utilizar la placa micro:bit, un sensor de presencia y un zumbador para crear un sistema antirrobo. Nuestro sistema funcionará de forma que cuando la placa detecte movimiento alrededor haremos sonar una alarma.

En el diseño de este programa repasaremos el uso de la estructura condicional (si…entonces…), la conexión y lectura de un sensor externo, y la conexión y utilización de un zumbador. Además, aprenderemos a usar el bucle "mientras" (*while*) que nos permitirá repetir una instrucción mientras se cumpla una condición.

Los sensores de presencia que vamos a utilizar son PIR (*Passive Infrared Sensor*). Son sensores de infrarrojos digitales que detectan movimiento a su alrededor. Los sensores digitales ofrecen dos valores, 0 ó 1. Y dependiendo del sensor, estos valores significarán una cosa u otra. En el caso de nuestro sensor PIR, el 1 significa que detecta presencia y 0 que no lo detecta.

Los zumbadores son unos pequeños dispositivos que actúan de altavoz, reproduciendo los sonidos que ejecutamos en la placa. No es un sensor, sino un actuador, al que le tenemos que indicar lo que queremos que haga.

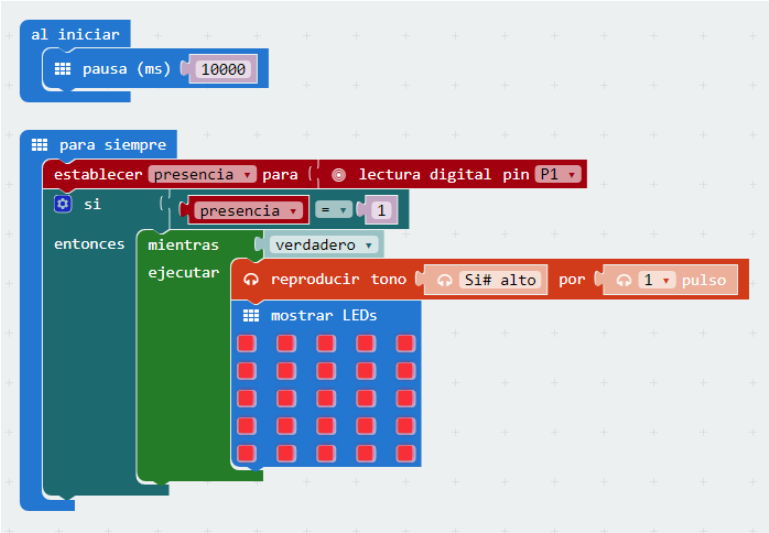
Desarrollo

1. Como queremos que nuestro programa se esté ejecutando en la placa continuamente, lo primero será englobar todo nuestro código dentro de un bloque "para siempre". Nuestro problema nos pide que la placa suene solamente cuando se detecta movimiento, por lo que tendremos que utilizar una estructura condicional "si…entonces…". En este caso la condición será que el PIR detecte movimiento, que ocurrirá cuando la señal digital que leamos del pin al que conectemos el sensor sea un 1. SI se detecta movimiento ENTONCES reproduciremos un tono en el zumbador.

También queremos que cuando comience a sonar la alarma, ésta no deje de sonar. Para ello utilizamos la estructura MIENTRAS… EJECUTAR… (en inglés WHILE). Es una estructura de repetición, bucle, que nos permite que mientras se cumpla una condición se ejecuten lo que le indiquemos de forma repetitiva. En este caso, una vez que comienza el bucle, como la condición "verdadero" no va a cambiar nunca, se va a ejecutar de forma repetitiva el tono en el zumbador.

Un detalle que tendremos que añadir a nuestro programa, es que cuando dejemos el dispositivo en la habitación que queremos monitorizar, también nos detectará cuando salgamos de la habitación, así que tendremos que programar que al iniciarse la placa espere un tiempo (por ejemplo 10 segundos) para que nos dé tiempo a salir antes de que se ponga a funcionar.

Selecciona los bloques que necesites de la caja de herramientas y configúralos para que funcione como deseamos (puedes utilizar el tono que quieras para el zumbador)



2. Antes de ejecutar el programa en la placa tenemos que conectar el PIR y el zumbador.

El zumbador tiene dos cables:

- Cable rojo (alimentación y señal): a través de este cable nuestro zumbador se alimenta y también nos podemos comunicar con él, diciéndole qué tonos tiene que reproducir. Lo conectaremos al pin 0, ya que es el pin recomendado para conectar este dispositivo en micro:bit
- Cable negro (tierra) que conectaremos a la tierra (GND) de la placa. En el caso de la tierra, no importa que varios cables estén conectados al GND.

El PIR tiene tres conexiones:

- GND (tierra) que conectaremos a la tierra de la placa
- VCC (alimentación) que conectaremos al pin de 3V de la placa
- OUT (señal) a través de esta conexión recibiremos la información del sensor, en este caso lo conectaremos al pin 1 de la placa aunque podrías usar otro que no esté en uso. En nuestro programa tendremos que leer del pin digital al que hayamos conectado el sensor para poder recibir su información.

3. Ahora sube el código a la placa y ejecútalo. Verás que una vez pase el tiempo de espera que hemos programado al principio, si el sensor detecta movimiento empezará a sonar. Si quieres que pare, puedes apretar el botón de *reset* para que vuelva a ejecutarse el programa como si la acabaras de conectar.

4. El siguiente reto que te proponemos es el de crear una alarma que nos avise con sonido de que la temperatura de la habitación sobrepasa cierto valor. Para ello puedes utilizar el sensor de temperatura que está incluido en la placa. Para probar el código puedes escoger los valores de temperatura que quieras para hacer saltar la alarma, pero te recomendamos que primero compruebes a qué temperatura está la habitación, y que utilices ese valor como límite. Luego puedes echar vapor a la placa para calentarla y comprobar que tu sistema funciona.

5. Te proponemos también el reto de crear un sistema de alamar para tu mochila en el que en lugar de sonar cuando detecte movimiento, suene cuando la placa se mueva. Pista: ¿recuerdas qué detectaba el acelerómetro?

P5 Tecnología 3º ESO
Control de motores

Introducción

En esta práctica vamos a manejar un actuador, es decir, un dispositivo al que vamos a decir qué tiene que hacer. Aunque puede haber múltiples ejemplos de actuadores (como un LED al que le decimos cuándo tiene que encenderse), en este caso vamos a utilizar un servo motor. Esto es un motor al que se le puede indicar los grados que tiene que girar. Esto puede ser interesante por ejemplo si quisiéramos programar el funcionamiento de una barrera.

Nuestro reto será el de programar la placa de forma que el servo motor gire 10 grados hacia un sentido cada vez que apretemos el botón A, y que gire 10 grados en sentido contrario cada vez que apretemos el botón B. Para que durante el programa podamos recordar en qué ángulo se encuentra en cada momento utilizaremos una variable.

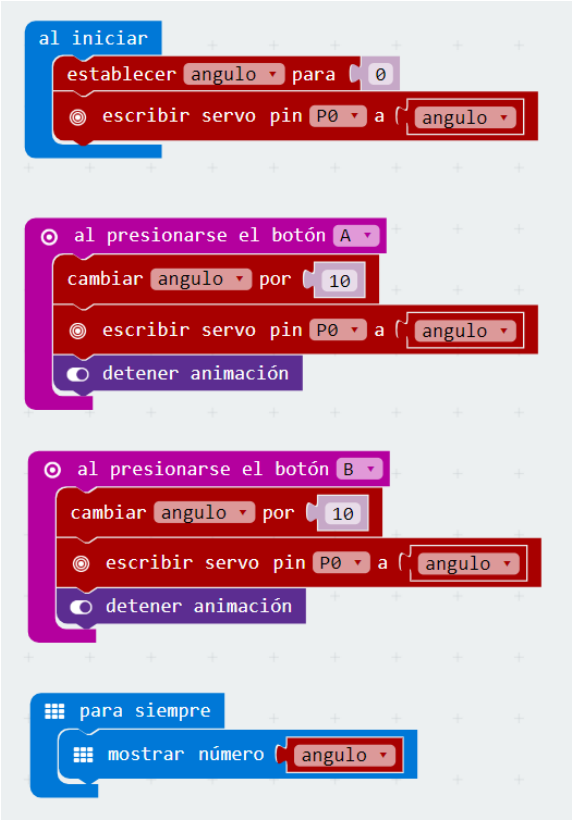
El servo motor tiene 3 conexiones, alimentación (3V), señal (lo podemos poner al pin 0), y tierra (GND). A través del pin señal al que lo conectemos podremos enviarle señales para indicarle cómo tiene que girar. En la caja de herramientas podremos encontrar las instrucciones para manejar los pines en el apartado de "Avanzado" en "Pines".

En esta práctica aprenderemos a enviar señales a un actuador externo, repasaremos el trabajo con variables y funciones, y en la última actividad recordaremos nuestros conocimientos sobre estructuras de decisión.

Desarrollo

1. Para realizar nuestro programa necesitaremos 4 cosas:
   a) Inicializar al conectar la placa nuestra variable a un ángulo concreto y mover el servo a ese ángulo. Para ello utilizaremos el bloque "al iniciar" y añadiremos instrucciones donde crearemos la variable a 90 grados y donde enviaremos ese valor de ángulo al pin donde conectamos el motor (en este caso el pin 0). De esta forma le mandaremos la señal a la placa de que tiene que colocar el servo con 90º de giro.
   b) programar qué tiene que ocurrir cuando se aprieta el botón A: para ello utilizaremos el evento "al presionar el botón A" y dentro de este bloque le indicaremos lo que tiene que ocurrir cuando se produzca ese evento, en este caso aumentar el valor del ángulo en 10 grados. Para ello utilizaremos la función "escribir servo" al que le indicaremos el pin al que queremos mandar la señal y el valor que queremos enviar.
   c) programar qué tiene que ocurrir cuando se aprieta el botón B: para ello utilizaremos el evento "al presionar el botón B" y dentro de este bloque le indicaremos lo que tiene que ocurrir cuando se produzca ese evento. En este caso el valor de nuestro ángulo cambiará en 10 grados menos. Para ello utilizaremos la función "escribir servo" al que le indicaremos el pin al que queremos mandar la señal y el valor que queremos enviar.
   d) mostrar en la matriz de LEDs cuál es el valor de nuestro ángulo en todo momento. Esto es opcional, pero así podemos ver cómo va cambiando el ángulo. Utilizaremos un bucle "para siempre" donde diremos que se muestre el valor de la variable ángulo.

Localiza todos los bloques en la caja de herramientas y crea tu propio código.



2. Conecta el servo motor a la placa, conectando el cable de señal al pin 0, el cable de alimentación al pin de 3V de la placa y el cable de tierra al pin GND.

3. Carga el programa en la placa y prueba si funciona de forma correcta.

4. Al principio dijimos que podíamos utilizar el servo motor para crear una barrera. Realiza un programa en el que cuando apretemos el botón A, si la barrera estaba cerrada (0 grados) gire 90º y se abra, y si estaba abierta (90º) gire 90º en sentido contrario y se cierre.

P0 Control y Robótica 3º ESO
Construyendo una ciudad inteligente

Introducción

A través de este conjunto de prácticas vamos a trabajar juntos para crear la maqueta de una ciudad inteligente que presentaremos al concurso CantabRobots.

Pero, ¿qué es una ciudad inteligente? En resumen, podemos decir que es una ciudad que utiliza la tecnología para mejorar la calidad de vida de los ciudadanos, mejorando los servicios y cuidando al medio ambiente.

Santander es una ciudad inteligente, en ella hay desplegados una gran cantidad de sensores de diferentes tipos que dan información a la ciudad para poder gestionarla mejor: sensores de temperatura y humedad, sensores de ruido, sensores de riego, sensores de llenado de basuras, sensores de aparcamiento, sensores de tráfico, etc.

En la ciudad inteligente que vamos a diseñar y construir nos vamos a centrar inicialmente en tres objetivos: riego sostenible, aparcamiento y alumbrado.

Entre todos diseñaremos, construiremos y programaremos soluciones a estos tres problemas con ayuda de la placa micro:bit y de distintos sensores y actuadores que nos ayudarán en esta tarea.

Cuando tengamos todos nuestros sistemas listos, los integraremos todos en una misma maqueta para mostrar nuestro prototipo de ciudad inteligente.

Además de nuestros tres objetivos, ¿se te ocurre alguna otra propuesta? ¿quieres que añadamos otro sistema más? Vamos a proponer, entre todos, una idea adicional. Cada uno va a pensar una idea. Dentro de una semana la presentaremos brevemente en clase: piensa qué quieres hacer y qué se te ocurre que podríamos utilizar sabiendo cómo funciona microbit. Entre todas aquellas ideas que encontremos factibles de realizar, votaremos una y la incluiremos en nuestra ciudad.

¡Manos a la obra!

P1 Control y Robótica 3º ESO
Alumbrado inteligente

En nuestras ciudades los servicios deben ser cada vez más sostenibles. En general, aunque algunas farolas tienen sensores de presencia para encenderse, se encienden y se apagan a horas concretas, dependiendo también de la estación del año. Este sistema tiene el problema de que algunos días con poca luz (mal tiempo) aunque todavía no sea la hora de encendido, deberían encenderse ya que las calles están oscuras. Nosotros vamos a crear un sistema que haga que las luces se enciendan o se apaguen dependiendo de la cantidad de luz que haya en el momento.

Para medir la cantidad de luz, utilizaremos una fotorresistencia, también llamada LDR (*Light Dependent Resistance*). Este componente es una resistencia variable que cambia su valor dependiendo de la cantidad de luz que reciba. El LDR, aunque sea sencillo, funciona como un sensor analógico, es decir, traduce la cantidad de luz que recibe en valores que van desde 0 (mucha luz) hasta un máximo 1024 (oscuridad).

Para simular la farola, utilizaremos un led blanco y los materiales que puedas encontrar en el aula o que quieras traer de casa. Para controlar bajo qué valores de nuestra resistencia (niveles de luz ambiental) encenderemos la farola, tendremos que aprender a calibrar nuestro sistema.

En esta práctica aprenderemos a leer de un sensor analógico y a calibrar el sistema. Repasaremos qué es una variable, para qué sirve y cómo utilizarlas, las estructuras condicionales y la programación de actuadores digitales (en este caso encender un LED). Además, para conectar los sensores aprenderemos a utilizar una placa de prototipado.

Desarrollo

1. Pensad en cómo diseñar la farola. Qué materiales necesitáis y cómo lo construiríais.

2. Ahora pensad el diseño del programa. Podéis escribir en un papel los diferentes pasos que tiene que realizar el programa, un pseudocódigo, para organizaros antes de poneros a programar. Como pista, tendréis que tener en cuenta algunos aspectos:

- Leer la información de un sensor analógico en el pin al que lo conectemos:

- Tendréis que calibrar la farola, es decir averiguar bajo qué valores de luminosidad la farola deberá encenderse. Para ello podéis jugar tapando y destapando la farola y viendo qué valores del sensor los relacionáis con oscuridad y qué valores con luminosidad.
- Una vez que ya sepáis qué valor utilizareis de límite para encender/apagar la farola, podréis usar las estructuras de programación que conocéis y programar el led de forma correspondiente.
- Para las conexiones podemos utilizar una placa de prototipado.

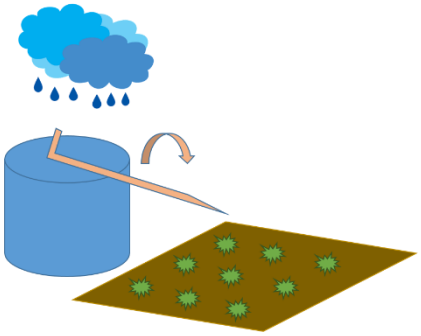
P2 Control y Robótica 3º ESO
Riego sostenible e inteligente

Nuestro reto ahora será el de crear la maqueta de un sistema de riego inteligente y sostenible que nos permita regar una zona de huertos urbanos en la ciudad.

La idea es la de crear un tanque de agua que usaremos para regar el huerto. Para hacerlo más ecológico, utilizaremos el agua de la lluvia para llenarlo y usaremos el agua sólo cuando el suelo necesite ser regado. Para saber cuál es la humedad del suelo, y así poder saber si necesitamos regar o no, utilizaremos un sensor de humedad del suelo.

Por otro lado, para saber el nivel de agua de nuestro tanque utilizaremos un sensor de nivel de agua. Lo usaremos porque nos interesa saber si el tanque está vacío y tenemos que utilizar el agua de otro sitio para regar, o si va a rebasar el tanque y necesitamos extraer el agua para que pueda ser utilizada para otros usos, como limpiar las calles.

Para nuestro sistema de riego os proponemos usar un servo motor que permita mover una especie de manguera que recoja el agua del tanque y la vuelque sobre el huerto. En la imagen tienes una propuesta de diseño.



En resumen, estos van a ser los sensores y actuadores que vamos a utilizar: sensor de humedad del suelo, sensor de nivel de agua y servo motor. Nuestra placa micro:bit recogerá la información de los sensores y de esta forma podremos programar cómo tiene que actuar el sistema.

Desarrollo

1. Pensad en cómo diseñar la farola. Qué materiales necesitáis y cómo lo construiríais.

2. Ahora pensad el diseño del programa. Podéis escribir en un papel los diferentes pasos que tiene que realizar el programa, un pseudocódigo, para organizaros antes de ponerte a programar. Como pista, tendréis que recordar algunos aspectos:

- Leer la información de un sensor analógico en el pin al que lo conectemos:

- Deberéis calibrar tanto el sensor de humedad del suelo como el de nivel de agua del tanque. Para el de humedad del suelo, deberéis averiguar qué valores devuelve el sensor cuando la tierra está seca y necesita ser regada, y para qué valores hay que parar de regar. Para el de nivel de agua del tanque, tendréis que averiguar para qué valores leídos por el sensor el tanque está vacío y para qué valores el tanque está lleno.
- Una vez que ya sepáis qué valores utilizareis de límites podéis usar las estructuras de programación que conocéis y programar el sistema de forma correspondiente.

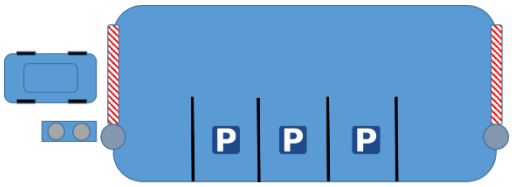
- Para las conexiones podéis utilizar una placa de prototipado.

P3 Control y Robótica 3º ESO
Aparcamiento inteligente

En esta práctica vamos a construir un aparcamiento que constará de una entrada y una salida con barreras y un área de aparcamiento. Tendrá que tener cumplir los siguientes requisitos:

- Detectar cuando llega un coche a la entrada
- Si hay espacios libres, se abrirá la barrera
- Si no hay espacios libres, no se abrirá la barrera
- Cuando un coche se acerca a la barrera de salida, la barrera siempre se abrirá
- Tendremos dos leds (rojo y verde) en la entrada, que indicarán si hay huecos libres o no

En el dibujo podéis ver una primera idea de lo que queremos hacer, pero por supuesto, una vez que vayamos programando los dispositivos, podremos ver cómo integrarlo todo dándole el aspecto que nosotros queramos.



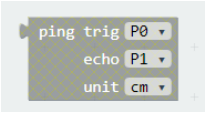
Para detectar en la entrada o en la salida si hay un coche utilizaremos un sensor de ultrasonidos. Estos sensores funcionan como un radar, enviando una señal y calculando el tiempo que tardan en recibir el eco. A partir de lo que tarda en volver la señal calcula la distancia a la que se encuentra el objeto. En la caja de herramientas no hay bloques para leer de este sensor, sin embargo, se pueden agregar paquetes a nuestra caja de herramientas. En este caso, se llama “sonar”, y lo podemos obtener en el apartado “Add package” (Añadir paquete).

Tendremos que llevar la cuenta de las plazas libres que hay en el aparcamiento. Para ello podemos utilizar una variable a la que iremos sumando o restando plazas dependiendo de qué barreras se abran. Cuando estén ocupadas todas las plazas, no podremos abrir la barrera de entrada, sólo la de salida en el caso de que quiera salir un coche.

Por último, utilizaremos dos leds para indicar si hay o no plazas en el aparcamiento. Encenderemos el led verde cuando haya plazas libres, y el rojo cuando no las haya.

Desarrollo

1. Pensad en cómo diseñar el aparcamiento. Qué dispositivos/materiales necesitáis y cómo lo construiríais.
2. Ahora pensad el diseño del programa. Podéis escribir en un papel los diferentes pasos que tiene que realizar el programa, un pseudocódigo, para organizaros antes de poneros a programar. Como pista, tendréis que tener algunos aspectos en cuenta:
   - Leer la información del ultrasonidos, indicando qué pin envía la señal (trig), qué pin recibe el eco (echo) y la unidad en la que queremos que nos devuelva la distancia:



   - Deberéis calibrar los sensores de ultrasonido para saber para qué valores de distancia devueltos por el sensor podéis asegurar que hay un coche delante.
   - Una vez que ya sepáis qué valores utilizareis de límites, podéis usar las estructuras de programación que conocéis y programar el sistema.
   - Para las conexiones podéis utilizar una placa de prototipado.

P0

TIC 4º ESO
Primeros pasos en Python con micro:bit

Introducción

En esta serie de prácticas vamos a aprender a programar la placa micro:bit utilizando el lenguaje de programación Python, y aprendiendo así diferentes conceptos e instrucciones útiles en programación.

En esta primera práctica aprenderemos a utilizar el entorno de programación Python que ofrece la placa micro:bit, y escribiremos nuestro primer programa, lo cargaremos en la placa y lo ejecutaremos.

Desarrollo

Abre el programa **mu editor** que estará instalado en el ordenador. Te encontrarás con el entorno de programación que vamos a utilizar para programar en Python nuestras placas.

1. **¡Vamos a crear nuestro primer programa en Python!**

Diseñaremos un programa en el que se muestre en la matriz de LEDs el mensaje: “Es mi primer programa!!”

```
#Programa que muestra un mensaje en la matriz de LEDs

from microbit import *

while True:
    display.scroll('Es mi primer programa!!')
    display.show(Image.HEART)
    sleep(2000)
```

Vamos a analizar brevemente qué es lo que hacen las diferentes instrucciones de este programa.

La primera línea es un comentario, se indica con el símbolo # al principio de la línea. Cuando la placa lo ejecute va a ignorarlo. Se utilizan los comentarios para que el programador pueda describir qué hace el código, o parte de él. No es obligatorio comentar el código, pero es una buena práctica en programación.

En la siguiente línea importamos el módulo microbit, que nos va a proporcionar instrucciones para programar la placa (mostrar cosas en la matriz de LEDs, leer los sensores de la placa, etc.)

Después, para que el programa se ejecute continuamente utilizamos "while True:" que significa "mientras Verdadero". Es un bucle que hace que todo lo que esté dentro de él se ejecute de forma repetitiva. Más adelante veremos más profundamente qué son los bucles.

Para iluminar los LEDs de la matriz se utiliza la palabra reservada "display". Esto es lo que llamamos en programación un objeto, ya que esa palabra tiene muchas acciones (métodos) disponibles a las que podemos acceder si utilizamos un punto. Por ejemplo:

display.scroll → muestra por pantalla un mensaje en forma de scroll

display.show → muestra por pantalla un mensaje o una imagen

Si en el editor escribes display y a continuación pones el punto, podrás ver todos los métodos que el objeto display tiene, es decir, todo lo que puedes hacer con el display.

Lo mismo ocurre con la palabra Image, es un objeto que tiene una gran cantidad de métodos que muestran diferentes imágenes.

Finalmente, antes de volver a ejecutar este trozo de código hacemos una pausa de 2 segundos con la instrucción "sleep" (dormir). La instrucción sleep necesita el tiempo en milisegundos, por eso ponemos 2000.

Fíjate en el sangrado del programa. Python se usa el sangrado para saber qué código va dentro de qué instrucción. Por ejemplo, todo el código que está dentro del "while True" está con el mismo sangrado.

2. **Ahora vamos a cargar y ejecutar el programa en la placa.**

   a) Conecta la placa con el cable microUSB-USB al ordenador
   b) Haz clic en el botón Flash para que el programa se transfiera en la placa.

3. **Listo!**

Mientras se carga el programa en la placa parpadea un LED en la parte trasera. Una vez que deja de parpadear el programa ya estará cargado y lo podrás probar. Verás que de forma continua se muestra el mensaje y el corazón.

4. **Probemos otros programas….**

Ahora puedes probar a mostrar otros mensajes. También puedes mostrar otras imágenes utilizando los distintos métodos del objeto "Image"

P1 TIC 4º ESO
Variables en Python

Introducción

Una variable es un contenedor donde almacenamos un valor, y una vez que creamos esa variable ésta se almacena en la memoria del ordenador (en este caso nuestra micro:bit). De esta forma podremos utilizarla y actualizarla todas las veces que queramos en nuestro programa.

Por ejemplo, imagina que tienes un juego en el que tienes 3 vidas y tienes que ir ganando puntos.

vidas = 3
puntos = 0

Cuando cometas un error, perderás una vida:
vidas = vidas – 1 → vidas ahora guardará el valor 2, porque vidas tenía guardado el valor 3

Si consigues monedas, ganarás un punto:
puntos = puntos + 1 → puntos ahora guardará el valor 1, porque puntos guardaba el valor 0

Si consigues una vida por haber conseguido un trofeo en el juego:
vidas = vidas + 1 → vidas guardará el valor 3, porque el último valor guardado de vidas era 2

Pero las variables no guardan sólo número, sino que pueden guardar diferentes tipos de valores:
- Números
- Cadenas de texto, que son palabras o frases (siempre entre comillas simples ‘ o dobles “)
- Listas (conjuntos de variables ordenados)
- Booleanos (verdadero o falso)

Ahora vamos a realizar algún programa en el que juguemos con las variables y podamos ver cómo funcionan.

Desarrollo

1. **Cuentapasos.** Vamos a crear un programa que cuente nuestros pasos y ejecutarlo en la placa micro:bit. Para esto utilizaremos el sensor acelerómetro que tiene la placa. Cada vez que el sensor nos informe de que la placa ha sido agitada significará que hemos dado un paso.

En nuestro programa tendremos que crear una variable (pasos) que iremos actualizando cada vez que la placa detecte que se ha realizado un paso.

```
from microbit import *

pasos = 0 #variable que guarda los pasos, comienza en 0
```

```
        while True:
            if accelerometer.was_gesture('shake'): #si se detecta un paso
                pasos = pasos + 1  #sumamos 1 paso
                display.show(Image.HAPPY)
                sleep(500)
                display.clear()
            if button_a.is_pressed(): #si se aprieta el botón A mostramos pasos
    display.scroll(str(pasos))
```

2. **Generador de mensajes positivos.** Es importante ser positivo en la vida, por ello, vamos a crear un dispositivo que cada vez que toquemos un botón de la placa micro:bit nos muestre de forma aleatoria un mensaje diferente que nos anime y nos de fuerzas para empezar el día.

En este ejercicio tenemos varias variables, pero fíjate en la variable "opciones" que es una lista de valores, en este caso una lista de cadenas de texto.

```
from microbit import *
import random #es una función que nos devuelve números aleatorios

opciones = [
  "Menos dramas y mas ganas",
  "Eres genial",
  "Buenos dias con alegria",
  "Si quieres puedes",
  "Hoy sera un gran dia"
]

while True:
  display.show(Image.HEART)
  if button_a.is_pressed(): #si se toca el boton A
    aleatorio = random.randint(0, 4) #numero aleatorio entre 0 y 4
    mensaje = opciones[aleatorio] #leemos la posición dada por el numero aleatorio de la lista
    display.scroll(mensaje)
```

3. **Sumar dos números.** Vamos a crear con micro:bit una calculadora que sume dos valores. Los valores a sumar se los diremos a través de apretones en los botones. Primero la placa nos pedirá que introduzcamos el sumando A, y tendremos 5 segundos para apretar un número de veces igual al número que queramos introducir. Después nos pedirá que hagamos lo mismo con el sumando B. Finalmente los sumará y mostrará el resultado por pantalla.

```
from microbit import *

sumandoA = 0
sumandoB = 0

display.show('A')
sleep(5000)
sumandoA = button_a.get_presses()
display.show('B')
sleep(5000)
```

```
sumandoB = button_b.get_presses()
suma = sumandoA + sumandoB
display.scroll('suma = ' + str(suma))
```

P2 TIC 4º ESO
Estructuras condicionales en Python

Introducción

En esta práctica vamos a trabajar las estructuras condicionales en Python. Estas estructuras permiten que se ejecute un código u otro dependiendo de si se cumple o no una condición.

Por ejemplo, un monumento que sólo se iluminará si es de noche:

1. **SI** es de noche
   ENTONCES encender las luces

También podremos especificar qué ocurre cuando no se cumple la condición

1. **SI** es de noche
   ENTONCES encender las luces
2. **SI NO** apagar las luces

En el caso de que tengamos más de una opción. Por ejemplo, las notas de un alumno.

1. **SI** examen <5
   ENTONCES nota = suspenso
2. **SI NO, SI** examen < 7
   ENTONCES nota = aprobado
3. **SI NO, SI** examen <9
   ENTONCES nota = notable
4. **SI NO** nota = sobresaliente

Ahora vamos a realizar algunos programas para ejecutarlos en micro:bit que utilicen estructuras condicionales. Además, repasaremos el uso de variables.

Desarrollo

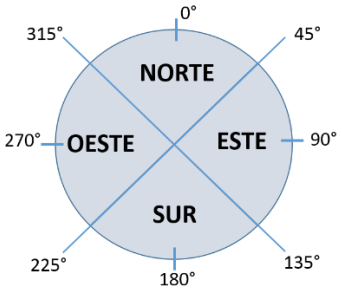
1. **Cambio de humor.** Tenemos que realizar un programa en el que si no se toca ningún botón se muestre una cara triste, si apretamos el botón A se muestra una cara feliz y cuando tocamos el botón B se dejará de ejecutar el programa y se limpiará la matriz de LEDs.

```
from microbit import *

while True:
    if button_a.is_pressed():
        display.show(Image.HAPPY)
    else if button_b.is_pressed():
        break
    else:
        display.show(Image.SAD)
display.clear()
```

2. **Brújula.** Tenemos que realizar un programa que lea del sensor brújula de la placa y dependiendo del ángulo que nos devuelva, mostraremos en la matriz de LEDs la orientación.

Fíjate en la figura para ver qué ángulos corresponden con el Norte, el Sur, el Este y el Oeste para poder programar las condiciones.




```
from microbit import *

compass.calibrate()

while True:
    angulo = compass.heading()
    if angulo < 45:
        display.show('N')
    elif angulo < 135:
        display.show('E')
    elif angulo < 225:
        display.show('S')
    elif angulo < 315:
        display.show('O')
    else:
        display.show('N')
```

3. **Bola 8 de la adivinación.** La Magic 8-Ball (en español Bola 8 Mágica) es un juguete usado para echar la fortuna o buscar consejo diseñado por la compañía juguetera Mattel en 1946. Ahora podremos fabricar la nuestra propia. Nuestro programa tendrá una lista de mensajes posibles a responder ante una pregunta que haga el jugador. Cuando el jugador agite la placa, escogeremos un número aleatorio entre 0 y el número total de mensajes, y escogeremos el mensaje que esté en esa posición de la lista para que se muestre en la pantalla.

```
from microbit import *
import random

opciones = [
    "Es cierto",
    "Es asi, sin dudas",
    "Sin duda",
    "Si",
    "Puedes confiar en ello",
    "Como yo lo veo, si",
    "Es muy probable",
    "Si",
    "Las señales apuntan que si",
    "Respuesta confusa, prueba otra vez",
    "Pregunta mas tarde",
    "Mejor no te lo digo",
    "No los se",
```

```
    "Concentrate y pregunta otra vez",
    "No cuentes con ello",
    "Mi respuesta es no",
    "Mis fuentes dicen que no",
    "Tengo dudas"
]

while True:
    display.show('8')
    if accelerometer.was_gesture('shake'):
        display.clear()
        sleep(1000)
        respuesta = random.choice(opciones)
        display.scroll(respuesta)
    sleep(10)
```

P3 TIC 4º ESO
Estructuras de repetición en Python

Introducción

En esta práctica vamos a trabajar las estructuras de repetición en Python, también llamadas bucles. Estas estructuras permiten que se repita parte del código un número concreto de veces o de manera infinita. Nos permite ser más eficientes a la hora de escribir código.

Por ejemplo, imagínate que quieres programar el mandar el mismo email a 2000 personas. Tienes dos opciones:

1. listaPersonas = [email1, email2, email3,….email2000]
2. enviar(mensaje, email1)
3. enviar(mensaje, email2)
4. enviar(mensaje, email3)

…..

2001. enviar(mensaje, email2000)

Tendrías que escribir 2001 líneas de código. Otra opción sería utilizar un bucle FOR

```
1. listaPersonas = [email1, email2, email3,….email2000]
2. FOR (i=0, i<2000, i++)
       enviar(mensaje, listaPersonas[i])
```

Sólo han sido 3 líneas de código. Lo que hacemos es repetir la instrucción enviar(mensaje, listaPersonas[i]) tantas veces como indicamos en el paréntesis de la estructura FOR. En este caso desde que i valga 0 (i=0), hasta que valga 1999 (i<2000) y que en cada repetición a i le sumemos 1 (i++). En la primera vuelta i valdrá 1, en la segunda 2, en la tercera 3, etc. hasta que i llegue a valer 1999 y termine el bucle.

También podemos querer que se repita un código mientras se cumpla una condición. Para ello utilizaremos la estructura WHILE, que significa mientras: mientras se cumpla la condición repetiremos el código. Por ejemplo, cuando salte una alarma queremos que ésta suena hasta que alguien la apague.

```
1. WHILE alarmaEncendida = verdadero
2.     sirena = encendida
3.     SI botonApagado = pulsado
4.         ENTONCES alarmaEncendida = falso
5.     SI NO alarmaEncendida = verdadero
```

Ahora vamos a realizar algunos programas para ejecutarlos en micro:bit que utilicen bucles. También recordaremos el uso de variables y de condicionales.

Desarrollo

1. **Tiempo de reacción.** Vamos a construir un juego en el que los jugadores van a medir sus reflejos. El juego va a constar de 5 rondas, es decir, se va a repetir 5 veces. Cuando se ilumine la pantalla, los

dos jugadores deberán apretar su botón (A o B) más rápido que su rival. Cada ronda ganada supone 1 punto para el ganador. Gana el jugador que tenga más puntos.

```
from microbit import *

puntosA = 0  #jugador A empieza con 0 puntos
puntosB = 0  #jugador B empieza con 0 puntos

display.scroll('3 2 1') #mensaje inicial

for i in range(1, 9):   #9 rondas
   rondaActiva = True
   display.show(Image.SQUARE_SMALL)
   while rondaActiva: #mientras espero a que aprieten botón
      if button_a.is_pressed(): #si toca botón A
         display.show('A')
         puntosA = puntosA + 1 #sumo 1 pto a A
         sleep(2000) #espero 2 segundos para la siguiente ronda
         rondaActiva = False #para salir del bucle a la siguiente ronda
      elif button_b.is_pressed():
         display.show('B')
         puntosB = puntosB + 1
         sleep(2000)
         rondaActiva = False
      else:
         display.show(Image.SQUARE_SMALL) #imagen si nadie toca botón

display.scroll("GAME OVER")

if (puntosA > puntosB): #calculo quién ha ganado
   ganador = 'A'
else:
   ganador = 'B'
display.scroll(ganador)
```

2. **Creando sonidos.** En los bucles FOR, el intervalo dentro de la condición no tiene por qué ser siempre 1. Podemos especificar el intervalo que queramos. Por ejemplo, si queremos hacer sonar una sirena, que suene desde una frecuencia de 880Hz hasta 1760Hz saltando de 16Hz en 16Hz, podemos utilizar un salto de 16 en nuestro ciclo. Para este programa conecta un zumbador para poder escuchar los sonidos.

```
from microbit import *
import music  #importamos la librería para música

while True: #de forma infinita
   for freq in range(880, 1760, 16):  #para valores desde 880 hasta 1760 en saltos de 16
      music.pitch(freq, 6) #reproducir tono durante 6 milisegundos
   for freq in range(1760, 880, -16): #para valores desde 1760 hasta 880 en saltos de -16
      music.pitch(freq, 6)
```

P4 TIC 4º ESO
Mascota robot

Introducción

En esta práctica vamos a desarrollar una mascota robot para repasar los conceptos que hemos visto anteriormente.

Realizaremos uno de los dos tipos de robots propuestos. Un robot deberá emitir un sonido y cambiar la iluminación de su matriz LED cuando detecte un obstáculo a menos de X cm. El otro robot también emitirá sonidos y cambiará la iluminación de la matriz, pero cuando escuche palmas.

Para el cálculo de distancias utilizaremos un sensor de ultrasonidos y para la detección de las palmadas utilizaremos un sensor de sonido.

El sensor de ultrasonidos funciona de forma que envía una señal y detecta el tiempo que tarda en llegarle su eco. En función de ese tiempo, el sensor calcula a cuántos centímetros se encuentra el obstáculo.

El sensor de sonido cuenta con un potenciómetro (una resistencia a la que podemos cambiar de valor girando el tornillo) que tendremos que ajustar para calibrar su sensibilidad.

Trabajaremos en grupo para abordar las diferentes partes del trabajo: análisis de las especificaciones, diseño del programa, testeo en la placa, diseño de la apariencia del robot y su construcción. Pero a continuación os damos unas pistas para aprender a trabajar con los sensores de ultrasonidos y de sonido.

Para trabajar con estos sensores necesitaremos unos módulos especiales. Un módulo de un sensor es un fichero que contiene una serie de funciones que nos permite recoger la información del sensor, procesarla y poder obtener valores entendibles por nosotros. Por ejemplo, en el sensor de ultrasonidos, el sensor envía una señal y recoge el tiempo que tarda en recibir el eco. El módulo nos ofrece funciones que traducen ese tiempo en centímetros.

Desarrollo

1. **Detectando distancias con el sensor HCSR-04**

a) Descarguemos el módulo Python para el HCSR-04. Encontrarás el código en https://github.com/fizban99/microbit_hcsr04/blob/master/hcsr04.py. Copia el código en un archivo que se llame hcsr04.py (la extensión .py indica que es Python) y guárdalo en la carpeta mu-editor dentro de la carpeta C:/Usuarios/UsuarioX/mu_code.

b) Abre el entorno de programación mu Editor

c) Estas son las instrucciones básicas que te permite leer del sensor y obtener distancias. Como ves hemos importado el módulo (hcsr04) y lo que necesitamos de él (HCSR04)

```
from hcsr04 import HCSR04
from microbit import sleep
sonar=HCSR04()
while True:
    distancia_mm=sonar.distance_mm() #leer distancia en milímetros
    distancia_cm = distancia_mm/10 #calculamos la distancia en centímetros
```

```
display.scroll(str(distancia_cm)) #str para transformar el número en texto
sleep(1000)
```

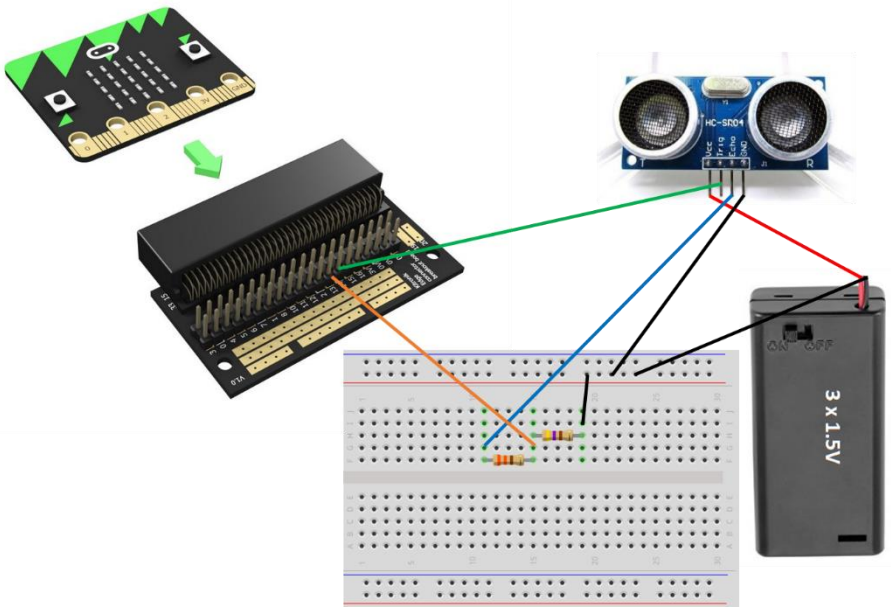
d) Carga el programa en la placa (“Flash”)

e) Ahora verás que la placa te da un error y te dice que no encuentra el módulo hcsr04. Eso es porque todavía no lo hemos copiado en la placa para que ésta lo pueda usar. Haz clic en el botón “Files” del editor mu. Verás que se abren dos nuevos espacios. A la izquierda la micro:bit y a la derecha la carpeta donde guardamos el módulo. Arrástralo hasta la micro:bit.

f) Ahora el programa se ejecutará correctamente.

g) Conexión del sensor

El sensor tiene 4 conexiones: VCC (alimentación), GND (tierra), TRIG (señal), ECHO (eco). A la hora de conectar este sensor a la placa micro:bit tendremos que tener en cuenta que la placa micro:bit sólo puede alimentar con 3V y el sensor necesita 5V para trabajar correctamente.



2. Detectando sonido con el sensor KY038

a) Descarguemos la librería Python para el KY038. Encontrarás el código en https://github.com/fizban99/microbit_ky038/blob/master/ky038.py Copia el código en un archivo con el nombre ky038.py (la extensión .py indica que es Python) y guárdalo en la carpeta mu-editor dentro de la carpeta C:/Usuarios/UsuarioX/mu_code.

b) Abre el entorno de programación mu Editor

c) Estas son las instrucciones básicas que te permite leer del sensor y detectar palmadas (o ruidos continuos). Como ves hemos importado el módulo (ky038) y lo que necesitamos de él (KY038). La instrucción “str” se utiliza para transforman en texto un número, ya que la instrucción display.scroll sólo muestra texto. El método calibrate() nos indicará cómo tenemos que girar el tornillo del potenciómetro hasta que esté calibrado nuestro sensor.

```
from microbit import *
from ky038 import KY038

KY038.calibrate() #calibrar sensor
```

```
while True:
    aplausos = KY038.count_claps()) #calcula número de palmadas seguidas
    display.scroll(str(aplausos))
```

d) Carga el programa en la placa (“Flash”)

e) Ahora verás que la placa te da un error y te dice que no encuentra el módulo ky038. Eso es porque todavía no lo hemos copiado en la placa para que ésta lo pueda usar. Haz clic en el botón “Files” del editor mu. Verás que se abren dos nuevos espacios. A la izquierda la micro:bit y a la derecha la carpeta donde guardamos el módulo. Arrástralo hasta la micro:bit.

f) Ahora el programa se ejecutará correctamente.

g) <u>Conexión del sensor</u>

El sensor tiene 3 conexiones: VCC (alimentación), GND (tierra), y OUT (salida). A la hora de conectar este sensor a la placa micro:bit tendremos que tener en cuenta que la placa micro:bit sólo puede alimentar con 3V y el sensor necesita 5V para trabajar correctamente.