



**UNIVERSIDAD
DE GRANADA**

Escuela Internacional de Posgrado

Máster Universitario en Profesorado

Especialidad: **Tecnología, Informática y Procesos Industriales.**

Campus: **Granada.**

TRABAJO FIN DE MÁSTER

Uso de una Plataforma Online para el Aprendizaje de la Programación.

Programa un Robot en la Luna.

Presentado por:

Elena Álvarez Castro

Tutor:

Miguel Ángel Rubio Escudero

Curso académico 2020/2021

Declaración de Originalidad del TFM

Dña **Elena Álvarez Castro**, con DNI **75576450 Z**, declaro que el presente Trabajo de Fin de Máster es original, no habiéndose utilizado fuentes sin ser citadas debidamente. De no cumplir con este compromiso, soy consciente de que, de acuerdo con la Normativa de Evaluación y de Calificación de los estudiantes de la Universidad de Granada de 20 de mayo de 2013, *esto conllevará automáticamente la calificación numérica de cero [...] independientemente del resto de las calificaciones que el estudiante hubiera obtenido. Esta consecuencia debe entenderse sin perjuicio de las responsabilidades disciplinarias en las que pudieran incurrir los estudiantes que plagien.*

Y para que así conste firmo el presente documento.

En Granada a 4 de junio de 2021.

Firma del alumno



Elena Álvarez Castro

Título: Uso de una Plataforma Online para el aprendizaje de la programación. Programa un Robot en la Luna

Autora: Elena Álvarez Castro

Tutor: Miguel Ángel Rubio Escudero

Resumen

La programación basada en bloques se ha convertido en una herramienta esencial en la enseñanza de programación. Programar un robot, ayuda a comprender la estructura de un programa y mostrar su aplicación en la vida real. Utilizar simuladores aumenta la accesibilidad y disminuye la brecha digital. En este trabajo, se desarrollan un conjunto de actividades, que utilizan la plataforma Open Roberta Lab, que permite simular el funcionamiento del robot y una programación intuitiva por bloques. Las actividades están organizadas en un guion de prácticas para el alumnado y una guía básica para el profesorado. Después de dos ejemplos de aplicación de estas actividades, los guiones cuentan con modificaciones resultantes de la retroalimentación obtenida.

Palabras clave

Programación en contexto, laboratorios virtuales, programación basada en bloques, plataforma online, programación, simulación, robótica.

Title: Use of an online platform for learning coding. Programming a robot on the Moon

Author: Elena Álvarez Castro

Advisor: Miguel Ángel Rubio Escudero

Abstract

Block-based programming has become an essential tool in teaching programming. Programming a robot helps to understand coding structure and show its application in real life. Using simulators increases accessibility and reduces technical hurdles. This work presents a set of activities using the Open Roberta Lab platform. This tool allows robots simulation and an intuitive block-based programming. The activities are organized in students' worksheets and a teacher's guide. After two examples of the application of these activities, guides have modifications from the feedback obtained.

Keywords

Context programming, virtual laboratories, block-based programming, online platform, coding, simulation, robotics.

Índice general

1	Introducción	2
2	Fundamento Teórico.....	3
3	Objetivos	7
4	Metodología	8
4.1	Marco Legal	8
4.2	Objetivos	8
4.3	Contenidos, criterios de evaluación y estándares de aprendizaje.....	9
4.4	Estrategias Metodológicas.....	12
4.5	Plataforma Open Roberta Lab.....	12
4.5.1	Entorno de programación.....	14
4.5.2	Entorno de simulación	15
5	Material Desarrollado.....	17
5.1	Objetivos	17
5.2	Competencias, Contenidos y Planificación	18
5.3	Herramientas de Evaluación.....	21
5.4	Guiones.....	23
5.4.1	Guion Alumnado.....	23
5.4.2	Guion para el profesorado.....	35
6	Ejemplos de Aplicación del material desarrollado	37
6.1	Feria de las ingenierías	37
6.1.1	Antecedentes	37
6.1.2	Feria de las Ingenieras 2021.....	38
6.1.3	Realización del Taller	39
6.2	Periodo de Prácticas	41
6.2.1	Realización de la Actividad	41
6.3	Resultados	43
6.4	Lecciones Aprendidas	45
7	Conclusiones	47

Bibliografía

Anexo I

Anexo II

1 Introducción

La programación en contexto, el uso de lenguajes de programación basados en bloques y los laboratorios virtuales, han transformado la forma de enseñar y aprender programación. En este trabajo, se desarrollan una serie de actividades, que tienen el objetivo de introducir conceptos de programación, contextualizando la actividad a través de un entorno virtual de simulación.

La misión es programar un robot, que ha sido enviado a la Luna para tomar muestras de la superficie. Se controla desde la Tierra de forma remota, tal y como se realiza en la Misión Heracles de la Agencia Espacial Europea. Esta contextualización ayuda a captar el interés del alumnado en la actividad y mostrar una de las aplicaciones de la programación, la robótica espacial. Las profesiones asociadas a la industria del espacio, en general se consideran muy complejas y la percepción por parte de la sociedad es que sólo son accesibles para personas con una alta cualificación académica.

La plataforma para realizar esta actividad es Open Roberta Lab [1]. Es una herramienta online, que aparte de ofrecer una programación basada en bloques, permite la simulación de varios sistemas. En concreto, el utilizado en esta actividad es el LEGO Mindstorm EV3. La interfaz de simulación, permite ver los movimientos que sigue el robot con la programación realizada. Por esta razón, no es necesario contar con este sistema físicamente. Este aspecto es especialmente importante en la situación de pandemia actual; además de que hace accesible la actividad para cualquier persona que disponga de un ordenador con conexión a internet, ya que, tampoco necesita instalación de software.

En este proyecto, se presenta, en primer lugar, el fundamento teórico, haciendo referencia a los laboratorios virtuales, la programación por bloques y la contextualización del aprendizaje. En el siguiente punto, la legislación vigente, para contextualizar las actividades dentro de los contenidos curriculares establecidos. Después, se muestran las características y posibilidades de Open Roberta Lab. A continuación, el material desarrollado. Por último, se exponen los resultados y las lecciones aprendidas de cada una de las aplicaciones que se han llevado a cabo.

2 Fundamento Teórico

En la sociedad actual, la programación juega un papel muy importante, pues, es necesario saber programar, aunque ésta no sea tu profesión. Instrumentos que utilizamos en nuestra vida cotidiana, como son los robots de cocina, aspiradoras, televisores y asistentes virtuales como Alexa o Siri, cada vez incluyen más opciones de configuración y programación. También es el caso de la domotización de viviendas, el sistema de iluminación, persianas o alarmas. En ámbitos profesionales, a primera vista alejados de la programación, también está muy presente; cajeros automáticos, sistema digital para pedir citas, herramientas de marketing y una gran infinidad de ámbitos, dónde programar es una herramienta más de la profesión.

Está claro que todos los sistemas que utilizamos están cada vez más digitalizados y, por tanto, necesitan una programación. A pesar de que las nuevas generaciones han crecido en un mundo digital, en general muestran poco interés por la programación y la informática, es decir, por las tecnologías de la información y la comunicación, así como por las disciplinas STEM¹.

Es difícil introducir conceptos de programación a esta población joven y mantener la motivación y el interés, pues la mayoría de sistemas tradicionales de programación, ofrecen entornos poco intuitivos y sin interfaz gráfica. Para las nuevas generaciones, las pantallas, y los entornos gráficos son parte de su vida cotidiana. La mayoría de dispositivos tienen grandes pantallas táctiles para utilizarlos a través de una interfaz gráfica interactiva. Además, incluyen sistemas de interacción con el mundo exterior, como son las cámaras o el reconocimiento por voz. Lo más importante hasta ahora han sido las líneas de código, mantener la sintaxis y la estructura de programación. Para lograr que estudiantes de niveles de secundaria se interesen por la programación, es necesario relacionarla con sus experiencias en el mundo tecnológico y digital [2].

Los entornos de programación por bloques transforman los comandos, funciones, variables y demás objetos de programación en bloques. Proporcionan pistas visuales sobre cómo y dónde se pueden usar estos bloques, como si fueran piezas de un puzzle [3]. Actualmente, existe una larga lista de herramientas que permiten una programación por bloques. Algunos ejemplos son Scratch, Snap, Blockly, Mbloc, Arduino Blocks y Tynker. Todas estas plataformas utilizan un lenguaje visual para acercar la programación a la población más joven. Existe un gran número de entornos que, aparte de ofrecer una programación por bloques, permiten la transformación en tiempo real a código, generando plataformas que permiten la transición entre lenguaje de programación por bloques y lenguajes de programación como C++ y Python.

Las herramientas que permiten programación por bloques son una forma de introducir conceptos de programación, ya que, empezar a programar con estos entornos visuales, acelera el aprendizaje y la comprensión de los conceptos abstractos de computación [3]. Sin embargo, todos los lenguajes

¹ Science, Technology, Engineering and Mathematics (Ciencia, Tecnología, Ingeniería y Matemáticas).

visuales tienen limitaciones, una programación demasiado compleja no es posible, pues los bloques están definidos y no se pueden modificar ciertas características. A pesar de esto, para niveles educativos de educación secundaria, la complejidad que ofrecen es más que suficiente.

En un estudio realizado por las Universidades de Chicago y de Northwestern en 2017 [3] se demostró que la interfaz visual que ofrecen los entornos de programación por bloques, permiten mejorar el ritmo de aprendizaje de la programación. Este estudio se realizó durante cinco semanas, en las que dos grupos de estudiantes de secundaria sin experiencia previa realizaron un curso de programación. Un grupo de estudiantes lo realizó con interfaces de programación por bloques y el otro grupo en código. Los estudiantes que utilizaron programación por bloques, mostraron una mejor curva de aprendizaje, motivación e iniciativa a la hora de programar. Es decir, los estudiantes aprenden todos los conceptos esenciales en programación: secuencia de acciones, sentencias condicionales y bucles, de una forma más rápida. La razón es que estas herramientas permiten construir un programa relativamente complejo de forma fácil y rápida, lo que hace que se sientan recompensados rápidamente por sus esfuerzos [4], aumentando su motivación por seguir aprendiendo. Además de facilitar la interacción con la programación, los bloques esconden los problemas de sintaxis, haciendo que los estudiantes puedan centrarse en el pensamiento lógico, la programación y la resolución de problemas [4].

El resultado más impactante es que los estudiantes que trabajan con lenguaje de programación por bloques muestran más interés en continuar sus estudios en el campo de la informática y la programación, porque tienen una percepción menos compleja de la programación [3]. Así pues, la modalidad y la forma de mostrar la programación, tiene un gran impacto en el aprendizaje y además juega un papel clave en que estos estudiantes consideren la programación como una profesión en el futuro.

Por tanto, la plataforma y lenguaje utilizado para introducir la programación es muy importante para aumentar la motivación y el interés por aprender. Sin embargo, aparte de la plataforma utilizada, el contexto también juega un papel crucial a la hora de introducir conceptos complejos [4].

Estudiantes de todas las edades, muestran una gran motivación cuando pueden comprender la utilidad de lo que están aprendiendo y que realmente pueden utilizar esos conocimientos en el futuro o en su vida cotidiana [2]. Esta afirmación se aplica también a la programación. El objetivo de contextualizar el aprendizaje es mostrar una utilidad práctica de los conceptos expuestos. Por otro lado, también permite ‘distraer’ a los estudiantes de la complejidad del tema tratado. La programación está llena de conceptos abstractos, que pueden llegar a ser comprendidos contextualizando el aprendizaje. De este razonamiento nace la llamada programación en contexto, la enseñanza de la programación a través de un contexto motivador que muestre una utilidad [2].

Aun así, la programación en contexto no ayuda a los estudiantes a aprender más en menos tiempo, ni tampoco significa que los estudiantes pueden aprender sólo con el contexto. Ahora bien, proporciona e incrementa la atención de la clase [2]. Además de mostrar una aplicación práctica, que puede ser el detonante para fomentar vocaciones científicas y tecnológicas.

Los conceptos de computación, pueden resultar complicados, debido a que los estudiantes encuentran dificultades en los conceptos teóricos abstractos de la programación. La robótica es una herramienta

que se utiliza en educación para contextualizar la programación, ya que, es un área interdisciplinar, donde se aplican conceptos de mecánica, electrónica y programación [5].

La robótica educativa, proporciona herramientas para aprender a programar, a la vez que los estudiantes se divierten en el contexto, pues estos robots son muy atractivos y populares para los más jóvenes [6]. Además, el contexto de la robótica tiene un gran potencial para desarrollar habilidades en matemáticas, física, tecnología e informática, sirviendo como una herramienta multidisciplinar en las escuelas [5]. Manipular objetos reales y ver el funcionamiento de la programación de forma real, ayuda a motivar y a fomentar el interés por la programación [7]. Es más fácil saber si funciona tu código, viendo si un robot realiza un determinado movimiento. La programación de un robot, materializa el comportamiento abstracto de los algoritmos y los convierte en algo atractivo para los jóvenes [6].

Un ejemplo claro de un robot creado para ser utilizado en el entorno educativo es LEGO Mindstorm. Este robot es programable y fomenta el pensamiento crítico y la habilidad para resolver problemas. A su vez, tiene un entorno gráfico para ser programado, basado en la programación por bloques, que facilita su programación y aumenta el interés [7].

Utilizar robots, junto con un lenguaje básico de programación por bloques, aumenta la curva de aprendizaje y el interés. Además, estudiantes y profesorado perciben que usar robots para aprender programación, ayuda a entender conceptos de programación y algoritmos, aparte de mejorar el ambiente de clase y aplicar los conocimientos adquiridos [6]. Completar retos robóticos, como pueden ser, moverse hacia delante, utilizar algún sensor, o encender una luz, incrementa la autoeficacia, la motivación y el interés por las disciplinas STEM [6].

Sin embargo, utilizar este material en clase, presenta ciertos problemas. Factores relacionados con aspectos técnicos, como son el movimiento de los motores, pues depende de la carga de las baterías; o las condiciones de iluminación, que afectan al funcionamiento de los sensores y a la percepción de las luces LEDs [6]. Por otro lado, el coste de estos robots educativos no es especialmente asequible para centros públicos de educación primaria y secundaria. Es más, estos robots suelen quedarse obsoletos en muy poco tiempo, al igual que cualquier otra tecnología, como por ejemplo los smartphones, cada año salen al mercado nuevos robots, con más prestaciones y nuevas características.

Los laboratorios virtuales son una solución a este tipo de problemas. En el mundo de la ingeniería, existen una gran cantidad de herramientas para simular el comportamiento de sistemas y dispositivos. Desde herramientas para comprobar el funcionamiento de circuitos, modelado 3D, diseños CAD, hasta grandes simuladores como aquellos que simulan el funcionamiento de una planta de producción, robots de automatización industrial o trayectorias de vuelo [8].

Algunas ventajas de utilizar laboratorios virtuales son [8]:

- Proporcionan una solución asequible para trabajar disciplinas STEM en universidades y centros educativos
- Ofrece flexibilidad y la posibilidad de cambiar la configuración fácilmente
- Varios estudiantes pueden utilizar el sistema al mismo tiempo

Sin embargo, también presentan ciertas desventajas. La complejidad del software de simulación desarrollado, pues necesita sistemas y características especiales para funcionar en una computadora; la falta de cuidado y responsabilidad a la hora de programar, puesto que, al ser virtual, el sistema no se puede ‘romper’ [8].

A pesar de estas desventajas, en niveles de secundaria, no se necesita un gran nivel de precisión ni de configuración. Permitir funciones de programación básica y un entorno atractivo para los estudiantes es más que suficiente, pues están acostumbrados al mundo virtual de los videojuegos, donde todo es gráfico, llamativo e intuitivo.

Por otra parte, la importancia y el interés en las plataformas virtuales ha aumentado debido a la situación ocasionada por la COVID-19. Durante más de tres meses, los estudiantes no han tenido ningún tipo de acceso e interacción con material y herramientas del aula. Las herramientas de la información y la comunicación, han sido el gran salvavidas para la educación durante la segunda parte del curso 2019/2020 y el curso 2020/2021. Plataformas como Moodle o Classroom, unidas a los sistemas de videoconferencia, han sido esenciales para mantener la comunicación y el sistema educativo, dentro de las posibilidades. Los laboratorios virtuales, ofrecen la posibilidad al alumnado de realizar actividades desde cualquier parte, contando con un dispositivo electrónico y conexión a internet.

Por tanto, para aprender a programar, no sólo es necesario enseñar las estructuras y posibilidades del código, es necesario aprender a pensar de forma computacional, usando la abstracción, modularidad, jerarquía y demás conceptos abstractos que nos ayudan a solucionar los problemas que aparecen en programación [4]. Para lograr todo esto, contamos con las plataformas de programación por bloques, el contexto y laboratorios virtuales.

Aprender robótica en cursos de educación secundaria, de un modo interdisciplinar y práctico, ayuda atraer a los más jóvenes hacia la tecnología, y en especial al público femenino. El porcentaje de chicas que deciden dedicarse al mundo de la tecnología, informática y electrónica sigue siendo muy bajo [5].

Las chicas que inicialmente no están interesadas en la tecnología, se sienten motivadas cuando las actividades de robótica se introducen con un contexto que cuenta una historia o está conectado con otras asignaturas o áreas que si le interesan [5]. Es el caso de plataformas gráficas de diseño 3D como Tinkercad. En general, debido a la presión social, las chicas se sienten menos seguras a la hora de utilizar herramientas informáticas y tecnológicas, pero el entorno gráfico y la vinculación que tiene con disciplinas como el arte y el diseño, ayudan a que las chicas se acerquen cada vez más a este tipo de herramientas. Otro ejemplo son los entornos que permiten personalizar de alguna forma los gráficos, como es el caso de Scratch. Realizar una parte ‘gráfica’, contando una historia, hace que se sientan más seguras y poco a poco descubren las posibilidades de la programación.

La robótica, combinada con una programación lógica, ayuda al desarrollo del pensamiento computacional y deductivo. Desarrollar estas habilidades en educación secundaria es esencial aunque los estudiantes, chicas y chicos, acaben eligiendo otro tipo de carreras profesionales, estos conocimientos son útiles e importantes para las profesiones del futuro [5] y para desenvolverse en una sociedad cada vez más digitalizada.

3 Objetivos

El objetivo principal de este proyecto es crear un conjunto de actividades para introducir conceptos de programación a través de un contexto atractivo, en una plataforma de simulación robótica y programación por bloques.

Los objetivos específicos son:

- Crear un guion de prácticas para que el alumnado pueda desarrollar las actividades planteadas, con gráficos e instrucciones de apoyo.
- Proporcionar al profesorado herramientas para aplicar esta actividad en el aula.
- Utilizar una herramienta que permita realizar la actividad en clases online.
- Desarrollar todo el material teniendo en cuenta el contexto en todo momento, para no perder el hilo de la actividad.
- Aplicar la actividad desarrollada, para obtener retroalimentación y mejorar el material y su accesibilidad.

4 Metodología

4.1 Marco Legal

Esta actividad está contextualizada dentro de los contenidos curriculares establecidos en la asignatura Tecnologías de la Información y Comunicación I (TIC I), que se imparte en el primer curso de bachillerato. El marco legal de esta asignatura es:

- Real Decreto 1105/2014 de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y del Bachillerato [9].
- Decreto 110/2016, de 14 de junio, por el que se establece la ordenación y el currículo del Bachillerato en la comunidad Autónoma de Andalucía [10].
- Orden de 14 de julio de 2016, por la que se desarrolla el currículo correspondiente al Bachillerato en la Comunidad Autónoma de Andalucía, se regulan determinados aspectos de la atención a la diversidad y se establece la ordenación de la evaluación del proceso de aprendizaje del alumnado [11].

4.2 Objetivos

Según lo establecido en la Orden de 14 de julio de 2016 [11], la asignatura Tecnologías de la información y la comunicación, tanto en primero como en segundo de bachillerato, debe cumplir los siguientes objetivos:

- Comprender la importancia de las tecnologías de la información y la comunicación en nuestra sociedad actual y cómo influyen a nivel social, económico y cultural.
- Entender cómo funcionan los ordenadores y los dispositivos digitales, así como las redes de comunicación e Internet.
- Utilizar herramientas para crear producciones digitales, bajo unos requisitos, realizando análisis, contraste y evaluación de datos.
- Componer un proyecto web para una finalidad determinada, teniendo en cuenta el diseño, la accesibilidad y analizando su uso.
- Utilizar las tecnologías de forma segura y responsable, reconociendo e informando de un mal uso de éstas.

- Fomentar el uso de herramientas Open Source² y proyectos colaborativos.
- Utilizar motores de búsqueda y navegadores con responsabilidad, contrastando y seleccionando la información deseada.
- Entender los algoritmos. Cómo se implantan, almacenan y ejecutan las instrucciones; representación y manipulación digital de datos.
- Desarrollo de aplicaciones informáticas y programación; aplicando ingeniería de software.
- Reconocer la normativa de protección de datos, asegurando la privacidad y aplicando medidas de seguridad en las tecnologías de la información y la comunicación.

4.3 Contenidos, criterios de evaluación y estándares de aprendizaje

En el currículo establecido por el BOE en el Real Decreto 1105/2014 del 26 de diciembre [9], y la Orden de 14 de julio de 2016 [11], se definen los contenidos curriculares correspondientes a la asignatura TIC de primero de bachillerato. En la Tabla 4.1, se muestran los contenidos, criterios de evaluación y estándares de aprendizaje definidos para esta asignatura.

La asignatura TIC I es una de las asignaturas específicas, que se ofertan en los centros de forma optativa y comprende cuatro horas a la semana [10].

El material desarrollado está relacionado con el Bloque 5 de contenidos establecidos para la asignatura TIC I, Programación (Sec. I. Pág. 533). En este bloque de contenidos se incluyen conceptos relacionados con la aplicación de algoritmos, analizar y resolver problemas, identificar elementos y construcciones básicas de lenguajes de programación; y además realizar una programación para solucionar determinadas aplicaciones y problemas reales. Todos estos aspectos están de manifiesto en las actividades propuestas y son reforzados gracias a un contexto y una plataforma de programación visual; trabajando indirectamente objetivos transversales.

² Código abierto. Desarrollo de software de forma colaborativa, con el objetivo de hacer accesibles programas informáticos al mayor número de personas posible

Bloque	Contenidos [11]	Criterios de Evaluación [11] [9]	Estándares de aprendizaje evaluables [9]
1. La sociedad de la información y el ordenador	La Sociedad de la Información y la Sociedad del Conocimiento. Impacto de las TIC: aspectos positivos y negativos. Ejemplos y exponentes: las redes sociales, el comercio electrónico, la publicidad en Internet, la creatividad digital, protección de datos, etc. Nuevos sectores laborales: marketing en buscadores (SEO/SEM), gestión de comunidades, analítica web, etc. Áreas emergentes: Big Data, Internet de las Cosas, etc.	1. Analizar y valorar las influencias de las tecnologías de la información y la comunicación en la transformación de la sociedad actual, tanto en los ámbitos de la adquisición del conocimiento como en los de la producción. CSC, CD, SIEP.	1.1. Describe las diferencias entre lo que se considera sociedad de la información y sociedad del conocimiento. 1.2. Explica que nuevos sectores económicos han aparecido como consecuencia de la generalización de las tecnologías de la información y la comunicación.
2. Arquitectura de ordenadores	Hardware y Software. Sistemas propietarios y libres. Arquitectura: Concepto clásico y Ley de Moore. Unidad Central de Proceso. Unidad de control. Unidad aritmético-lógica. Memoria principal. Memoria secundaria: estructura física y estructura lógica. Dispositivos de almacenamiento. Fiabilidad. Sistemas de entrada/salida: Periféricos. Clasificación. Periféricos de nueva generación. Buses de comunicación: datos, control y direcciones. Sistemas operativos: Arquitectura. Funciones. Normas de utilización (licencias). Gestión de procesos. Sistema de archivos. Usuarios, grupos y dominios. Gestión de dispositivos e impresoras. Compartición de recursos en red. Monitorización. Rendimiento. Instalación de SS.OO: requisitos y procedimiento. Configuración. Software de aplicación: Tipos. Clasificación. Instalación. Uso.	1. Configurar ordenadores y equipos informáticos identificando los subsistemas que los componen, describiendo sus características y relacionando cada elemento con las prestaciones del conjunto. CCL, CMCT, CD, CAA. 2. Instalar y utilizar software de propósito general y de aplicación evaluando sus características y entornos de aplicación. CCL, CMCT, CD, CAA. 3. Utilizar y administrar sistemas operativos de forma básica, monitorizando y optimizando el sistema para su uso. CD, CMCT, CAA.	1.1. Describe las características de los subsistemas que componen un ordenador identificando sus principales parámetros de funcionamiento. 1.2. Realiza esquemas de interconexión de los bloques funcionales de un ordenador describiendo la contribución de cada uno de ellos al funcionamiento integral del sistema. 1.3. Describe dispositivos de almacenamiento masivo utilizados en sistemas de ordenadores reconociendo su importancia en la custodia de la información. 1.4. Describe los tipos de memoria utilizados en ordenadores analizando los parámetros que las definen y su aportación al rendimiento del conjunto. 2.1. Elabora un diagrama de la estructura de un sistema operativo relacionando cada una de las partes las funciones que realiza. 2.2. Instala sistemas operativos y programas de aplicación para la resolución de problemas en ordenadores personales siguiendo instrucciones del fabricante.
3. Software para sistemas informáticos.	Procesadores de texto: Formatos de página, párrafo y carácter. Imágenes. Tablas. Columnas. Secciones. Estilos. Índices. Plantillas. Comentarios. Exportación e importación. Hojas de cálculo: Filas, columnas, celdas y rangos. Referencias. Formato. Operaciones. Funciones lógicas, matemáticas, de texto y estadísticas. Ordenación. Filtrado. Gráficos. Protección. Exportación e importación. Base de datos: Sistemas gestores de bases de datos relacionales. Tablas, registros y campos. Tipos de datos. Claves. Relaciones. Lenguajes de Definición y Manipulación de Datos, comandos básicos en SQL. Vistas, informes y formularios. Exportación. e importación. Presentaciones. Multimedia. Formatos de imágenes, sonido y vídeo. Aplicaciones de propósito específico.	1. Utilizar aplicaciones informáticas de escritorio o web, como instrumentos de resolución de problemas específicos. CCL, CMCT, CD, CAA. 2. Buscar y seleccionar aplicaciones informáticas de propósito general o específico, dados unos requisitos de usuario. CD, CAA, SIEP, CED.	1.1. Diseña bases de datos sencillas y /o extrae información, realizando consultas, formularios e informes. 1.2. Elabora informes de texto que integren texto e imágenes aplicando las posibilidades de las aplicaciones y teniendo en cuenta el destinatario. 1.3. Elabora presentaciones que integren texto, imágenes y elementos multimedia, adecuando el mensaje al público objetivo al que está destinado. 1.4. Resuelve problemas que requieran la utilización de hojas de cálculo generando resultados textuales, numéricos y gráficos. 1.5. Diseña elementos gráficos en 2D y 3D para comunicar ideas.

			1.6. Realiza pequeñas películas integrando sonido, vídeo e imágenes, utilizando programas de edición de archivos multimedia.
4. Redes de ordenadores.	Redes de ordenadores e Internet. Clasificación de las redes. Modelo de referencia OSI y arquitectura TCP/IP. Capa de enlace de datos. Capa de Internet. Capa de Transporte. Capa de Aplicación. Redes cableadas y redes inalámbricas. Direccionamiento de Control de Acceso al Medio. Dispositivos de interconexión a nivel de enlace: concentradores, conmutadores y puntos de acceso. Protocolo de Internet (IP). Enrutadores. Direcciones IP públicas y privadas. Modelo Cliente/Servidor. Protocolo de Control de la Transmisión (TCP). Sistema de Nombres de Dominio (DNS). Protocolo de Transferencia de Hipertexto (HTTP). Servicios: World Wide Web, email, voz y video. Buscadores. Posicionamiento. Configuración de ordenadores y dispositivos en red. Monitorización. Resolución de incidencias básicas.	<p>1. Analizar las principales topologías utilizadas en el diseño de redes de ordenadores relacionándolas con el área de aplicación y con las tecnologías empleadas. CMCT, CD, CSC.</p> <p>2. Analizar la función de los equipos de conexión que permiten realizar configuraciones de redes y su interconexión con redes de área extensa. CMCT, CD, CAA.</p> <p>3. Describir los niveles del modelo OSI, relacionándolos con sus funciones en una red informática. CCL, CD, CAA.</p> <p>4. Explicar el funcionamiento de Internet, conociendo sus principales componentes y los protocolos de comunicación empleados. CMCT, CD, CAA.</p> <p>5. Buscar recursos digitales en Internet, conociendo cómo se seleccionan y organizan los resultados, evaluando de forma crítica los contenidos recursos obtenidos. CD, CCL, CMCT, CSC, SIEP.</p>	<p>1.1. Dibuja esquemas de configuración de pequeñas redes locales seleccionando las tecnologías en función del espacio físico disponible. 1.2. Realiza un análisis comparativo entre diferentes tipos de cableados utilizados en redes de datos. 1.3. Realiza un análisis comparativo entre tecnología cableada e inalámbrica indicando posibles ventajas e inconvenientes. 2.1. Explica la funcionalidad de los diferentes elementos que permiten configurar redes de datos indicando sus ventajas e inconvenientes principales. 3.1. Elabora un esquema de cómo se realiza la comunicación entre los niveles OSI de dos equipos remotos.</p>
5. Programación	Lenguajes de programación: Estructura de un programa informático y elementos básicos del lenguaje. Tipos de lenguajes. Tipos básicos de datos. Constantes y variables. Operadores y expresiones. Comentarios. Estructuras de control. Condicionales e iterativas. Estructuras de datos. Funciones y bibliotecas de funciones. Reutilización de código. Facilidades para la entrada y salida de datos de usuario. Manipulación de archivos. Programación orientada a objetos: objetos, atributos y métodos. Interfaz gráfico de usuario. Programación orientada a eventos. Metodologías de desarrollo de software: Enfoque Top-Down, fragmentación de problemas y algoritmos. Pseudocódigo y diagramas de flujo. Depuración. Entornos de desarrollo integrado. Trabajo en equipo y mejora continua.	<p>1. Aplicar algoritmos a la resolución de los problemas más frecuentes que se presentan al trabajar con estructuras de datos. CMCT, CD.</p> <p>2. Analizar y resolver problemas dividiéndolos en sub-problemas y definiendo algoritmos que los resuelven. CMCT, CD.</p> <p>3. Analizar la estructura de programas informáticos, identificando y relacionando los elementos propios del lenguaje de programación utilizado. CMCT, CD.</p> <p>4. Conocer y comprender la sintaxis y la semántica de las construcciones básicas de un lenguaje de programación. CMCT, CD.</p> <p>5. Realizar pequeños programas de aplicación en un lenguaje de programación determinado aplicándolos a la solución de problemas reales. CMCT, CD, SIEP.</p>	<p>1.1. Desarrolla algoritmos que permitan resolver problemas aritméticos sencillos elaborando sus diagramas de flujo correspondientes.</p> <p>2.1. Escribe programas que incluyan bucles de programación para solucionar problemas que implique la división del conjunto en parte más pequeñas.</p> <p>3.1. Obtiene el resultado de seguir un pequeño programa escrito en un código determinado, partiendo de determinadas condiciones.</p> <p>4.1. Define qué se entiende por sintaxis de un lenguaje de programación proponiendo ejemplos concretos de un lenguaje determinado.</p> <p>5.1. Realiza programas de aplicación sencillos en un lenguaje determinado que solucionen problemas de la vida real.</p>

Tabla 4.1. Contenidos, Criterios de evaluación y estándares de aprendizaje de la asignatura TIC I [9] [11].

4.4 Estrategias Metodológicas

En la Orden de 14 de julio de 2016, por la que se desarrolla el currículo correspondiente al Bachillerato en la Comunidad Autónoma de Andalucía, se regulan determinados aspectos de la atención a la diversidad y se establece la ordenación de la evaluación del proceso de aprendizaje del alumnado [11], se definen una serie de estrategias metodológicas que son una guía para impartir la asignatura TIC en el aula. Los aspectos más importantes, referentes al nivel de bachillerato, a tener en cuenta durante la asignatura son:

- Centrar la metodología en mostrar un uso productivo, seguro y responsable de las tecnologías de la información y comunicación, con el objetivo de desarrollar la competencia digital y trabajar al resto de competencias clave.
- Realizar proyectos cooperativos, para aplicar los conocimientos adquiridos de forma teórica. Redactar y presentar una documentación, además de presentar el proyecto en público, haciendo uso de herramientas digitales.
- El alumnado, en la medida de lo posible, puede elegir la temática de los proyectos mencionados anteriormente, promoviendo así la diversidad de temáticas y trabajando elementos transversales. Esto ayuda a la motivación e interés por realizar el proyecto.
- Utilizar entornos de aprendizaje online, con el objetivo de dinamizar las clases e interactuar con el alumnado, utilizando herramientas TIC.
- Ajustarse a los diferentes ritmos de aprendizaje.

4.5 Plataforma Open Roberta Lab

Open Roberta Lab [12] es una plataforma de programación y simulación de código abierto desarrollada en 2002 por Fraunhofer IAIS (Instituto Fraunhofer de Análisis Inteligente y Sistemas de Información IAIS) y financiada por Google Alemania dentro de la iniciativa “Roberta – Learning with Robots” [13]. El Instituto Fraunhofer se encuentra en el municipio de Sankt Augustin, en Alemania. Es uno de los institutos científicos líderes en el campo de la inteligencia artificial, aprendizaje automático y big data en Alemania y Europa.

Open Roberta Lab nació con el objetivo de reducir la brecha digital, facilitando el acceso a la robótica educativa y el aprendizaje de la programación. Se puede utilizar de forma gratuita desde cualquier dispositivo que tenga un navegador web, a través de este enlace: <https://lab.open-roberta.org/>. De esta forma, desaparecen los obstáculos técnicos para docentes y alumnado.

Ofrece opciones para configuración de cuentas. Permite guardar programas, configurarlos como públicos o compartirlos con usuarios determinados de open Roberta Lab. También tiene opciones para crear y gestionar grupos. Al igual que la herramienta clases de Tinkercad. Un usuario puede crear un grupo y los miembros del grupo pueden utilizar la plataforma y guardar sus programas en la

cuenta del grupo; sin necesidad de que cada miembro del grupo tenga que registrarse con una cuenta propia. Esta funcionalidad es muy útil a la hora de llevar esta plataforma al aula.

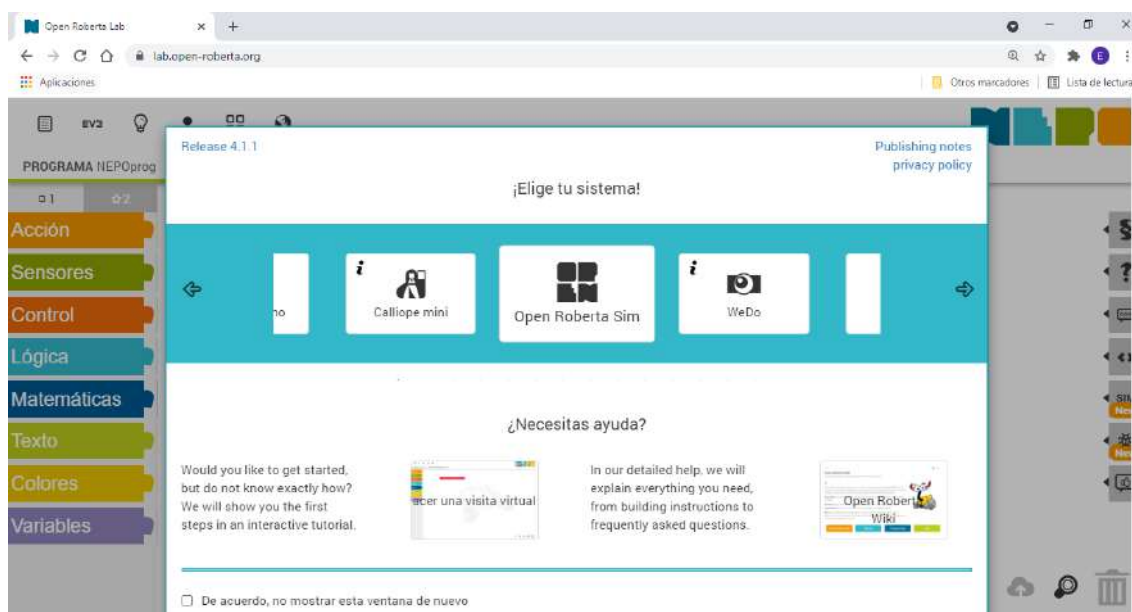


Figura 4.1. Plataforma Open Roberta Lab

En la Figura 4.1, aparece la página de inicio de la plataforma Open Roberta Lab, dónde permite escoger el sistema con el que se va a trabajar. En la Tabla 4.2, se muestran las opciones del menú de Open Roberta Lab: Editar, cambiar de sistema, ayuda, control de usuario, galería de proyectos y configuración de idiomas. Por ejemplo, desde la opción Usuario, se puede crear una nueva cuenta, entrar en una cuenta existente y acceder a la configuración de grupos que se ha mencionado anteriormente.

	Editar. Esta opción permite: ejecutar en simulación, ejecutar en hardware (en el caso de tener el robot/sistema conectado) guardar programa, abrir un programa nuevo, abrir mi lista de programas, cambiar los bloques de programación a modo avanzado, exportar e importar un programa.
EV3	Este botón aparece con el símbolo del sistema escogido en el inicio. Permite cambiar de sistema y conectar con el hardware en caso de conectar el sistema físico.
	Ayuda. Enlaza con la página wiki de Open Roberta Lab.
	Usuario: Crear usuario, registrarse, crear una cuenta de grupo y cerrar sesión.
	Galería de proyectos Open Roberta Lab.
	Cambiar el idioma. Open Roberta está disponible actualmente en 21 idiomas diferentes.

Tabla 4.2. Open Roberta Lab. Menú de Opciones

4.5.1 Entorno de programación

El lenguaje de programación que utiliza esta plataforma es NEPO (onliNE Open Programming), un lenguaje basado en programación por bloques. Con este lenguaje, se pueden crear programas simples y complejos en muy poco tiempo. Ofrece dos niveles de dificultad para realizar la programación. En el modo avanzado, aparte de añadir más bloques a las categorías existentes (Acción, Sensores, Control, Colores, Lógica, Matemáticas, Textos y Variables), se incluyen tres categorías más: listas, funciones y mensajes. Estos bloques avanzados permiten realizar una programación más compleja.

En la Tabla 4.3, aparecen las categorías de bloques de programación disponibles en la plataforma, tanto en nivel básico y como en avanzado. El nivel de los bloques de programación se puede cambiar en la opción Editar del Menú de la plataforma [Tabla 4.2]. Los bloques de acción incluyen todo lo relacionado con el movimiento de los motores: mover hacia delante/atrás, girar, mover hacia delante/atrás una determinada distancia; además incluyen bloques en los que se puede establecer la velocidad de cada una de las ruedas por separado. En la Figura 4.2, se puede observar la interfaz de programación de Open Roberta Lab, con los bloques de Acción desplegados.

Bloques básicos			
Acción	Movimiento de los motores	Lógica	Operadores lógicos y aritméticos. Construir condiciones
Sensores	Uso de los sensores disponibles (simulación y/o sistema físico)	Matemáticas	Operaciones matemáticas
Control	Bucles y sentencias	Texto	Opciones para mostrar texto
Colores	Colores detectados por el sensor de color	Variables	Definir variables locales y globales
Bloques Avanzados			
Listas	Crear una lista, ordenar y buscar elementos	Funciones	Crear funciones
Mensajes	Enviar y recibir mensajes vía bluetooth		

Tabla 4.3. Bloques de Programación Open Roberta Lab

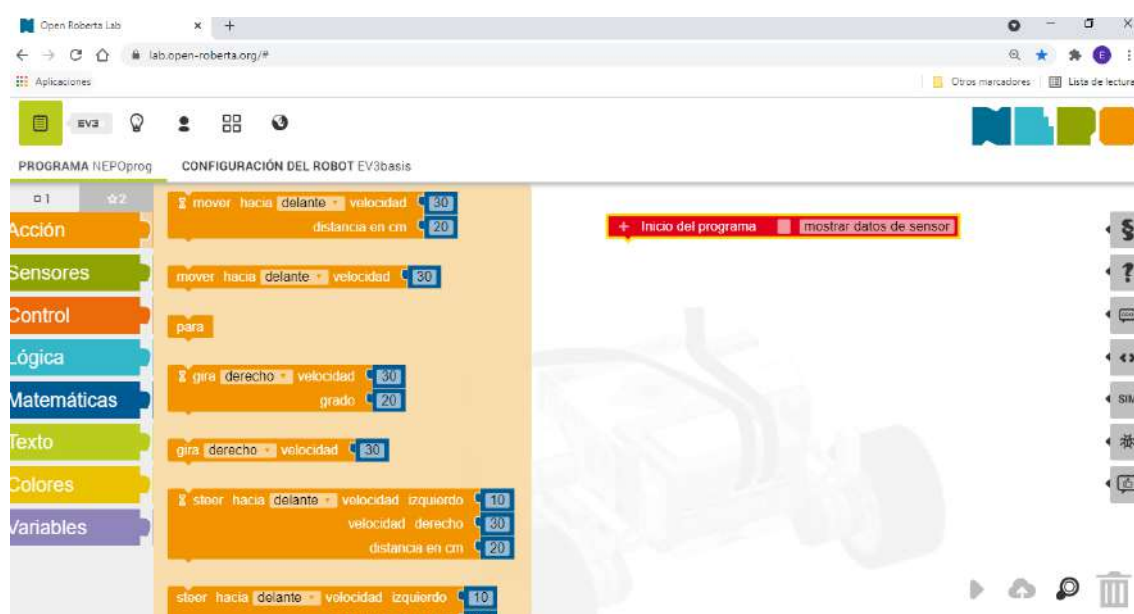


Figura 4.2. Open Roberta Lab. Interfaz de programación

4.5.2 Entorno de simulación

En Open Roberta Lab, se pueden programar diferentes sistemas. Sin embargo, la simulación no está disponible para todos. El simulador que ofrece más funcionalidades es el simulador para el robot LEGO MINDSTORMS EV3. La plataforma permite programar este robot y ver como ejecuta las instrucciones, con un fondo o plano de simulación. Este es el simulador que se utiliza en el material desarrollado en esta práctica. La simulación permite dos opciones:

- Configurar el robot virtual para que se ajuste a nuestras necesidades. La Figura 4.3, muestra la configuración del robot virtual. Se puede cambiar el diámetro de las ruedas, la distancia que hay entre ellas y configurar el número de motores y sensores de los que dispone. Los sensores están limitados en este caso a cuatro: sensor de contacto, de color, de ultrasonidos y un giroscopio. Los bloques de programación para controlar los sensores aparecen en la sección Sensores [Tabla 4.3].
- Configurar el entorno de simulación. Además de los controles básicos de inicio y Stop, en el entorno de simulación se puede cambiar el fondo de la simulación, añadir obstáculos, bloques de color, cargar una configuración, ver el estado de los sensores y demás opciones. Todas estas opciones aparecen descritas en la Tabla 4.4.

Otras opciones del entorno de simulación de Open Roberta Lab son:

- Visualizar código: abrir una pestaña para visualizar la ventana de comandos; es decir, traducir la programación por bloques a código.
- Añadir notas: las notas añadidas se guardan dentro del programa, lo que permite documentar la programación realizada.
- Simulación en modo Debug: está disponible la simulación paso a paso del programa desarrollado. Esta opción es útil para verificar el correcto funcionamiento de todas las partes de la programación.

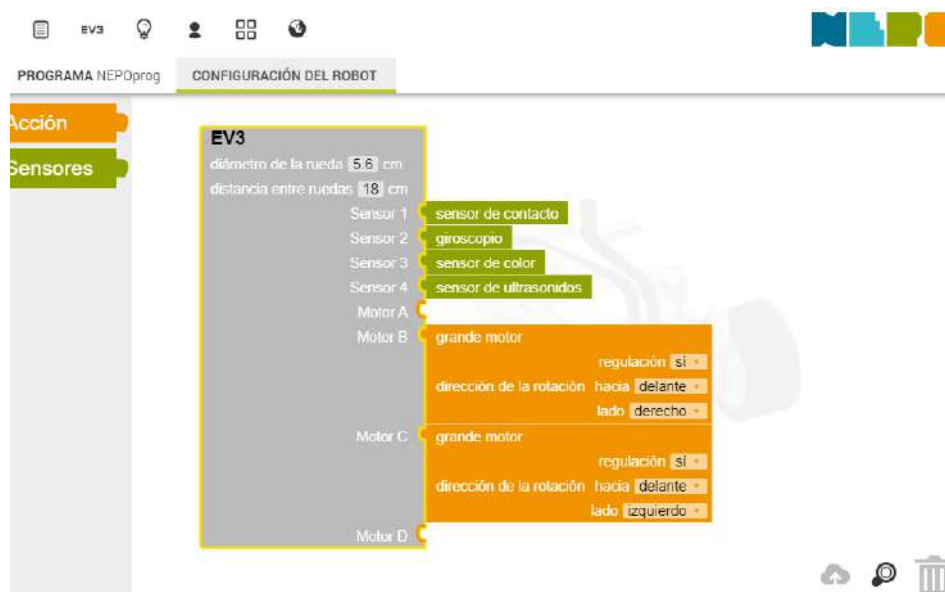


Figura 4.3. Opciones de Configuración del Robot. Simulador EV3











	Cambiar fondo de simulación. Con esta opción, se puede elegir entre los fondos de simulación predeterminados de Open Roberta.
	Subir un fondo de simulación. Permite poner el fondo de simulación desde una imagen concreta.
	Importar, exportar condiciones de simulación.
	Añadir un obstáculo, reconocible por el sensor de ultrasonidos.
	Añadir un área de color, reconocible por el sensor de color.
	Elegir el color del obstáculo/área de color añadidas a la simulación.
	Eliminar obstáculo/área de color.
	Iniciar simulación.
	Vista de estado de los sensores.
	Volver a la posición inicial. Pone el robot en la posición inicial por defecto.

Tabla 4.4. Open Roberta Lab. Opciones de Simulación

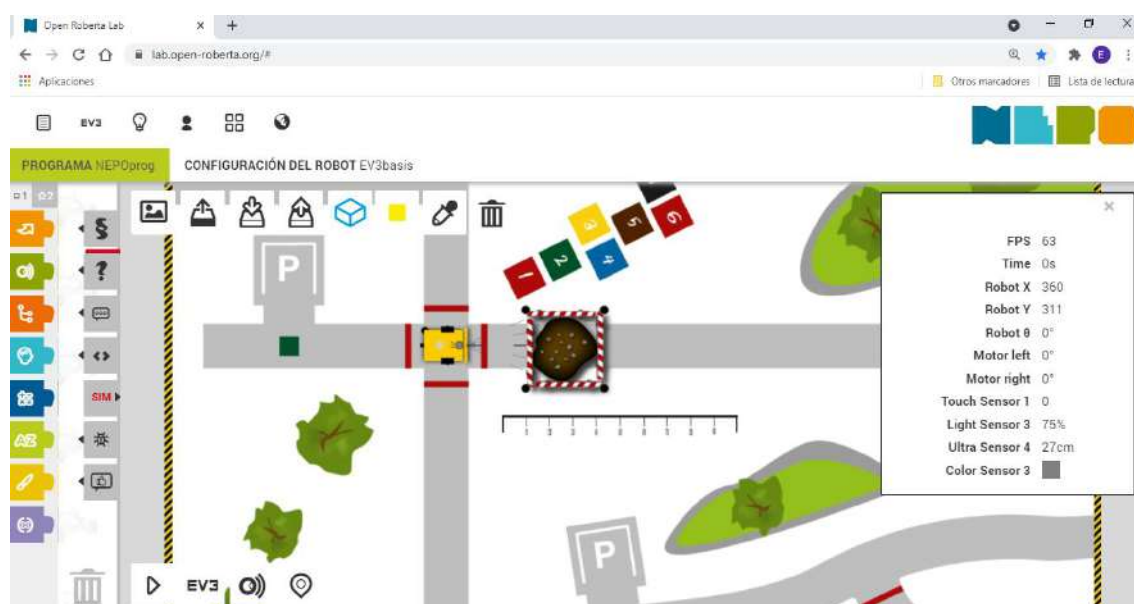


Figura 4.4. Open Roberta Lab. Entorno de Simulación

5 Material Desarrollado

El material desarrollado en este proyecto es un guion de prácticas para el alumnado y un guion con ciertas propuestas e información, destinado al profesorado que aplique las actividades en el aula.

El objetivo es programar un robot, a través de la plataforma Open Roberta Lab, contextualizado dentro del sector de la robótica espacial y las misiones espaciales. El robot, se convierte en un Rover Lunar, que ha sido enviado a la Luna para recoger muestras de la superficie, con el fin de estudiar si podríamos vivir allí en un futuro.

5.1 Objetivos

Los objetivos de las actividades propuestas, se dividen en generales, como son, motivar al alumnado y fomentar el interés en las carreras espaciales; y en específicos, centrados en los contenidos curriculares que se trabajan y refuerzan realizando esta actividad.

En la Tabla 5.1, se muestra el listado de objetivos generales y específicos que cumple el material desarrollado. Uno de los objetivos generales es aumentar el interés por las carreras espaciales, mostrando que pueden estar a nuestro alcance. Algunos de los objetivos específicos, son desarrollar el pensamiento computacional, trabajar la programación y fomentar la creatividad y el espíritu crítico.

Objetivos Generales	Objetivos Específicos
Fomentar el interés en las carreras espaciales	Trabajar conceptos básicos de programación y robótica
Contextualizar la programación en el mundo de la exploración espacial; con el objetivo de despertar el interés y la curiosidad de los estudiantes por la robótica y la exploración espacial	Desarrollar el pensamiento computacional
Motivar al alumnado para que considere una carrera técnica en el futuro	Aplicar a un caso práctico bucles y sentencias de programación
	Fomentar la creatividad y el espíritu crítico
	Desarrollar habilidades en programación y robótica a través de un entorno gráfico de simulación
	Trabajar la competencia digital
	Conocer las posibilidades profesionales dentro de la industria espacial

Tabla 5.1. Objetivos generales y Específicos del material desarrollado.

5.2 Competencias, Contenidos y Planificación

En el guion para el alumnado, se realizan un total de siete actividades, entre Misiones y actividades de repaso, refuerzo e introducción.

En la Tabla 5.2, aparece el listado de actividades, mostrando para cada una de ellas, una breve descripción, los objetivos, los conocimientos previos requeridos y la duración aproximada de la actividad. Las actividades 1 y 2, son de introducción y su objetivo es que alumnado se familiarice con el entorno de programación y simulación.

Después de estas actividades de introducción, aparece la Misión 01 (actividad 3), en la que hay que volver a la base desde la posición inicial. Tras la Misión 01, se incluyen las actividades de repaso y refuerzo 4 y 5, destinadas a mostrar el funcionamiento de los bucles y de la sentencia if/else. Para realizar la Misión 03 (actividad 7), es necesario completar con éxito todas las actividades anteriores, ya que, su objetivo es que el alumnado sea capaz de aplicar todos los conocimientos previos. En la columna destinada a mostrar los conocimientos previos requeridos, se puede observar que, excluyendo las actividades 1 y 2 de introducción, para completar una actividad es necesario realizar la actividad anterior.

También, en la Tabla 5.2, se muestra un cronograma, indicando el tiempo medio que necesita el alumnado para realizar cada una de las actividades; sin tener en cuenta explicaciones previas, introducción al contexto y demás factores externos que ocurren en el aula. Por esta razón, en la Tabla 5.3, se propone una planificación para cuatro sesiones de una hora. Este tiempo permite introducir de forma adecuada el contexto y, sobre todo, ayuda a que el alumnado que muestre un menor ritmo de aprendizaje, pueda completar las misiones sin problemas. Mientras tanto, el resto puede ampliar su programación y probar nuevos bloques y opciones de la plataforma.

Por último, la Tabla 5.4, se relacionan las actividades propuestas, con los criterios de evaluación y los estándares de aprendizaje definidos en BOE [9] y en BOJA [11] para el Bloque 5 de contenidos de la asignatura TIC I. También se incluye una columna indicando las Competencias Clave relacionadas [11]. En las actividades 3, 4, 5, 6 y 7, el alumnado resuelve pequeños problemas de programación, con un lenguaje determinado, dentro de un determinado contexto y aplicación. Por tanto, estas actividades están relacionadas con el criterio de evaluación número 5 de la asignatura TIC I, y con el estándar de aprendizaje 5.1. Además, se trabaja la competencia digital (CD), la competencia matemática y competencia básica en ciencia y tecnología (CMCT) y el sentido de la iniciativa y espíritu emprendedor (SIEP).

Es claro que, la realización de estas actividades, no cubre todas las necesidades curriculares para el Bloque 5 de contenidos de la asignatura TIC I. Sin embargo, la Tabla 5.4, tiene como objetivo señalar la estrecha relación que existe entre el currículo y las actividades presentadas, mostrando todas sus posibilidades y flexibilidad. Así como, los contenidos que se pueden reforzar y trabajar con estas actividades.

Act	Título	Descripción	Objetivo	Conocimientos previos	Tiempo (min)
1	¿Cómo funciona nuestro simulador?	Entrar en el simulador y seleccionar el sistema adecuado. No hay que hacer ninguna tarea.	Introducir funciones básicas y la interfaz gráfica del simulador	-	10
2	Misión Inicial. Nuestro Simulador	Programar el robot para que se mueva hacia delante con bloques de acción y utilizando el sensor de color	Conocer el entorno de programación y simulación	-	20
3	Misión 1. ¡Tenemos que volver a la base!	Programar el robot para que realice una trayectoria determinada y vuelva a la base lunar desde la posición inicial	Conocer el funcionamiento de los bloques de acción y los ángulos de giro	Actividades 1 y 2	30
4	¿Qué es un bucle?	Programar el robot para que repita indefinidamente una determinada instrucción	Utilizar y recordar el funcionamiento de un bucle	Actividades 1, 2 y 3	15
5	¿Qué hace la sentencia if/else?	Programar el robot para realizar diferentes acciones dependiendo de si se cumple una determinada condición	Utilizar y recordar el funcionamiento de la sentencia if/else	Actividades 1, 2, 3 y 4	15
6	Misión 2. Busquemos muestras de la superficie Lunar	Programar el robot para que pase por cuatro puntos diferentes, utilizando ángulos de giro y el sensor de color	Conocer el funcionamiento del sensor de color y utilizarlo para realizar una determinada trayectoria	Actividades 1, 2,3,4 y 5	30
7	Misión 3. ¡Hagamos una carrera en la Luna!	Programar el robot para seguir una línea negra	Aplicar los conocimientos adquiridos en las actividades previas, utilizando bloques de programación de acción, control, lógica y sensores	Actividades 1, 2, 3, 4, 5 y 6	30

Tabla 5.2. Resumen de Actividades Propuestas: Descripción, objetivos, conocimientos requeridos y actividades tiempo aproxima

Sesión	Tiempo (h)	Contenidos propuestos
1	1	Introducción al contexto: utilizando videos y material de apoyo Conocer el entorno de simulación Misión Inicial. Nuestro Simulador
2	1	Misión 01 ¡Tenemos que volver a la base!
3	1	Refuerzo de Contexto Actividades de repaso (Actividades 4 y 5) Inicio Misión 02. Busquemos muestras de la superficie Lunar
4	1	Misión 02. Busquemos muestras de la superficie Lunar Misión 03. ¡Hagamos una carrera en la Luna! Cierre y debate sobre la exploración y la robótica Espacial ¿Llegaremos a vivir en la Luna en el futuro?

Tabla 5.3. Propuesta de Planificación

Actividades Relacionadas	Estándares de Aprendizaje. TIC I, Bloque 5: Programación [9] [11]	Criterios de Evaluación. TIC I, Bloque 5: Programación [9] [11]	CC [11]
2, 3	1.1. Desarrolla algoritmos que permitan resolver problemas aritméticos sencillos	1. Aplicar algoritmos a la resolución de los problemas más frecuentes que se presentan al trabajar con estructuras de datos.	CMCT, CD
4, 5, 7, 6	2.1. Escribe programas que incluyan bucles de programación para solucionar problemas que impliquen la división del conjunto en parte más pequeñas.	2. Analizar y resolver problemas de tratamiento de información dividiéndolos en sub-problemas y definiendo algoritmos que los resuelven.	CMCT, CD
3, 6, 7	3.1. Obtiene el resultado de seguir un pequeño programa escrito en un código determinado, partiendo de determinadas condiciones.	3. Analizar la estructura de programas informáticos, identificando y relacionando los elementos propios del lenguaje de programación utilizado.	CMCT, CD
2, 3, 6, 7	4.1. Define qué se entiende por sintaxis de un lenguaje de programación proponiendo ejemplos concretos de un lenguaje determinado.	4. Conocer y comprender la sintaxis y la semántica de las construcciones básicas de un lenguaje de programación.	CMCT, CD
3, 4, 5, 6, 7	5.1. Realiza programas de aplicación sencillos en un lenguaje determinado que solucionen problemas de la vida real.	5. Realizar pequeños programas de aplicación en un lenguaje de programación determinado aplicándolos a la solución de problemas reales.	CMCT, CD, SIEP

Tabla 5.4. Relación de las actividades con el currículo y las competencias clave.

5.3 Herramientas de Evaluación

Para evaluar el trabajo realizado durante el desarrollo de la práctica, la plataforma Open Roberta Lab ofrece una herramienta de creación de grupos. Esto permite compartir programas con el grupo y visualizar los programas que los miembros del grupo han creado, indicando fecha y hora de creación [Sección 4.5 Plataforma Open Roberta Lab].

Aparte de esta herramienta, para llevar estas actividades al aula y disponer de instrumentos para evaluar y calificar, se propone una rúbrica de evaluación. Aparece en la Tabla 5.5, y está separada en tres niveles: nivel inicial, en desarrollo, conseguido y excelente. Esta separación en cuatro niveles permite adaptar la rúbrica a las características de cada grupo-clase.

Los niveles están basados en completar las tareas que se proponen en las actividades. Por ejemplo, en la misión inicial (actividad 2) se proponen dos tareas. Por tanto, si el alumnado completa las dos tareas correctamente, su nivel es que ha ‘Conseguido’ el reto; por el contrario, si sólo realiza una de las dos tareas, su nivel está ‘En Desarrollo’. Su nivel es ‘Inicial’, cuando completa una de las tareas con dificultades. El nivel ‘Excelente’ se alcanza cuando, además de conseguir los retos, el alumnado es capaz de introducir bloques adicionales a la programación.

La actividad 1, no se incluye en la rúbrica, ya que, no propone realizar ninguna tarea. Sólo tiene el objetivo de mostrar al alumnado como entrar en la plataforma y elegir el sistema adecuado para la simulación, tal y como aparece en su descripción de la Tabla 5.2.

Act	Título	Nivel Inicial	En Desarrollo	Conseguido	Excelente
2	Misión Inicial. Nuestro Simulador	Realiza la primera tarea con dificultades	Realiza la primera tarea correctamente. La segunda tarea con dificultades	Realiza las dos tareas correctamente	Realiza las dos tareas y añade bloques de ampliación, como alarmas y/o melodía
3	Misión 1. ¡Tenemos que volver a la base!	No completa la primera misión	Realiza la primera misión con dificultades y/o haciendo uso de la solución	Realiza la primera misión correctamente	Realiza la primera misión con la trayectoria de ejemplo, incluyendo ampliaciones como bloques de alarmas y/o melodías. Realiza otra trayectoria adicional para completar la misión
4	¿Qué es un bucle?	No completa la tarea	Completa la tarea con dificultades	Completa la tarea correctamente	Completa la tarea y modifica los bloques para realizar otra acción
5	¿Qué hace la sentencia if/else?	No completa la tarea	Completa la tarea con dificultades	Completa la tarea correctamente	Completa la tarea y modifica los bloques para realizar otra acción
6	Misión 2. Busquemos muestras de la superficie Lunar	Realiza la primera parte de la misión con dificultades	Realiza la misión completa con dificultades y/haciendo uso de la solución	Realiza la misión correctamente	Realiza la misión completa y modifica la programación para cambiar la trayectoria, añadir alarmas y/o melodías
7	Misión 3. ¡Hagamos una carrera en la Luna!	Realiza la programación básica indicada en el guion	Realiza la misión completa con dificultades y/haciendo uso de la solución	Realiza la misión correctamente	Realiza la misión completa, optimiza el código y añade bloques de alarma y/o melodía

Tabla 5.5. Propuesta Rúbrica de Evaluación

5.4 Guiones

En este punto, se muestra una breve descripción de la estructura de los guiones realizados, para el alumnado y para el profesorado. Los guiones completos desarrollados, se encuentran en el Anexo I y el Anexo II.

5.4.1 Guion Alumnado

El guion dirigido al alumnado está basado en una metodología tipo Scaffolding³, dónde, a través de instrucciones guiadas, se va mostrando cómo realizar las actividades. Para realizar las actividades propuestas en el guion, no es necesario que el alumnado tenga experiencia previa en programación, pues está estructurado y planteado para un nivel inicial en programación en robótica.

Está dividido en dos partes: una primera parte, donde aparece el contexto, los primeros pasos en el entorno de simulación, el lenguaje de programación por bloques y las opciones de simulación; y una segunda parte, centrada en desarrollar una programación más compleja. En la Tabla 5.6 se muestra la estructura del guion completo.

Todo el guion incluye gráficos de apoyo y cuadros de texto en color, para llamar la atención sobre aspectos importantes y proponer retos adicionales. A continuación, se explica brevemente el contenido de cada parte del guion, su estructura y actividades.

Contenidos	Descripción
Parte I	
Contexto	Robótica espacial, control remoto de un Rover Lunar para recoger muestras en la superficie de la Luna.
Misión Inicial. Nuestro Simulador	Actividad de Introducción. Realizar una programación inicial sencilla para controlar el entorno de programación y simulación
Misión 1 ¡Tenemos que volver a la base!	Programar el Rover, para que vuelva a la base.
Parte II	
¿Qué es un bucle?	Actividad de refuerzo/introducción
¿Qué hace la sentencia if/else?	Actividad de refuerzo/introducción
Misión 2 Busquemos muestras de la superficie Lunar	Utilizar el sensor de color, para que el robot encuentre las muestras en la superficie lunar
Misión 3 ¡Hagamos una carrera en la Luna!	Utilizar el sensor de color, para que el robot siga una línea negra

Tabla 5.6. Tabla resumen de los contenidos del guion del alumnado

³ Scaffolding, en español ‘Andamiaje’ es una metodología que se basa en separar en bloques o escalones el proceso de aprendizaje, de tal forma que, en cada bloque, es necesario aplicar lo aprendido en el anterior; tal y como se construye un andamio.

5.4.1.1 Parte I

El aspecto más importante es la primera parte del guion, porque muestra la contextualización, pieza clave para motivar y captar la atención del alumnado. El contexto presentado está basado en la guía didáctica Misión en la Luna, de la oficina de recursos educativos de la agencia espacial europea, ESERO Spain [14].

El contexto comienza con un párrafo donde se resumen las misiones espaciales a la Luna más importantes, haciendo mención a la misión Apollo 11 de la NASA. Después de esta misión, las agencias espaciales se dieron cuenta de que enviar astronautas a la Luna era muy peligroso, y los robots teledirigidos son una buena opción para inspeccionar la superficie Lunar. Continúa introduciendo la Misión Heracles, una misión conjunta entre la Agencia Espacial Europea y las agencias espaciales de Canadá y Japón. En esta misión, se plantea controlar de forma remota un robot todoterreno, que recorrerá la superficie Lunar para recoger diferentes muestras.

Dentro de este contexto, se lanza el reto [Figura 5.2]: *“En esta práctica, vamos a programar un Rover que recorre la superficie Lunar en busca de muestras. Utilizaremos un simulador, en el que nuestro Rover podrá moverse por una superficie Lunar simulada y recoger muestras gracias al sensor de color”*.

Como se puede ver en la Figura 5.1 y Figura 5.2, el contexto ocupa las dos primeras páginas del guion y está reforzado por imágenes del Rover de la misión Heracles y de Neil A. Armstrong en la Luna. Se presenta este contexto para motivar al alumnado aprovechando la fascinación que sienten por el mundo del espacio y en especial por la robótica espacial. Pues, las películas de ciencia ficción, muestran muchos robots y sistemas que funcionan de forma autónoma fuera de nuestro planeta.

Las imágenes para realizar los diferentes retos de programación han sido creadas especialmente para apoyar al contexto y que durante el desarrollo de las actividades no se quede en un segundo plano. Además, las actividades con más peso, se han llamado Misión 01, Misión 02 y Misión 03, para no perder el contexto espacial en ningún momento. Es más motivador y atractivo completar una Misión, que realizar una actividad.

Justo después del contexto, hay una breve explicación de cómo funciona el simulador [Figura 5.2]. Consiste en una serie de instrucciones para entrar en el simulador y escoger el sistema adecuado. En esta explicación inicial se muestran gráficos para que el alumnado pueda identificar claramente, las zonas de programación y simulación. Así como, abrir la pestaña de simulación e identificar los controles que nos ofrece el entorno de simulación de Open Roberta Lab.

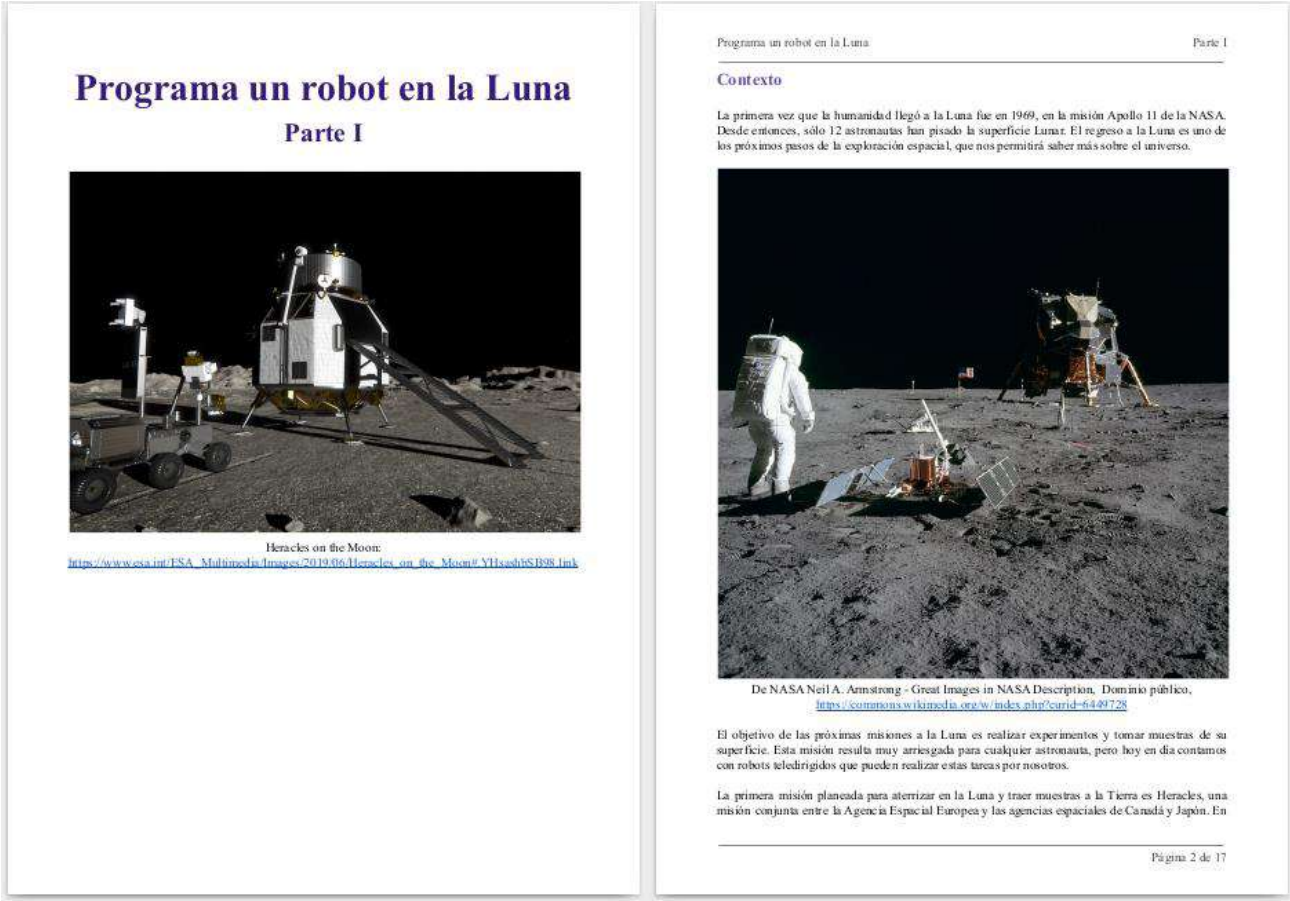


Figura 5.1. Guion del Alumnado Parte I. Portada y contexto

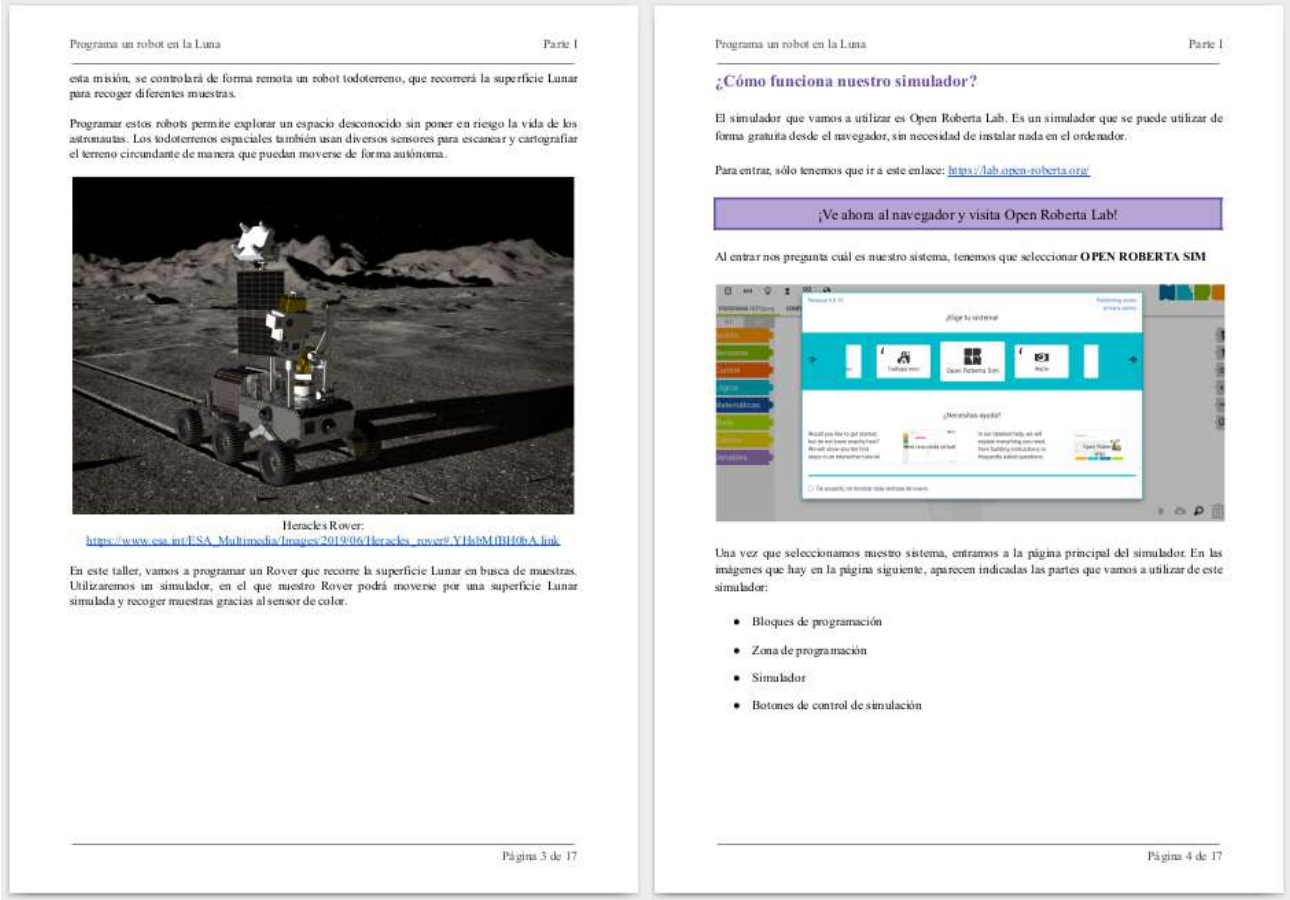


Figura 5.2. Guion del Alumnado Parte I. Contexto y Explicación Inicial.

Las Figura 5.3, Figura 5.4 y Figura 5.5 son imágenes que se incluyen en el guion del alumnado para mostrar de forma visual la interfaz de Open Roberta Lab. En la Figura 5.3 se muestra dónde se encuentran la Zona de Programación, los Bloques de Programación y la pestaña dónde se abre el Simulador.

En la Figura 5.4, se señalan dónde están los controles de simulación. Estas imágenes guían durante los primeros pasos en el entorno de programación y simulación, muy importante para que el alumando no pierda el interés al principio y perciba la plataforma como una herramienta fácil de utilizar.



Figura 5.3. Guion Alumnado. Gráfico para mostrar la zona de programación y abrir el simulador.

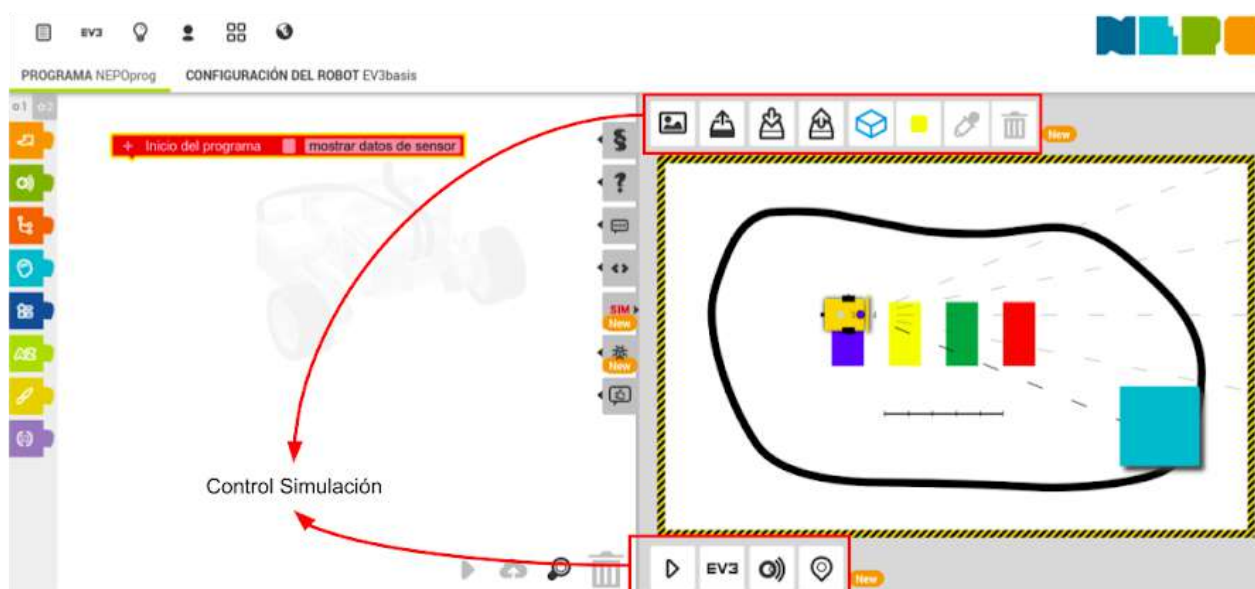


Figura 5.4. Guion del Alumnado Parte I. Gráfico para mostrar los controles de simulación

Para terminar con la introducción sobre el simulador, se incluye el gráfico de la Figura 5.5. En éste se indica la utilidad de los controles de simulación que se van a utilizar: Cambiar el fondo, Subir un fondo, Iniciar simulación, Ver sensores y Volver a la posición inicial.

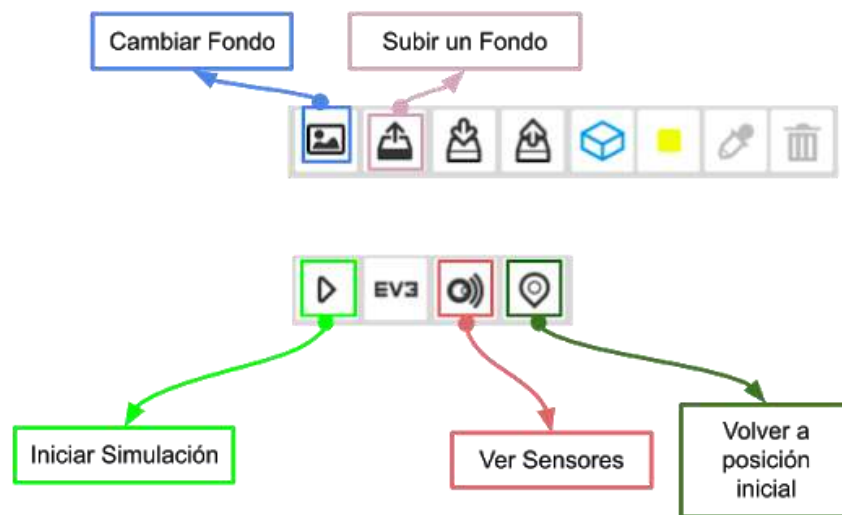


Figura 5.5. Guion del Alumnado Parte I. Gráfico para mostrar la utilidad de los controles de simulación

Después de esta breve explicación, comienza la actividad de introducción, **Misión Inicial: Nuestro Simulador**. Esta misión ha sido creada para introducir conceptos básicos de programación, a la vez que el alumnado se sigue familiarizando con el funcionamiento del simulador. La actividad está totalmente guiada, y las tareas se realizan paso a paso, con imágenes y gráficos de apoyo. Se realizan dos tareas: mover el robot hacia delante y utilizar el sensor de color. Estas dos estructuras básicas son las que se utilizan para programar el robot en las siguientes actividades.

Durante el desarrollo de la Misión Inicial, se incluye la Figura 5.6 que muestra de forma visual e intuitiva los pasos a seguir para iniciar la simulación, haciendo hincapié en que antes de pulsar Play, es necesario volver a la posición inicial. De lo contrario la misión no se finaliza de forma correcta.

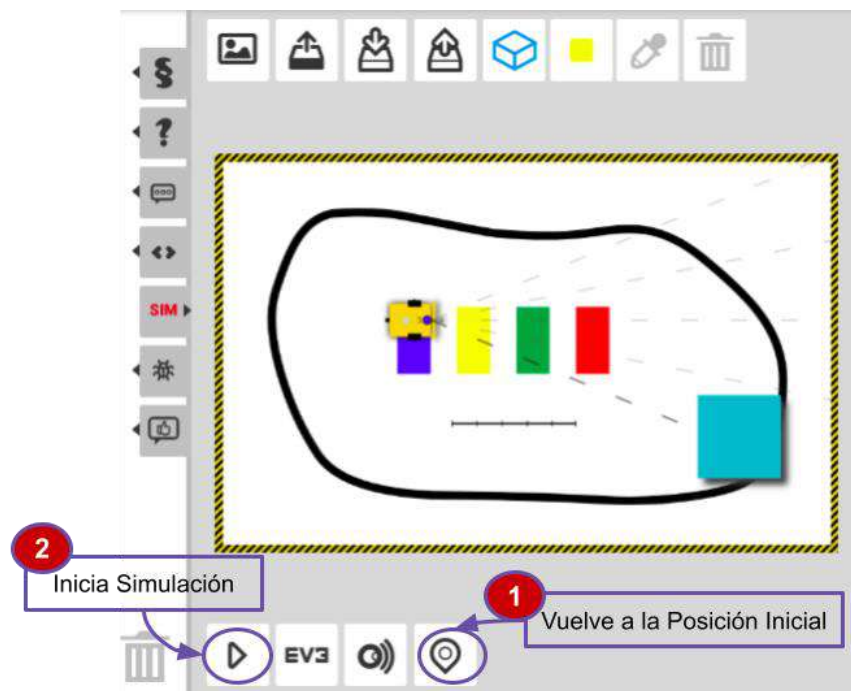


Figura 5.6. Guion del Alumnado Parte I. Gráfico para mostrar los pasos para iniciar la simulación. 1. Volver a la posición Inicial, 2. Iniciar simulación.

Una vez terminadas las dos tareas de la misión inicial, el guion avanza hacia la **Misión 01: Tenemos que volver a la base**. El primer paso en esta misión es mostrar cómo se descargan las imágenes necesarias para la simulación de nuestro Rover Lunar [Figura 5.7]. Para que estas imágenes sean accesibles y se puedan descargar desde cualquier dispositivo, están alojadas en una carpeta de Drive, en modo lectura para todas las personas que dispongan del enlace.


Después de guardar las imágenes, el guion muestra las instrucciones para cargarlas en el fondo de simulación de Open Roberta Lab. Como aparece en la Figura 5.7, se vuelve a incluir en el guion el gráfico de la Figura 5.5, dónde aparecen las funciones de los botones de la simulación; con el objetivo de recordar que botón que hay que utilizar para subir una imagen al fondo de simulación.

El primer fondo de simulación [Figura 5.8], está compuesto por la superficie Lunar y una base Lunar equipada con una antena de comunicaciones, una placa solar y un sistema de gestión de residuos.

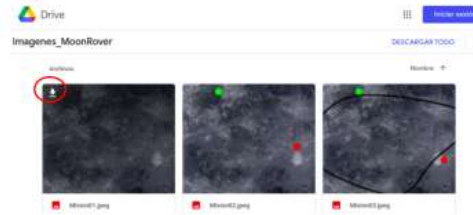
La base lunar, también forma parte del contexto y puede ayudar a introducir en clase conceptos relacionados con las telecomunicaciones, la importancia de la gestión de residuos y las formas de conseguir energía en el espacio exterior. Esto último puede enlazar con las diferentes fuentes de energía que disponemos en nuestro planeta.

Programa un robot en la Luna Parte I

Cuando entres en el enlace te aparecerán tres imágenes para descargar:



Lleva el ratón hacia el pico izquierdo de las imágenes y haz clic sobre el icono que aparece para descargarlas:

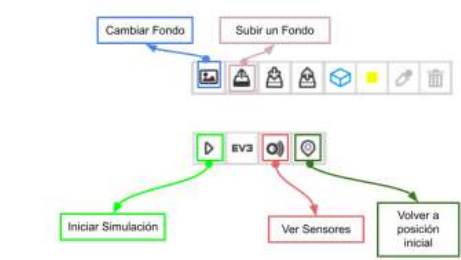


¡Es importante que recuerdes dónde guardas las imágenes!! Puedes guardar las imágenes en una carpeta en el **ESCRITORIO**. Si al descargar las imágenes, el navegador no te da opción para elegir en qué carpeta guardarlas, las encontrarás en la carpeta **DESCARGAS**.


Una vez hayas guardado las imágenes, abre la ventana de simulación de Open Roberta (la pestaña SIM del lateral derecho) y haz clic en el botón de **subir un fondo**, que está arriba a la izquierda:

Página 13 de 17

Programa un robot en la Luna Parte I



Al darle a este botón, se abrirá una ventana, dónde hay que buscar nuestras imágenes dónde las hemos guardado. Busca **Mision01.jpg** y selecciona abrir:



Pulsa **NO** en la ventana que te aparece:

Atención

The Open Roberta Lab can automatically load your simulation background on your next visits. For this purpose, we will save data in the Local Storage on your computer. [More information in our privacy policy.](#)

SI
NO

Ahora ya tenemos nuestro fondo, en el que nuestro robot se ha convertido en un Rover Lunar. Debes ver a tu robot en la superficie de la luna.

La base lunar, es la casita que puedes ver arriba a la derecha. Nuestra base lunar está equipada con una antena para las comunicaciones, placas solares para tener energía y un pequeño tubo para la gestión de residuos.

Página 14 de 17

Figura 5.7. Guion del Alumnado Parte I. Misión 01.



Figura 5.8. Fondo de Simulación de la Misión 01.

Una vez subida la imagen, el primer reto es programar el robot para llegar a la base, desde su posición inicial por defecto. En esta primera programación, se realiza de forma guiada un ejemplo de trayectoria a seguir para llegar a la base. Para este cometido se presenta una imagen, donde aparece dibujada la trayectoria, indicando el ángulo de giro y las direcciones que el Rover tiene que seguir [Figura 5.9].

A parte de programar el Rover para que vuelva a la base, se introduce un bloque de programación que reproduce una nota musical durante un determinado tiempo, cuando el Rover llega a la base. Este aspecto es divertido y, crear una melodía, puede ser una posible ampliación para el alumnado que presenta un mayor ritmo de aprendizaje.

Para llegar a la base desde la posición inicial, hay muchas formas y soluciones, por lo que seguramente una parte del alumnado, intentará llegar con otro camino, después de realizar la trayectoria de ejemplo. Con esta actividad termina la primera parte del guion de prácticas, que se realiza en aproximadamente 60 minutos, dependiendo del nivel y del ritmo de aprendizaje del alumnado.

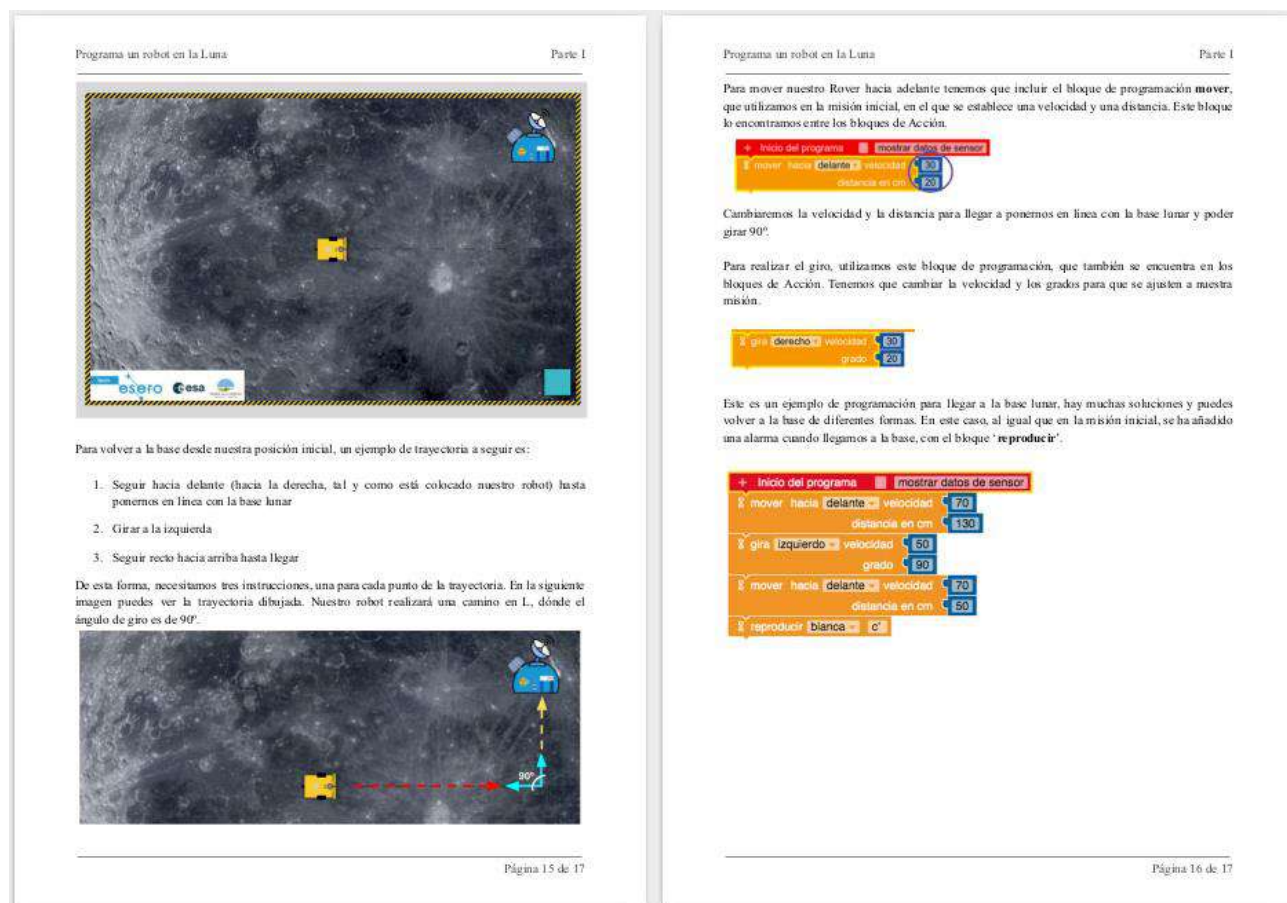


Figura 5.9. Guion del Alumnado Parte I. Misión 01.

5.4.1.2 Parte II

Esta parte del guion sigue teniendo muy presente el contexto, pero se centra más en conceptos de programación. Para realizar esta parte, es necesario que el alumnado complete las actividades previas, pues, aunque muestre conocimientos en lenguajes de programación por bloques, es necesario controlar la interfaz y los controles de programación.

Al igual que en la Parte I, este guion incluye actividades de introducción, donde se realizan ejemplos de la programación que habrá que realizar en los siguientes retos. La primera actividad de introducción es sobre bucles [Figura 5.10]. El objetivo es crear un bucle para que el robot gire de forma indefinida, hasta que se pare la simulación. La segunda actividad de repaso, es sobre la sentencia if/else [Figura 5.11], en la que se realiza una programación similar a la propuesta en los retos siguientes.

Las actividades de introducción, están guiadas, mostrando los pasos a seguir, con gráficos y bloques resaltados de apoyo. Este aspecto es importante, para que el alumnado pueda entender bien cómo funcionan estos bloques de programación y poder aplicar esos conocimientos en los siguientes retos del guion.

Programa un robot en la Luna

Parte II



Heracles on the Moon:

https://www.esa.int/ESA_Multimedia/Images/2019/06/Heracles_on_the_Moon#:~:YH4ashbSR98-link

Programa un robot en la Luna

Parte II

Una vez que hemos llevado nuestro robot hasta la base Lunar, vamos a recoger muestras de la superficie de la Luna, de dos formas diferentes:

- Programando nuestro robot para que vaya de una muestra a otra, hasta que pase por todas.
- Programando nuestro robot para que siga una línea que pasa por todas las muestras que tenemos que buscar.

¿Cuál te parece más fácil?

Antes de empezar a programar nuestro robot en la Luna, vamos a entrenar dos conceptos básicos de programación: los bucles y la sentencia if/else.

¿Qué es un bucle?

Un bucle es una serie de instrucciones que se repite siempre o hasta que se cumple una determinada condición. Cuando terminan todas las instrucciones del bucle, se vuelve a empezar de nuevo. Vamos a ver un ejemplo de bucle con nuestro simulador.

Entra en el simulador Open Roberta Lab: <https://lab.open-roberta.org/> y añade un bloque de código para que el robot gire:



Dale al botón de **PLAY** en la simulación y observa qué ocurre.

El robot ha girado hacia la derecha 20° a una velocidad del 30 % y se ha parado porque ya no tiene más instrucciones.

Ahora vamos a añadir un bucle. Entra en la pestaña de **Control** y añade el bloque de programación **'Repetir indefinidamente'**.

Página 2 de 16

Figura 5.10. Guion del Alumnado Parte II. Actividades de Introducción.

Programa un robot en la Luna

Parte II



Coloca el bloque que hace girar al robot, dentro de este bucle:



De esta forma, el bloque girar, se va a repetir indefinidamente, es decir, el robot va a girar 20°, después otros 20°, después 20° y así siempre, hasta que paremos la simulación, porque no hay nada que detenga el bucle. Dale a play y comprueba cómo funciona el bucle.

Todo lo que haya dentro del bloque **'hacer indefinidamente'**, se repetirá una y otra vez hasta que paremos la simulación

Página 3 de 16

Programa un robot en la Luna

Parte II

¿Qué hace la sentencia if/else?

Una sentencia es una instrucción que definimos para programar nuestro robot. La sentencia if/else, en español si/sino consiste en dos condiciones:

- Si ocurre la condición A, nuestro robot hace una cosa
- Si no ocurre la condición A, nuestro robot hará otra cosa diferente

La sentencia **if/else**, se utiliza dentro del bloque **'Hacer indefinidamente'**. Si el bucle no está, el robot sólo comprobará que se cumple la condición una vez, y nosotros queremos que esté continuamente comprobando la condición.

Vamos a programar un ejemplo para ver el funcionamiento de este bloque de programación.

Queremos que nuestro robot siga hacia delante si su sensor de color marca blanco, y si el sensor no marca blanco, que gire hacia la derecha.

Añade el bloque de control **'Si/hacer/Sino'**



Página 4 de 16

Figura 5.11. Guion del Alumnado Parte II. Actividades de Introducción.

Por ejemplo, para introducir el funcionamiento de una sentencia if/else, en el guion se incluye la imagen de la Figura 5.12. A partir de este gráfico, se muestran los pasos a seguir para ir construyendo la programación paso a paso, incluyendo un gráfico por cada bloque que se añade a la programación.

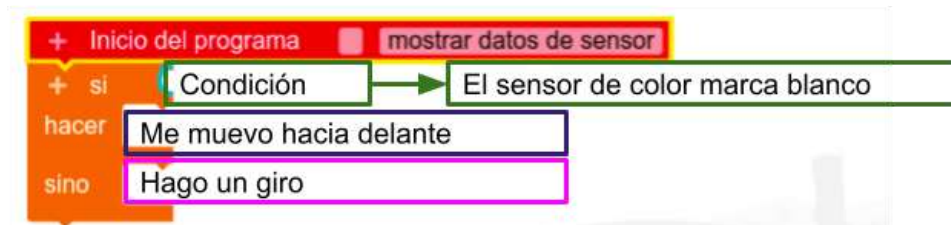


Figura 5.12. Guion Alumnado Parte II. Gráfico introducción Sentencia if/else.

Después de estas dos actividades de introducción y refuerzo, comienza la **Misión 02: Busquemos muestras en la superficie Lunar**. Para realizar esta misión, es necesario cargar la imagen correspondiente. En este caso, ya no aparecen las instrucciones, sólo se hace referencia a la carpeta de Drive, donde están alojadas las imágenes, y al proceso seguido anteriormente [Figura 5.13].

La imagen para esta simulación, está formada por la superficie Lunar, la base Lunar y cuatro puntos de color, que representan las muestras de la superficie Lunar que estamos buscando [Figura 5.13]. Esta misión consiste en programar el Rover, para que pase por todas las muestras, utilizando bloques de acción y el sensor de color. En este caso, se utiliza el bloque de programación ‘Esperar hasta’, descrito en la Parte I del guion.

En el guion, se muestran los pasos para que el robot siga una determinada trayectoria [Figura 5.13]. El Rover puede seguir cualquier trayectoria, siempre y cuando recoja todas las muestras de la superficie. La primera parte del reto, la cual consiste en llegar hasta la muestra roja, se realiza de forma totalmente guiada; mostrando paso a paso los bloques de programación necesarios. El proceso para llegar a las demás muestras es similar, por lo que cada vez, se muestran menos instrucciones de programación.

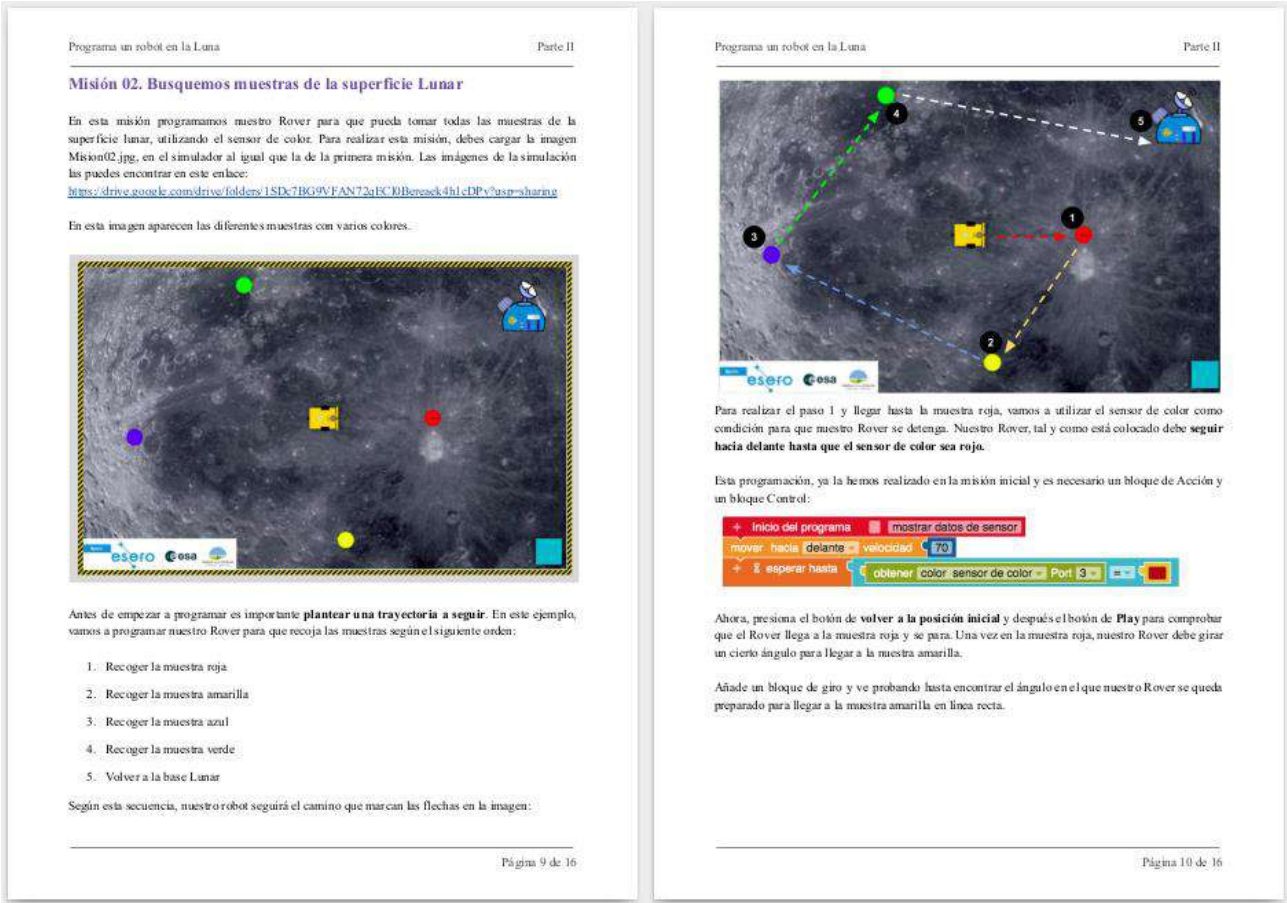


Figura 5.13. Guion del Alumnado Parte II. Misión 02.

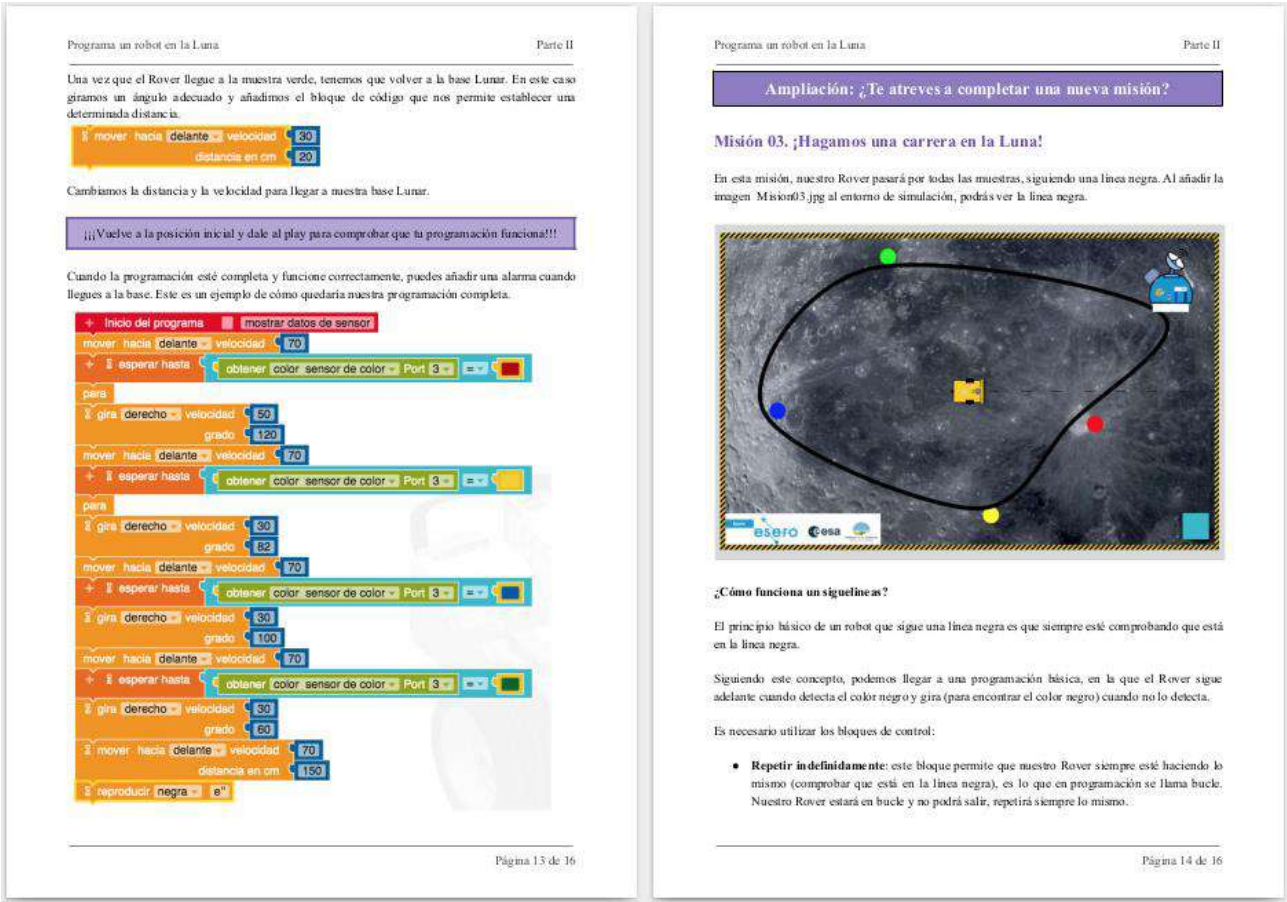


Figura 5.14. Guion del Alumnado Parte II. Misión 02 y Misión 03.

La **Misión 03: Hagamos una carrera en la Luna** está planteada como una ampliación. El nivel de complejidad que presenta es mayor y es necesario utilizar bucles y sentencias if/else. El objetivo es que el Rover Lunar, siga una línea negra que pasa por todas las muestras.

Para comenzar la Misión 03 se realiza una breve explicación del funcionamiento de un robot que sigue una línea [Figura 5.14, Figura 5.15], mostrando los bloques de programación necesarios para este cometido. En este caso, se muestra un ejemplo de programación básica, es decir, sólo el concepto de la programación de un robot siguelíneas: el robot debe seguir hacia delante si está en la línea negra y girar para encontrar la línea negra, si no está en ella. A partir de aquí, el guion ofrece breves instrucciones y pistas sobre cómo conseguir la programación.

Esta es la última Misión y la última actividad de la segunda parte del guion de prácticas. El guion completo de prácticas para el alumnado, se puede consultar en el Anexo I.

Programa un robot en la Luna

Parte II

- Si/hacer/Sino:** en este bloque, ponemos una condición y damos una instrucción en el caso de que se cumpla y una instrucción en el caso de que no se cumpla: **Si el robot está en la línea negra, entonces sigue adelante, si no está en la línea negra, giro a la izquierda para encontrar la línea.**

El bloque **Si/hacer/Sino**, se coloca dentro del bloque **Repetir indefinidamente**, de esta forma, nuestro Rover siempre está comprobando si está en la línea negra.

Pista: para establecer la condición de obtener el color del sensor de color y saber si es igual a negro, hay que utilizar un bloque de Lógica, un bloque de Sensores y un bloque de Colores.

Prueba esta programación. Coloca el robot directamente encima de la línea antes de comenzar la programación.

¿Cómo funciona?, ¿Es capaz el Rover de seguir la línea sin problemas?, ¿cuándo se desvía?, ¿Cómo podemos mejorar esta programación?

Esta programación no funciona del todo bien, el robot se va de la línea si hay curvas marcadas. Para mejorar esto, la clave está en los **giros**. Los bloques de giro que hemos añadido hasta ahora, no indican la rueda que gira, es decir, giran las dos ruedas delanteras a la vez.

Si queremos hacer el giro más suave, para que nuestro Rover no se desvíe, tenemos que utilizar este bloque de Acción:

En este bloque de programación, podemos establecer la velocidad de la rueda izquierda y de la rueda derecha.

¿Qué pasará cuando la rueda izquierda gira más rápido que la derecha? ¿Y cuando la derecha gira más rápido que la izquierda? Intenta mejorar la programación utilizando este bloque de programación en las dos condiciones del bloque **Si/hacer/Sino**.

Página 15 de 16

Figura 5.15. Guion del Alumnado Parte II. Misión 03

5.4.2 Guion para el profesorado

El guion para el profesorado, tiene la finalidad de mostrar la organización de las actividades, la temporización aproximada, el material necesario y demás aspectos útiles para su aplicación en clase. Además, presenta una serie de recomendaciones para utilizar las opciones disponibles de la plataforma, con el fin de aumentar la interactividad entre el alumnado y que las actividades se realicen con la mayor diversión y agilidad posible. Este guion incluye:

- Tabla resumen de actividades propuestas [Tabla 5.2]
- Objetivos de aprendizaje [Tabla 5.1]
- Rúbrica de Evaluación [Tabla 5.5]
- Propuesta de Planificación [Tabla 5.3]
- Soluciones propuestas a los retos presentados en los guiones para el alumnado

Además de toda esta información, en la primera página aparece una tabla resumen de los aspectos más generales de la actividad [Tabla 5.7], esencial para llevar a cabo las actividades en el aula.

Asignaturas de aplicación	TIC, TIC I, TIC II, Tecnología
Nivel Educativo Recomendado	Educación Secundaria Obligatoria y Bachillerato
Complejidad	Parte I: inicial-media Parte II: media-avanzada
Conocimientos Previos requeridos	Ninguno
Tiempo aproximado	Parte I: 60'' Parte II: 90''
Coste	Ninguno, se realiza a través de una herramienta online
Material necesario	Ordenadores o Tablets con conexión a internet

Tabla 5.7. Tabla resumen aspectos generales. Guion profesorado

Existe un apartado dedicado a mostrar herramientas adicionales para introducir el contexto en el aula de forma gráfica e interactiva:

- Videos de la Agencia Espacial Europea, que muestran simulaciones de las misiones espaciales a Luna; y en concreto de la Misión Heracles, dónde está constextualizada la actividad.
- Videos de la Funcación Airbus Space. Estos videos muestran cómo sería nuestra vida en la Luna y los retos a los que nos tendríamos que enfrentar, como son, la falta de agua potable, energía, alimentos, etc.

Todos estos videos, son visualmente atractivos y ayudarán a captar el interés en el alumando; así como a mantener la motivación durante el transcurso de la actividad.

Por último, se añade un apartado con instrucciones muy básicas para utilizar las opciones que ofrece la plataforma Open Roberta Lab, en cuanto a compartir programas, gestión y creación de grupos. En

el guion aparecen indicaciones para crear un grupo, compartir un programa y visualizar un programa creado por uno de los miembros del grupo. Estas opciones de la plataforma permiten al profesorado:

- Crear archivos compartidos con las soluciones a los retos y compartirlos con los miembros de un grupo
- Ver los programas que crean los miembros del grupo, con la hora y la fecha de creación

El guion completo para el profesorado está alojado en el Anexo II.

6 Ejemplos de Aplicación del material desarrollado

En este apartado, se presentan dos ejemplos de aplicación del material desarrollado: En la feria de las Ingenierías (abril 2021) y en el periodo de prácticas curriculares (mayo 2021).

6.1 Feria de las ingenierías

6.1.1 Antecedentes

La feria de las Ingenierías es un evento dentro del Proyecto 'Quiero Ser Ingeniera' promovido por el Vicerrectorado de Igualdad, Inclusión y Sostenibilidad, la Unidad de Cultura Científica y la Oficina de Software Libre de la Universidad de Granada. Dentro de este proyecto también se encuentra la actividad [Campus Tecnológico para chicas](#).

La [feria de las ingenierías](#) se celebró por primera vez en febrero de 2019, en las instalaciones de la facultad de Ciencias, la Escuela Técnica Superior de Caminos, Canales y puertos, la Escuela Técnica Superior de Informática y Telecomunicaciones y la Escuela Técnica Superior de Ingeniería de Edificación de la Universidad de Granada.



Figura 6.1. Cartel Feria de las Ingenierías 2019. I Edición.

El objetivo de esta feria es dar a conocer el mundo de la Ingeniería a estudiantes de Educación Secundaria y Bachillerato, y las diferentes titulaciones de Ingeniería a las que pueden optar. Este evento está dirigido especialmente al público femenino para mostrar que, las ingenierías son una salida profesional viable para todos los públicos.

Las actividades se llevan a cabo durante una semana (normalmente la semana del 11 de febrero, por el día de la Mujer y la Niña en la Ciencia). Estudiantes de todos los institutos de Andalucía visitan las principales facultades de ingeniería de la universidad de Granada, además de museos (incluido el Parque de las Ciencias) y/o instalaciones industriales. Este año, debido a la situación de Pandemia, esta feria se ha realizado en formato Online, durante la semana del 19 de abril.

Durante estos días se realizan talleres prácticos, charlas o conferencias, a cargo de investigadoras e investigadores de la UGR, empresas, entidades colaboradoras, estudiantes de último curso y/o egresados. En los últimos años se han impartido más de 50 talleres diferentes y conferencias, a más de 1.200 estudiantes de institutos de toda Andalucía.

Mi experiencia en la feria de las ingenierías comenzó cuando estaba terminando el grado, pues colaboré como voluntaria de organización en este evento en su primera edición de 2019.

En la [edición de 2020](#), tuve el honor de realizar la charla de bienvenida, en la facultad de Ciencias; ya desde mi puesto profesional, como miembro del departamento [ESERO Spain](#) (Oficina de Recursos Educativos de la Agencia Espacial Europea), cuya sede es el Parque de las Ciencias de Granada. Esta colaboración con la feria de las ingenierías está enmarcada dentro del Proyecto [Ellas Inspiran STEAM](#), de ESERO Spain.

6.1.2 Feria de las Ingenieras 2021

Este año, debido a la situación de pandemia, la feria de las ingenierías se ha llevado a cabo en formato totalmente online. Todos los talleres y actividades se grabaron previamente, para evitar problemas de conexión; quedando publicados en la [plataforma oficial](#).

Durante la semana del 19 al 23 de abril, se realizaron las conexiones en directo con los centros educativos. Este año, también he participado en esta feria como miembro del equipo ESERO Spain, con un taller relacionado con las carreras espaciales.

El guion que desarrollé para el alumnado asistente a este taller fue la primera versión del material desarrollado en este trabajo. El material proporcionado a los estudiantes para el taller, tanto el guion como el video, están publicados en la plataforma de la feria de las Ingenierías 2021, junto con el video correspondiente [Figura 6.2], preparado para que los centros que asistieron pudieran verlo previamente; ya que, la conexión en directo estaba limitada a una hora.



Figura 6.2. Página del Taller 'Programa un Robot de Exploración Lunar' en la plataforma oficial de la feria de las Ingenierías 2021.

Enlace directo: <https://educa.ugr.es/feriadelasingenierias/actividades/taller-programa-un-robot-de-exploracion-lunar/>

6.1.3 Realización del Taller

Los centros educativos interesados en atender a la sesión en directo de este taller, realizaron su inscripción durante la semana previa a la feria de las ingenierías. Los centros, alumnado y profesorado asistentes a la sesión en directo (online) realizada el día 21 de abril, aparecen en la Tabla 6.1.

La forma en la que los centros se conectaron a la sesión, fue muy diferente, dependiendo de la logística y el plan de asistencia de cada centro.

Por un lado, algunos centros se conectaron de forma colectiva desde sus aulas. Por otro lado, en algunos grupos el alumnado se conectó de forma individual para seguir la sesión desde sus casas. A su vez, debido al sistema semipresencial, en algunos centros la mitad de la clase estaba de forma colectiva desde el aula y la otra mitad de forma individual.

Centro	Curso	Alumnado	Profesorado
IES Cristo de la Yedra	3º E.S.O.	90	3
IES Mariana Pineda	3º E.S.O., 1º y 2º Bach.	30	2
IES Juan XXIII Cartuja	1º de Bach.	20	1
IES VALLE DE LECRÍN	4º E.S.O., 1º Bach.	30	2
IES "La Zafra"	1º y 2º de Bachillerato	100	4
Colegio Internacional de Granada	3º ESO	21	1
TOTAL		291	13

Tabla 6.1. Asistentes Taller Online 'Programa tu Robot de Exploración Lunar' 21/04/2021.

La conexión en directo se llevó a cabo durante el día 21 de abril, desde las 10:00 h hasta las 11:00 h.



Figura 6.3. Taller Feria de las Ingenierías. Conexión en directo. 21/04/2021



Figura 6.4. Taller Feria de las Ingenierías. Conexión en directo. 21/04/2021

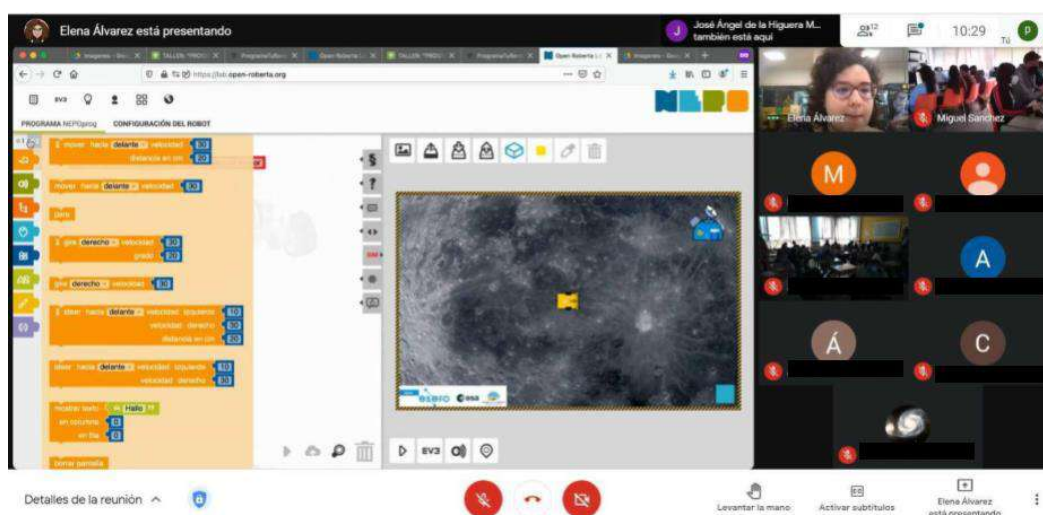


Figura 6.5. Taller Feria de las Ingenierías. Conexión en directo. 21/04/2021.

6.2 Periodo de Prácticas

Tuve la oportunidad de poner en práctica de nuevo el material desarrollado durante la realización de las prácticas curriculares en centros educativos. El material realizado para esta feria de las ingenierías quedó guardado en la plataforma online, tanto el guion de prácticas como el vídeo. Debido a esto, propuse a mi tutora profesional de prácticas realizar la actividad durante la fase de actuación de las prácticas curriculares en centros educativos.

La clase con la que he realizado las prácticas es el 1º Ciclo del Grado medio de Formación profesional en sistemas microinformáticos y redes (SMIR), en horario nocturno. Durante el primer curso de este grado profesional no se imparte programación como tal. Es en el segundo curso cuando se imparte el temario de programación.

Este nivel inicial del alumnado en programación, unido al tiempo que quedó libre tras terminar el temario curricular, hizo que aplicar esta actividad, fuera una oportunidad de volver a ponerla en práctica y motivar al alumnado, mostrándoles el potencial de las carreras espaciales como salida profesional; ya que, en todas las empresas de base tecnológica es necesario un perfil profesional técnico.

6.2.1 Realización de la Actividad

La actividad se llevó a cabo el día 26 de mayo, durante las dos horas de clase de Aplicaciones Ofimáticas. En este caso, se realizó de forma presencial, y con la estructura de guion desarrollada y presentada en este proyecto; dividido en dos partes, con actividades de repaso y refuerzo de bucles y sentencias.

En un principio, estaba planteado para realizarse de forma semipresencial, con alumnado en clase y con alumnado conectado a través de videoconferencia. Sin embargo, debido al cambio de restricciones, a partir del lunes 24, todo el alumnado asistió de forma presencial a clase.

Centro	Curso	Alumnado	Profesorado
IES Padre Suarez	1º Ciclo SMIR	10	1
TOTAL		10	1

Tabla 6.2. Tabla 6.1. Asistentes Taller 'Programa tu Robot en la Luna' 26/05/2021.

El grupo mostró mucho interés y motivación por completar las misiones. Mi expectativa era realizar sólo la primera parte del guion, pero lo completaron muy rápidamente y dio tiempo suficiente para que casi todos terminaran la Misión 02 y más de la mitad, la misión 03.



Figura 6.6. Aula 1º Ciclo SMIR. IES Padre Suarez

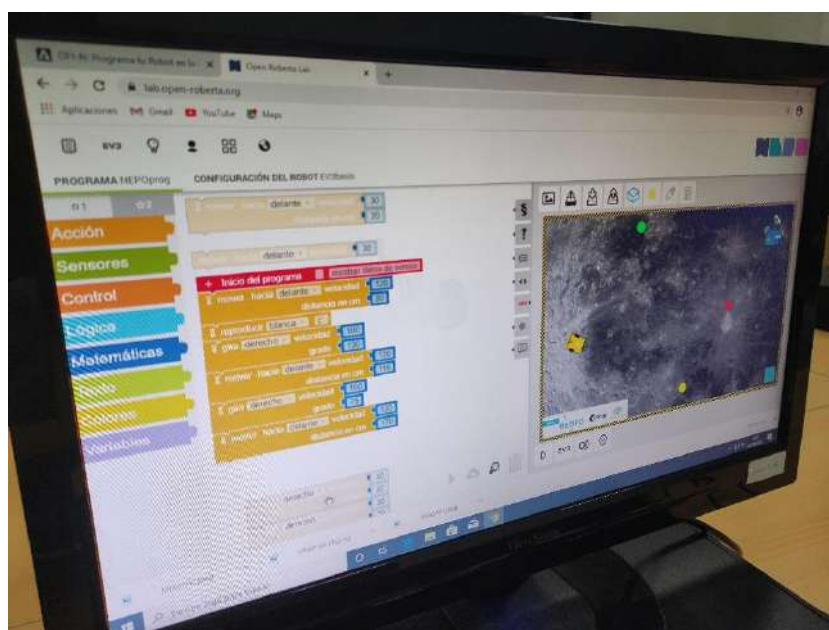


Figura 6.7. Programa tu Robot en la Luna. IES Padre Suarez. 26/05/2021

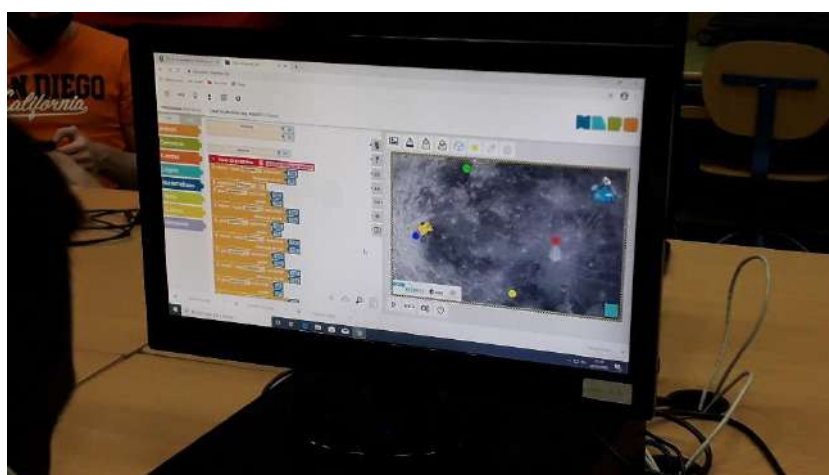


Figura 6.8. Programa tu Robot en la Luna. IES Padre Suarez. 26/05/2021

6.3 Resultados

Después de realizar cada taller, los asistentes, tanto profesorado como alumnado rellenaron una encuesta con algunas preguntas simples sobre el desarrollo de la actividad. En la Tabla 6.1, aparece un resumen de las características de cada aplicación.

	Feria de las Ingenierías	IES Padre Suarez
Alunando	291	10
Tiempo	1 hora	2 horas
Formato	Online	Presencial
Participación Encuestas	15.46 %	100 %

Tabla 6.3. Resumen Ejemplos de aplicación.

La encuesta, a contestar por el alumnado de cada grupo, cuenta con preguntas simples repartidas en dos partes, tal y como aparece en la Tabla 6.4.

	Preguntas	Opciones de respuesta
Primera Parte	Después de este taller, ¿ves más interesantes las carreras espaciales?	SI, NO, N/C
	¿Te gustaría trabajar en la Agencia Espacial Europea?	SI, NO, N/C
Segunda Parte	¿Has completado la Misión 01?	SI, NO
	¿Has completado la Misión 02?	SI, NO
	¿Has completado la Misión 03?	SI, NO

Tabla 6.4. Preguntas encuesta alumnado.

Los grupos y condiciones de aplicación han sido muy diferentes. Sin embargo, en las primeras preguntas realizadas al alumnado, los resultados obtenidos son muy similares. En las Figura 6.9 y Figura 6.10, se puede observar, que en ambos casos el interés por las carreras espaciales y la industria espacial creció significativamente.

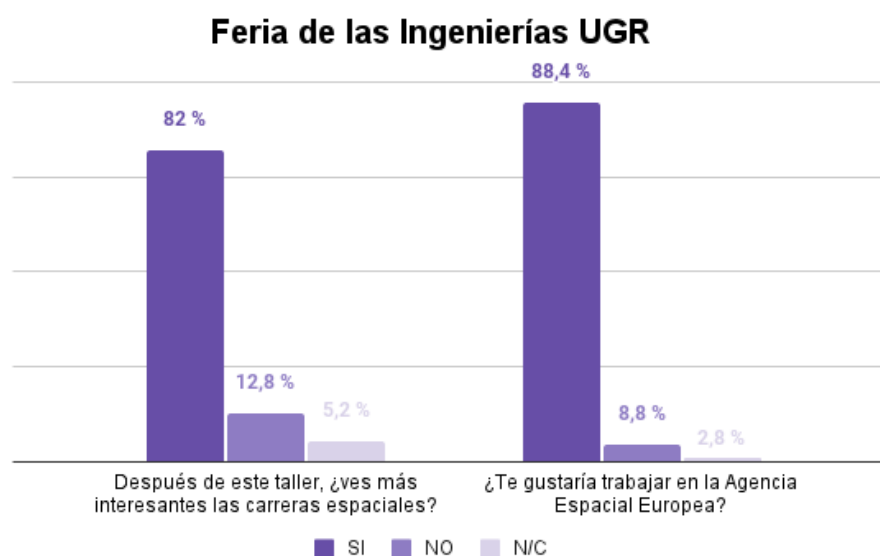


Figura 6.9. Resultados Encuesta Feria de las Ingenierías 2021. Primera parte

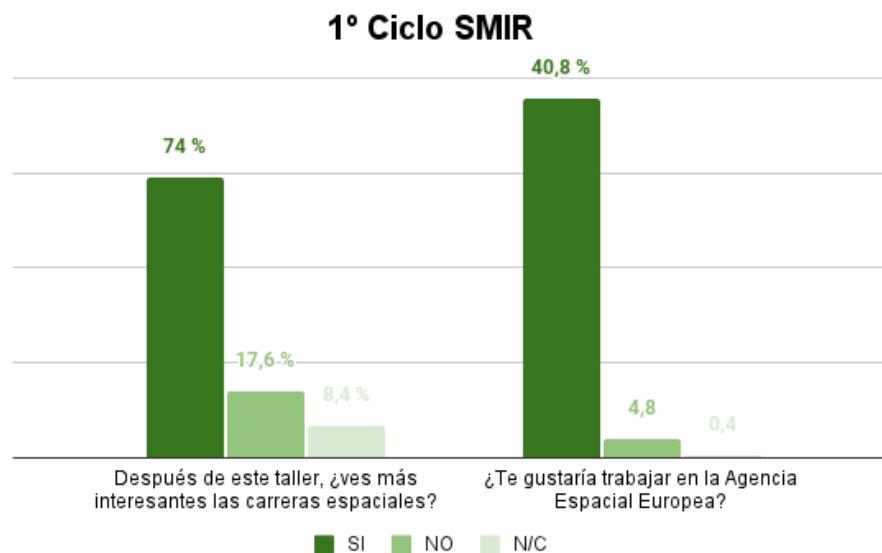


Figura 6.10. Resultados Encuesta 1º Ciclo SMIR. Primera parte

El objetivo de la segunda parte de la encuesta es obtener el porcentaje de alumnado que ha conseguido completar cada misión. Según los resultados [Figura 6.11], prácticamente todo el alumnado consiguió realizar la primera Misión planteada. Sin embargo, los porcentajes caen en las dos misiones siguientes. En la feria de las ingenierías, sólo del 31,5 % consiguieron completar la misión 03. En el caso del 1º Ciclo de SMIR, el 70 % realizaron la última misión.

Esta diferencia, se debe a varios factores. En primer lugar, en el ciclo de SMIR, había más tiempo y menos alumnado, lo que permitió un seguimiento continuo de la actividad, dando soporte a los alumnos que mostraban un menor ritmo de aprendizaje. En segundo lugar, en el guion utilizado por el segundo grupo, ya estaban incluidas las actividades de refuerzo relacionadas con bucles y sentencias, necesarias para completar la Misión 03.



Figura 6.11. Resultados Encuesta. Segunda parte

Los datos recogidos en la encuesta para el profesorado, corresponden al grupo de docentes que asistieron al taller, junto con su alumnado, en la feria de las ingenierías. En este caso, el porcentaje

de participación es del 92.6 %. Esta encuesta sólo consistió en dos preguntas: ¿Crees que esta actividad se ajusta al currículum de 1º de Bachillerato? y ¿utilizarías esta actividad en clase?

Los resultados son muy favorables [Figura 6.12] y la mayoría coincide en que la actividad se ajusta bien a los contenidos y es atractiva para aplicarla en el aula.

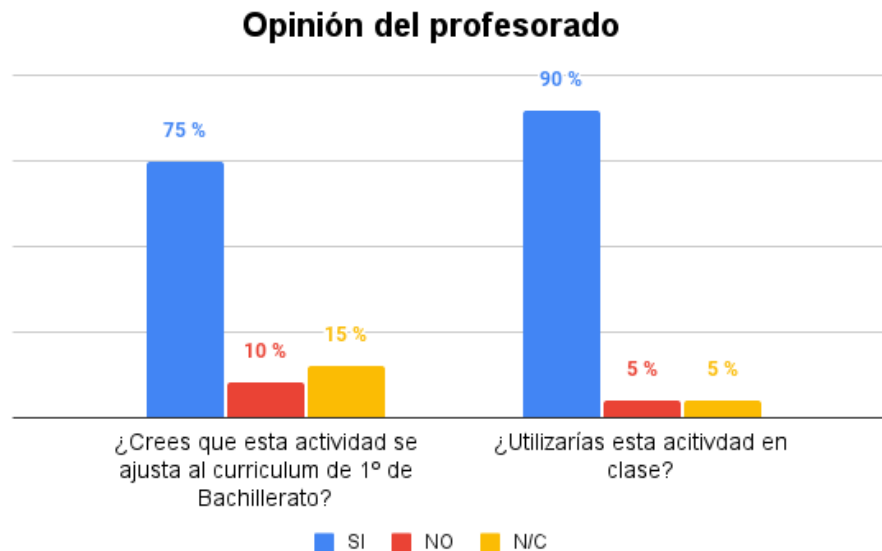


Figura 6.12. Resultados Encuesta Profesorado. Feria de las Ingenierías

6.4 Lecciones Aprendidas

El taller realizado para la feria de las ingenierías se llevó a cabo sin ningún problema y los resultados obtenidos son bastante buenos. La claridad del guion y las explicaciones paso a paso, ayudaron a los estudiantes a completar la Misión 01 y más del 50 %, consiguió completar la Misión 02. Sin embargo, el porcentaje de alumnado que completó la Misión 03 fue muy bajo. Una de las razones por las que sucedió esta caída de porcentaje, es que, el guion utilizado en la feria era muy denso para llevarlo a cabo en una hora, pues condensaba las tres misiones.

A raíz de esta experiencia, separé el guion del alumnado en dos partes. De esta forma, permite incluir actividades de repaso y/o refuerzo, muy útiles para programar la Misión 03.

Además, separando el guion en dos partes, hay más tiempo para introducir el contexto y asegurar que el alumnado maneja bien la plataforma de simulación y programación. Impartir este taller, fue clave para descubrir la adaptación que necesitaban las actividades para aplicarlas en el aula.

Después de la feria de las ingenierías, hubo una actualización de la plataforma Open Roberta Lab, y, por tanto, fue necesario hacer actualizaciones en todos los gráficos utilizados en las explicaciones, para que estuvieran en consonancia con la interfaz gráfica de la plataforma.

La aplicación en el 1º Ciclo de grado medio de SMIR, fue una experiencia totalmente diferente. La presencialidad y el grupo tan reducido que asistió a clase, lograron un clima más interactivo y colaborativo. La mayoría completó las misiones sin problema, mostrando buena actitud, interés y

curiosidad por las opciones que ofrece la plataforma para simular y programar al robot. En este caso, se puso de manifiesto que la plataforma permite muchas opciones de ampliación y flexibilidad de funcionamiento, para ajustarse a los diferentes ritmos de aprendizaje.

7 Conclusiones

El material desarrollado en este trabajo se caracteriza principalmente porque utiliza un contexto atractivo para la población más joven y una plataforma interactiva. Permite por un lado realizar una programación por bloques y por otro, cargar la programación en un robot virtual, visualizando los movimientos sin necesidad de tener un sistema real.

El contexto, basado en las misiones espaciales a la Luna, resulta atractivo y misterioso, captando la atención del alumnado. Este contexto también cumple la finalidad de mostrar una aplicación de la programación, ya que, controlar un robot de forma remota, es una misión real de la Agencia Espacial Europea. Contextualizar actividades dentro este entorno, el cual, desde fuera parece ‘difícil’ e incluso ‘inalcanzable’ es una forma de cambiar esta percepción y mostrar que nada es imposible. A parte del contexto, la plataforma utilizada en el desarrollo de las actividades también es clave.

Actualmente, existe una gran cantidad de herramientas online para programar, sin necesidad de instalar nada en los equipos. La pandemia ha hecho que muchas de ellas, antes desconocidas, salieran a la luz. Es el caso de la plataforma Open Roberta Lab creada en 2002, para disminuir la brecha digital. Ha tenido una popularidad muy baja hasta ahora. Esta plataforma incluye, aparte de programar mediante un lenguaje basado en bloques, simular el comportamiento de un robot. Este hecho es crucial en el sistema educativo online y semipresencial al que nos hemos enfrentado este último año. Además, Open Roberta Lab funciona en el navegador, independientemente del sistema operativo utilizado y sus prestaciones, lo que aumenta significativamente su accesibilidad.

Estas dos características, el contexto y la plataforma, unidas al aspecto gráfico y visual del guion de prácticas, ha hecho que el trabajo realizado cumpla todos los objetivos expuestos al principio y supere con creces mis expectativas.

Las dos experiencias de aplicación han sido claves para mejorar las primeras versiones de los guiones. Llevarlo a la práctica, en dos contextos totalmente diferentes, ha logrado que el aprendizaje sea mayor, y por tanto también las mejoras incluidas en los guiones gracias a estas experiencias. Como es habitual en este tipo de experiencias, aspectos que funcionan bien en la teoría, no lo hacen en la práctica.

El objetivo principal de la feria de las ingenierías es despertar vocaciones científicas en el público femenino. Creo que cumplí ese objetivo con mi taller, pues al final, pregunté cuántas chicas querían trabajar en la Agencia Espacial Europea y la respuesta fue bastante buena. Lamentablemente, en el 1º ciclo de SMIR, no pude cumplir este objetivo, porque sólo hay una chica matriculada, y ese día no asistió. Desde mi punto de vista y experiencia propia, lograr que las chicas pierdan el ‘miedo’ a las carreras técnicas es una labor muy difícil, y a la vez muy necesaria.

El mundo en el que vivimos está rodeado de programación, desde una simple alarma de horno, hasta los equipos de automatización industrial de las plantas de producción. Es necesario mostrarles a las

nuevas generaciones que la programación es la base del mundo tecnológico en el que viven y que, al ser un invento del ser humano, pueden dominarla sin problemas.

Por último, me gustaría señalar que realizar este máster ha sido una gran experiencia de aprendizaje. La educación es la base de una sociedad y la labor del docente, es una de las más importantes. Conocer el mundo que nos rodea, sus características y posibilidades es lo que nos hace libres de elegir el camino que queramos. Con el guion de prácticas que presento en este trabajo, quiero mostrar al alumnado que nada es inalcanzable y que los límites sólo están en nuestra mente, porque cualquiera puede programar.

Bibliografía

- [1] «Learning to program intuitively in the Open Roberta Lab - Roberta». <https://www.roberta-home.de/en/lab/> (accedido may 30, 2021).
- [2] M. Guzdial, «Does contextualized computing education help?», *ACM Inroads*, vol. 1, n.º 4, pp. 4-6, dic. 2010, doi: 10.1145/1869746.1869747.
- [3] D. Weintrop y U. Wilensky, «Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms», *ACM Trans. Comput. Educ.*, vol. 18, n.º 1, pp. 1-25, dic. 2017, doi: 10.1145/3089799.
- [4] J. Scott y A. Bundy, «Creating a new generation of computational thinkers», *Commun. ACM*, vol. 58, n.º 12, pp. 37-40, nov. 2015, doi: 10.1145/2791290.
- [5] C. B. Santos, D. J. Ferreira, M. C. Borim do Nascimento Rodrigues de Souza, y A. Rodrigues Martins, «Robotics and programming: Attracting girls to technology», en *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Jaipur, sep. 2016, pp. 2052-2056. doi: 10.1109/ICACCI.2016.7732353.
- [6] S. Magnenat, J. Shin, F. Riedo, R. Siegwart, y M. Ben-Ari, «Teaching a core CS concept through robotics», en *Proceedings of the 2014 conference on Innovation & technology in computer science education - ITiCSE '14*, Uppsala, Sweden, 2014, pp. 315-320. doi: 10.1145/2591708.2591714.
- [7] A. Alvarez y M. Larranaga, «Using LEGO mindstorms to engage students on algorithm design», en *2013 IEEE Frontiers in Education Conference (FIE)*, Oklahoma City, OK, USA, oct. 2013, pp. 1346-1351. doi: 10.1109/FIE.2013.6685052.
- [8] V. Potkonjak *et al.*, «Virtual laboratories for education in science, technology, and engineering: A review», *Comput. Educ.*, vol. 95, pp. 309-327, abr. 2016, doi: 10.1016/j.compedu.2016.02.002.
- [9] Boletín Oficial del Estado, «Real Decreto 1105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y del Bachillerato.»
- [10] Boletín Oficial Junta de Andalucía, «Decreto 110/2016, de 14 de junio, por el que se establece la ordenación y el currículo del Bachillerato en la Comunidad Autónoma de Andalucía.»

- [11] Boletín Oficial Junta de Andalucía, «Orden de 14 de julio de 2016, por la que se desarrolla el currículo correspondiente al Bachillerato en la Comunidad Autónoma de Andalucía, se regulan determinados aspectos de la atención a la diversidad y se establece la ordenación de la evaluación del proceso de aprendizaje del alumnado.»

- [12] Fraunhofer IAIS Confluence, «Open Roberta Wiki - Open Roberta Wiki». <https://jira.iais.fraunhofer.de/wiki/display/ORInfo/Open+Roberta+Wiki> (accedido may 30, 2021).

- [13] Fraunhofer Institute for Intelligent Analysis and Information Systems, «ROBERTA – LEARNING WITH ROBOTS».

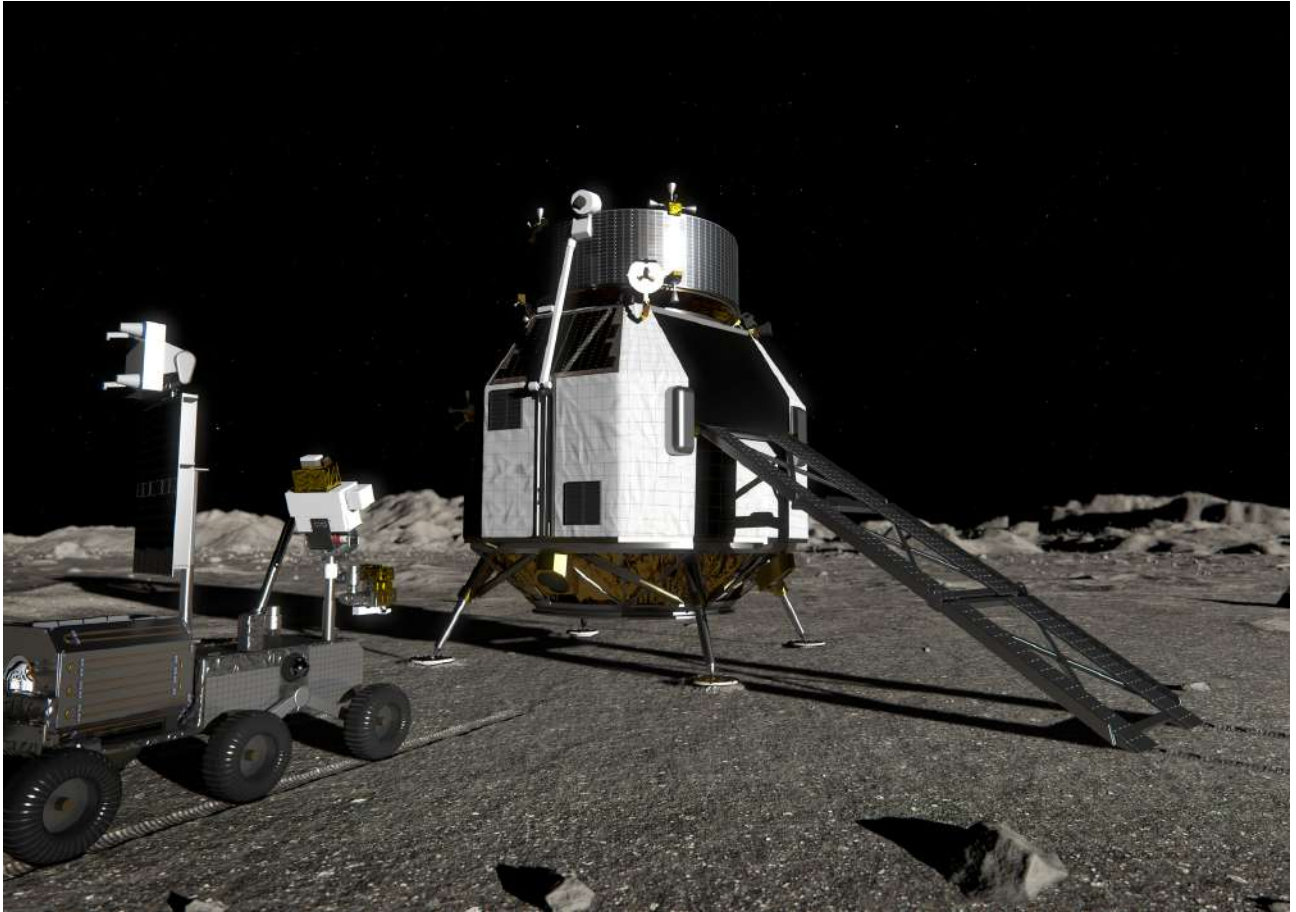
- [14] ESERO Spain, ESA, «Misión en la Luna. Planifica una misión en la Luna para un compañero de clase. Cuaderno del Profesorado.»

Anexo I

En este anexo se adjunta el guion desarrollado para que el alumnado realice las actividades planteadas.

Programa un robot en la Luna

Parte I

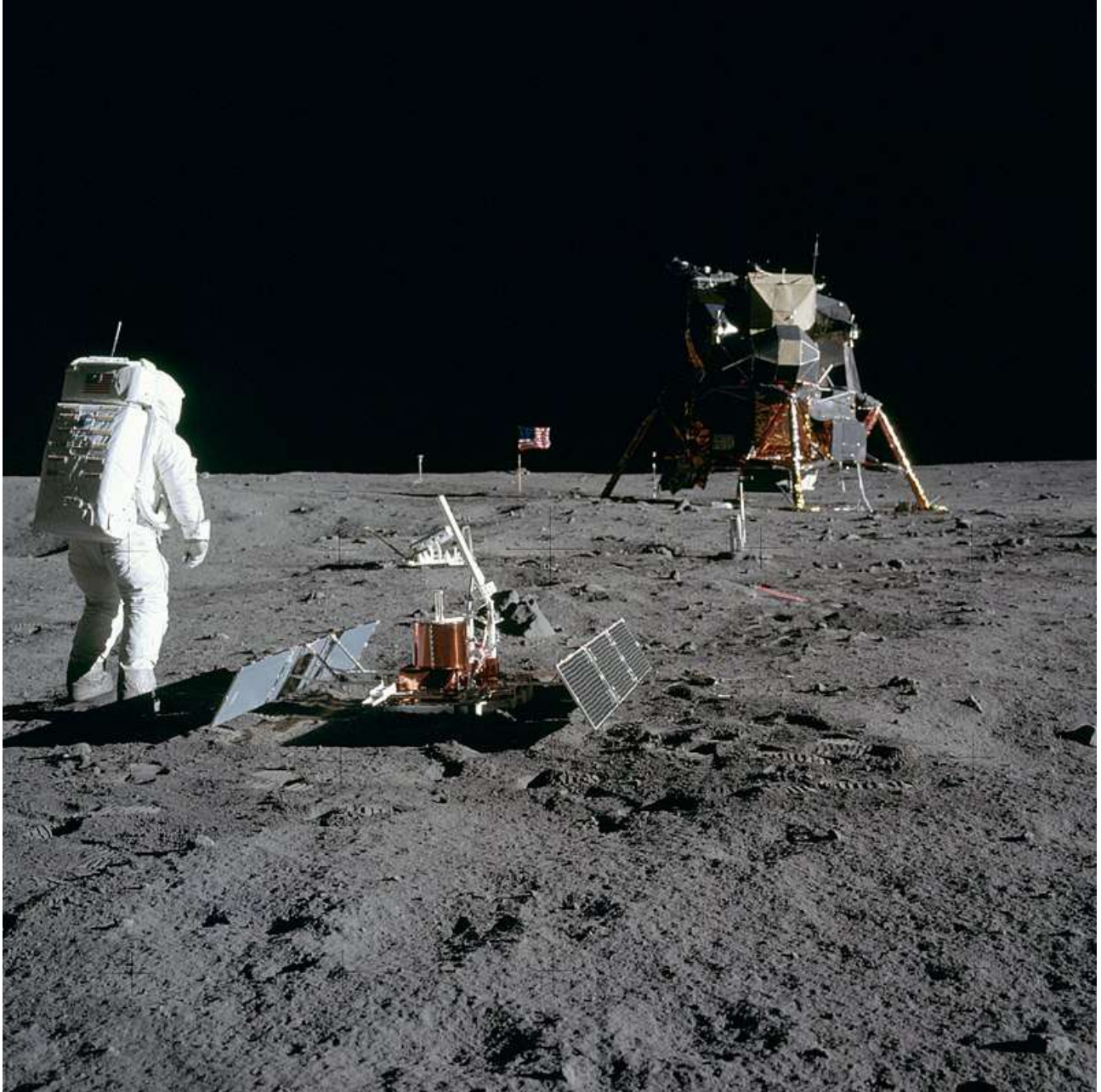


Heracles on the Moon:

https://www.esa.int/ESA_Multimedia/Images/2019/06/Heracles_on_the_Moon#.YHsashbSB98.link

Contexto

La primera vez que la humanidad llegó a la Luna fue en 1969, en la misión Apollo 11 de la NASA. Desde entonces, sólo 12 astronautas han pisado la superficie Lunar. El regreso a la Luna es uno de los próximos pasos de la exploración espacial, que nos permitirá saber más sobre el universo.



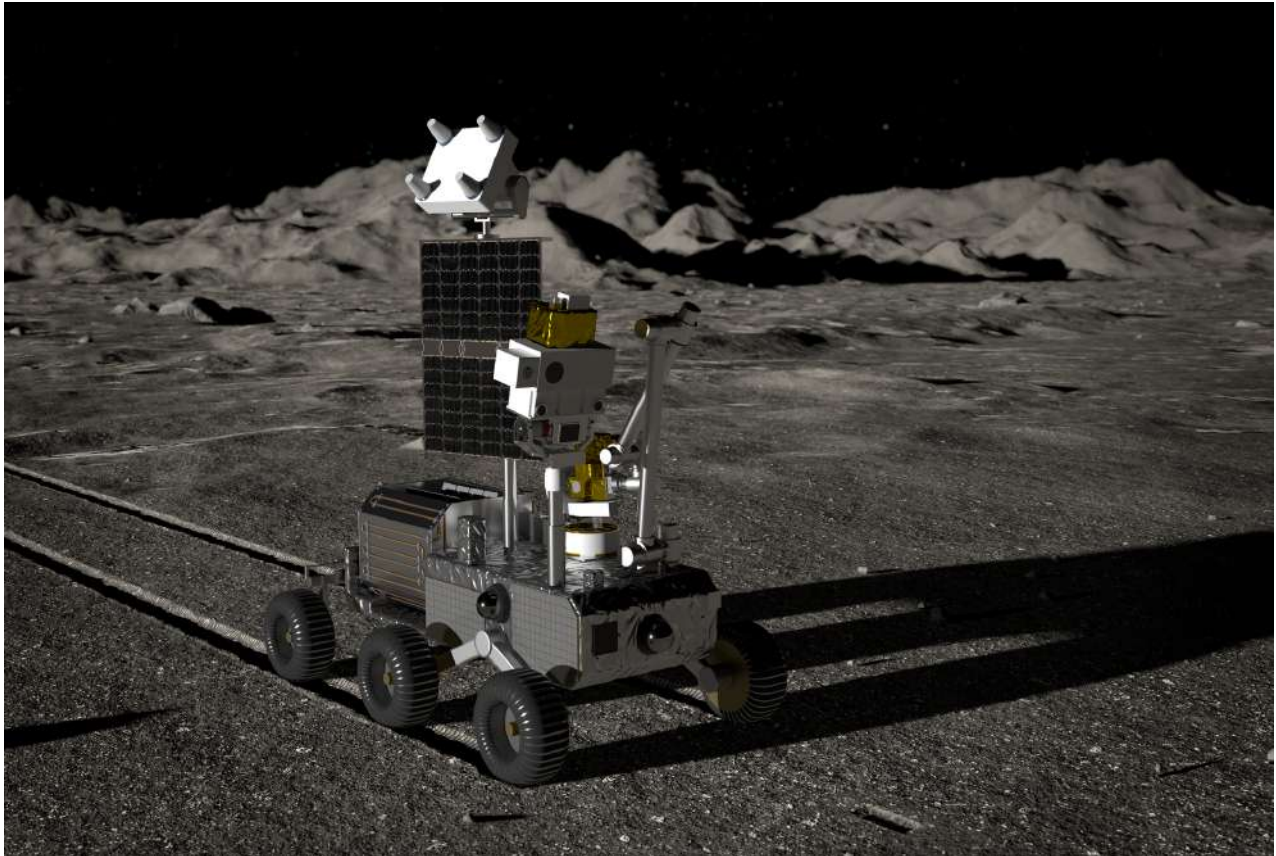
De NASA Neil A. Armstrong - Great Images in NASA Description, Dominio público,
<https://commons.wikimedia.org/w/index.php?curid=6449728>

El objetivo de las próximas misiones a la Luna es realizar experimentos y tomar muestras de su superficie. Esta misión resulta muy arriesgada para cualquier astronauta, pero hoy en día contamos con robots teledirigidos que pueden realizar estas tareas por nosotros.

La primera misión planeada para aterrizar en la Luna y traer muestras a la Tierra es Heracles, una misión conjunta entre la Agencia Espacial Europea y las agencias espaciales de Canadá y Japón. En

esta misión, se controlará de forma remota un robot todoterreno, que recorrerá la superficie Lunar para recoger diferentes muestras.

Programar estos robots permite explorar un espacio desconocido sin poner en riesgo la vida de los astronautas. Los todoterrenos espaciales también usan diversos sensores para escanear y cartografiar el terreno circundante de manera que puedan moverse de forma autónoma.



Heracles Rover:

https://www.esa.int/ESA_Multimedia/Images/2019/06/Heracles_rover#.YHsbMfBH0bA.link

En esta práctica, vamos a programar un Rover que recorre la superficie Lunar en busca de muestras. Utilizaremos un simulador, en el que nuestro Rover podrá moverse por una superficie Lunar simulada y recoger muestras gracias al sensor de color.

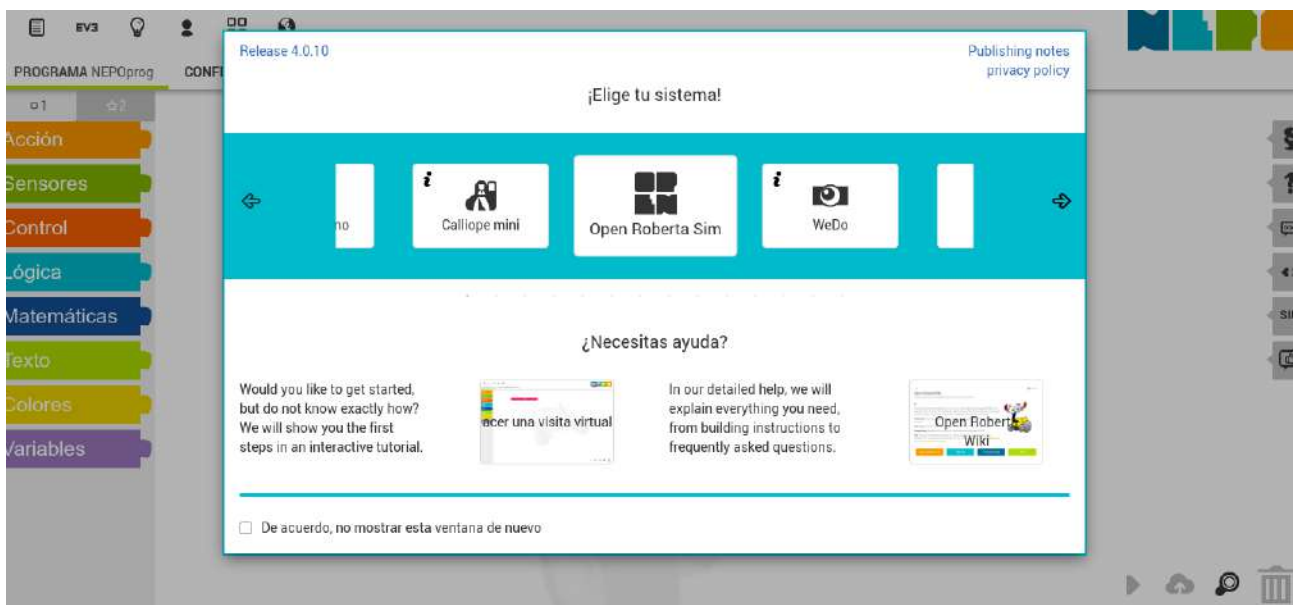
¿Cómo funciona nuestro simulador?

El simulador que vamos a utilizar es Open Roberta Lab. Es un simulador que se puede utilizar de forma gratuita desde el navegador, sin necesidad de instalar nada en el ordenador.

Para entrar, sólo tenemos que ir a este enlace: <https://lab.open-roberta.org/>

¡Ve ahora al navegador y visita Open Roberta Lab!

Al entrar nos pregunta cuál es nuestro sistema, tenemos que seleccionar **OPEN ROBERTA SIM**



Una vez que seleccionamos nuestro sistema, entramos a la página principal del simulador. En las imágenes que hay en la página siguiente, aparecen indicadas las partes que vamos a utilizar de este simulador:

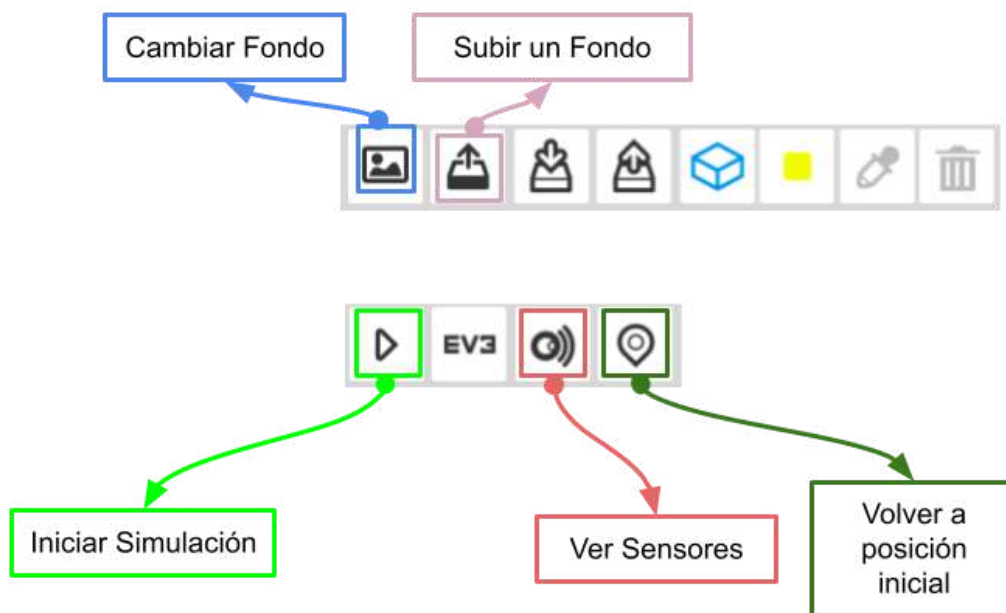
- Bloques de programación
- Zona de programación
- Simulador
- Botones de control de simulación



Para abrir el simulador, tenemos que hacer click en la pestaña **SIM** de la derecha.



En la ventana de simulación podemos ver nuestro robot con una imagen de fondo y varias opciones de control en la parte de abajo y en la parte de arriba. Aquí aparecen indicadas las que vamos a utilizar:



- **Iniciar Simulación:** este botón nos permite empezar con la simulación.
- **Cambiar fondo:** haciendo clic en este botón, puedes cambiar de fondo de simulación y elegir entre uno de los que hay por defecto en el simulador.
- **Ver sensores:** cuando hacemos clic aquí, aparece una nueva ventana con el valor de nuestros sensores en cada momento. Nosotros sólo vamos a utilizar el sensor de color para nuestra misión.
- **Volver a la posición inicial:** esta opción es muy útil, nos permite volver al estado inicial. La utilizaremos para volver a empezar de nuevo en nuestras pruebas.
- **Subir un fondo:** con este botón podemos incluir un fondo de simulación nuevo y personalizado y es lo que nos va a permitir añadir nuestro fondo de la superficie de la luna, dónde tiene lugar nuestra misión.

Para comenzar y saber cómo funciona nuestro entorno de programación, vamos a realizar una Misión inicial.

Misión Inicial. Nuestro Simulador

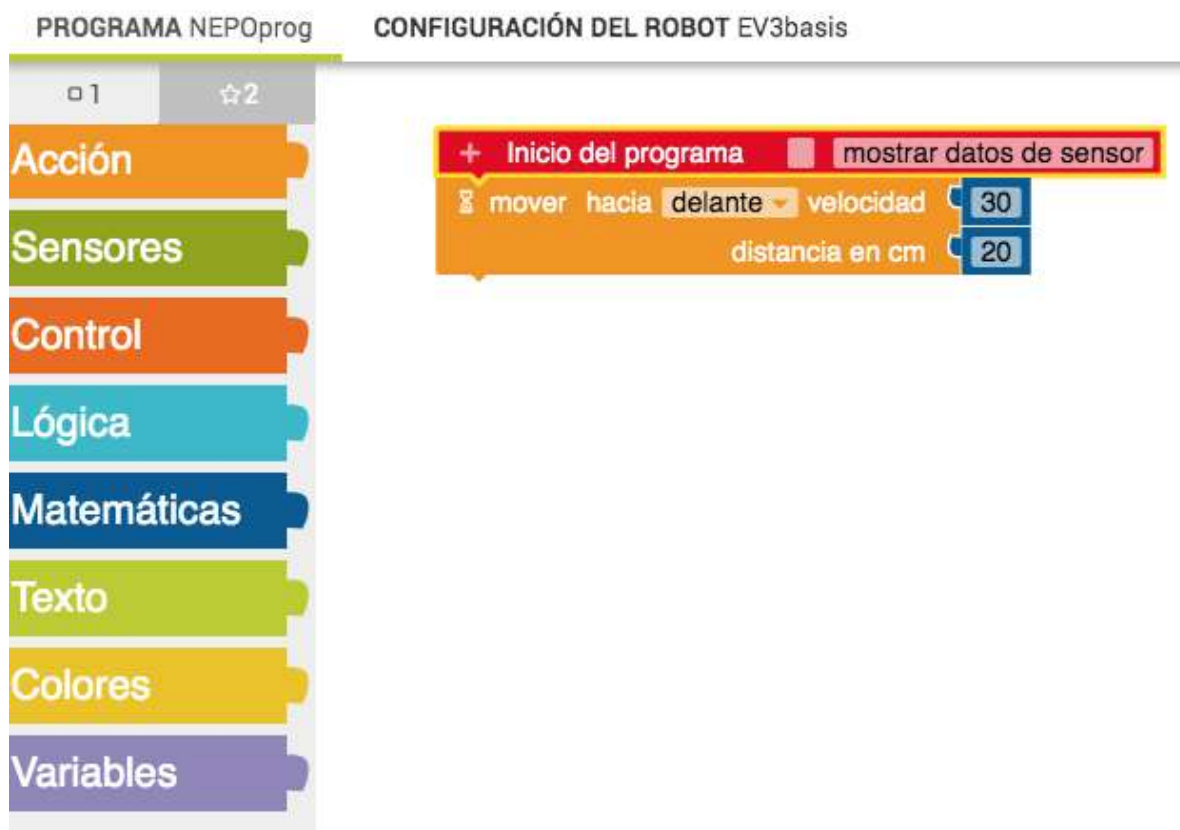
La misión inicial es un entrenamiento para conocer mejor cómo programar nuestro robot dentro de nuestro simulador. Sigue los siguientes pasos para completar la misión.

Paso 1: Añade el bloque de código '**Mover hacia delante con una velocidad y una distancia**' a la zona de programación. Este bloque lo encontramos en la pestaña de Acción, de color naranja.

Tal y como aparece en la siguiente imagen, haz clic en la pestaña de Acción y selecciona el bloque. Debes mantener pulsado el ratón encima del bloque para poder arrastrarlo al entorno de programación y encajarlo debajo del bloque rojo de inicio de programa.

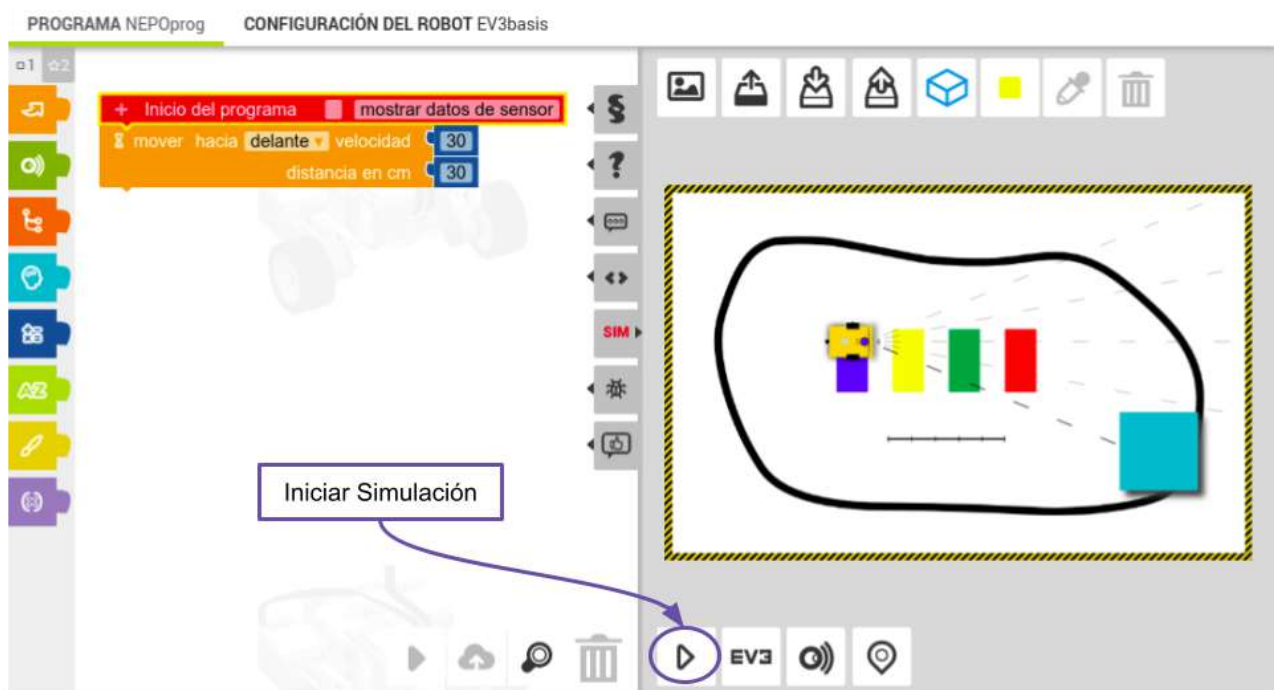


Una vez encajes el nuevo bloque debajo del bloque de inicio de programa, tu entorno de programación quedará como el de la siguiente imagen:



Paso 2: Abrir la pestaña de simulación y hacer clic en el botón **Iniciar Simulación**. La simulación comenzará en el primer fondo por defecto.





¿Puedes ver a tu robot moverse? ¿Va muy lento?

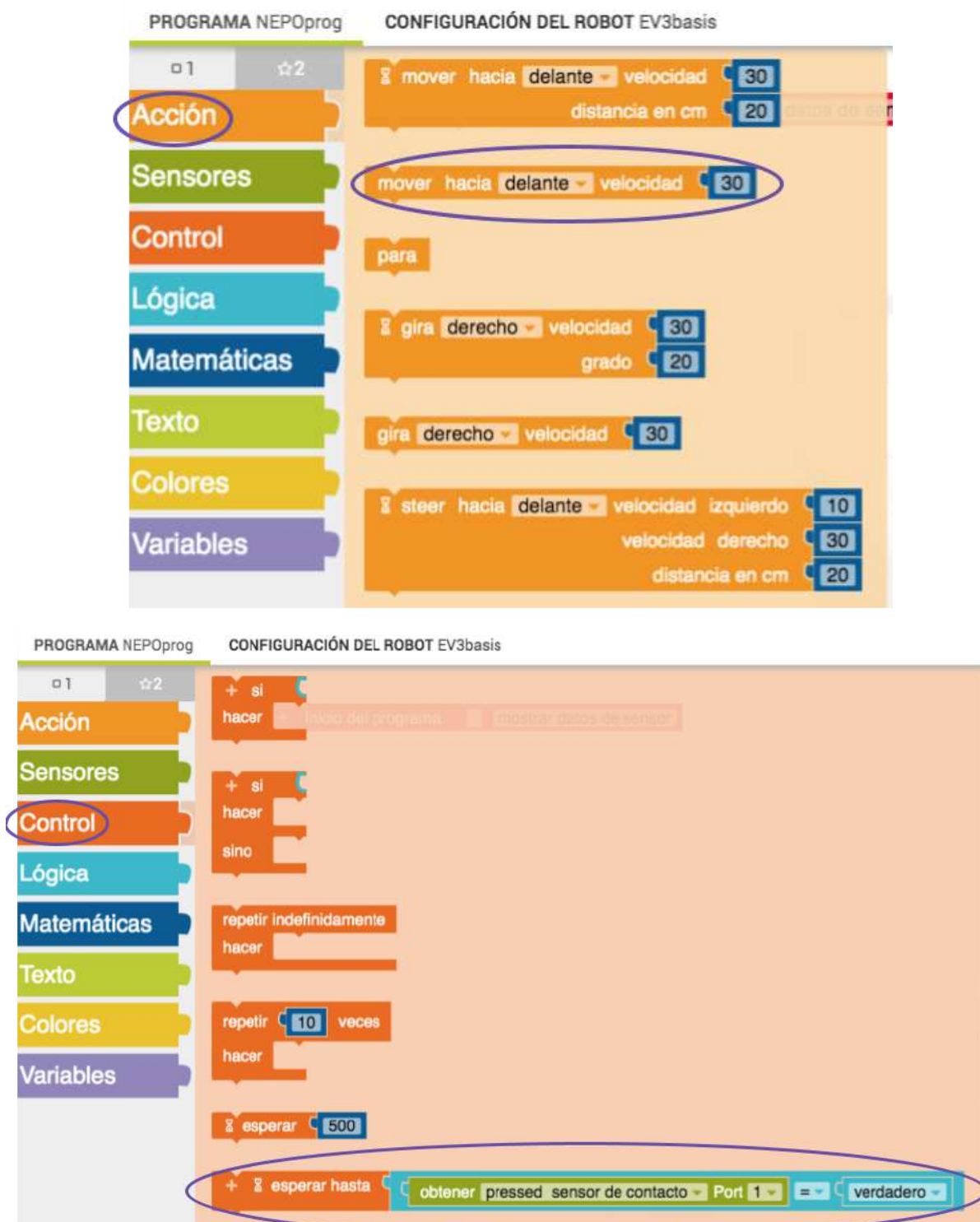
Esta instrucción le está diciendo al robot que se mueva hacia delante, con una velocidad de 30 % a una distancia de 20 cm. Si cambiamos los números de velocidad y distancia podemos controlar estas variables.

Modifica el programa para que el robot se mueva a una velocidad del 50 %, una distancia de 100 cm. ¿Se mueve más rápido? ¿Qué pasa si cambias la velocidad al 100 %?

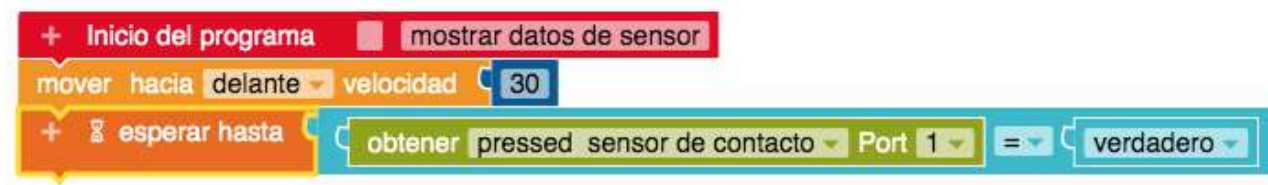
Ahora vamos a cambiar un poco el código para comprobar que nuestro sensor de color funciona correctamente.

¿Qué es un sensor de color? Es un aparato con el que nuestro robot puede saber de qué color es el suelo.

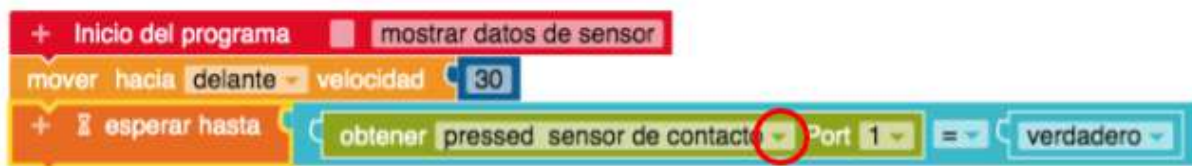
Paso 3: Cambia el bloque anterior por el de simplemente '**mover hacia delante velocidad**'. Busca en el apartado de control el bloque '**esperar hasta**' y añádelo a la zona de programación.



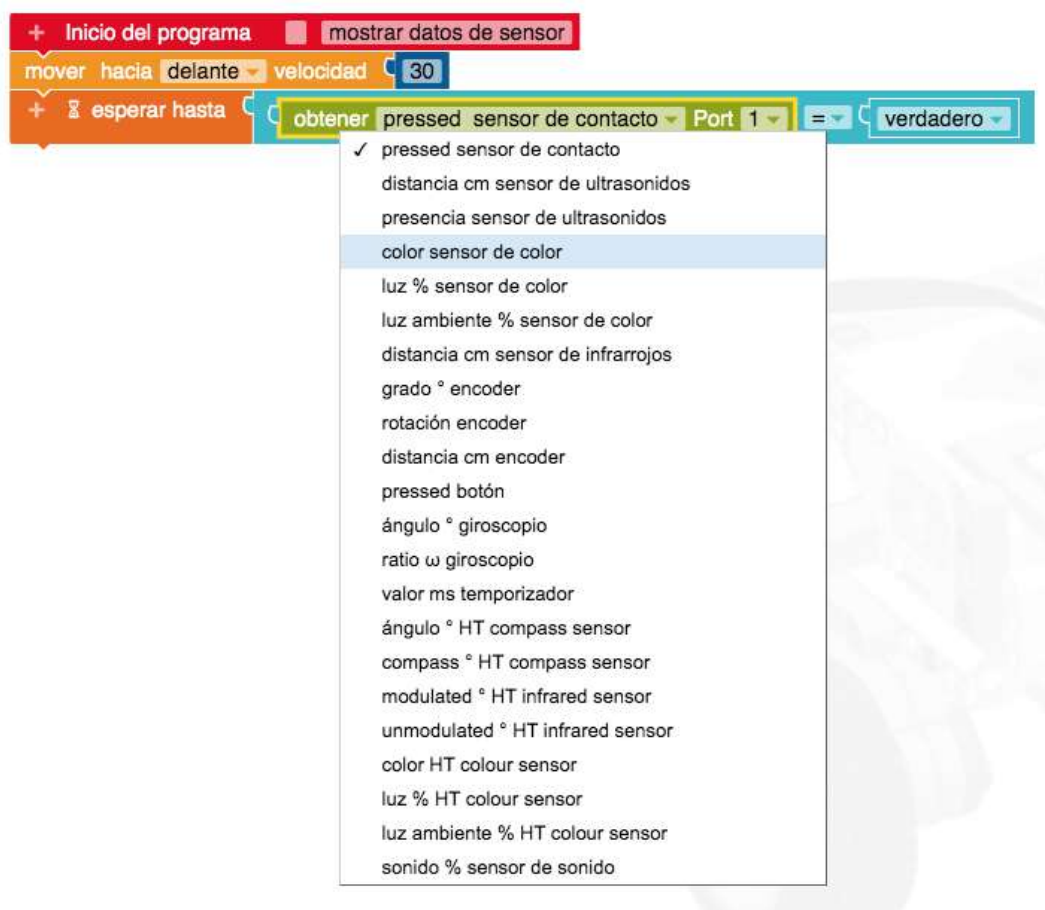
Después de añadir estos dos bloques, en el entorno de programación debes tener el sistema de bloques siguiente:



Paso 4: Vamos a cambiar el bloque verde (dentro del bloque ‘esperar hasta’), para obtener lo que nuestro robot ve desde el sensor de color. Para esto, hacemos clic en la flecha pequeña, como se ve en la imagen:



Se abrirá un menú, tenemos que elegir ‘color sensor de color’

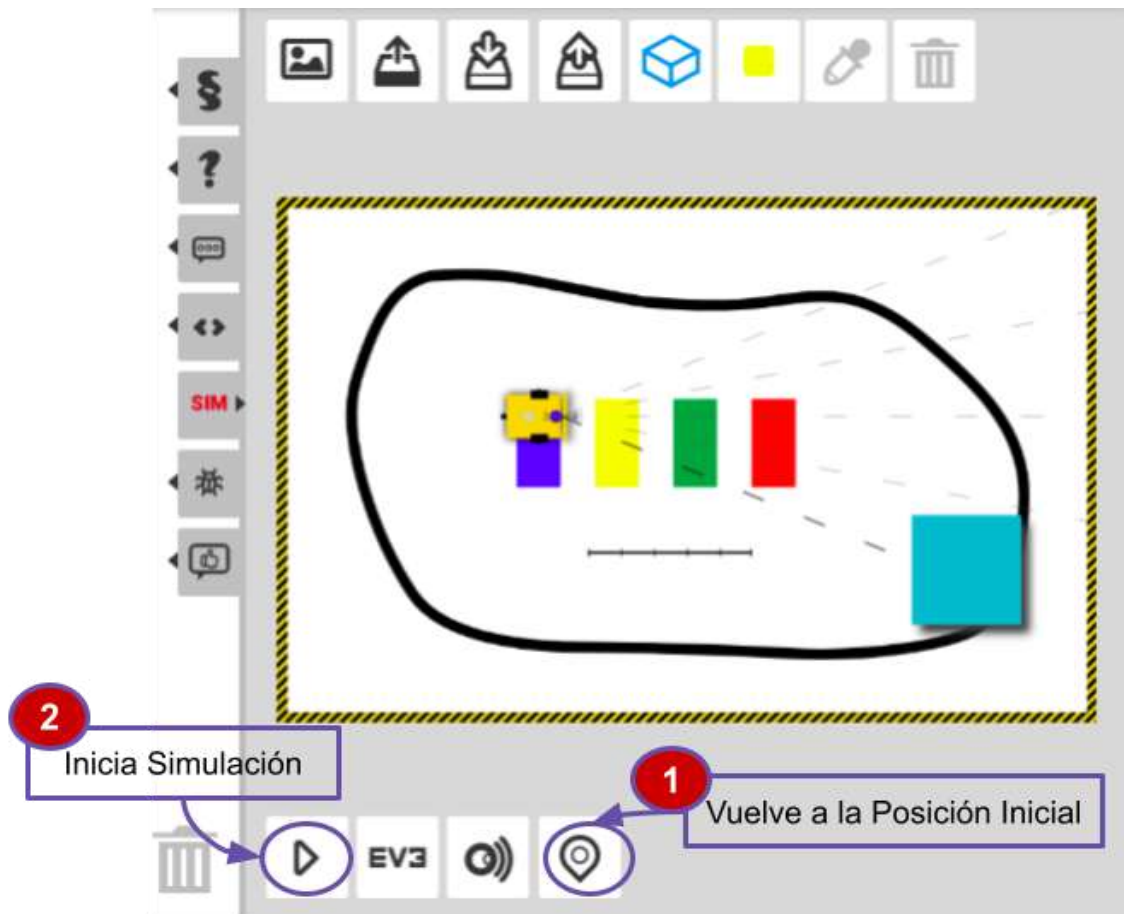


Al seleccionar el sensor de color, nuestro bloque quedará así:

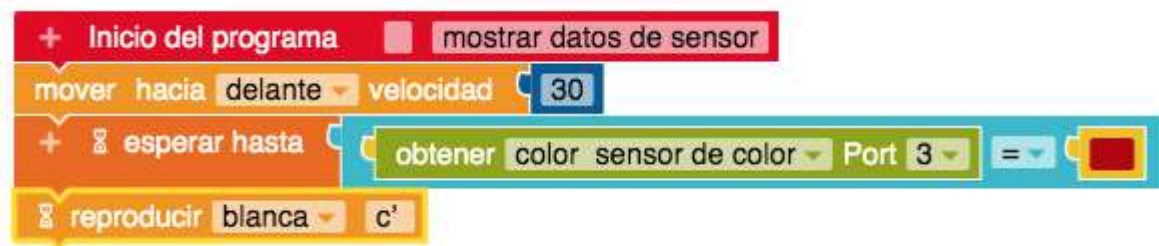


Si iniciamos nuestra simulación, nuestro robot se moverá hacia delante hasta que esté encima del bloque rojo.

!!! Antes de iniciar la simulación, recuerda pulsar el botón de volver a la posición inicial !!!



Paso 5: ¿Quieres incluir una alarma?, añade el bloque ‘**reproducir blanca c**’ (en el apartado de Acción) y vuelve a iniciar la simulación, ¿qué escuchas? Prueba a cambiar las opciones de este último bloque hasta encontrar un sonido que te guste.



¡Ya podemos comenzar a realizar nuestra Misión en la Luna!

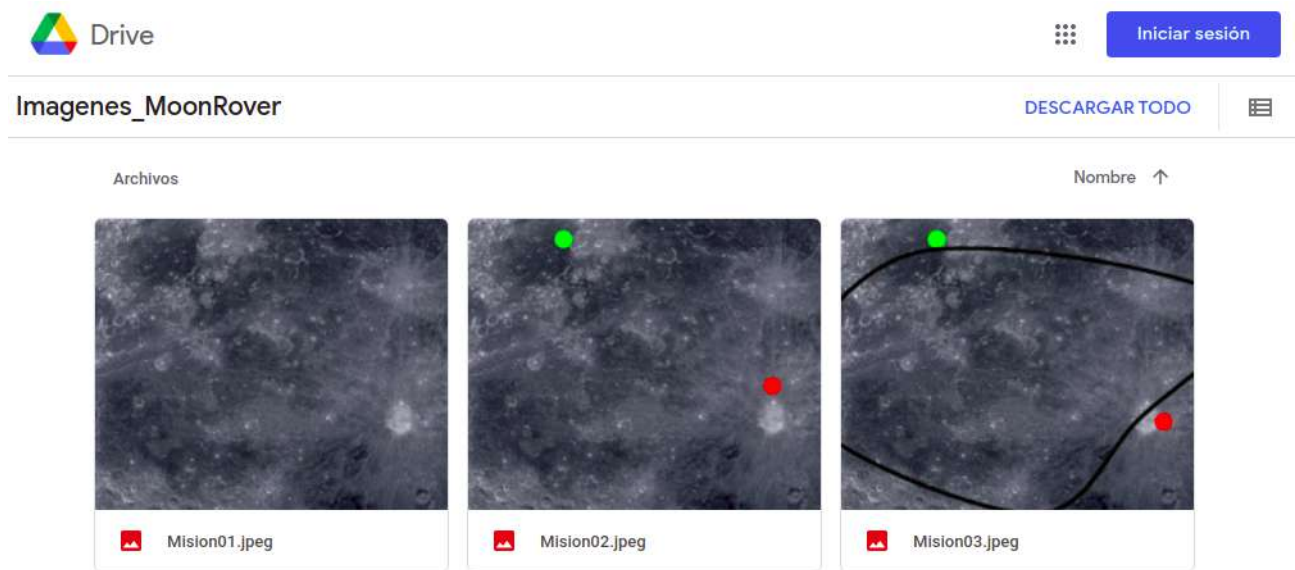
Misión 01. ¡Tenemos que volver a la base!

En esta misión vamos a regresar desde nuestra posición inicial hasta nuestra base lunar. Lo primero que tenemos que hacer es añadir nuestro fondo de simulación, donde veremos nuestro robot en la luna y nuestra base lunar.

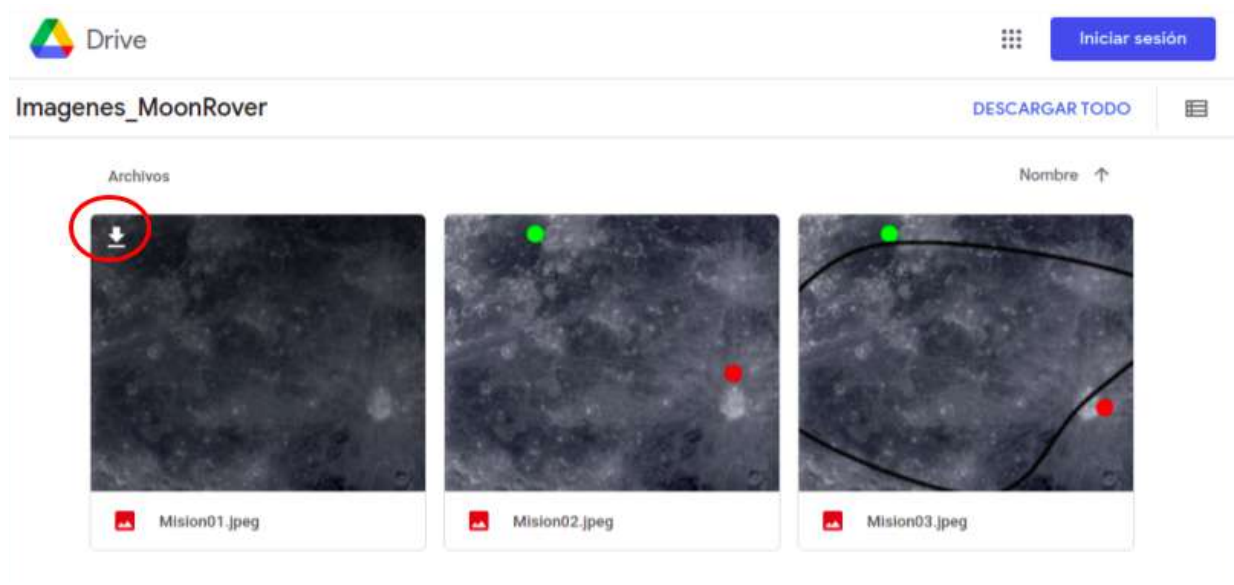
Paso 1: Ve a este enlace y descarga las imágenes de simulación:

<https://drive.google.com/drive/folders/1SDc7BG9VFAN72qECI0Bereaek4h1cDPv?usp=sharing>

Cuando entres en el enlace te aparecerán tres imágenes para descargar:

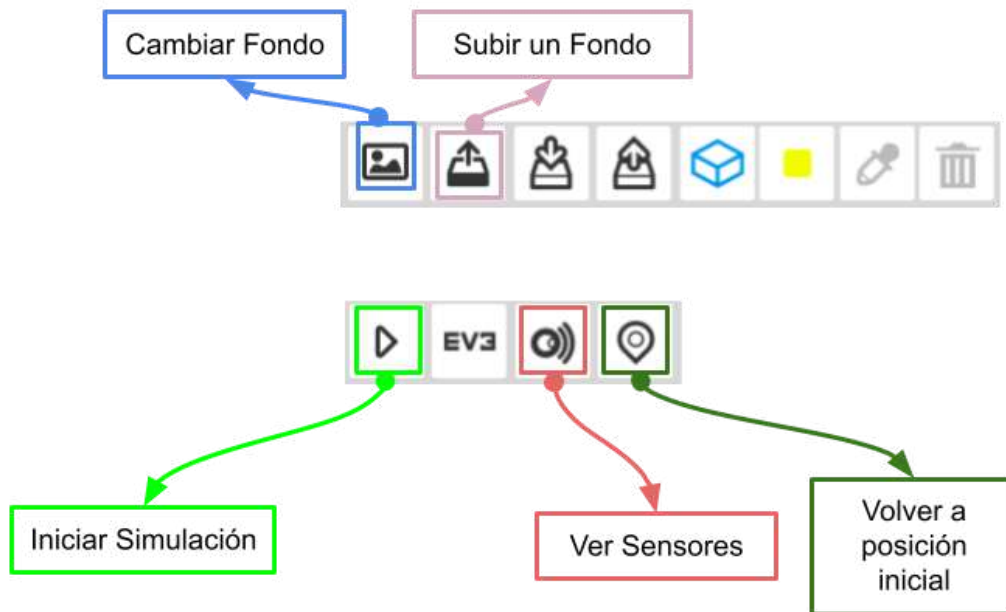


Lleva el ratón hacia el pico izquierdo de las imágenes y haz clic sobre el icono que aparece para descargarlas:



¡Es importante que recuerdes dónde guardas las imágenes!! Puedes guardar las imágenes en una carpeta en el **ESCRITORIO**. Si al descargar las imágenes, el navegador no te da opción para elegir en qué carpeta guardarlas, las encontrarás en la carpeta **DESCARGAS**.

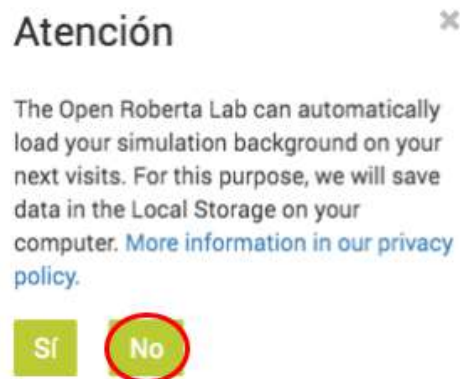
Una vez hayas guardado las imágenes, abre la ventana de simulación de Open Roberta (la pestaña SIM del lateral derecho) y haz clic en el botón de **subir un fondo**, que está arriba a la izquierda:



Al darle a este botón, se abrirá una ventana, dónde hay que buscar nuestras imágenes dónde las hemos guardado. Busca Mision01.jpeg y selecciona abrir:



Pulsa **NO** en la ventana que te aparece:



Ahora ya tenemos nuestro fondo, en el que nuestro robot se ha convertido en un Rover Lunar. Debes ver a tu robot en la superficie de la luna.

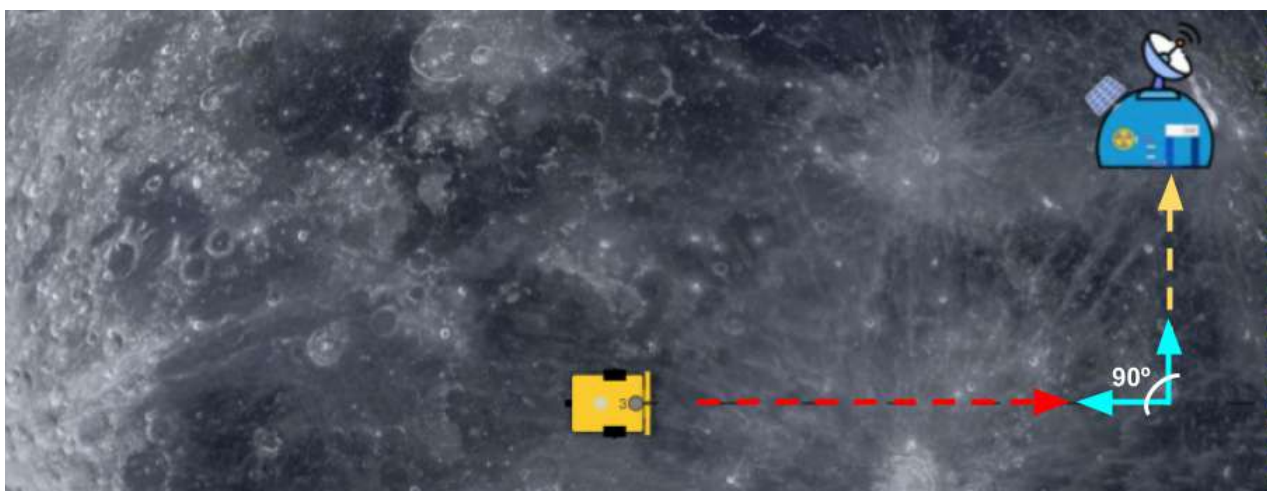
La base lunar, es la casita que puedes ver arriba a la derecha. Nuestra base lunar está equipada con una antena para las comunicaciones, placas solares para tener energía y un pequeño tubo para la gestión de residuos.



Para volver a la base desde nuestra posición inicial, un ejemplo de trayectoria a seguir es:

1. Seguir hacia delante (hacia la derecha, tal y como está colocado nuestro robot) hasta ponernos en línea con la base lunar
2. Girar a la izquierda
3. Seguir recto hacia arriba hasta llegar

De esta forma, necesitamos tres instrucciones, una para cada punto de la trayectoria. En la siguiente imagen puedes ver la trayectoria dibujada. Nuestro robot realizará una camino en L, dónde el ángulo de giros es de 90° .



Para mover nuestro Rover hacia adelante tenemos que incluir el bloque de programación **mover**, que utilizamos en la misión inicial, en el que se establece una velocidad y una distancia. Este bloque lo encontramos entre los bloques de Acción.



Cambiaremos la velocidad y la distancia para llegar a ponernos en línea con la base lunar y poder girar 90°.

Para realizar el giro, utilizamos este bloque de programación, que también se encuentra en los bloques de Acción. Tenemos que cambiar la velocidad y los grados para que se ajusten a nuestra misión.



Este es un ejemplo de programación para llegar a la base lunar, hay muchas soluciones y puedes volver a la base de diferentes formas. En este caso, al igual que en la misión inicial, se ha añadido una alarma cuando llegamos a la base, con el bloque **reproducir**.



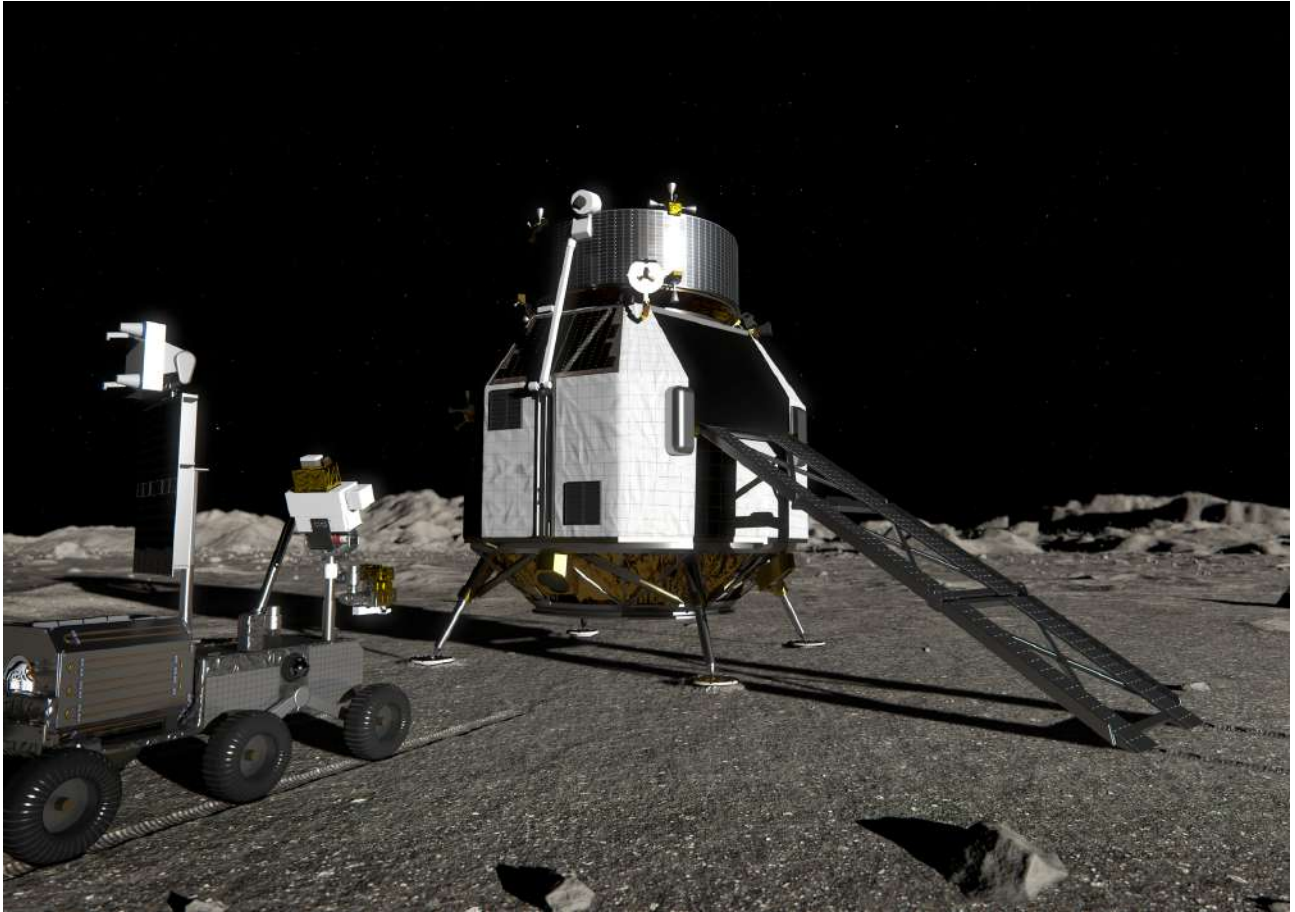
Referencias

Guía didáctica Misión en la Luna:

<http://esero.es/wp-content/uploads/2019/08/Mision-en-la-Luna-062019.pdf>

Programa un robot en la Luna

Parte II



Heracles on the Moon:

https://www.esa.int/ESA_Multimedia/Images/2019/06/Heracles_on_the_Moon#.YHsashbSB98.link

Una vez que hemos llevado nuestro robot hasta la base Lunar, vamos a recoger muestras de la superficie de la Luna, de dos formas diferentes:

- Programando nuestro robot para que vaya de una muestra a otra, hasta que pase por todas
- Programando nuestro robot para que siga una línea que pasa por todas las muestras que tenemos que buscar

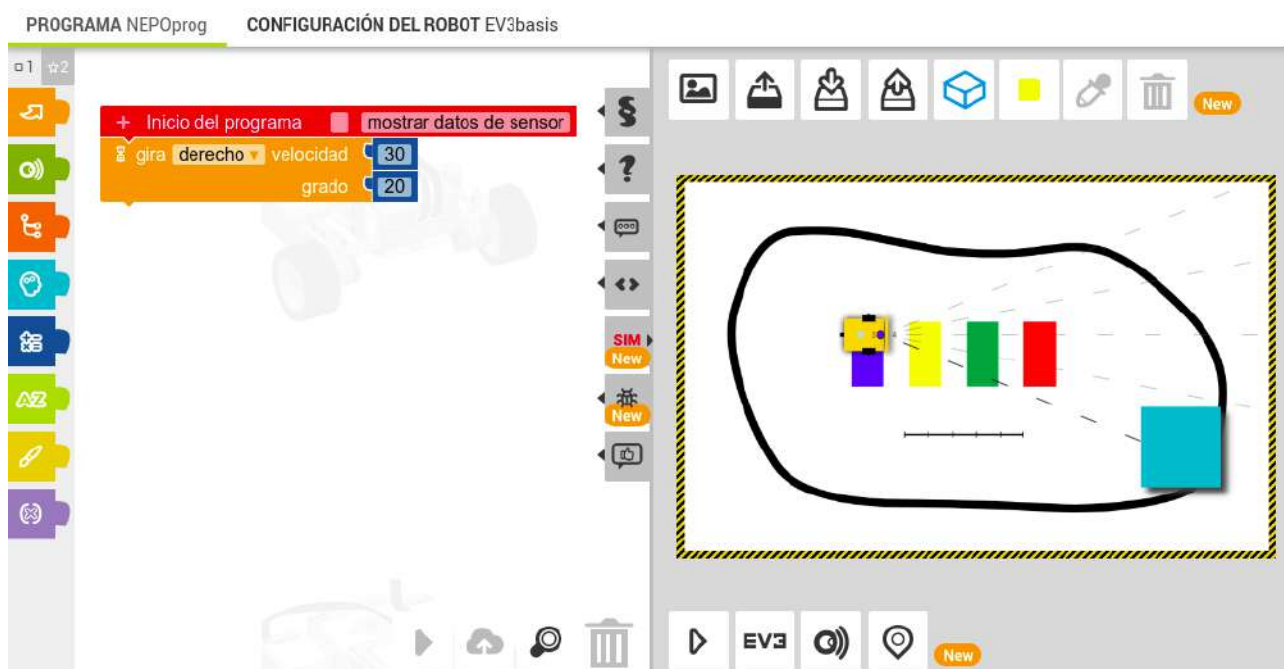
¿Cuál te parece más fácil?

Antes de empezar a programar nuestro robot en la Luna, vamos a entrenar dos conceptos básicos de programación: los bucles y la sentencia if/else.

¿Qué es un bucle?

Un bucle es una serie de instrucciones que se repite siempre o hasta que se cumple una determinada condición. Cuando terminan todas las instrucciones del bucle, se vuelve a empezar de nuevo. Vamos a ver un ejemplo de bucle con nuestro simulador.

Entra en el simulador Open Roberta Lab: <https://lab.open-roberta.org/> y añade un bloque de código para que el robot gire:



Dale al botón de **PLAY** en la simulación y observa qué ocurre.

El robot ha girado hacia la derecha 20° a una velocidad del 30 % y se ha parado porque ya no tiene más instrucciones.

Ahora vamos a añadir un bucle. Entra en la pestaña de **Control** y añade el bloque de programación '**Repetir indefinidamente**'



Coloca el bloque que hace girar al robot, dentro de este bucle:



De esta forma, el bloque girar, se va a repetir indefinidamente, es decir, el robot va a girar 20°, después otros 20°, después 20° y así siempre, hasta que paremos la simulación, porque no hay nada que detenga el bucle. Dale al play y comprueba cómo funciona el bucle.

Todo lo que haya dentro del bloque '**hacer indefinidamente**', se repetirá una y otra vez hasta que paremos la simulación

¿Qué hace la sentencia if/else?

Una sentencia es una instrucción que definimos para programar nuestro robot. La sentencia if/else, en español si/sino consiste en dos condiciones:

- Si ocurre la condición A, nuestro robot hace una cosa
- Si no ocurre la condición A, nuestro robot hará otra cosa diferente

La sentencia **if/else**, se utiliza dentro del bloque '**Hacer indefinidamente**'. Si el bucle no está, el robot sólo comprobará que se cumple la condición una vez, y nosotros queremos que esté continuamente comprobando la condición.

Vamos a programar un ejemplo para ver el funcionamiento de este bloque de programación.

Quemos que nuestro robot siga hacia delante si su sensor de color marca blanco, y si el sensor no marca blanco, que gire hacia la derecha.

Añade el bloque de control '**Si/hacer/Sino**'





Para establecer la condición, necesitamos un bloque de **Lógica**, para comprobar que el sensor de color es igual a blanco:



Nuestra programación, va quedando así:



Ahora tenemos que introducir en el bloque de lógica azul, a un lado el valor de nuestro sensor de color y al otro el color blanco. De esta forma nuestra condición es que el sensor de color sea igual blanco. Para esto utilizamos:

- De los bloques de sensores, el bloque obtener color sensor de color
- De los bloques de colores, el color blanco



Después de añadir nuestros dos bloques, la programación quedará así:

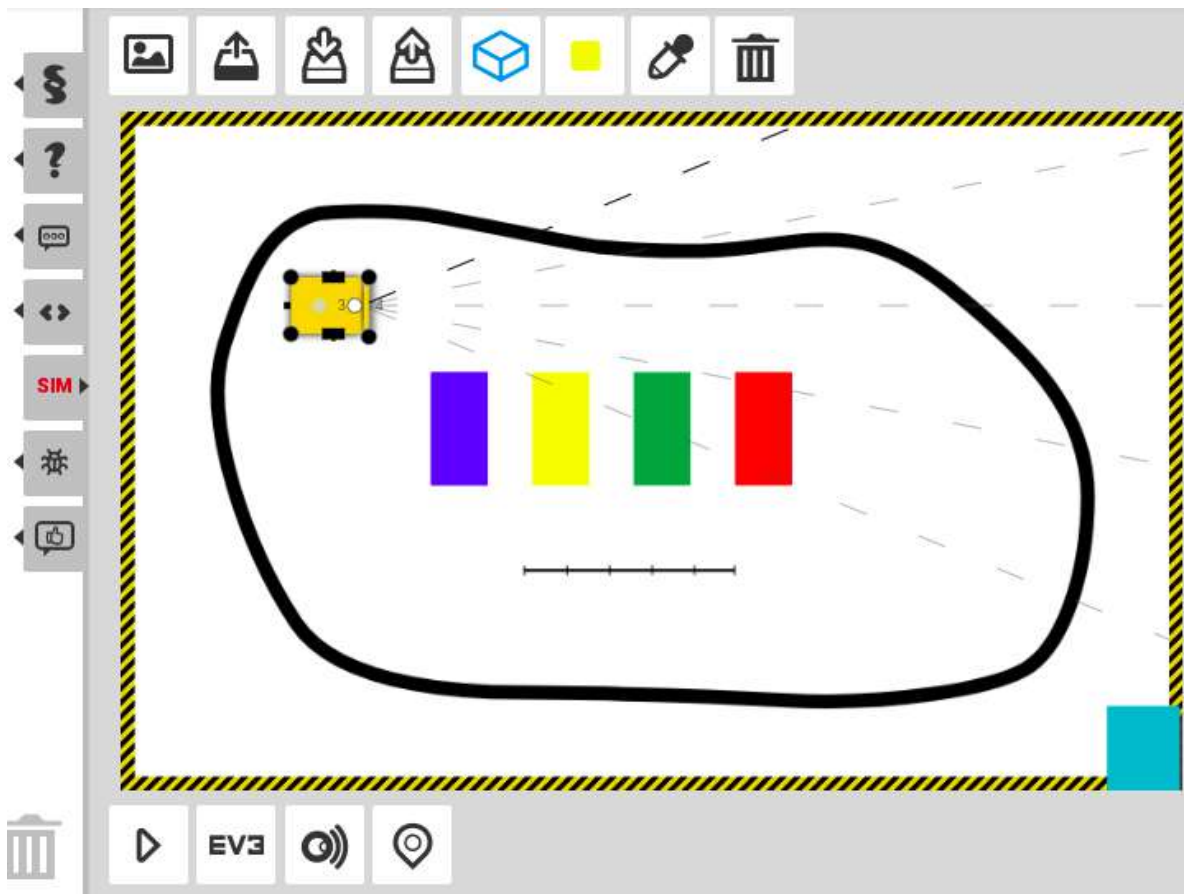


Si el sensor de color es igual a blanco, nuestro robot hará la opción que aparezca en la línea de HACER, y si no es blanco, hará la instrucción que pongamos en la línea de SINO.

En este ejemplo, vamos a poner que si el sensor de color es blanco, que siga hacia delante; y si no es blanco, que gire hacia la derecha. Esto lo hacemos con bloques de **Acción**:



!!!Coloca a tu robot en la parte blanca de la simulación y dale PLAY!!!



¿Qué ocurre? ¿funciona correctamente la programación? ¿Qué crees que está fallando?

Lo que pasa es que nuestro robot sólo está haciendo las instrucción una vez, y por eso:

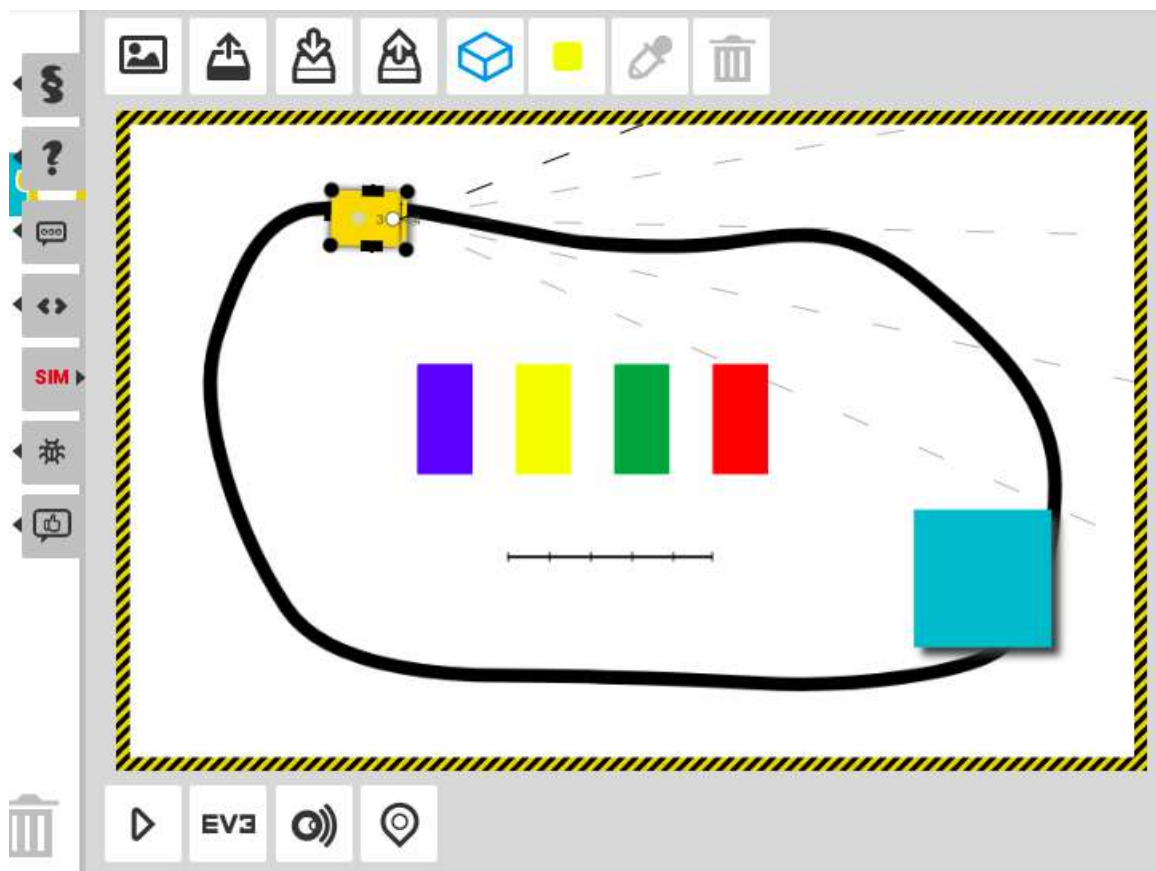
1. Comprueba que el sensor es igual a blanco
2. Como es igual, se mueve un paso hacia delante
1. Termina el programa

Para que nuestro robot esté **continuamente comprobando que el sensor es blanco** y se mueva hacia adelante hasta que no sea blanco y tenga que girar, tenemos que añadir un bucle 'repetir indefinidamente'



Mueve tu robot cerca de los colores de la imagen de simulación y comprueba que el programa funciona.

Lleva a tu robot cerca de la línea negra, ¿Que ocurre? ¿Parece que tu robot sigue la línea? Esto pasa porque nuestro robot sigue hacia delante si el suelo es blanco, al acercarse a la línea negra, tiene que girar porque el suelo es negro; y esto hace que siga la trayectoria de la línea.



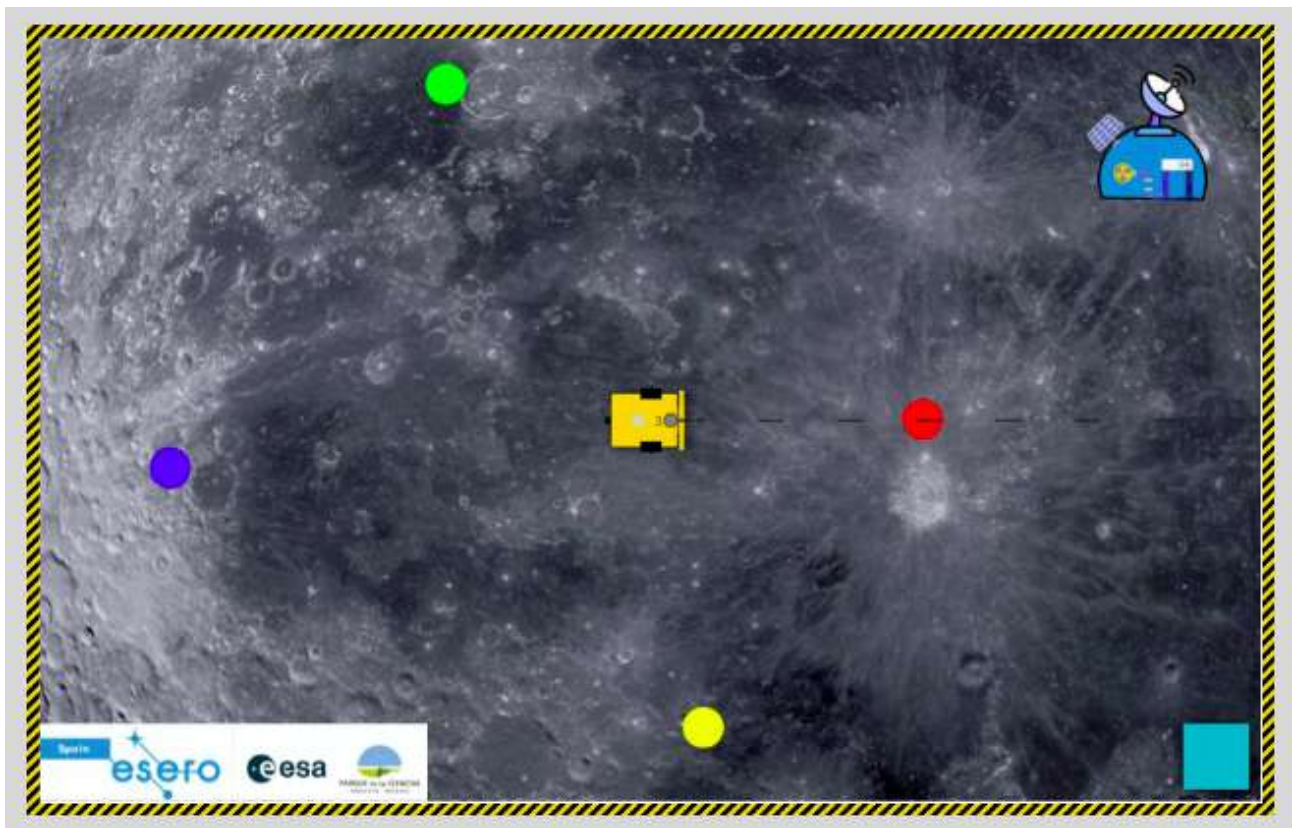
Después de este entrenamiento, ya puedes seguir con las misiones.

Misión 02. Busquemos muestras de la superficie Lunar

En esta misión programamos nuestro Rover para que pueda tomar todas las muestras de la superficie lunar, utilizando el sensor de color. Para realizar esta misión, debes cargar la imagen Mision02.jpg, en el simulador al igual que la de la primera misión. Las imágenes de la simulación las puedes encontrar en este enlace:

<https://drive.google.com/drive/folders/1SDc7BG9VFAN72qECI0Bereaek4h1cDPv?usp=sharing>

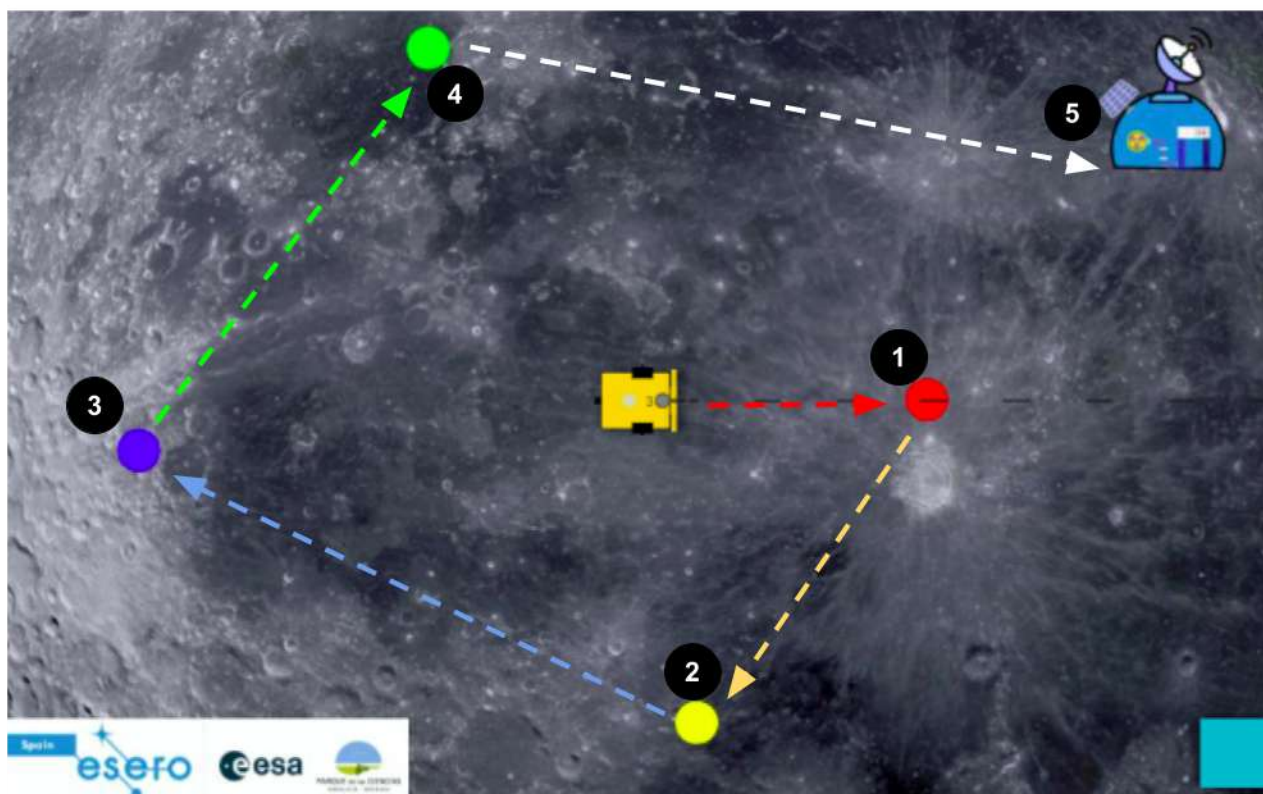
En esta imagen aparecen las diferentes muestras con varios colores.



Antes de empezar a programar es importante **plantear una trayectoria a seguir**. En este ejemplo, vamos a programar nuestro Rover para que recoja las muestras según el siguiente orden:

1. Recoger la muestra roja
2. Recoger la muestra amarilla
3. Recoger la muestra azul
4. Recoger la muestra verde
5. Volver a la base Lunar

Según esta secuencia, nuestro robot seguirá el camino que marcan las flechas en la imagen:



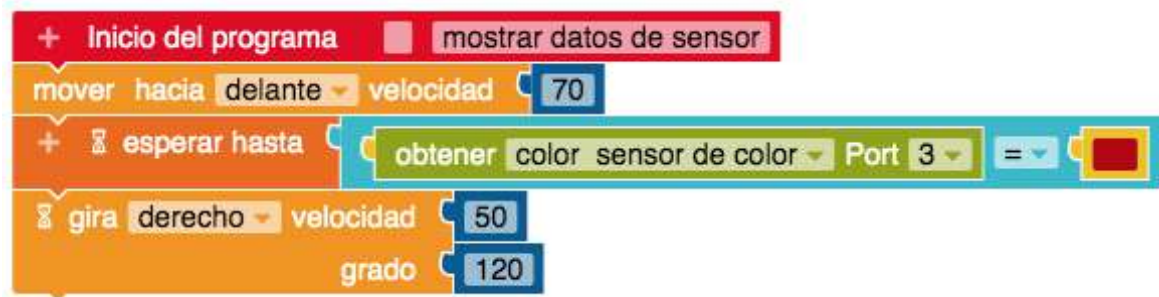
Para realizar el paso 1 y llegar hasta la muestra roja, vamos a utilizar el sensor de color como condición para que nuestro Rover se detenga. Nuestro Rover, tal y como está colocado debe **seguir hacia delante hasta que el sensor de color sea rojo**.

Esta programación, ya la hemos realizado en la misión inicial y es necesario un bloque de Acción y un bloque Control:

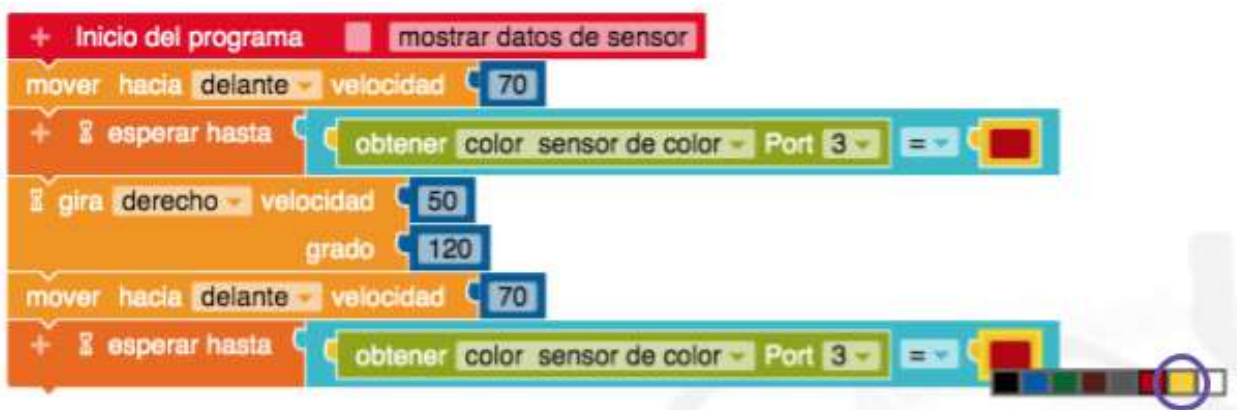
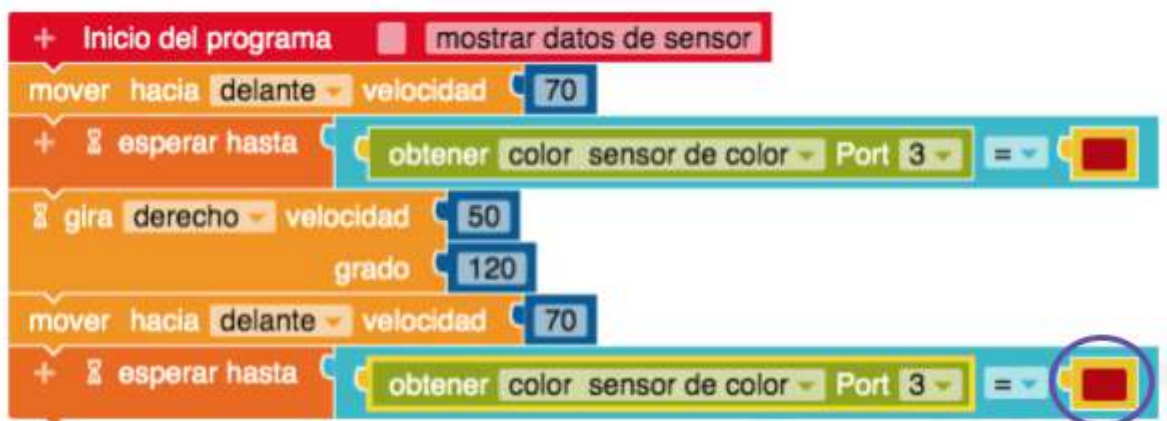


Ahora, presiona el botón de **volver a la posición inicial** y después el botón de **Play** para comprobar que el Rover llega a la muestra roja y se para. Una vez en la muestra roja, nuestro Rover debe girar un cierto ángulo para llegar a la muestra amarilla.

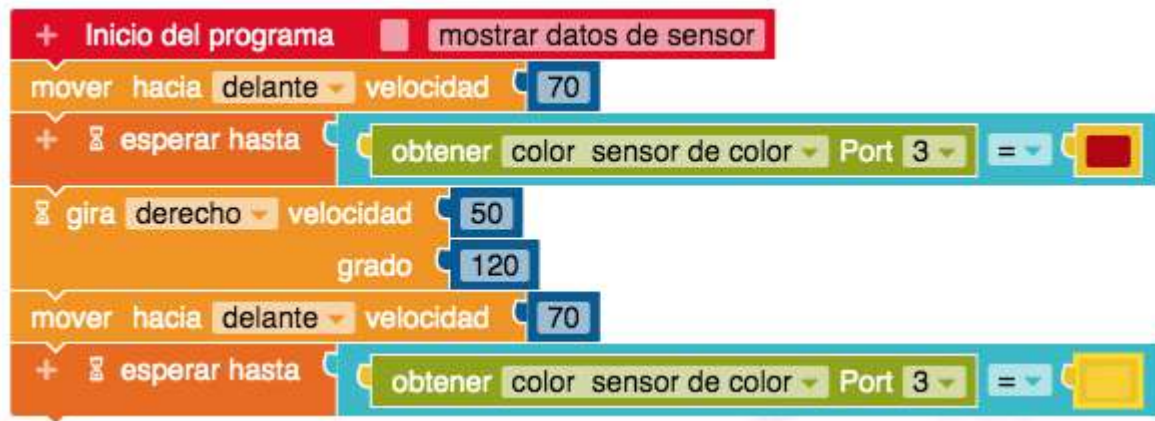
Añade un bloque de giro y ve probando hasta encontrar el ángulo en el que nuestro Rover se queda preparado para llegar a la muestra amarilla en línea recta.



Cuando nuestro Rover esté orientado, tiene que seguir hacia delante **hasta que encuentre la muestra amarilla**. Para esto, volvemos a repetir la secuencia anterior de bloques, pero cambiando el color que detecta el sensor por **amarillo**. Si haces clic en el color, te aparecerán varios para elegir.

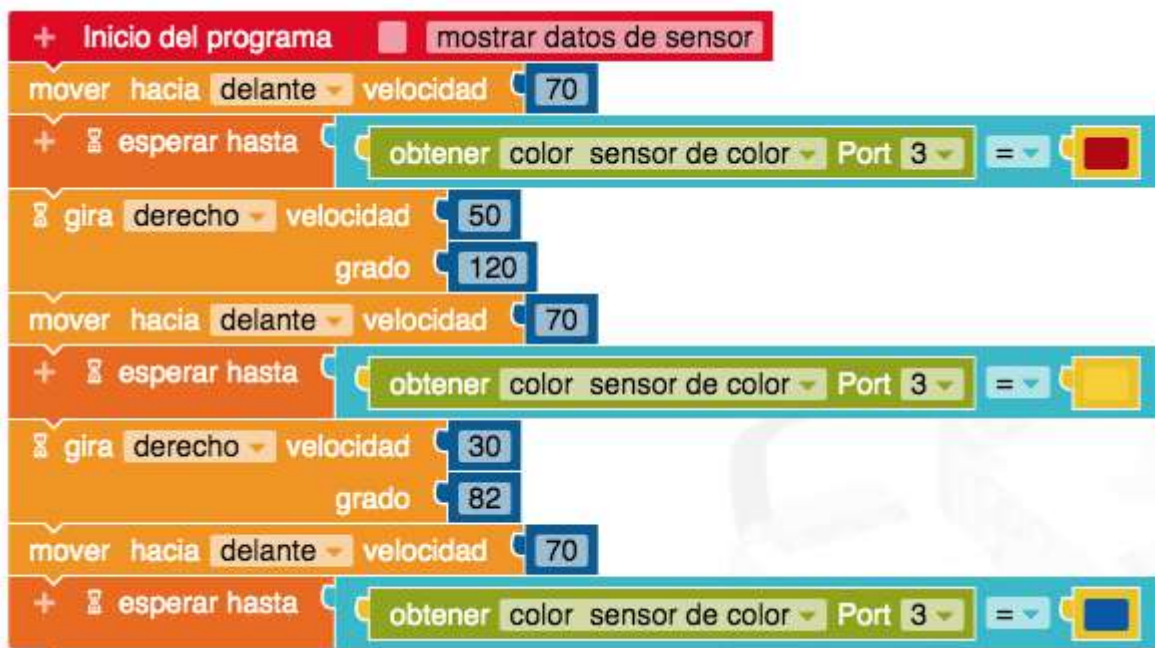


Una vez elegido el color amarillo, las instrucciones son que el Rover siga hacia delante hasta que el sensor de color detecte color amarillo.



!!!Vuelve a la posición inicial y dale al Play para comprobar que tu programación funciona!!!

Ahora, al igual que antes hay que realizar un giro para poner a nuestro Rover mirando hacia la muestra azul. Prueba varios grados hasta encontrar uno que se ajuste. Después añade los bloques de código para que **siga hacia delante hasta que encuentre la muestra azul**.



Continúa con la programación para encontrar la muestra verde. Recuerda los pasos a seguir para programar al Rover.

1. Encuentra el ángulo de giro
2. Sigue hacia delante hasta que el sensor de color detecte la muestra
3. Vuelve a la posición inicial y dale al Play para comprobar si la programación funciona

Una vez que el Rover llegue a la muestra verde, tenemos que volver a la base Lunar. En este caso giramos un ángulo adecuado y añadimos el bloque de código que nos permite establecer una determinada distancia.



Cambiamos la distancia y la velocidad para llegar a nuestra base Lunar.

!!!Vuelve a la posición inicial y dale al play para comprobar que tu programación funciona!!!

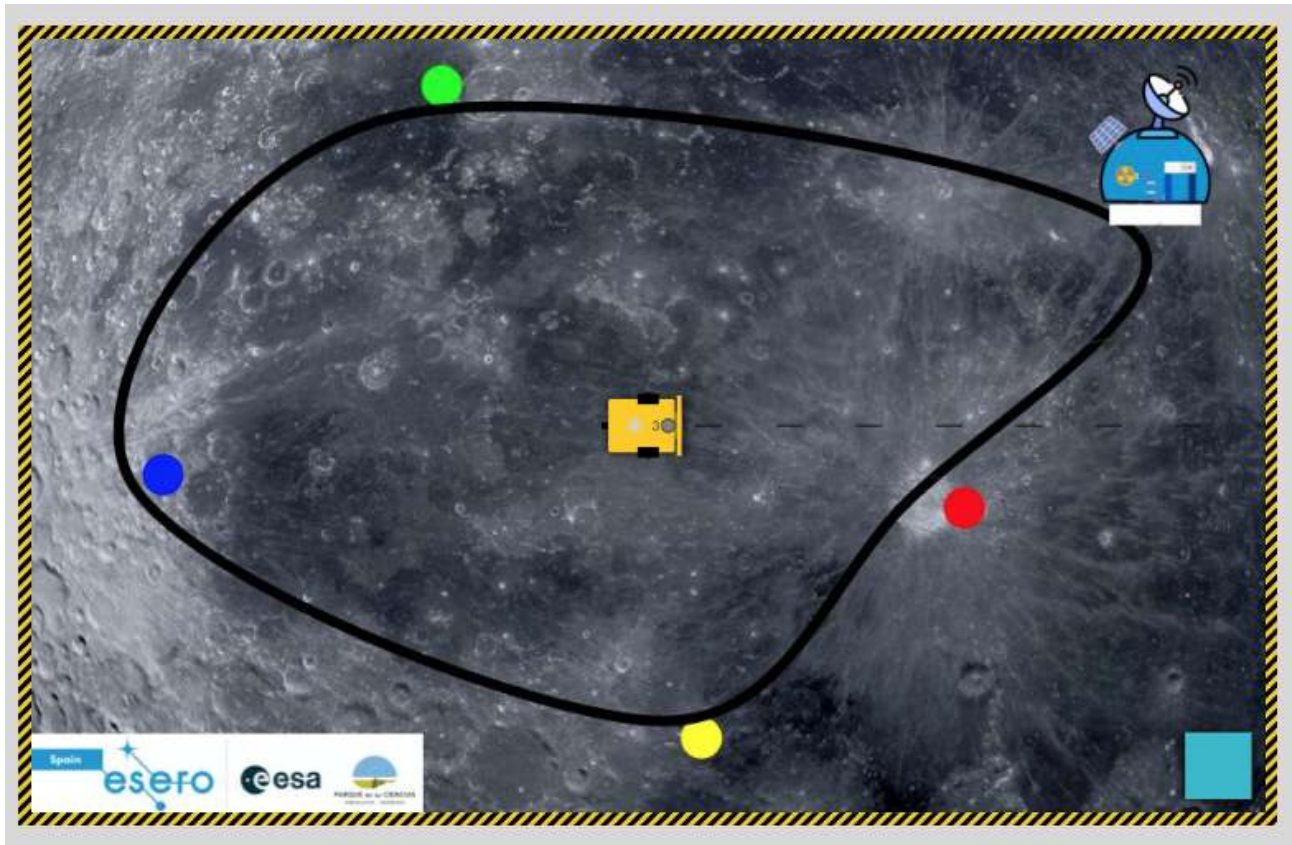
Cuando la programación esté completa y funcione correctamente, puedes añadir una alarma cuando llegues a la base. Este es un ejemplo de cómo quedaría nuestra programación completa.



Ampliación: ¿Te atreves a completar una nueva misión?

Misión 03. ¡Hagamos una carrera en la Luna!

En esta misión, nuestro Rover pasará por todas las muestras, siguiendo una línea negra. Al añadir la imagen Mision03.jpg al entorno de simulación, podrás ver la línea negra.



¿Cómo funciona un siguelineas?

El principio básico de un robot que sigue una línea negra es que siempre esté comprobando que está en la línea negra.

Siguiendo este concepto, podemos llegar a una programación básica, en la que el Rover sigue adelante cuando detecta el color negro y gira (para encontrar el color negro) cuando no lo detecta.

Es necesario utilizar los bloques de control:

- **Repetir indefinidamente:** este bloque permite que nuestro Rover siempre esté haciendo lo mismo (comprobar que está en la línea negra), es lo que en programación se llama bucle. Nuestro Rover estará en bucle y no podrá salir, repetirá siempre lo mismo.

- **Si/hacer/Sino:** en este bloque, ponemos una condición y damos una instrucción en el caso de que se cumpla y una instrucción en el caso de que no se cumpla: **Si el robot está en la línea negra, entonces sigue adelante, si no está en la línea negra, giro a la izquierda para encontrar la línea.**

El bloque **Si/hacer/Sino**, se coloca dentro del bloque **Repetir indefinidamente**, de esta forma, nuestro Rover siempre está comprobando si está en la línea negra.



Pista: para establecer la condición de obtener el color del sensor de color y saber si es igual a negro, hay que utilizar un bloque de Lógica, un bloque de Sensores y un bloque de Colores.

Prueba esta programación. Coloca el robot directamente encima de la línea antes de comenzar la programación.

¿Cómo funciona? ¿Es capaz el Rover de seguir la línea sin problemas? ¿cuándo se desvía? ¿Cómo podemos mejorar esta programación?

Esta programación no funciona del todo bien, el robot se va de la línea si hay curvas marcadas. Para mejorar esto, la clave está en los **giros**. Los bloques de giro que hemos añadido hasta ahora, no indican la rueda que gira, es decir, giran las dos ruedas delanteras a la vez.

Si queremos hacer el giro más suave, para que nuestro Rover no se desvíe, tenemos que utilizar este bloque de Acción:



En este bloque de programación, podemos establecer la velocidad de la rueda izquierda y de la rueda derecha.

¿Qué pasará cuándo la rueda izquierda gira más rápido que la derecha? ¿Y cuándo la derecha gira más rápido que la izquierda? Intenta mejorar la programación utilizando este bloque de programación en las dos condiciones del bloque **Si/hacer/Sino**.

Referencias

Guía didáctica Misión en la Luna:

<http://esero.es/wp-content/uploads/2019/08/Mision-en-la-Luna-062019.pdf>

Anexo II

En este Anexo, se incluye el guion para el profesorado, con instrucciones y aspectos a tener en cuenta para llevarlo al aula.

GUIÓN PARA EL PROFESORADO

Programa un Robot en la Luna

Contenidos

1.-Descripción	1
2.-Aspectos Generales	1
3.-Actividades Propuestas	2
4.- Objetivos de aprendizaje	3
5.- Evaluación. Propuesta de Rúbrica	4
6.- Herramientas para introducir el contexto	5
7.- Propuesta de Planificación	5
8.-Propuestas de organización. Grupos Plataforma Open Roberta Lab	6
9.- Propuestas de solución para las misiones	15

1.-Descripción

El objetivo es programar un Rover Lunar, enviado a la Luna para recoger muestras de la superficie Lunar y estudiar si sería posible que vivamos allí en el futuro. Las actividades se realizan a través de la plataforma de programación y simulación web Open Roberta Lab. En total se realizan tres misiones, acompañadas de actividades iniciales relacionadas con el funcionamiento del simulador y algunos bloques de programación.

2.-Aspectos Generales

Asignaturas de aplicación	TIC, TIC I, TIC II, Tecnología
Nivel Educativo Recomendado	Educación Secundaria Obligatoria y Bachillerato
Complejidad	Parte I: inicial-media Parte II: media-avanzada
Conocimientos Previos requeridos	Ninguno
Tiempo aproximado	Parte I: 60'' Parte II: 90''
Coste	Ninguno, se realiza a través de una herramienta online
Material necesario	Ordenadores o Tablets con conexión a internet

3.-Actividades Propuestas

Actividad	Título	Descripción	Objetivo	Conocimientos previos	Tiempo
1	¿Cómo funciona nuestro simulador?	Entrar en el simulador y seleccionar el sistema adecuado. No hay que hacer ninguna tarea.	Introducir funciones básicas y la interfaz gráfica del simulador	-	10 min
2	Misión Inicial. Nuestro Simulador	Programar el robot para que se mueva hacia delante con bloques de acción, utilizando el sensor de color	Conocer el entorno de programación y simulación	-	20 min
3	Misión 1 ¡Tenemos que volver a la base	Programar el robot para que realice una trayectoria determinada y vuelva a la base lunar desde la posición inicial	Conocer el funcionamiento de los bloques de acción y los ángulos de giro	Actividad previas	30 min
4	¿Qué es un bucle?	Programar el robot para que repita indefinidamente una determinada instrucción	Utilizar y recordar el funcionamiento de un bucle	Actividades previas	15 min
5	¿Qué hace la sentencia if/else?	Programar el robot para realizar diferentes acciones dependiendo de si se cumple una determinada condición	Utilizar y recordar el funcionamiento de la sentencia if/else	Actividades previas	15 min
6	Misión 2 Busquemos muestras de la superficie Lunar	Programar el robot para que pase por cuatro puntos diferentes, utilizando ángulos de giro y el sensor de color	Conocer el funcionamiento del sensor de color y utilizarlo para realizar una determinada trayectoria	Actividades previas	30 min
7	Misión 3 ¡Hagamos una carrera en la Luna	Programar el robot para seguir una línea negra	Aplicar los conocimientos adquiridos en las actividades previas, utilizando bloques de programación de acción, control, lógica y sensores	Actividades previas	30 min

4.- Objetivos de aprendizaje

Objetivos Generales	Objetivos Específicos
<p>Fomentar el interés en las carreras espaciales</p> <p>Contextualizar la programación en el mundo de la exploración espacial; con el objetivo de despertar el interés y la curiosidad de los estudiantes por la robótica y la exploración espacial</p> <p>Motivar al alumnado para que considere una carrera técnica en el futuro</p>	<p>Trabajar conceptos básicos de programación y robótica</p> <p>Desarrollar el pensamiento computacional</p> <p>Aplicar a un caso práctico bucles y sentencias de programación</p> <p>Fomentar la creatividad y el espíritu crítico</p> <p>Desarrollar habilidades en programación y robótica a través de un entorno gráfico de simulación</p> <p>Trabajar la competencia digital</p> <p>Conocer las posibilidades profesionales dentro de la industria espacial</p>

5.- Evaluación. Propuesta de Rúbrica

Act	Título	Nivel Iniciado	En Desarrollo	Conseguido	Excelente
2	Misión Inicial. Nuestro Simulador	Realiza la primera tarea con dificultades	Realiza la primera tarea correctamente. La segunda tarea con dificultades	Realiza las dos tareas correctamente	Realiza las dos tareas y añade bloques de ampliación, como alarmas y/o melodía
3	Misión 1 ¡Tenemos que volver a la base	No completa la primera misión	Realiza la primera misión con dificultades y/o haciendo uso de la solución	Realiza la primera misión correctamente	Realiza la primera misión con la trayectoria de ejemplo, incluyendo ampliaciones como bloques de alarmas y/o melodías. Realiza otra trayectoria adicional para completar la misión
4	¿Qué es un bucle?	No completa la tarea	Completa la tarea con dificultades	Completa la tarea correctamente	Completa la tarea y modifica los bloques para realizar otra acción
5	¿Qué hace la sentencia if/else?	No completa la tarea	Completa la tarea con dificultades	Completa la tarea correctamente	Completa la tarea y modifica los bloques para realizar otra acción
6	Misión 2 Busquemos muestras de la superficie Lunar	Realiza la primera parte de la misión con dificultades	Realiza la misión completa con dificultades y/haciendo uso de la solución	Realiza la misión correctamente	Realiza la misión completa y modifica la programación para cambiar la trayectoria, añadir alarmas y/o melodías
7	Misión 3 ¡Hagamos una carrera en la Luna	Realiza la programación básica indicada en el guión	Realiza la misión completa con dificultades y/haciendo uso de la solución	Realiza la misión correctamente	Realiza la misión completa, optimiza el código y añade bloques de alarma y/o melodía

6.- Herramientas para introducir el contexto

Videos relacionados con el contexto, para motivar y ayudar a que el alumnado se implique en las actividades.

- El Rover de la misión Heracles en la Luna:
<https://www.youtube.com/watch?v=75MQSSJHieY&t=37s>
- Robótica Espacial en la Luna: <https://www.youtube.com/watch?v=RZ06r9aWhfg>

Videos de la Fundación Airbus Space: Misión a la Luna: Empieza a aprender sobre el Espacio. Es una serie de videos creados para mostrar cómo sería nuestra futura vida en la Luna.

- Videos en Español:
<https://www.airbus.com/company/sustainability/airbus-foundation/discovery-space-es/ninos/mission-a-la-luna.html>
- Vídeos en Inglés (subtitulados en español):
https://www.youtube.com/watch?v=h5X3CTfShyE&list=RDCMUCJ1cS4hALCDHPnygfv6VOhQ&start_radio=1&rv=h5X3CTfShyE&t=3

7.- Propuesta de Planificación

Nº sesión	Tiempo Aproximado	Contenidos propuestos
1	1 h	Introducción al contexto: utilizando videos y material de apoyo Conocer el entorno de simulación Misión Inicial:
2	1 h	Misión 01
3	1 h	Refuerzo de Contexto Actividades de repaso (Actividades 4 y 5) Inicio Misión 02
4	1 h	Misión 02 Misión 03 Cierre y debate sobre la exploración y la robótica Espacial ¿Llegaremos a vivir en la Luna en el futuro?

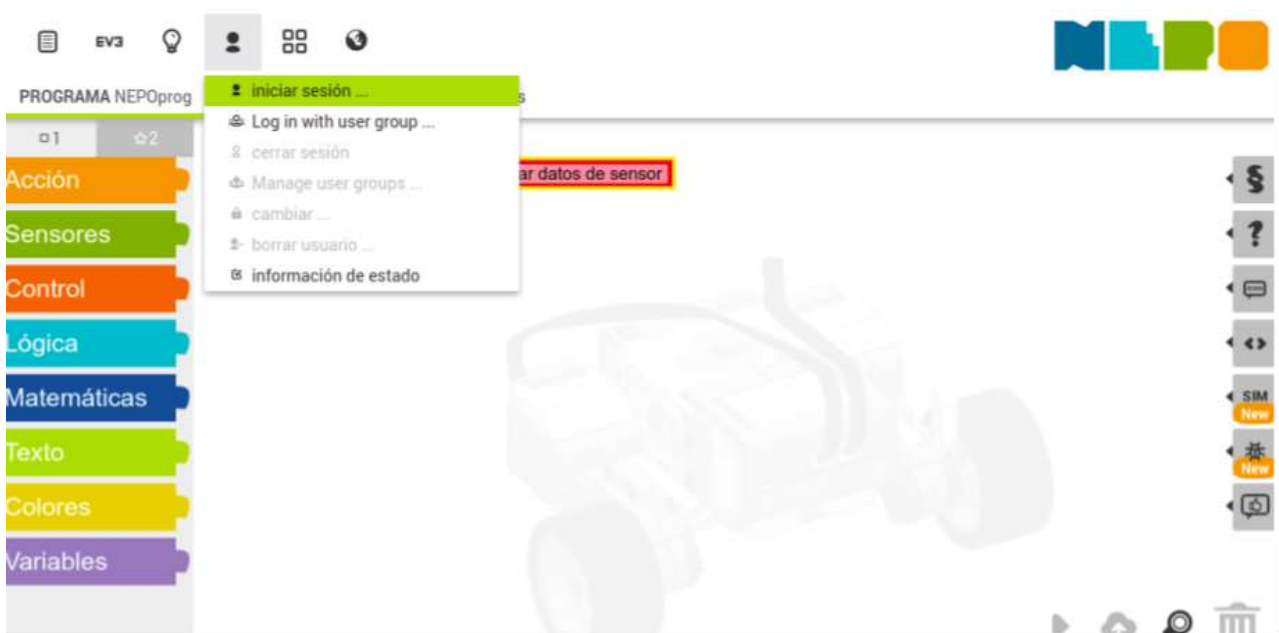
8.-Propuestas de organización. Grupos Plataforma Open Roberta Lab

La plataforma se utiliza directamente desde el navegador, sin necesidad de instalar ningún programa y ofrece una serie de posibilidades para ser utilizado en el grupo clase. Para utilizarlo, no es necesario contar una cuenta. Sin embargo, registrarse es gratuito y te permite generar un repositorio de programas y compartirlos. Al terminar una programación, se puede guardar como privado o público.

Tener una cuenta, da acceso a la opción crear un grupo. Esto permite guardar programas para que sean visibles entre los usuarios de un determinado grupo; y que los usuarios del grupo puedan guardar nuevos programas. De esta forma, el creador del grupo, puede ver los programas que crean cada uno de los miembros del grupo.

Para crear un grupo, primero es necesario registrarse con una cuenta de usuario e indicar un correo de referencia. A continuación, se muestran las instrucciones

1. Crear una cuenta de usuario. Entra en <https://lab.open-roberta.org/> y selecciona iniciar sesión en la siguiente imagen:



2. Selecciona nuevo usuario e introduce los datos de usuario correspondientes. Una vez seleccionado el botón 'Registrar ahora', recibirás un correo electrónico de confirmación a la cuenta que hayas indicado. Una vez confirmes tu cuenta, introduce tus datos para iniciar sesión.

iniciar sesión ...

Nombre de usuario

Contraseña

OK

resetear password ... nuevo ...

nuevo ...

Nombre de usuario

»AmBotman« or »RobellaStracciatella«? Not everyone needs to know your real name. Think of a cool nickname that you can easily remember.

Contraseña

Repetir contraseña

Nombre

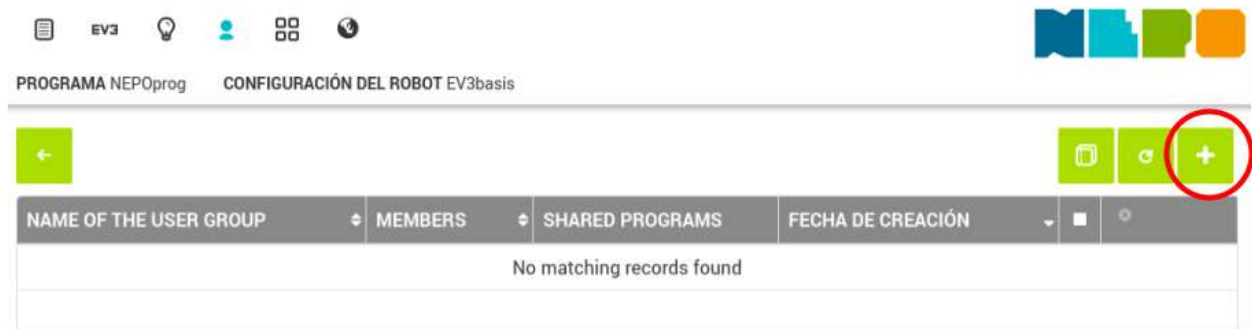
Correo electrónico

Registrar ahora

3. Selecciona la opción 'Manage user groups' para abrir la pestaña de gestión de grupos.



4. Se abrirá una pestaña, donde aparecen los grupos creados. La primera vez que entramos, no tendremos ningún grupo. Para crear un grupo nuevo, hacemos clic en el + de la parte superior derecha.



5. Se abrirá una ventana, en la que indicamos el nombre del grupo y el número de miembros que tendrá nuestro grupo. Seleccionamos 'Create new user group' para guardar los cambios.

Create new user group

Name of the user group

Grupo A

Automatically generated members

15

Automatically generated members are normal members of a user group. They are generated with a number as username, which you can change afterwards.

Create new user group Cancelar

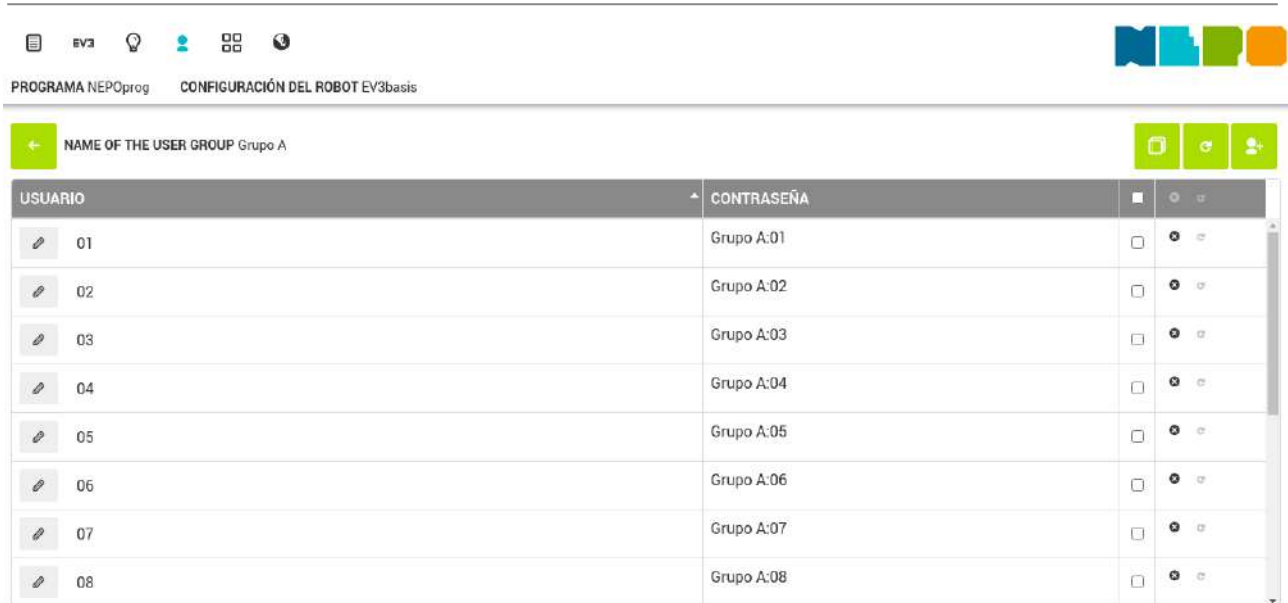
Nombre del grupo

Número de miembros del grupo

6. Una vez creado el grupo, ya aparece en nuestro listado de grupos. Para ver el listado de los miembros del grupo, hacemos clic sobre el nombre del grupo

NAME OF THE USER GROUP	MEMBERS	SHARED PROGRAMS	FECHA DE CREACIÓN
Grupo A	15	-	23.05.2021, 18:52

7. En el listado podemos ver el nombre de usuario y la contraseña generados para cada uno de nuestros miembros del grupo. Por defecto el nombre usuario es un número, pero se puede cambiar haciendo clic en el lápiz de la derecha de cada nombre. Las contraseñas se crean de forma automática.



PROGRAMA NEPOprog CONFIGURACIÓN DEL ROBOT EV3basis

NAME OF THE USER GROUP Grupo A

USUARIO	CONTRASEÑA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
01	Grupo A:01	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
02	Grupo A:02	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
03	Grupo A:03	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
04	Grupo A:04	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
05	Grupo A:05	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
06	Grupo A:06	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
07	Grupo A:07	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
08	Grupo A:08	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>



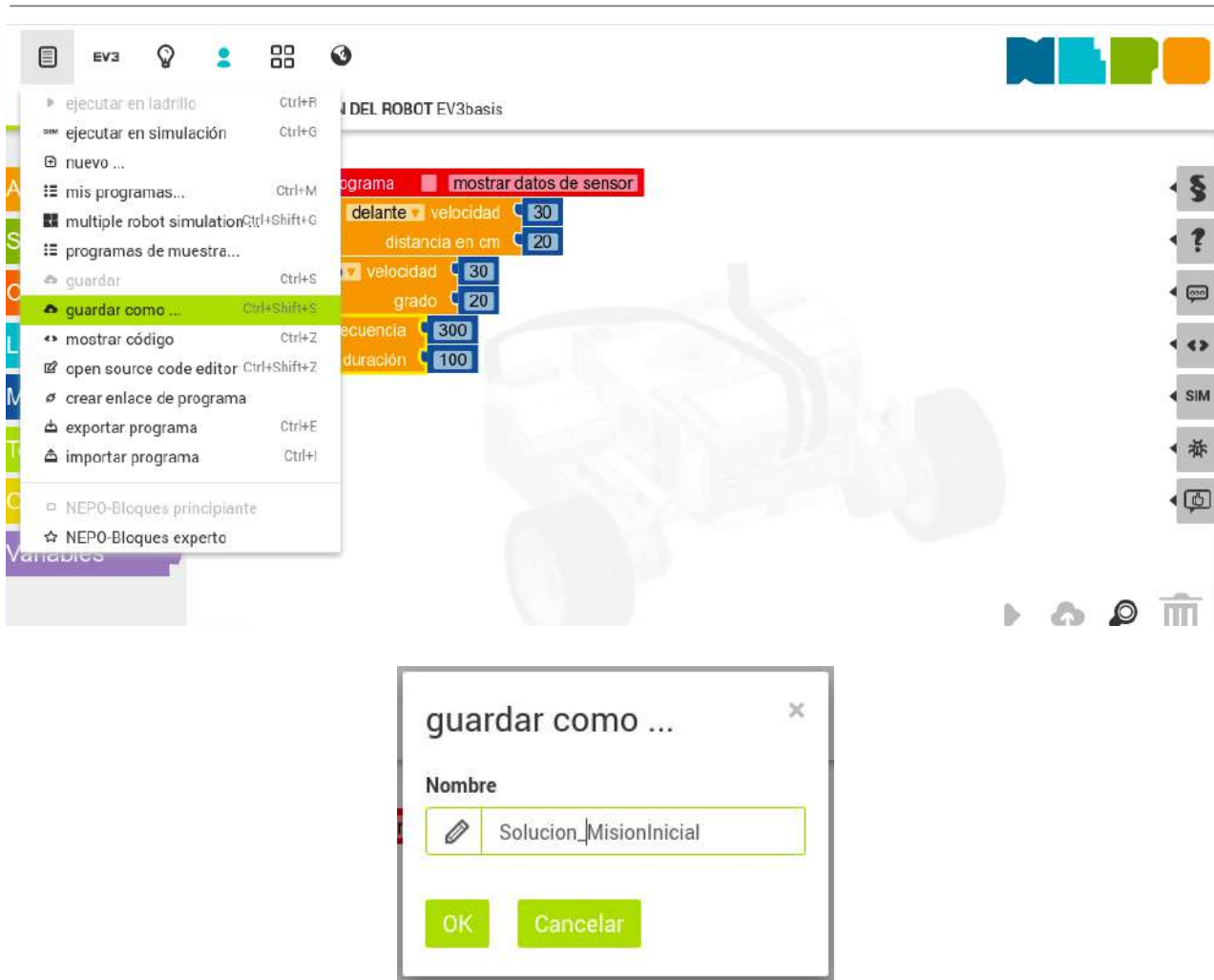
PROGRAMA NEPOprog CONFIGURACIÓN DEL ROBOT EV3basis

NAME OF THE USER GROUP Grupo A

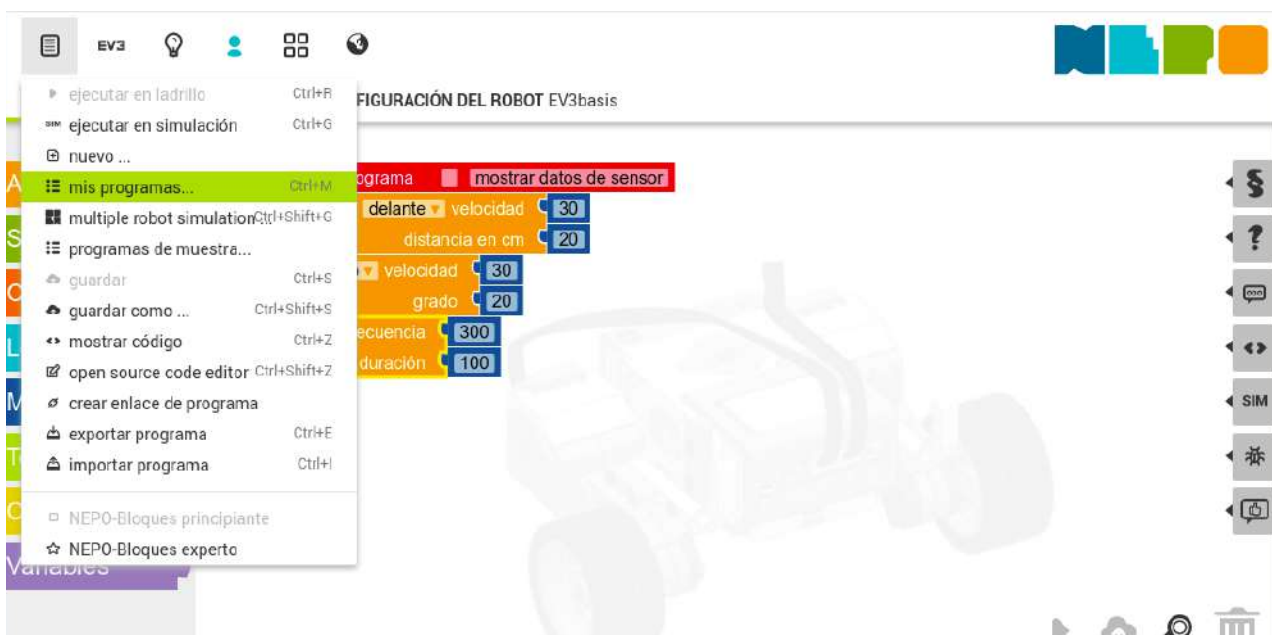
USUARIO	CONTRASEÑA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alumnd01	Grupo A:01	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
02	Grupo A:02	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
03	Grupo A:03	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
04	Grupo A:04	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Una vez hecho esto, ya tenemos nuestro grupo creado con un usuario y contraseña asignados a cada miembro del grupo.

Ahora podemos volver a la pestaña de programa y guardar la solución de la misión inicial. Para guardar un programa, abrimos el menú de la izquierda y seleccionamos 'guardar como'; e introducimos el nombre del programa.



Una vez está guardado, entramos en la pestaña de 'Mis programas', tal y como aparece en la siguiente imagen.



Se abre la ventana donde aparece el listado de los programas creados por mi usuario. A la derecha, aparece la opción de compartir el programa.



Compartir Programa

Compartimos el programa con el Grupo A. De esta forma todos los miembros del grupo A pueden ver este programa.

Compartir »Solucion_MisionInicial«

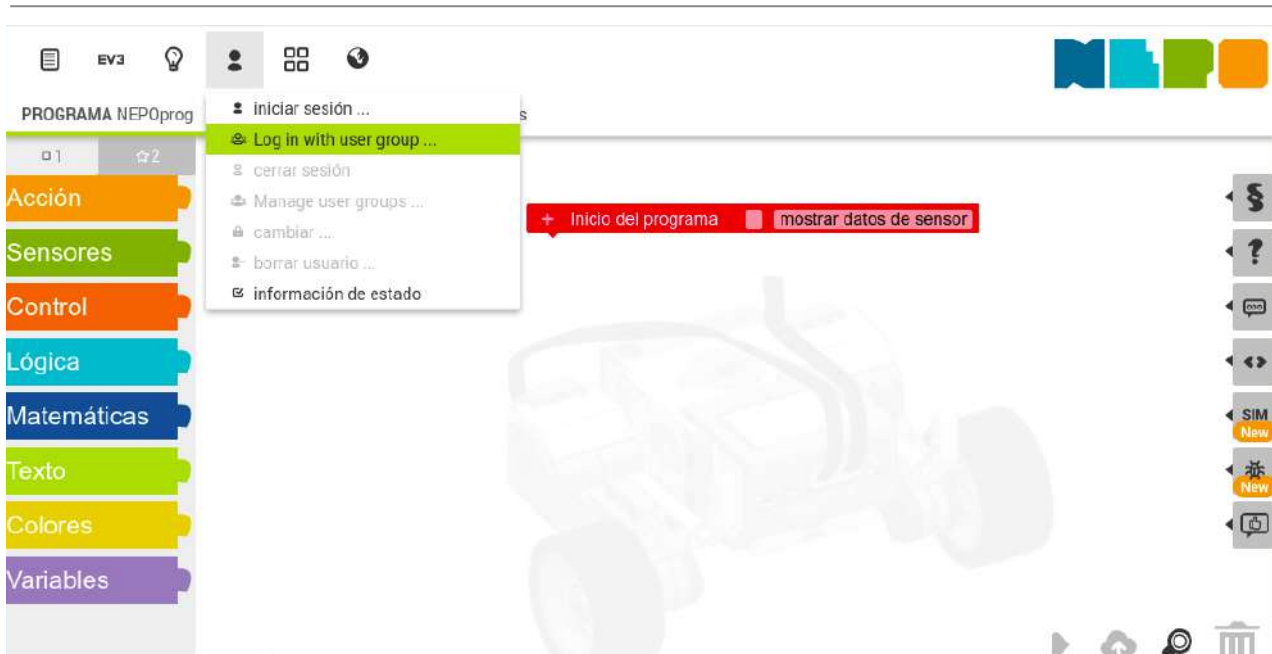
COMPARTIDO CON			
	<input type="text" value="Share your program with another user ..."/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	<input type="text" value="Grupo A"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>



En la pestaña de Gestión de grupos, aparece que el programa Solucion_MisionInicial está compartido con el grupo A.



A continuación, se muestra el proceso de registro de un miembro del grupo A.



Log in with user group ×

...

Name of the owner of the user group

 elenaElectronica

Name of the user group

 Grupo A

Nombre de usuario

 Alumno01

Contraseña



OK

[resetear password ...](#)

cambiar contraseña ... ×

The password for your account is not safe. Set your own password, so that only you can login to your account.

Nueva contraseña



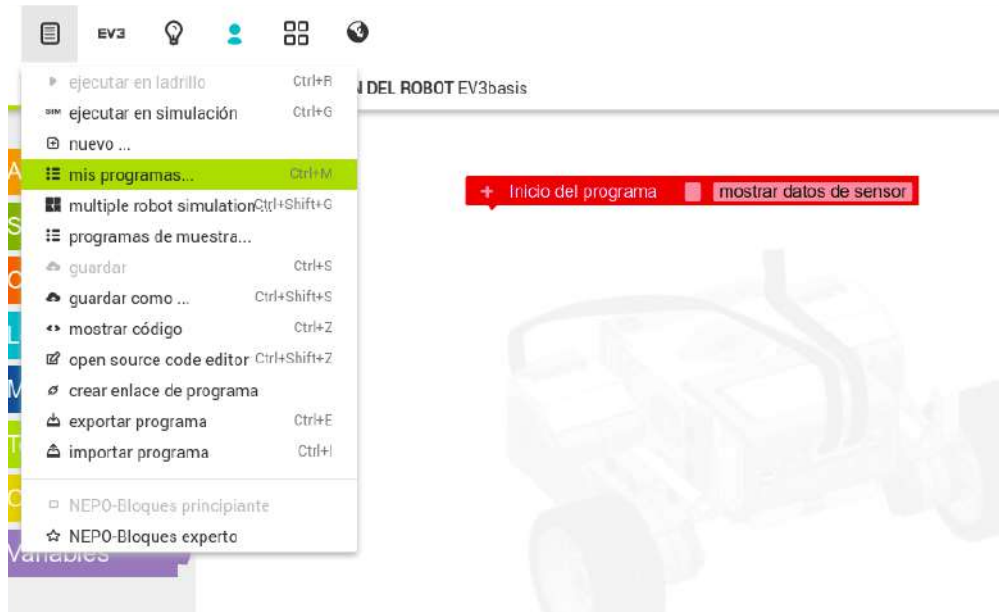
Repetir contraseña



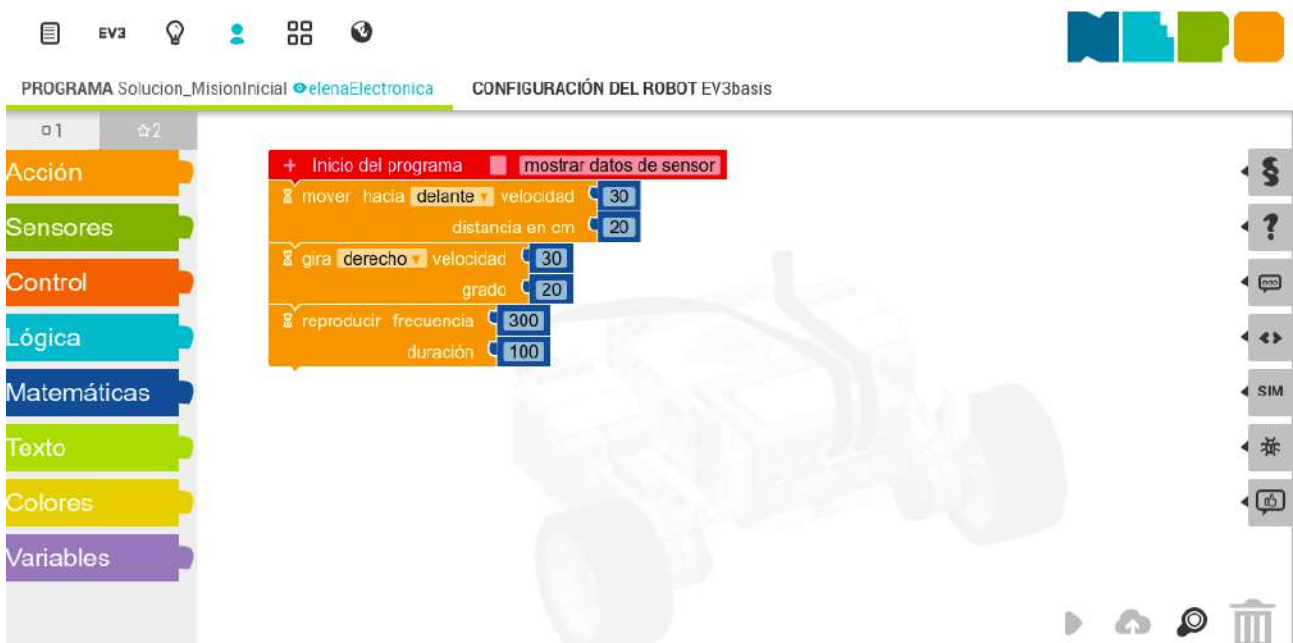
OK

Cancelar

El usuario, puede cambiar su contraseña la primera vez que se registra. Al entrar en 'Mis programas', podrá ver la solución de la misión inicial y cargar ese programa en su entorno de programación.



Cargar el programa



Cuando el alumno, modifique o cree un nuevo programa, podrá guardarlo con un nombre nuevo (no permite modificar el programa compartido) a través de la opción 'Guardar como'.

The screenshot shows the NEPO software interface. On the left, a menu is open with the option 'guardar como ...' (Save as...) highlighted. The main area displays a block-based programming environment for a robot configuration, titled 'CONFIGURACIÓN DEL ROBOT EV3basis'. The blocks include 'mostrar datos de sensor', 'hacia delante' (forward) with velocity 30 and distance 20, 'detener' (stop), 'hacia atrás' (backward) with velocity 30 and distance 20, 'girar' (turn) with frequency 300 and duration 100, and 'hacia delante' (forward) with velocity 30 and distance 20. The bottom status bar shows 'PROGRAMA MisionInicial_Alumno01' and 'CONFIGURACIÓN DEL ROBOT EV3basis'.

En tu cuenta de profesor, puedes ver todos los programas que han creado los usuarios del grupo, por tanto, podrás ver que el Alumno01 ha realizado un programa.

The screenshot shows the NEPO software interface. At the top, a dropdown menu is open with the option 'mis programas...' (my programs...) selected. The main area displays a table listing programs created by users in the group. The table has columns: NOMBRE DEL PROGRAMA, CREADOR, FECHA DE CREACIÓN, and ÚLTIMA ACTUALIZACIÓN. The table shows two programs: 'MisionInicial_Alumno01' created by 'Grupo A:Alumno01' and 'Solucion_MisionInicial' created by 'elenaElectronica'.

NOMBRE DEL PROGRAMA	CREADOR	FECHA DE CREACIÓN	ÚLTIMA ACTUALIZACIÓN
MisionInicial_Alumno01	Grupo A:Alumno01	23.05.2021, 19:14	23.05.2021, 19:14
Solucion_MisionInicial	elenaElectronica	23.05.2021, 18:57	23.05.2021, 18:57

Además te aparece la hora y el día en que el alumno lo ha creado. Puedes entrar en el programa y ver que todo funciona correctamente.

Con esta herramienta, se puede gestionar el desarrollo y actividades que ha completado cada alumno.

9.- Propuestas de solución para las misiones

Misión 01

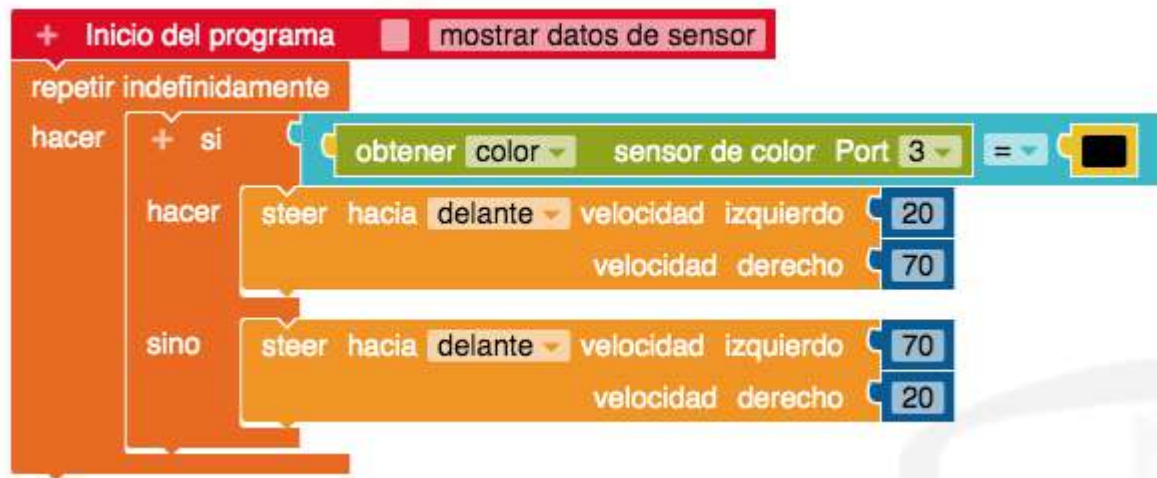


Misión 02



Misión 03

Solución Normal



Solución nivel Avanzado

