



FACULTAD DE EDUCACIÓN DE PALENCIA
UNIVERSIDAD DE VALLADOLID

PROPUESTA DE INTRODUCCIÓN DE LA PROGRAMACIÓN EN 6º CURSO DE EDUCACIÓN PRIMARIA

TRABAJO FIN DE GRADO
EN EDUCACIÓN PRIMARIA

AUTOR/A: FRANCISCO JAVIER GARCÍA LUENGO

TUTOR/A: MARÍA ASTRID CUIDA GÓMEZ

Palencia, 18 Junio 2019

RESUMEN

El presente Trabajo de Fin de Grado (TFG) constituye una propuesta de proyecto para la introducción de la programación en 6º curso de Educación Primaria. Para ello, y tras plantear la situación de la programación en Europa y en España, se plantea una propuesta metodológica que va de lo teórico a lo práctico.

Debido a las aportaciones que la programación puede hacer al resto de asignaturas o aspectos del currículo, el trabajo se plantea bajo un claro prisma de transversalidad.

Dada la edad y las características de los alumnos junto con las opciones que plantea el software y el hardware elegidos, he optado por utilizar metodología STEAM, que, además de conectar el mundo teórico con el práctico, fomenta la creatividad y la participación activa del alumnado, junto con una gamificación apoyada por una web que refleje los progresos y logros de los alumnos en esta propuesta de proyecto de introducción de la programación en un aula de 6º curso de Educación Primaria.

Palabras clave: programación, lógica, TIC, transversalidad, STEAM, gamificación, Pensamiento Computacional, Educación Primaria, Cody&Roby, Blockly, CODE.org, Micro:bit.

ABSTRACT

The following Final Degree Dissertation is a project proposal to introduce coding in Key Stage 3 (UK) or 6th grade (USA). To do so, and after considering the situation of coding in Europe and Spain, a methodological proposal, from a theoretical to a practical approach, will be proposed.

Due to the contributions that coding can provide to the rest of the curriculum subjects, the project is considered under a clear crosscutting perspective.

Given the age and characteristics of the students along with the options posed by the software and hardware chosen, I opted for using a STEAM methodology, that, as well as connecting the theoretical world with the practical one, promotes creativity and an active participation from students, together with a gamification supported by a web that shows the advances and achievements of the students in this project proposal for the introduction of coding in a Key Stage 3 (UK) or 6th grade (USA) classroom.

Keywords: coding, logic, ICT, crosscutting elements, STEAM, gamification, Computational Thinking, Primary Education, Cody&Roby, Blockly, CODE.org, Micro:bit.

Tabla de contenido

1. JUSTIFICACIÓN.....	5
2. PROGRAMACIÓN.....	5
A. Definición.....	5
B. Efectos cognitivos del aprendizaje de la programación.....	6
C. Influencia de la programación en el desarrollo del Pensamiento Computacional.....	7
D. La programación en Educación Primaria.....	8
i. Postura de la Unión Europea respecto a la enseñanza de la programación.....	8
ii. Situación de la programación en el currículo.....	9
iii. Proyectos sobre Programación desarrollados o que se están llevando a cabo en Ed. Primaria en Castilla y León	10
3. OBJETIVOS PERSEGUIDOS.....	11
A. General.....	11
B. Específicos.....	11
4. CONTENIDOS TRABAJADOS.....	12
A. Actividades desenchufadas.....	12
i. Juegos de seguir-dar explicaciones.....	12
ii. Robots Humanos.....	12
iii. Cody&Roby.....	13
B. Actividades enchufadas.....	13
i. Blockly.com.....	13
ii. Code.org.....	13
iii. Micro:bit.....	14
5. COMPETENCIAS DEL GRADO DE EDUCACIÓN PRIMARIA A DESARROLLAR..	15
A. Competencias del Grado.....	15
B. Módulo Didáctico-Disciplinar.....	15
C. Módulo de Prácticum y Trabajo Fin de Grado.....	16
6. PROPUESTA METODOLÓGICA.....	16
A. Contexto del centro.....	16
B. Principios metodológicos más relevantes.....	17
C. Dinámica de actividades propuesta.....	19
i. Actividades desenchufadas.....	20
1. Simplificar comandos / dar y seguir explicaciones.....	20
2. Robots humanos.....	25
3. Cody&Roby.....	30

ii.	Actividades enchufadas.....	37
1.	Introducción a la programación basada en bloques – Blockly games.....	37
2.	Trabajando la programación basada en bloques - Code.org.....	42
3.	Micro:bit.....	49
D.	Evaluación.....	62
7.	CONCLUSIONES.....	63
8.	REFERENCIAS BIBLIOGRÁFICAS.....	64
9.	ANEXO I.....	70
10.	ANEXO II.....	73
11.	ANEXO III.....	80
12.	ANEXO IV.....	82
13.	ANEXO V.....	89
14.	ANEXO VI.....	96
15.	ANEXO VII.....	102
16.	ANEXO VIII.....	116
17.	ANEXO IX.....	129
18.	ANEXO X.....	132

1. JUSTIFICACIÓN

La programación de computadores es una competencia básica cuya importancia, ya de por sí grande, se espera aumente en el futuro. De hecho, y siguiendo a Rushkoff (2010) y a Jenkins (2006), “en el mundo actual, pero sobre todo, en un futuro no muy lejano, saber programar va a ser una competencia básica que toda persona va a necesitar dominar no sólo para encontrar un trabajo, sino también para desenvolverse en su día a día”. Además, Davidson (2011) afirma que alrededor del 65% de los alumnos que acceden actualmente a la escuela podrían tener en un futuro trabajos que todavía no existen.

Es por ello que creo fundamental iniciar y trabajar la programación con los alumnos, no sólo de Ed. Primaria, sino de cualquier ciclo educativo.

La programación, por otra parte, cumple un requisito que a mi modo de ver y por mi experiencia personal en el aula, es capital: convierte a los alumnos de consumidores de tecnología en creadores de la misma. Este aspecto es algo fundamental, ya que los alumnos pasan de la teoría a la práctica, de ver a hacer, de tener una actitud pasiva en su proceso de enseñanza/aprendizaje a una activa.

Este TFG ha sido enfocado teniendo en mente tres cuestiones básicas: la transversalidad, la gamificación y la metodología STEAM. Considero la transversalidad como algo esencial en cualquier aula, pues permite a los alumnos conectar entre sí los distintos contenidos aprendidos en las diferentes asignaturas, realizando de esta manera un aprendizaje más ajustado a la realidad, más duradero en el tiempo y más significativo. En cuanto al carácter lúdico, pienso que es primordial que los alumnos se diviertan y disfruten en el aula. Para ello, he utilizado una gamificación que, como iré desgranando en esta propuesta de proyecto, lo impregna en su totalidad. La metodología STEAM permitirá desarrollar una dinámica de aprendizaje que vaya del desarrollo teórico a la práctica. Del mundo de las ideas al mundo real.

2. PROGRAMACIÓN

A. DEFINICIÓN

Según LaBoda (1985), la programación es “el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales”. Siguiendo a este autor, el propósito de la programación sería el de “crear programas que exhiban un comportamiento deseado”, es decir,

que estén diseñados de tal forma que realicen exactamente aquello que el programador quiere que realicen.

B. EFECTOS COGNITIVOS DEL APRENDIZAJE DE LA PROGRAMACIÓN

Existen muchas investigaciones que enumeran los efectos cognitivos positivos de utilizar la programación en el aula de Ed. Primaria. Según estas investigaciones, la programación fomenta:

- El aprendizaje de estrategias para solucionar problemas, diseñar proyectos y comunicar ideas (Resnick, 2013).
- La capacidad de atención, la autonomía y el placer por el descubrimiento de nuevos conceptos (Clements, 1986).
- La creatividad (Liao y Bright, 1991; Clements, 1995).
- La respuesta emocional y la creatividad de niños con dificultades de aprendizaje (Douglas y Swaminathan, 1995).
- La socialización entre compañeros (Wartella y Jennings, 2000).
- La creación de menos estereotipos de género en relación a las carreras STEM (Burke y Mattis, 2007).

En cuanto al área de Matemáticas en concreto, y siguiendo a Ross y Howe (1981) cuando sintetizan los postulados de Feurzeig et al's (1959), la programación:

- Da ideas claves sobre ciertos conceptos matemáticos.
- Proporciona un contexto para la resolución de problemas y un lenguaje con el cual el alumno puede describir dicha resolución de problemas.
- Proporciona habilidades para justificar e ilustrar las explicaciones matemáticas con más rigor formal.
- Anima a los alumnos a estudiar matemáticas desde un punto de vista más exploratorio.

Sin embargo, y como postula Scherer (2016), hay que resaltar que la mayoría de estos estudios e investigaciones son de corte teórico, por lo que, aun defendiendo ideas razonables, todavía faltan estudios con un matiz más empírico.

C. INFLUENCIA DE LA PROGRAMACIÓN EN EL DESARROLLO DEL PENSAMIENTO COMPUTACIONAL

Una de las herramientas que ayudan al desarrollo del Pensamiento Computacional es la programación. Por lo tanto, creo que es pertinente por mi parte incluir algunos apuntes sobre él en esta propuesta.

Lo primero que hay que matizar es que no hay un consenso internacional sobre la *definición* de Pensamiento Computacional. Destaco, por su relevancia, las siguientes definiciones:

- “El Pensamiento Computacional son aquellos procesos de pensamiento involucrados en la formulación de problemas y sus soluciones de tal forma que las soluciones estén representadas en una forma que pueda ser efectivamente llevada a cabo por un agente de procesamiento de información” (Cuny et al., 2010).
- “El pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano basándose en los conceptos fundamentales de la informática” (Wing, 2006).

En cuanto a su *relación* con la programación, comentar que durante la programación los estudiantes están expuestos a elementos tales como abstracción, depuración, remezcla e iteración para resolver problemas, elementos que, para diversos autores, también están relacionados con el Pensamiento Computacional (Brennan y Resnick, 2012; Ioannidou, Bennett, Repenning, Koh, y Basawapatna, 2011; Wing, 2008).

Finalmente, entre las *ventajas* que ofrece desarrollar el Pensamiento Computacional en nuestros alumnos, podemos mencionar las siguientes:

- El pensamiento computacional está en línea con muchos aspectos de las competencias del siglo XXI, tales como la creatividad, el pensamiento crítico y la resolución de problemas (Ananiadou y Claro, 2009; Binkley et al., 2012). Una de nuestras labores como docentes es asegurarnos de que nuestros alumnos adquieren las habilidades que sabemos que van a necesitar en su futuro.
- Lu y Fletcher (2009) defienden que utilizar lenguaje de Pensamiento Computacional permite a los alumnos entender procesos computacionales a la vez que desarrollan las habilidades de abstracción y representación de la información.

- La utilización del Pensamiento Computacional en el aula mejora la capacidad de pensar de manera más sistemática (Kafai y Burke, 2013) y la pericia matemática y científica (Sengupta, Kinnebrew, Basu, Biswas y Clark, 2013).

D. PROGRAMACIÓN EN EDUCACIÓN PRIMARIA

i. Postura de la unión europea respecto a la enseñanza de la programación

El 25 de julio de 2014, Neelie Kroes y Androulla Vassiliou, Comisionados Europeos de Agenda Digital y Educación respectivamente, enviaron a los Ministros de Educación de toda Europa una carta instándoles a introducir la programación en las aulas. En la carta se afirmaba que, de este modo, se reduciría drásticamente la ratio de paro juvenil además de reducirse la brecha entre competencias y puestos de trabajo en el sector de las nuevas tecnologías. Uno de sus argumentos de más peso señalaba que “la programación no solo ayuda a los estudiantes con las matemáticas, la ciencia, la tecnología y la ingeniería, sino que también les ayudará directamente a desarrollar habilidades transversales, tales como el pensamiento analítico, la resolución de problemas, el trabajo en equipo y la creatividad”.

Desde ese momento, la Comisión Europea ha seguido insistiendo en dicho aspecto promoviendo varias iniciativas dentro del mundo educativo que nos indican la importancia que para ella tiene este tema. Entre todas, destaco las siguientes:

- Lanzando un **portal** dedicado a alumnos y docentes con multitud de herramientas para programar:

<http://www.allyouneediscodes.eu/>

- Publicando el documento **Comunicación de la Comisión al Parlamento Europeo, al Consejo, al Comité Económico y Social Europeo y al Comité de las Regiones** (2016), en el que se afirmaba que “la transformación digital está cambiando la estructura del mercado de trabajo y la propia naturaleza del trabajo. Preocupa el que estos cambios puedan afectar a las condiciones del empleo, los niveles y distribución de los ingresos. Abordar estos desafíos requiere un diálogo global sobre los aspectos sociales de la digitalización en el que participen todas las partes interesadas relacionadas con todos los aspectos del trabajo, la educación y la formación”. En dicho documento, se presenta una serie de medidas para llevar a cabo la mejora de las habilidades digitales requeridas en el mismo.

- Publicando el documento **Agenda Europea de Nuevas Habilidades** (2016), que versa, entre otras cuestiones, sobre cómo mejorar la calidad y la relevancia de las habilidades en la formación. En el punto 3 de esta agenda, se hace referencia a que la meta debe ser mejorar la competencia digital de la mayoría de la población y no sólo de los profesionales de las Ciencias de la Computación.
- Promoviendo en el ámbito educativo la celebración de la **Semana del Código**, cuyo objetivo fundamental es acercar las Ciencias de la Computación a los alumnos a través de actividades sencillas. Dentro de la Semana del Código también se puede celebrar la **Hora del Código**, en la que se anima a los docentes a que acerquen la programación a sus alumnos durante 1 hora durante la celebración de esa semana. Hay multitud de recursos ya preparados para facilitar la celebración de ambas:

<https://code.org/learn>

<https://codeweek.eu/resources/teach>

Esta iniciativa fue lanzada por primera vez en 2013 y su meta es alcanzar el 50% de las escuelas europeas para 2020.

ii. Situación de la programación en el currículum

En Europa

En 2015, según European Schoolnet (organización sin ánimo de lucro que agrupa a 34 Ministerios Europeos de Educación y cuya meta es promocionar la innovación en educación), los siguientes 15 países de la Unión Europea ya habían integrado la programación en su currículum: Austria, Bulgaria, República Checa, Dinamarca, Estonia, Francia, Hungría, Irlanda, Lituania, Malta, España, Polonia, Portugal, Eslovaquia e Inglaterra. Ese mismo año, Finlandia anunció que la integraría en 2016, mientras que Bélgica (más en concreto la región de Flandes) estaba inmersa en un debate sobre su inclusión o no, cosa que finalmente hizo.

De esos 15 países, 8 la incluían a nivel de Ed. Primaria: Bélgica, Estonia, Finlandia, Francia, Israel, Eslovaquia, España y Reino Unido.

En España

Ni en la legislación educativa vigente en Educación Primaria (Ley Orgánica 8/2013, de 9 de diciembre, para la Mejora de la Calidad Educativa / LOMCE) ni en su desarrollo (Real Decreto 126/2014, de 28 de febrero, por el que se establece el currículum básico de la Educación Primaria) se hace mención explícita a la inclusión de la programación dentro del currículum.

Sin embargo, algunas Comunidades Autónomas han aprovechado la oportunidad que la LOMCE brinda al permitirles fijar una serie de asignaturas de libre configuración (artículo 9, punto 4). De este modo:

En Cataluña, el Decreto de Ordenación de las Enseñanzas para la Educación Primaria, establece en sus orientaciones metodológicas que “la programación provee de una serie de habilidades que favorecen la adquisición de estrategias que mejoran la resolución de problemas, el diseño de estrategias, el razonamiento, la justificación de las opciones tomadas, el trabajo en grupo, la toma de acuerdos y la comprensión de las razones de los otros”.

La Comunidad de Madrid, en el anexo III de su Decreto por el que se establece el Currículo de la Educación Primaria, hace referencia al aprendizaje de los “fundamentos de la programación y a la creación de pequeños programas informáticos utilizando el lenguaje de programación Scratch”.

La Comunidad de Navarra, establece en el artículo 7.7 de su Currículo de las Enseñanzas de Educación Primaria que “los centros educativos establecerán a través de proyectos acciones destinadas a la adquisición y mejora de las destrezas básicas de programación informática”.

En Castilla y León

De momento, en nuestra Comunidad Autónoma la programación sólo ha sido introducida en el currículo de 3º y 4º de Educación Secundaria Obligatoria a través de la Orden EDU/589/2016, de 22 de junio.

iii. Proyectos sobre programación desarrollados o que se están llevando a cabo en educación primaria en Castilla y León

La Junta de Castilla y León ha realizado o está realizando una serie de proyectos educativos y actividades relacionadas con la programación:

- *Ingenia Primaria*: proyecto de innovación educativa que pretende desarrollar las competencias de los alumnos a través del uso de la programación aplicada a robots y a las nuevas metodologías.
- *Proyecto Creando Código*: proyecto educativo en colaboración con Telefónica que pretende que los docentes adquieran una serie de competencias sobre colaboración, comunicación, creatividad, pensamiento crítico y resolución de problemas a través de, entre otras, la programación.

- *Escuela de Pensamiento Computacional*: proyecto educativo cuyo objetivo fundamental es ofrecer recursos educativos y formación a los docentes para que éstos utilicen con garantías de éxito la programación en su aula.
- *Proyecto de innovación TIC STEAM*: proyecto educativo realizado durante el curso 2016/2017 que pretendía formar a los docentes en programación para que implementaran la misma en su aula mediante la creación de proyectos utilizando metodología STEAM.
- *Clementoni*: proyecto educativo que pretende, a través de la utilización de la programación aplicada a robots, dotar a los alumnos de herramientas y competencias para luchar contra el bullying.
- *Escuela de Verano (Itinerario II de Escuela aprendizaje TIC 4.0)*: curso que se desarrollará durante los días 1, 2 y 3 de Julio del curso 2018/2019 sobre "Robótica y Programación" y que pretende ser una introducción para los docentes en ambos mundos con la intención última de que éstas sean introducidas en el aula.
- *Cursos de formación en los CFIEs Provinciales*: en los 13 CFIEs provinciales de la región se están impartiendo en la actualidad multitud de cursos referidos a la introducción de la programación en el aula.

3. OBJETIVOS PERSEGUIDOS

A. GENERAL

- Realizar una propuesta metodológica que ayude y motive a los docentes a introducir la programación en su aula.

B. ESPECÍFICOS

- Conocer el papel de la programación en entornos educativos en Europa y en España.
- Dotar a los docentes de ejemplos de uso de la programación en el aula para que les sirvan de inspiración y modelo.
- Animar a docentes a introducir la programación en su aula a través de herramientas sencillas, gratuitas, motivadoras y que fomentan la creatividad.
- Difundir algunas de las herramientas más utilizadas en el mundo para introducir la programación en el aula.
- Enseñar a los alumnos cómo crear un programa sencillo utilizando una metodología adaptada a su edad, nivel y capacidades.

- Enseñar a los alumnos algunos de los elementos más característicos en programación: bucles, condicionales, variables y funciones.
- Animar a los docentes a introducir la metodología STEAM en sus aulas.
- Dotar a los docentes de un ejemplo de actividad gamificada.

4. CONTENIDOS TRABAJADOS

De forma general, estos han sido los contenidos trabajados en las distintas actividades propuestas:

A. ACTIVIDADES DESENCUFADAS

i. Juegos de seguir-dar explicaciones

1. Actividades iniciales: simplificando comandos

En las 3 actividades:

- Lengua: Estructuras básicas de gramática.
- Lengua: Tiempos verbales: el imperativo.
- Lengua: Sinónimos y antónimos.

2. Actividades de seguir/dar instrucciones

- Actividad 1: nivel fácil:
 - E.F: Conciencia y control del cuerpo.
 - E.F.: Hábitos saludables.
 - E.F.: Prevención de lesiones en la actividad física: el calentamiento.
- Actividad 2: nivel medio.
 - Lengua Inglesa: Estructuras básicas de gramática inglesa.
 - Lengua Inglesa: Tiempos verbales: el imperativo.
 - Lengua Inglesa: Sinónimos y antónimos.
- Actividad 3: nivel difícil.
 - Competencia digital: Elementos básicos de un editor de textos: texto, imágenes y enlaces.

ii. Robots Humanos

- **Actividad 1: nivel fácil:**
 - Matemáticas: Números romanos (trabajados en 5º).

- **Actividad 2: nivel medio:**
 - a. Matemáticas: Operaciones matemáticas básicas: sumas, restas, multiplicaciones y divisiones (repaso de cursos anteriores).
- **Actividad 3: nivel difícil:**
 - a. Lengua Inglesa: Formas en pasado de los verbos regulares e irregulares (repaso de cursos anteriores).

iii. **Cody&Roby**

En las 3 actividades:

- a. Matemáticas: Algoritmos.

B. ACTIVIDADES ENCHUFADAS

i. **Blockly.com**

- **Rompecabezas:**
 - a. Ciencias Naturales: Características básicas de algunos animales.
- **Laberinto:**
 - a. Matemáticas: Bucles y condicionales, ambos conceptos referidos a programación.
 - b. E.F.: Nociones espaciales básicas.
- **Pájaro:**
 - a. Matemáticas: Condicionales en programación; grados contenidos en una circunferencia; representación gráfica de los conceptos mayor y menor; representación gráfica de los ejes cartesianos.
- **Tortuga:**
 - a. Matemáticas: Bucles en programación; figuras geométricas básicas; grados contenidos en una circunferencia.
 - b. E.F.: Nociones espaciales básicas.
- **Música:**
 - a. Matemáticas: Funciones en programación.
 - b. Música: Notas musicales: notación, valor y representación gráfica de las mismas.

ii. **Code.org**

Las 11 sesiones comparten un mismo objetivo:

- a. Lengua Inglesa: Vocabulario y estructuras gramaticales básicas.

- **Sesión 1**
 - a. Matemáticas: Errores (bugs) en programación.
- **Sesión 2**
 - a. Matemáticas: Bucles.
- **Sesión 3**
 - a. Matemáticas: Bucles anidados.
- **Sesión 4 y 5:**
 - a. Matemáticas: Bucles condicionados.
- **Sesión 6, 7 y 8:**
 - a. Matemáticas: Condicionales.
- **Sesión 9, 10 y 11:**
 - a. Matemáticas: Funciones.

iii. **Micro:bit**

Las 6 sesiones comparten un mismo objetivo:

- a. Ed. Artística: Elaboración de producciones tridimensionales teniendo en cuenta la aplicación de estrategias creativas, la responsabilidad en el trabajo cooperativo y el respeto a las aportaciones de los demás.

- **Sesión 1: Creamos un saludo**
 - a. Lengua: creación de un saludo básico.
- **Sesión 2: Creamos un termómetro**
 - a. Matemáticas: Variables, condicionales y comparaciones.
- **Sesión 3: Creamos un cuenta pasos**
 - a. Matemáticas: Variables y funciones.
- **Sesión 4: Bola mágica de respuestas**
 - a. Matemáticas: Variables, funciones, condicionales y comparaciones.
- **Sesión 5: ¿Cómo andas de pulso?**
 - a. Matemáticas: Variables.
- **Sesión 6: Creamos una brújula**
 - a. Matemáticas: Variables, variables booleanas, comparaciones y grados contenidos en una circunferencia y su localización.

5. COMPETENCIAS DEL GRADO DE EDUCACIÓN PRIMARIA A DESARROLLAR

A. COMPETENCIAS DEL GRADO

De entre todas las competencias contenidas en la Orden ECI/3857/2007, de 27 de diciembre, por la que se establecen los requisitos para la verificación de los títulos universitarios oficiales que habiliten para el ejercicio de la profesión de Maestro en Educación Primaria, considero que el presente TFG desarrolla en mayor medida las siguientes:

- *Conocer las áreas curriculares de la Educación Primaria, la relación interdisciplinar entre ellas, los criterios de evaluación y el cuerpo de conocimientos didácticos en torno a los procedimientos de enseñanza y aprendizaje respectivos.*
- *Diseñar, planificar y evaluar procesos de enseñanza y aprendizaje, tanto individualmente como en colaboración con otros docentes y profesionales del centro.*
- *Conocer y aplicar en las aulas las tecnologías de la información y de la comunicación.*

Las dos primeras tienen una relación clara y directa con el aspecto transversal que impregna esta propuesta de proyecto, aspecto que, como ya he comentado con anterioridad, su inclusión me parece fundamental en cualquier aula; mientras que la última competencia tiene una relación más que evidente con la temática elegida para el mismo.

B. MÓDULO DIDÁCTICO-DISCIPLINAR

En cuanto a las competencias contenidas dentro del Módulo Didáctico-Disciplinar de la citada Orden, decir que me he centrado en estos dos aspectos:

- *Adquirir competencias matemáticas básicas:* me he focalizado en que los alumnos desarrollen, entre otros, los siguientes contenidos del Bloque 1 “Procesos, métodos y actitudes en matemáticas” de la legislación educativa vigente en Educación Primaria:
 - Planificación del proceso de resolución de problemas.

- Disposición para desarrollar aprendizajes autónomos y confianza en sus propias capacidades para desarrollar actitudes adecuadas y afrontar las dificultades propias del trabajo científico.
- Utilización de algoritmos estándar en los contextos de resolución de problemas y valoración de otras posibilidades de resolución.
- Utilización de los procedimientos matemáticos estudiados para resolver problemas en situaciones reales.
- Interés y curiosidad por el aprendizaje y utilización de las Matemáticas.
- *Plantear y resolver problemas vinculados con la vida cotidiana, sobre todo con la inclusión de las actividades que utilizan metodología STEAM.*

C.MÓDULO DE PRÁCTICUM Y TRABAJO FIN DE GRADO

Las competencias contenidas en la Orden ECI/3857/2007, de 27 de diciembre, que la presente propuesta de proyecto desarrolla sobre el Módulo de Prácticum y Trabajo Fin de Grado son las siguientes:

- *Ser capaces de relacionar teoría y práctica con la realidad del aula y del centro.*
- *Participar en la actividad docente y aprender a saber hacer, actuando y reflexionando desde la práctica, con la perspectiva de innovar y mejorar la labor docente.*
- *Adquirir hábitos y destrezas para el aprendizaje autónomo y cooperativo y promoverlo en los estudiantes.*

6. PROPUESTA METODOLÓGICA

A.CONTEXTO DEL CENTRO

El centro educativo en el que esta propuesta de proyecto se llevaría a cabo es el CEIP Padre Claret de la capital palentina. Digo “se llevaría a cabo” porque actualmente no estoy ejerciendo la docencia directa, ya que mi puesto de trabajo actual es el de asesor de formación en el CRFPTIC de Palencia; pero me parecía interesante ofrecer una visión general de mi centro de destino para que se puedan comprender mejor algunas de las decisiones tomadas en este TFG. Para conocer el contexto del centro, ver el **ANEXO I**.

B. PRINCIPIOS METODOLÓGICOS MÁS RELEVANTES

El presente TFG está impregnado de 3 principios metodológicos básicos: transversalidad, metodología STEAM y gamificación.

Transversalidad

Coincido con De la Vega (2012) cuando afirma que: “La transversalidad es un enfoque dirigido al mejoramiento de la calidad educativa, buscando superar la fragmentación de las áreas del conocimiento, a partir del desarrollo de una visión holística”. Es por ello que en la mayoría de los proyectos que he realizado en mi aula he incluido el aspecto transversal en la programación de los mismos, ya que creo firmemente que la transversalidad ayuda enormemente a los alumnos a que adquieran aprendizajes significativos y a que sitúen adecuadamente los contenidos de una asignatura, al compararlos y relacionarlos con los contenidos del resto de ellas.

Metodología STEAM

STEAM es un acrónimo formado por la primera letra de las siguientes disciplinas: Science, Technology, Engineering, Arts y Mathematics (en español, Ciencias, Tecnología, Ingeniería, Arte/Plástica y Matemáticas).

Meeth (1978) define la metodología STEAM como “una aproximación de enseñanza y aprendizaje interdisciplinar donde se parte de un problema, y en la que, mediante procesos de resolución de problemas, se aplican los conocimientos de varias disciplinas contribuyendo así a la resolución de dicho problema”.

La metodología STEAM busca relacionar, de una forma práctica, el mundo teórico con el mundo real a través de la participación activa de los alumnos. En este sentido, coincido con Cilleruelo y Zubiaga (2014) cuando aseveran que “la educación STEAM permite una aproximación al proceso de enseñanza/aprendizaje desde una posición activa impulsada por un juego experimental que promueve la ruptura de barreras entre disciplinas e incluye múltiples posibilidades en la encrucijada arte, ciencia y tecnología”.

Diversas organizaciones defienden y promueven la introducción de la metodología STEAM en el aula. Así:

- *European Schoolnet* ofrece soporte y recursos a los docentes a través de varias iniciativas:
 - Ofreciendo proyectos realizados siguiendo la metodología STEAM:

<http://www.eun.org/projects/stem>

- Ofreciendo ayuda, soporte, ejemplos, información, recursos,... a través del siguiente portal:

<http://scientix.eu/>

- La *Organización Educativa de Micro:bit* promueve la metodología STEAM a través del uso de su dispositivo Micro:bit.
- La organización internacional *FIRST* organiza en todo el mundo el concurso First Lego League, que mezcla robótica y metodología STEAM.
- La *Junta de Castilla y León*, a través del Proyecto de Innovación Educativa Ingeniería Primaria, ofrece formación a 30 centros durante el curso 2018/2019 utilizando material de Lego WeDo 2.0 para la realización de proyectos de robótica siguiendo la metodología STEAM.

Otro aspecto a tener muy en cuenta es que la metodología STEAM promueve y fomenta la *creatividad* entre el alumnado, ya que pone a disposición de los alumnos herramientas atractivas de creación, conocimientos para utilizar estas herramientas y una gran motivación.

Creo firmemente que la creatividad es una de las competencias más necesarias que deben desarrollar los alumnos del siglo XXI, pues será un aspecto fundamental en su futuro tanto educativo como laboral.

Gamificación

Según Ferrán Teixes (2014), la gamificación es “la aplicación de recursos de los juegos (diseño, dinámicas, elementos, etc.) en contextos no lúdicos para modificar comportamientos de los individuos mediante acciones sobre su motivación”.

Entre sus principales beneficios, podemos enumerar los siguientes:

- Incrementa la motivación y la implicación del alumnado (Hakulinen, Auvinen y Korhonen, 2017; Lee y Hammer, 2011; Muntean, 2011).
- Puede ser usada para proporcionar incentivos que desarrollen los comportamientos educativos deseados y para asegurar que estos comportamientos ayuden a los estudiantes a alcanzar los resultados de aprendizaje previstos (Lee y Hammer, 2011; Simões, Redondo y Vilas, 2013).

- Los alumnos mejorarán sus habilidades de resolución de problemas al aplicarlas y trabajarlas mediante el juego (Gee, 2003).
- Los alumnos, al enfrentarse en los juegos con sentimientos de decepción y curiosidad, se preparan para enfrentarse a los fracasos en su proceso de enseñanza/aprendizaje (Lazzaro, 2014).
- El uso de mecánicas del juego mejora un 40% la capacidad para aprender nuevas habilidades (Giang, 2013).
- La gamificación, normalmente, provee a los alumnos con feedback inmediato sobre sus logros, lo que facilita la comprensión y la retención de lo aprendido (Pandey, 2015).

En última instancia, tengo que admitir que la inclusión de la gamificación en mi propuesta de proyecto responde también a mi propia experiencia docente, ya que la he incluido en mi aula durante mis 3 últimos años en activo y tengo que decir que el resultado es muy satisfactorio, en cuanto a motivación por parte de los alumnos. Las actividades gamificadas, desde mi experiencia, dotan a éstas de un aura de juego, de situarte al margen de actividades más formales, por lo que los alumnos, sin darse cuenta, se involucran mucho más en ellas. El resultado son aprendizajes más significativos y duraderos en el tiempo.

La propuesta de gamificación cuenta con una web (<https://hackersdelclaret.weebly.com/>) donde se reflejarán, entre otros, los progresos y los resultados de los alumnos. En el **ANEXO II** se puede encontrar más información sobre la gamificación aplicada.

C. DINÁMICA DE ACTIVIDADES PROPUESTA

Las actividades se han dividido en 2 grandes tipos: actividades “desenchufadas” y actividades “enchufadas”:

- Las actividades desenchufadas son aquellas en las que no se utilizan dispositivos electrónicos.
- Las actividades enchufadas son aquellas en la que se utilizan mini portátiles, PCs, Tablets o los dispositivos Micro:bit.

Para facilitar la navegación entre múltiples actividades propuestas, consulten el índice de las mismas que podrán encontrar en el **ANEXO III**.

Actividades desenchufadas

i. Simplificar comandos / dar y seguir explicaciones

1. Consideraciones previas

Saber ordenar una secuencia de comandos, así como saber seguir (decodificar) dichas órdenes, evitando errores en la lectura (debugging), es algo básico no ya sólo en matemáticas, si no en cualquier asignatura o ámbito dentro y fuera de la escuela. Por lo tanto, está claro que trabajar este aspecto con los alumnos es fundamental.

Para ello, vamos a realizar una serie de actividades sobre cómo secuenciar órdenes, basándonos en la premisa de que las órdenes para la realización de cualquier actividad tienen que ser lo más claras y sencillas posibles para minimizar los riesgos de error a la hora de decodificar y seguir estas órdenes.

Si los alumnos interiorizan los aspectos mencionados en el párrafo anterior, podrán escribir código denominado “limpio”, es decir, código que al ser más simple y claro permite su lectura de una forma más sencilla a la vez que los errores se identifican con mayor rapidez. En este sentido, hay que recordar que una de las máximas en programación es KISS (Keep It Simple Stupid).

Por lo tanto, esta propuesta de proyecto comienza con el trabajo previo de aprendizaje, por parte de los alumnos, de simplificar órdenes, para continuar posteriormente con el aprendizaje de dar y seguir instrucciones de forma simple pero eficiente.

Para motivar a los alumnos, evitando situaciones de miedo o ansiedad ante lo desconocido, comenzaremos por algo conocido: simplificar textos.

2. Actividades iniciales: simplificando comandos

Las 3 actividades propuestas comparten diversos aspectos:

Objetivos:

- Conocer las estructuras básicas de la lengua.
- Usar el vocabulario más simple en cada situación.
- Usar sinónimos y antónimos.
- Aprender la importancia de dar órdenes sencillas y claras para que te entiendan correctamente.
- Aprender a simplificar órdenes recibidas para comprenderlas mejor.
- Encontrar errores (bugs) en las órdenes (en el código).

- Corregir dichos errores (debugging).

Contenidos:

- Estructuras básicas de gramática.
- Tiempos verbales: el imperativo.
- Sinónimos y antónimos.

Temporalización: 30/40 minutos.

Espacios: aula.

Materiales: pizarra y cuadernos.

Evaluación gran grupo. El maestro pregunta a la clase si las órdenes utilizadas les parecen válidas al resto, es decir, si el “código de programación es correcto”. Se corrigen los posibles errores entre todos.

Gamificación: damos 1 punto para la gamificación “HACKERS DEL CLARET” a cada alumno que realice de forma correcta la actividad.

➤ *Actividad 1: comenzamos a simplificar frases*

Desarrollo: actividad individual.

- Comenzamos la actividad explicando la importancia de la claridad y la simpleza a la hora de dar y entender órdenes.
- Explicamos la dinámica del ejercicio: los alumnos van a recibir una serie de órdenes medianamente complejas y las tienen que simplificar para hacerlas lo más sencillas posibles de entender y de seguir.
- A modo de ejemplo, simplificamos la primera orden en la pizarra en gran grupo.
 - *Ve a mi mesa y acércame los papeles, Juan.* Se podría simplificar en 3 frases:
Juan, vete a mi mesa. Coge los papeles. Tráeme los papeles.
- Resto de oraciones a simplificar:
 - Desmonta la caja con el martillo de arriba, Leticia.
 - Sal a la calle Sara y cómprame la revista con el dinero que está encima de la mesa.
 - Dibuja un sol con 5 rayos en esa pizarra de ahí, José, con la tiza de color rojo.

- Corta ese cacho de plastilina Laura y construye una casa con él que tenga 1 puerta y 2 ventanas.

➤ *Actividad 2: avanzamos en la tarea de simplificar (textos sencillos).*

Desarrollo: actividad por parejas.

- Volvemos a recordar la importancia de la claridad y la simpleza a la hora de dar y entender órdenes.
- Recordamos la dinámica del ejercicio.
- Texto a simplificar: *Mario, enciende el ordenador y busca el icono de Word para escribir una redacción. La redacción tiene que tener 200 palabras y tratar sobre las plantas. Quiero que, una vez finalizada, me la envíes por correo electrónico.*

➤ *Actividad 3: avanzamos en la tarea de simplificar (textos más complejos).*

Desarrollo: actividad por parejas.

- Recordamos la dinámica del ejercicio.
- Texto a simplificar: *Para hacer una ensalada tienes que ir al frigorífico y coger todos los ingredientes. Los ingredientes son: tomates, lechuga, aceitunas y cebolla. Luego, y después de lavarlos bien, tienes que cortarlos en trozos no muy grandes y ponerlos en una ensaladera. No te olvides de añadir aceite, vinagre y sal. ¡¡Ya tienes lista la ensalada!!*

Variaciones/Ampliaciones: dependiendo de la respuesta de los alumnos, se podría continuar con algún ejercicio de simplificación más, aunque creo que con los 3 ejercicios propuestos más las actividades que les siguen, los alumnos interiorizarán sin problema los conceptos pretendidos. En el caso de necesitar más ejercicios sería interesante dejar que fueran los propios alumnos los que creasen los textos a simplificar, ya que involucrar al alumnado en su proceso de enseñanza/aprendizaje siempre es positivo.

3. Actividades de dar/seguir instrucciones

Una vez comprendida e interiorizada la necesidad y la manera de simplificar comandos, los alumnos pasarán de ser receptores de información a creadores. Para ello deberán elaborar sus propias secuencias de órdenes partiendo de una acción que nosotros les propongamos.

Las 3 actividades propuestas comparten diversos aspectos:

Objetivo común a las 3 actividades:

- Aprender la importancia de dar órdenes sencillas y claras para que te entiendan correctamente.

Temporalización: 1 hora.

Materiales: cuadernos y cámara, webcam, Tablet o Smartphone.

Desarrollo: actividad por parejas.

- Las mejores actividades se grabarán en vídeo y los vídeos resultantes se subirán a la web del proyecto.

Evaluación: se realizaría en gran grupo: el maestro, después de ver cada actividad, pregunta al resto de alumnos si el código (las órdenes) les han parecido correctas o se pueden mejorar (búsqueda de errores – debugging). Se corrigen los posibles fallos/errores entre todos.

Gamificación: damos 1 punto para la gamificación “HACKERS DEL CLARET” a cada alumno que realice de forma correcta la actividad.

- *Actividad 1: nivel fácil (realizar un calentamiento – repaso de E.F.).*

Objetivos propios de la actividad:

- Conocer y controlar el propio cuerpo.
- Adaptar hábitos saludables en referencia al mantenimiento del propio cuerpo.
- Valorar el calentamiento con una medio eficaz para prevenir lesiones en la actividad física.

Contenidos:

- Conciencia y control del cuerpo.
- Hábitos saludables.

- Prevención de lesiones en la actividad física: el calentamiento.

Espacios: gimnasio o patio.

Desarrollo:

- Recordamos a los alumnos que las órdenes dadas serán más efectivas e inducirán a un menor número de errores cuanto más sencillas sean.
- Explicamos la dinámica del ejercicio: los alumnos tienen que explicar con órdenes muy sencillas una rutina de calentamiento (repaso de cursos anteriores): calentamiento de tobillos, rodillas, cadera, cuádriceps, isquiotibiales, espalda, hombros, tríceps, pecho y cuello.

➤ *Actividad 2: nivel medio (realizar un dibujo sencillo utilizando comandos en Lengua Inglesa).*

Objetivos propios de la actividad:

- Conocer las estructuras básicas de la lengua inglesa.
- Usar el vocabulario más simple en cada situación.
- Usar sinónimos y antónimos.

Contenidos:

- Estructuras básicas de gramática inglesa.
- Tiempos verbales: el imperativo.
- Sinónimos y antónimos.

Espacios: aula.

Materiales: lapiceros y pinturas.

Desarrollo:

- Explicamos la dinámica del ejercicio: los alumnos van a tener que crear una serie de órdenes en inglés para dibujar uno de los siguientes objetos: perro, casa, coche y ordenador.

- *Actividad 3: nivel difícil (crear un documento utilizando un ordenador que contenga texto y una imagen prediseñada).*

Objetivos propios de la actividad:

- Repasar cómo se crea un documento básico en un editor de texto (Libre Office).
- Repasar cómo se añade texto en un documento.
- Repasar cómo se inserta una imagen en un documento.
- Repasar cómo se añade un hipervínculo a un texto en un documento.

Contenidos:

- Elementos básicos de un editor de textos: texto, imágenes e hipervínculos.

Espacios: aula.

Desarrollo:

- Explicamos la dinámica del ejercicio: los alumnos tienen que explicar con órdenes muy sencillas cómo crear un documento básico con un editor de texto (Libre Office). El documento debe constar de texto (al menos 5 líneas), una imagen prediseñada y dos enlaces (a elegir por los alumnos).

ii. Robots humanos

1. Consideraciones previas

A continuación, los alumnos pasarán a “programar” a sus compañeros como si fueran robots. Con la realización de estos ejercicios busco que los alumnos conozcan de primera mano (los robots son ellos mismos) la importancia que tiene el saber ordenar correctamente y sin error una serie de comandos.

Crearemos nuevamente 3 niveles de dificultad, dejando la posibilidad de que los alumnos se inventen sus propios ejercicios al finalizar los mismos o elegir una de nuestras propuestas de modificación de cada juego.

En cada actividad intento incluir contenidos de repaso del curso anterior ya sea de matemáticas o de otras asignaturas. Una de las razones de esto es que, al utilizar contenidos de otras asignaturas, podemos involucrar a más docentes, además de hacer ver a los alumnos que este tipo de ejercicios no tienen solamente aplicación en el área de matemáticas, si no en cualquier materia (además de estar relacionados intrínsecamente con el mundo real).

Las 3 actividades propuestas comparten diversos aspectos:

Objetivos comunes a las 3 actividades:

- Entender la dinámica y la mecánica del juego.
- Dar instrucciones precisas para lograr el objetivo.
- Seguir las instrucciones para lograr el objetivo.
- Reflexionar sobre la necesidad de las órdenes precisas cuando queremos que alguien (o algo) realice una acción determinada.

Contenido común a las 3 actividades:

- Órdenes básicas.

Temporalización: 1 sesión de 1 hora. Dependiendo de la respuesta y de las actividades realizadas, se podría ampliar a 1 hora y 30 minutos.

Espacios: aula y dependiendo del tamaño de la misma, se podría considerar utilizar el patio el pasillo o incluso el gimnasio.

Material común a las 3 actividades:

- Grid (realizado previamente en Plástica).
- Papel y bolígrafo.
- Cámara, webcam, Tablet o Smartphone.

Desarrollo:

- Elegimos a tres alumnos voluntarios y explicamos los papeles de cada participante en el juego:
 - o El primer alumno elige la respuesta a la que hay que llegar. Este alumno, mediante rúbrica, va a evaluar a sus dos compañeros.
 - o El segundo alumno localiza la respuesta y da las indicaciones para llegar a la misma.
 - o El tercer alumno sigue las instrucciones del segundo alumno para llegar a la respuesta elegida por el primero.
- Las órdenes posibles son **adelante, gira a la derecha, gira a la izquierda.**
- Una vez llegado al objetivo correspondiente, el primer alumno comprobará que el proceso se ha realizado correctamente (el objetivo al que han llegado es el correcto y se han seguido las indicaciones de modo adecuado).

- Una vez comprobado, se invierten los papeles para que los tres alumnos cumplan con los tres papeles de la actividad y puedan ser evaluados.
- Para hacer la actividad más difícil y entretenida, los diferentes objetivos estarán escritos en fichas que iremos poniendo en cada grid y cambiando de ubicación en cada turno. Podrán ver un ejemplo de fichas a utilizar para los diferentes juegos en el **ANEXO IV**.
- Se grabarán vídeos de los alumnos realizando la actividad para subir a la web del proyecto.

A la hora de crear las fichas, y al tratarse de contenidos de repaso, el docente podría actuar de dos modos:

- a) Crear él o ella las fichas con los números.
- b) Pedir a los alumnos que traigan una ficha creada por ellos mismos. Para dificultar el juego, los alumnos intercambiarán sus fichas con otro grupo.

Variaciones:

- Involucrar al alumnado en su proceso de enseñanza aprendizaje siempre es positivo, por lo que se podría demandar a los alumnos que creen una variación del juego.
- Los alumnos tienen que llegar al resultado y volver al punto de partida (START).
- Cambiamos el punto de partida (START) en cada turno.
- Los alumnos tienen que llegar a varios resultados en un mismo juego.
- Pedimos a los alumnos que den todas las indicaciones por escrito a sus compañeros.

Evaluación: Rúbrica de evaluación (ver **ANEXO V**).

Gamificación: daremos 1 punto a cada alumno por cada intervención correcta. El tercer alumno del grupo se encargará de verificar esta circunstancia.

Ejemplos de grid de las 3 actividades: ver **ANEXO IV**.

2. Actividad 1: nivel fácil

Los alumnos tendrán que llegar a diferentes números romanos.

Objetivo propio de la actividad:

- Repasar los números romanos.

Contenido propio de la actividad:

- Números romanos (vistos y trabajados el curso anterior).

Material propio de la actividad:

- Números romanos escritos en tarjetas (ver **ANEXO IV**).

Variaciones: para adaptar el juego a las necesidades de los alumnos, hacerlo más difícil podríamos:

- Utilizar grupos de números romanos determinados: utilizar sólo del 1 al 100 en un turno; del 1 al 10.000 en otro turno; del 1 al 100.000; del 100 al 10.000;...

3. Actividad 2: nivel medio

La dinámica de esta actividad sería la misma que la de la actividad anterior pero incluyendo “islas” en el grid, para aumentar la dificultad del ejercicio.

Objetivo propio de la actividad:

- Repasar las operaciones matemáticas básicas: sumas, restas, multiplicaciones y divisiones.

Contenido propio de la actividad:

- Operaciones matemáticas básicas: sumas, restas, multiplicaciones y divisiones (repasso de cursos anteriores).

Material propio de la actividad:

- Resultados de operaciones escritos en tarjetas (ver **ANEXO IV**).

Variaciones: para adaptar el juego a las necesidades de los alumnos, hacerlo más difícil podríamos:

- La situación de las “islas” puede ir cambiando en cada partida.
- Utilizar distintos bancos de operaciones, con operaciones cada vez más complicadas.

4. Actividad 3: nivel difícil

Seguiríamos utilizando las “islas” en el grid e introduciríamos el concepto de bucle. Un bucle es una acción que se repite un determinado número de veces. Los bucles son básicos en

programación ya que facilitan enormemente la tarea de programar además de permitir crear un código más “limpio” y, por lo tanto, con menos posibilidades de errores de lectura e interpretación. Un ejemplo de bucle podría ser el siguiente:

avanza x5 (el alumno debería avanzar 5 casillas en el grid).

En esta actividad repasaríamos las formas verbales inglesas en pasado (verbos regulares e irregulares). Los alumnos tendrían que encontrar la forma verbal en pasado correcta de un determinado verbo en presente. Para aumentar la dificultad del ejercicio, pondríamos en el grid dos formas verbales en pasado de un mismo verbo, una correcta y otra incorrecta. Ejemplo: Presente – LIVE / Pasado – LIVED (forma correcta) y LIVEN (forma incorrecta).

Objetivo propio de la actividad:

- Repasar las formas regulares e irregulares de los verbos ingleses en pasado.

Contenido propio de la actividad:

- Formas en pasado de los verbos regulares e irregulares en lengua inglesa (repaso de cursos anteriores). Ver **ANEXO IV**.

Material propio de la actividad:

- Formas verbales (correctas e incorrectas) en pasado de verbos regulares e irregulares.

Variaciones: para adaptar el juego a las necesidades de los alumnos, hacerlo más difícil podríamos:

- Utilizar distintos bancos de verbos con más verbos irregulares que regulares, con verbos cada vez menos comunes,...
- La situación de las “islas” puede ir cambiando en cada partida.

5. Actividades extra

Este tipo de actividades, dependiendo de la respuesta y aceptación por parte de los alumnos, se puede ir desarrollando a lo largo de todo trimestre, pues es un método sencillo, lúdico y motivante para repasar diversos contenidos trabajados los cursos anteriores. Así, se podrían repasar contenidos de:

- Matemáticas (descomposición de números, redondeo, unidades del Sistema Métrico Decimal, los ángulos y sus elementos, figuras planas, polígonos,...).

- Lengua Castellana (conjugaciones verbales, ortografía, conectores,...).
- Ciencias Naturales (aparatos y sistemas, hábitos saludables, ecosistemas,...).
- Ciencias Sociales (el sistema solar, el clima, el paisaje,...).
- Lengua Inglesa (rutinas, preposiciones, vocabulario,...).
- Educación Artística (notas y su representación gráfica, formas planas y espaciales, tipos de instrumentos,...).
- Educación Física (sistemas relacionados con la actividad física, juegos populares y tradicionales, hábitos saludables,...).

iii. Cody&Roby

1. Consideraciones previas

Para conocer las características de este juego, ver **ANEXO VI**.

Tras haber comenzado a trabajar la secuenciación de comandos con la actividad de Robots Humanos, el siguiente paso será afianzar dicha capacidad. Las actividades propuestas en este apartado nos posibilitarán la consecución de dicho objetivo.

En lo referente a *contenidos, temporalización, espacios, materiales, evaluación y gamificación* todas las actividades comparten la siguiente información:

Contenidos:

- Algoritmos del juego Cody&Roby.

Temporalización: 45 minutos / 1 hora.

Espacios: aula

Materiales:

- Juego de Cody&Roby: tablero, fichas (robots) y cartas de movimiento (la caja para guardar el material es opcional).
- Fichas y cartas de movimiento fotocopiadas y aumentadas.
- Masilla.
- Cámara, webcam, Tablet o Smartphone.

Desarrollo: se jugará por parejas.

- Uso de las cartas:

- Carta de avance: el robot avanza una casilla.
- Carta de giro derecha: el robot gira, dentro de la misma casilla y sin avanzar, hacia su derecha.
- Carta de giro izquierda: el robot gira, dentro de la misma casilla y sin avanzar, hacia su izquierda.
- Carta de bucle: indica las veces que un mismo movimiento se va a repetir (el número de veces se escribirá en el círculo de la esquina superior derecha de la carta).

Con la combinación de varias cartas, los alumnos lograrán que su robot vaya avanzando a través del tablero hasta lograr el objetivo pretendido.

- En el **ANEXO VII** se puede encontrar ejemplos de códigos.
- Los mejores códigos de cada actividad se grabarán en vídeo y se subirán a la web del proyecto.

Evaluación: observación directa por parte del docente de la resolución de cada actividad al exponer los alumnos a la clase sus códigos de programación del robot (resolución del ejercicio).

Gamificación: daremos 1 punto a cada alumno por cada código correcto o juego ganado.

2. Actividad 1: nivel fácil (tablero sin islas)

Planteamos esta actividad como modo de iniciarse en la mecánica del juego de Cody&Roby, por lo que el objetivo principal es que los alumnos interioricen el tamaño del tablero y cómo funcionan las cartas de movimiento.

Objetivos:

- Conocer el funcionamiento básico (algoritmos) del juego Cody&Roby: tablero y movimientos (“lenguaje de programación”).

Desarrollo:

- Presentamos a los alumnos el juego Cody&Roby.
- Explicamos la dinámica del juego: tendremos que crear una secuencia de movimientos ayudándonos de las cartas a nuestra disposición para llevar al robot desde la casilla de salida hasta la de meta.

- Hacemos una demostración práctica en la pizarra: dibujamos el tablero y nos ayudamos de fichas de robots y de cartas de movimiento (que pegaremos en la pizarra utilizando masilla) para lograr el objetivo pretendido.
- Pediremos a los alumnos que realicen la actividad varias veces instándolos a crear cada vez secuencias de movimiento más complejas para que se habitúen al uso y la mecánica de las cartas (incluso dando rodeos). Este ejercicio va en el sentido opuesto de los realizados en las anteriores actividades (en los ejercicios de “dar y recibir explicaciones” y “robots humanos” los alumnos tenían que intentar crear un código lo más sencillo y “limpio” posible), pero en esta ocasión nos interesa que los alumnos interioricen y conozcan bien la mecánica y el uso de las cartas de movimiento para que, en los ejercicios posteriores (niveles medio y difícil), tengan menos dificultades.

3. Actividad 2: nivel medio (tablero con islas)

Una vez interiorizada la mecánica del juego, introduciremos islas dentro del tablero para hacer el juego más interesante, motivante y difícil. Al contrario que en el modo fácil, esta vez primaremos la limpieza del código, por lo que instaremos a los alumnos a crear un código correcto utilizando el menor número posible de cartas.

Objetivos:

- Afianzar el conocimiento sobre las reglas (algoritmos) del juego Cody&Roby.

Desarrollo: se jugará por parejas.

- Recordamos a los alumnos la dinámica del juego Cody&Roby.
- Explicamos que van a volver a jugar al juego, pero que en esta ocasión en el tablero podrán encontrar islas.
- La dinámica del juego es la misma que en su versión fácil.
- Hacemos una demostración práctica en la pizarra.
- Pediremos a los alumnos que realicen la actividad varias veces, instándolos esta vez a crear secuencias de movimiento cuanto más simples mejor, puesto que la mecánica del juego ya está interiorizada. Lo que ahora nos interesa es que los alumnos se vayan habituando a crear código sencillo y funcional.

4. Actividad 3: nivel difícil (ejercicios competitivos y de colaboración)

Una vez que todos los alumnos comprenden y conocen las reglas del juego y el uso de las cartas, llega la hora de proponer una serie de juegos en los que poner en práctica, de una forma cooperativa o competitiva, las habilidades adquiridas.

Los **contenidos** de los 4 juegos son los mismos: dominar las reglas (algoritmos) del juego.

En cuanto al **desarrollo**, los 4 juegos tienen ciertos puntos en común:

- Los propios alumnos serán los que comprobarán la corrección del código. Si el código propuesto es correcto, su creador gana una partida. Si por el contrario el código no es correcto, no podrá participar más en esta, dejando vía libre a su compañero para que pueda pensar en su código sin las restricciones del tiempo que conlleva el aspecto competitivo.
- Al principio de cada sesión, haremos una demostración práctica en la pizarra compitiendo el docente y un alumno voluntario.
- Pediremos a los alumnos que realicen cada juego varias veces.

La **variación** común en los 4 juegos es la siguiente:

- Se podría crear una especie de campeonato donde los alumnos ganadores (los que hubieran recorrido todas las casillas) se irían enfrentado entre sí hasta que quedara una sola pareja (la cual recibiría, además de 1 punto por cada victoria parcial, 10 puntos como premio final).

Los juegos que realizaremos son los que el Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF) propone en la web del proyecto Cody&Roby. Siguiendo mi propio criterio, y de menor a mayor dificultad, el orden elegido para realizar dichos juegos ha sido el siguiente: La Carrera, La Serpiente, El Duelo y Rellenar el Tablero.

Juego “La Carrera”:

Desarrollo: se jugará por parejas en modo competitivo (aunque también es posible el modo cooperativo en parejas para competir contra otra pareja).

- Reglas del juego:
 - o El tablero contará con un “camino” formado por islas. Los alumnos, turnándose, serán los que construyan este camino.
 - o El código que creen los alumnos deberá lograr que su robot siga, sin salirse, dicho camino.

- El alumno que sea el primero en crear correctamente un código que resuelva el ejercicio de forma satisfactoria será el ganador. Para ello, deberá ordenar todas las cartas de movimiento necesarias para seguir el camino formado por las islas y “pulsar” en el triángulo verde que está encima del tablero para indicar que ha resuelto el ejercicio.
- Cada alumno tendrá a su disposición una baraja completa de cartas de movimiento.

Variaciones:

- Se podrían crear dos caminos y que cada alumno eligiese el que más le gustara para realizar el ejercicio.

Juego “La Serpiente”:

Desarrollo: se jugará por parejas en modo competitivo (aunque también es posible el modo cooperativo en parejas y competir contra otra pareja).

- Reglas del juego:
 - La finalidad del juego es “cortar” el camino del contrario.
 - Para facilitar el juego, el tablero estará vacío de islas.
 - Cada alumno tendrá a su disposición una baraja completa de cartas de movimiento que tendrá que barajar antes de cada partida. En el juego de “La Serpiente”, y por motivos obvios, no se podrá utilizar la carta de bucle.
 - Cada jugador comenzará el juego cogiendo 3 cartas de su montón.
 - Los alumnos jugarán por turnos.
 - Los alumnos podrán utilizar solamente 1 carta por turno.
 - Cuando un jugador use una carta de movimiento, tendrá que coger otra de su montón, de manera que siempre tiene que tener 3 cartas en su mano.
 - Cada vez que un alumno avance con su robot una casilla, tendrá que dejar una ficha de isla en la casilla anterior. De este modo, su compañero podrá ver de forma clara cuál es el camino que va recorriendo (para así poder cortar dicho camino).
 - El jugador que corte el camino de su adversario, será el ganador.
- El uso de las cartas es el mismo que en las actividades anteriores, con la salvedad de no poder utilizar la carta bucle.

Variaciones:

- Podemos dar la vuelta al ejercicio: en lugar de tener que cortar el camino del oponente, cada alumno tendrá que evitarlo, y el perdedor será el que lo corte finalmente.

Juego “El Duelo”:

Desarrollo: se jugará por parejas en modo competitivo (aunque también es posible el modo cooperativo en parejas y competir contra otra pareja).

- Reglas del juego:
 - o La finalidad del juego es dar alcance al robot del contrario.
 - o El tablero contará con islas para hacerlo más emocionante y difícil. Cada alumno podrá poner un máximo de 2 islas en casillas de su elección.
 - o Cada alumno tendrá a su disposición una baraja completa de cartas de movimiento que tendrá que barajar antes de cada partida.
 - o Cada jugador comenzará el juego cogiendo 5 cartas de su montón.
 - o Los alumnos jugarán por turnos.
 - o Los alumnos podrán utilizar un máximo de 5 cartas por turno.
 - o Después de cada turno, el alumno deberá coger de su montón el mismo número de cartas que acaba de utilizar, de manera que siempre tiene que tener 5 cartas en su mano.
 - o El jugador que alcance al robot de su adversario, será el ganador.
- El uso de las cartas es el mismo que en las actividades anteriores, con la salvedad de no poder utilizar la carta de bucle.

Variaciones:

- Podemos aumentar o disminuir el número de cartas que los alumnos puedan usar en cada turno (por ejemplo, sólo pueden usar un máximo de 3 cartas).
- Podemos aumentar o disminuir el número de cartas que los alumnos tengan en la mano en cada turno (por ejemplo, en lugar de 5 cartas, sólo 4).

Juego “Rellenar el Tablero”:

Desarrollo: se jugará por parejas en modo cooperativo.

- Reglas del juego:
 - La finalidad del juego es recorrer todas las casillas del tablero con el robot sin repetir el paso por ninguna casilla.
 - El tablero no contará con islas.
 - Los alumnos tendrán a su disposición una baraja completa de cartas de movimiento que tendrán que barajar antes de cada partida. En el juego de “Rellenar el Tablero” se podrá utilizar la carta de bucle.
 - En cada pareja, un jugador será el encargado de sostener las cartas en su mano y de utilizarlas, y el otro de ir poniendo las islas por las casillas que el robot vaya recorriendo para saber en todo momento qué casillas faltan por recorrer.
 - Ambos jugadores podrán crear el código.
 - El jugador encargado de sostener las cartas comenzará el juego cogiendo 3 cartas de su montón.
 - Los jugadores sólo podrán utilizar 1 carta por turno, a excepción de la carta de bucle, que podrá utilizarse en combinación con cualquiera de las otras.
 - Después de usar la carta, el jugador encargado de sostener las cartas cogerá una nueva carta del montón (o 2 si ha utilizado también una carta de bucle), de manera que siempre deberá tener 3 cartas en su mano.
 - Si antes de que se acaben las cartas del montón, los alumnos han logrado que su robot recorra todas las casillas del tablero, habrán ganado la partida.

Variaciones:

- Dada la dificultad de realizar correctamente el ejercicio sin pasar 2 o más veces por una misma casilla, se podría eliminar esta regla, haciendo, de este modo, mucho más accesible la consecución final del mismo.
- Al tratarse del juego final del proyecto, y dados los conocimientos que sobre el juego se les presupone a los alumnos, si alguna pareja cometiera un error grave en la creación del código necesario para resolver el ejercicio, se les restarían a cada alumno 3 puntos de la gamificación.
- Podemos aumentar o disminuir el número de cartas de cada montón para hacer más fácil o difícil el ejercicio.

Actividades enchufadas

i. Introducción a la programación basada en bloques – Blockly games

1. Consideraciones previas

Después de haber adquirido los conceptos de simplificación y secuenciación de comandos, comenzaremos a trabajar el lenguaje de programación propiamente dicho. Esto lo haremos de una manera lo más accesible posible para los alumnos, evitando así el riesgo de bloqueos o rechazos. Es por ello que he elegido la actividad Blockly ya que trabaja el lenguaje de programación por bloques. Para conocer las características de esta aplicación, ver **ANEXO VI**.

De todos los juegos y actividades que la web nos ofrece, no cumplimentaremos la totalidad de los mismos, ya que nuestro objetivo con estas actividades no es tanto la masterización de los conceptos pretendidos, sino una introducción a los mismos. Por lo tanto, solamente utilizaremos los siguientes: Rompecabezas, Laberinto, Pájaro, Tortuga y Música.

Juegos a trabajar en cada sesión:

- **SESIÓN 1:** *Rompecabezas y Laberinto.*
- **SESIÓN 2:** *Pájaro, Tortuga y Música.*
- **SESIÓN 3:** repasaremos todos los juegos trabajados.

Las soluciones a todos los juegos se pueden encontrar en el **ANEXO VIII**.

En todas las sesiones, tendremos una serie de elementos comunes en cuanto a *espacios, materiales, desarrollo, evaluación y gamificación*. Así:

Espacios: aula.

Materiales:

- Mini portátiles.
- Pizarra interactiva.
- Cronómetro online:

<https://www.online-stopwatch.com/spanish/full-screen-stopwatch.php>

Evaluación: se realizará mediante observación directa por parte del docente.

Gamificación: los alumnos recibirán 1 punto por juego resuelto de forma correcta.

Las sesiones 1 y 2 comparten además el mismo **desarrollo** en los siguientes juegos: Laberinto, Pájaro, Tortuga y Música.

- Explicamos la mecánica de cada actividad realizando el primer juego nosotros en la pizarra interactiva.
- Los alumnos, de forma individual, tendrán que resolver el resto de juegos.
- Para facilitar la comprensión de la actividad por parte de los alumnos, las actividades se irán realizando 1 por 1 (no hay que olvidar que estamos comenzando con el lenguaje por bloques). De esta manera se pierde en agilidad a la hora de realizar el ejercicio, pero se gana en confianza y en comprensión de los contenidos.
- Dinámica a seguir:
 - o Damos un tiempo determinado para la realización de una actividad (en función de la dificultad de la actividad, daremos más o menos tiempo).
 - o Ponemos en marcha el cronómetro online.
 - o Una vez finalizado el tiempo, corregimos la actividad en la pizarra interactiva para que todos los alumnos puedan ver la corrección.
 - o Pasamos a la siguiente actividad.
- A través de esta dinámica, nos aseguramos de que todo el alumnado comprende y aplica correctamente la dinámica de la programación basada en bloques.

2. Sesión 1

Rompecabezas

Se trata de un juego muy sencillo y simple que pretende ser una entrada bastante accesible a la programación por bloques. Su objetivo es enseñar a los alumnos cómo se mueven y encajan los bloques unos en otros.

Para evitar fallos de conexión y para no sobrecargar la red del centro, en esta sesión los alumnos se descargarán e instalarán la aplicación en sus mini portátiles.

Objetivos:

- Ver el funcionamiento básico de los bloques en el lenguaje de programación basado en bloques.
- Repasar contenidos básicos sobre animales.

Contenidos:

- Funcionamiento de los bloques en el lenguaje de programación basado en bloques.
- Características de animales.

Temporalización: 5/7 minutos.

Desarrollo:

- Explicamos la actividad completando los datos sobre un animal (por ejemplo, el pato), en la pizarra interactiva.
- Los alumnos, una vez visto cómo es el funcionamiento básico de los bloques, completarán el resto de la actividad de forma individual.
- Corrección en gran grupo.

Laberinto

El juego cuenta con 10 niveles, pero nosotros solo realizaremos la actividad **de los niveles 1 al 9**, pues el nivel 10 es para programadores más avanzados.

Objetivos:

- Repasar el concepto de bucle en programación.
- Introducir el concepto de condicional en programación.
- Repasar las nociones espaciales básicas.

Contenidos:

- Bucles en programación.
- Condicionales en programación.
- Nociones espaciales básicas.

Temporalización: 40 minutos.

3. Sesión 2

Pájaro

El juego cuenta con 10 niveles, pero nosotros solo realizaremos **de los niveles 1 al 5**, pues el resto de niveles presentan una gran complejidad para las características de nuestro alumnado.

Objetivos:

- Repasar el concepto de condicional en programación.
- Repasar los grados de la circunferencia.
- Repasar la representación gráfica de los conceptos mayor y menor.
- Repasar la representación gráfica de los ejes cartesianos (x / y).

Contenidos:

- Condicionales en programación.
- Grados contenidos en una circunferencia.
- Representación gráfica de los conceptos mayor y menor.
- Representación gráfica de los ejes cartesianos (x / y).

Temporalización: 20 minutos.

Tortuga

El juego cuenta con 10 niveles, pero nosotros solo realizaremos **de los niveles 1 al 4**, pues el resto de niveles presentan una gran complejidad para las características de nuestro alumnado.

Objetivos:

- Repasar el concepto de bucle en programación.
- Repasar figuras geométricas básicas.
- Repasar los grados contenidos en una circunferencia.
- Repasar las nociones espaciales básicas.

Contenidos:

- Bucles en programación.
- Figuras geométricas básicas.
- Grados contenidos en una circunferencia.
- Nociones espaciales básicas.

Temporalización: 20 minutos.

Música

El juego cuenta con 10 niveles, pero nosotros solo realizaremos **de los niveles 1 al 6**, pues el resto de niveles presentan una gran complejidad.

Objetivos:

- Introducir el concepto de función en programación.
- Repasar las notas musicales.
- Repasar la notación de las notas musicales.
- Repasar el valor de las notas musicales y su representación gráfica.

Contenidos:

- Funciones en programación.
- Notas musicales: notación, valor y representación gráfica de las mismas.

Temporalización: 20 minutos.

4. Sesión 3

Objetivos:

- Terminar los juegos no realizados.
- Reforzar y repasar los conceptos trabajados.

Contenidos:

- Concepto de bucle aplicado a programación.
- Concepto de función aplicado a programación.
- Concepto de condicional aplicado a programación.

Temporalización: 30/40 minutos.

Desarrollo:

- Comprobamos que todos los alumnos tienen realizados de forma correcta los juegos propuestos en las 2 sesiones anteriores.
- Si no fuera así, damos un tiempo para que los alumnos que lo necesiten los terminen. Nombraríamos alumnos ayudantes para que echaran una mano en la resolución de los juegos a los alumnos rezagados.

- Una vez comprobado que todos los alumnos han realizado todos los juegos, procederíamos a borrar los progresos de los alumnos. Para ello pulsamos el icono “*Borrar datos*”.
- Para repasar los conceptos trabajados, cumplimentaremos los juegos de nuevo. Para evitar que los alumnos pierdan interés en esta tarea, propondremos un **juego** basado en la rapidez con la que se vayan resolviendo los juegos. La dinámica de este juego sería la siguiente:
 - Los alumnos tienen que resolver correctamente todos los juegos trabajados en cada actividad.
 - Dividimos el juego en 5 partes, tantas partes como actividades diferentes hay (Rompecabezas, Laberinto, Pájaro, Tortuga y Música).
 - Damos un tiempo determinado para resolver todos los juegos de cada parte.
 - Cuando un alumno acabe todos los juegos de una actividad, levantará la mano avisando al docente de esta circunstancia.
 - El docente tomará nota del orden en que los alumnos van acabando las actividades.
 - Damos puntos a los alumnos por resolver correctamente todos los juegos de cada actividad: si en aula hay 25 alumnos, daremos 25 puntos al primer alumno que acabe, 24 al segundo, 23 al tercero... y así sucesivamente.
 - Si un alumno no resolviera correctamente algún juego de una actividad, no recibirá ningún punto en esa actividad, pudiendo participar en las siguientes.

ii. Trabajando la programación basada en bloques - Code.org

1. Consideraciones previas

Tras introducir a nuestros alumnos en la programación por bloques, el siguiente paso será interiorizar los mismos y adquirir nuevos conceptos más avanzados que permitan a nuestros alumnos aumentar sus capacidades programadoras. Para ello, he elegido los juegos y actividades que la web Code.org propone. Para más información sobre esta web, ver **ANEXO VI**.

La web Code.org cuenta con numerosísimos juegos, actividades y lecciones para comenzar, repasar y avanzar en los conocimientos sobre la programación basada en bloques. Para repasar los contenidos trabajados en la actividad Blockly, he elegido utilizar el recurso “**Curso Rápido en su versión 2018**”. Las razones de tal elección son las siguientes:

- Los contenidos se encuentran muy bien agrupados, seleccionados y graduados de menor a mayor complejidad.

- Hay un buen equilibrio entre ejercicios a realizar y vídeos para repasar los contenidos adquiridos.
- Al final de cada lección se puede acceder a unos ejercicios adicionales para repasar los contenidos trabajados.
- Cuenta con un feedback inmediato sobre los ejercicios resueltos, cómo los has resuelto y los que faltan por resolver.
- Los ejercicios de cada lección cuentan con un enlace para ver la solución (sólo en modo profesor).
- Se trabaja con personajes atractivos para los alumnos: personajes de la Guerra de las Galaxias, Minecraft, zombis, personajes de las películas de Ice Age,... Esto aumenta el interés y la motivación por parte del alumnado.

Dado que resolver todos los ejercicios de todas las lecciones de este recurso nos llevaría muchísimo tiempo (son 47 lecciones), nos vamos a centrar exclusivamente en resolver los ejercicios de las lecciones donde se repasan los contenidos que previamente hemos trabajado en las actividades anteriores y que son los que más nos interesan: encontrar errores (debugging), bucles, condicionales y funciones.

Dada la complejidad de alguno de los contenidos, ciertos ejercicios se resolverán de forma individual y otros en pareja.

La resolución de los ejercicios se organizará en 11 sesiones: 1 sesión por cada lección, dejando 3 sesiones más para completar ejercicios y repasar y afianzar los conceptos más complejos, que son los trabajados en las lecciones 14, 18 y 21.

En todas las sesiones, tendremos una serie de elementos comunes en cuanto a *espacios*, *materiales*, *temporalización*, *desarrollo* y *evaluación*.

Espacios: aula.

Materiales:

- Mini portátiles.
- Pizarra interactiva.
- Cronómetro online.

Temporalización: 1 hora.

Desarrollo: debido a la naturaleza de los contenidos, las sesiones tendrán siempre la misma estructura. Mi experiencia en el aula me ha enseñado que con este tipo de contenidos

(repetitivos y bien estructurados) lo mejor es una metodología de aula un tanto “rutinaria”, es decir, clases bien organizadas y estructuradas, donde los alumnos saben con antelación lo que va a pasar y lo que se espera de ellos.

En relación con lo anterior, las clases tendrán la siguiente estructura: *introducción, desarrollo y planteamiento final*.

- *Introducción (5/10 minutos)*: comenzaremos cada sesión introduciendo el contenido que se trabajará en la misma. Para ello, nos serviremos de los juegos, ejercicios o programas trabajados con anterioridad en Blockly games o en una lección previa de Code.org.
- *Desarrollo de la sesión (40/45 minutos)*: el docente da un tiempo para realizar cada ejercicio y después lo irá corrigiendo en la pizarra interactiva a medida que los alumnos los vayan resolviendo (los alumnos resuelven un ejercicio, el docente lo corrige). De esta manera nos aseguramos de que los todos los alumnos entienden todos los conceptos y de que ningún alumno se queda atrás. Para ello usaremos el cronómetro online:

<https://www.online-stopwatch.com/spanish/full-screen-stopwatch.php>

- *Planteamiento final (5/10 minutos)*: en gran grupo debatimos sobre lo aprendido en la sesión, su utilidad y su relación con futuras actividades relacionadas con la programación o cualquier otro contenido o asignatura.

Evaluación: al haber creado una clase dentro de Code.org en la que se encontrarán todos los alumnos, el docente observará de forma directa los avances de los alumnos, los ejercicios resueltos, las lecciones trabajadas y los contenidos adquiridos. Para más información a este respecto, consultar el **ANEXO IX**.

Gamificación: por cada ejercicio resuelto correctamente (es decir, reconocido como tal por la plataforma), otorgaremos 1 punto a cada alumno.

2. Sesión 1 – trabajo individual

Durante esta sesión se trabajará la **LECCIÓN 4: *debugging*** (encontrar fallos en la programación). Se resolverán los ejercicios del 2 al 10.

Objetivos:

- Encontrar errores (bugs) en el código.
- Corregir dichos errores (debugging).
- Gusto por crear código “limpio”.

Contenidos:

- Errores en programación (bugs).

3. Sesión 2 – trabajo individual

Durante esta sesión se trabajará la LECCIÓN 8: *bucles simples*. Se resolverán los ejercicios del 2 al 14.

Objetivos:

- Conocer los bucles en programación.
- Usar los bucles para crear un código más “limpio”.
- Gusto por crear código “limpio”

Contenidos:

- Bucles en programación.

4. Sesión 3 – trabajo en parejas

Durante esta sesión se trabajará la LECCIÓN 12: *bucles anidados*. Se resolverán los ejercicios del 1 al 2 y del 4 al 13.

Objetivos:

- Conocer los bucles anidados en programación.
- Usar los bucles anidados para crear un código más “limpio”.
- Gusto por crear código “limpio”

Contenidos:

- Bucles anidados en programación.

5. Sesión 4 – trabajo en parejas

Durante esta sesión se trabajará la LECCIÓN 14: *bucles condicionados – while (mientras)*. Se resolverán los ejercicios del 1 al 3 y del 5 al 13.

La comprensión y el aprendizaje del concepto trabajado en esta sesión requiere tiempo, por lo que esta sesión es candidata a una sesión de repaso y refuerzo a mayores de las 3 que hemos programado.

Objetivos:

- Conocer los bucles condicionados en programación.
- Usar los bucles condicionados para crear un código más “limpio”.
- Gusto por crear código “limpio”

Contenidos:

- Bucles condicionados en programación.

6. Sesión 5 – trabajo en parejas

Durante esta sesión se repasarán y afianzarán los contenidos vistos en la sesión anterior. Se terminarán de resolver los ejercicios del 1 al 3 y del 5 al 13 (si fuera necesario) y se profundizará en la explicación y la comprensión del concepto de bucle condicionado con variaciones de los ejercicios trabajados.

Objetivos:

- Afianzar el conocimiento de los bucles condicionados en programación.
- Usar los bucles condicionados para crear un código más “limpio”.
- Gusto por crear código “limpio”

Contenidos:

- Bucles condicionados en programación.

7. Sesión 6 – trabajo individual

Durante esta sesión se trabajará la LECCIÓN 17: *condicionales*. Se resolverán los ejercicios del 1 al 7 y del 9 al 13.

Objetivos:

- Conocer el concepto de condicional en programación.
- Usar condicionales para crear un código más “limpio”.
- Gusto por crear código “limpio”

Contenidos:

- Condicionales en programación.

8. Sesión 7 – trabajo en parejas

Durante esta sesión se trabajará la LECCIÓN 18: *condicionales complejos*. Se resolverán los ejercicios del 2 al 10.

La comprensión y el aprendizaje del concepto trabajado en esta sesión requiere tiempo, por lo que esta sesión es candidata a una sesión de repaso y refuerzo a mayores de las 3 que hemos programado.

Objetivos:

- Profundizar en el concepto de condicional en programación.
- Usar condicionales complejos para crear un código más “limpio”.
- Gusto por crear código “limpio”

Contenidos:

- Condicionales en programación.

9. Sesión 8 – trabajo en parejas

Durante esta sesión se repasarán/afianzarán los contenidos vistos en la sesión anterior. Se terminarán de resolver los ejercicios del 2 al 10 (si fuera necesario) y se profundizará en la explicación y la comprensión del concepto de condicional complejo con variaciones de los ejercicios trabajados.

Objetivos:

- Afianzar el conocimiento del concepto de condicional en programación.
- Usar condicionales complejos para crear un código más “limpio”.
- Gusto por crear código “limpio”

Contenidos:

- Condicionales en programación.

10. Sesión 9 – trabajo individual

Durante esta sesión se trabajará la LECCIÓN 20: funciones. Se resolverán los ejercicios del 1 al 12 (las funciones están ya creadas en el ejercicio).

Objetivos:

- Introducir el concepto de función en programación.
- Usar de un modo básico funciones para crear un código más “limpio”.
- Gusto por crear código “limpio”

Contenidos:

- Funciones en programación.

11. Sesión 10 – trabajo en parejas

Durante esta sesión se trabajará la LECCIÓN 21: *funciones*. Se resolverán los ejercicios del 2 al 7 (las funciones están ya creadas en los ejercicios) y del 9 al 14 (el alumno tendrá que crear las funciones).

La comprensión y el aprendizaje del concepto trabajado en esta sesión (sobre todo de los 5 últimos ejercicios) requiere tiempo, por lo que esta sesión es candidata a una sesión de repaso y refuerzo a mayores de las 3 que hemos programado.

Objetivos:

- Profundizar en el concepto de función en programación.
- Usar funciones para crear un código más “limpio”.
- Comenzar a crear funciones sencillas.
- Gusto por crear código “limpio”

Contenidos:

- Funciones en programación.

12. Sesión 11 – trabajo en parejas

Durante esta sesión se repasarán y afianzarán los contenidos vistos en la sesión anterior. Se terminarán de resolver los ejercicios del 2 al 7 y del 9 al 14 (si fuera necesario) y se

profundizará en la explicación y la comprensión del concepto de función y en la creación de funciones mediante variaciones de los ejercicios trabajados.

Objetivos:

- Afianzar el conocimiento del concepto de función en programación.
- Usar funciones para crear un código más “limpio”.
- Crear funciones sencillas.
- Gusto por crear código “limpio”

Contenidos:

- Funciones en programación.

iii. Micro:bit

1. Consideraciones previas

Finalmente, y después de haber interiorizado y adquirido los conceptos trabajados en las actividades anteriores (simplificación, secuenciación de comandos, búsqueda de errores, bucles, bucles anidados, bucles condicionados, condicionales y funciones) vamos a pedir nuestros alumnos que pongan todos estos conocimientos en práctica a través de la utilización de un dispositivo llamado Micro:bit que nos permitirá la introducción en nuestra aula de la metodología STEAM. Para conocer más a fondo este dispositivo y sus posibilidades, ver **ANEXO VI**.

En total, nuestros alumnos crearán 6 proyectos. Cada proyecto se dividirá en 2 partes: la parte de programación y la parte de STEAM, donde los alumnos crearán un dispositivo que albergue el Micro:bit para que pueda cumplir correctamente la función para la que ha sido programado. Dado el carácter creativo y manipulativo de la parte STEAM, cada proyecto se podrá desarrollar tanto en una clase de Matemáticas como en una de Plástica (o, dependiendo del proyecto, en alguna clase de otra asignatura).

Si algún proyecto no pudiera terminarse durante el tiempo previsto, existen varias posibilidades de finalización del mismo:

- Terminarlo durante la clase de la siguiente asignatura.
- Terminarlo en los “tiempos muertos” de otras asignaturas.
- Terminarlo en un recreo.
- Terminarlo en la siguiente sesión, antes de comenzar el siguiente proyecto.

En cuanto a la temporalidad de cada proyecto, decir que es aproximada, pues dependerá de múltiples factores: recursos que se encuentren en el aula, materiales elegidos para la realización del proyecto STEAM, implicación por parte del alumnado en cada proyecto, variaciones propuestas,... Estos factores podrían modificar el tiempo requerido aumentándolo o disminuyéndolo.

En el **ANEXO X** se puede encontrar toda la programación necesaria para elaborar cada proyecto.

Todos los proyectos comparten diversos aspectos:

Objetivos comunes a todos los proyectos:

- Conocer las peculiaridades de la interfaz del editor de código de Micro:bit.
- Conocer algunos de los bloques específicos de Micro:bit.
- Crear un dispositivo práctico para poder utilizar de un modo real el código y el hardware (proyecto STEAM).
- Desarrollar el interés y el gusto por trabajar con el editor de código, con el hardware y por la creación de una aplicación real que aúne ambos (proyecto STEAM).

Contenidos comunes a todos los proyectos:

- Editor de código de Micro:bit (algunos bloques específicos).
- Aplicación práctica del código creado y del hardware.

Espacios: aula.

Materiales comunes a todos los proyectos:

- Mini portátiles.
- Pizarra interactiva.

Evaluación:

Comprobación directa del resultado final tanto del código creado como del dispositivo para mostrar el proyecto STEAM creado.

Gamificación:

- Parte de *programación*:
 - Los alumnos que realicen la actividad correctamente, recibirán 1 punto para la actividad de gamificación.
- Parte *STEAM*:
 - Los alumnos que creen los 3 dispositivos más ingeniosos recibirán puntos para la actividad de gamificación:
 - Mejor dispositivo: 10 puntos.
 - Segundo mejor dispositivo: 6 puntos.
 - Tercer mejor dispositivo: 3 puntos.

2. Proyecto 1: creamos un saludo. Trabajo individual y por parejas.

Comenzaremos a trabajar con Micro:bit realizando una actividad muy sencilla y simple. Con esto pretendo lograr tres objetivos básicos:

- Que los alumnos venzan su miedo y reticencia iniciales a trabajar con el hardware y el editor de código.
- Que los alumnos se familiaricen con el manejo de la placa Micro:bit.
- Que los alumnos vean la metodología STEAM como una oportunidad de poner en práctica lo aprendido realizando un proyecto con una aplicación en el mundo real.

De esta forma, la primera actividad consistirá en conocer cómo usar los leds para crear saludos, ya sean en forma de dibujo o de texto. Dada la sencillez del código necesario para crear el saludo, la parte de programación se realizará de forma individual; mientras que el proyecto de STEAM se hará en parejas (no hay que olvidar que se trata del primer proyecto).

Objetivo específico del proyecto:

- Utilizar los bloques de Micro:bit para crear un saludo.

Temporalización: 1 hora.

Materiales específicos del proyecto:

- Tijeras.
- Lana.
- Carpetas clasificadoras de plástico.

- Grapadora y grapas.
- ...

Desarrollo:

- Los alumnos acceden al editor de código de Micro:bit y pulsan en “NEW PROJECT” para acceder al apartado donde crearán el código.
- En gran grupo:
 - Al tratarse de la primera vez que los alumnos crean código con este editor, les explicaremos de forma práctica sus peculiaridades y sus bloques específicos más significativos. Esto lo iremos realizando en la pizarra interactiva al mismo tiempo que los alumnos van creando su primer código. De esta manera comprobamos y nos aseguramos de que todos los alumnos van entendiendo los conceptos pretendidos y que ningún alumno se queda atrás.
 - Explicación práctica y creación del saludo (ver **ANEXO X**): se trata de un programa que emitirá un saludo basado en un texto, un icono predeterminado del programa o un icono creado por los propios alumnos.

Proyecto STEAM (por parejas):

Una vez creado el código, los alumnos tendrán que construir un dispositivo para mostrar al resto de alumnos los saludos creados. Para ello, los alumnos tendrán que idear un dispositivo para esta tarea.

Nosotros les propondremos el siguiente *ejemplo*:

- Recortamos una carpeta clasificadora de plástico de tal modo que se cree una especie de “bolsa” que pueda contener el Micro:bit y su fuente de alimentación.
- Metemos el Micro:bit y la fuente de alimentación dentro de la “bolsa”.
- Mediante grapas, cerramos la “bolsa” para evitar que se caiga el hardware.
- Grapamos un cordón de lana a la “bolsa” con la largura suficiente para poder llevar ésta alrededor de nuestro cuello como si fuera un collar.
- Comprobamos que el conjunto queda sujeto y sin riesgo de caída.

Los alumnos podrán crear este dispositivo o elaborar cualquier otro de su invención.

Para evitar que la actividad se alargue mucho en el tiempo, los alumnos tendrán que usar materiales que se encuentren en el aula: cartulina, celo, pegamento,...

3. Proyecto 2: creamos un termómetro. Trabajo individual.

El código de esta actividad está inspirado en una actividad vista en:

<https://makecode.microbit.org/lessons/temperature/>

Nuestro segundo proyecto sigue en la misma línea de sencillez y simplicidad que el primero para continuar dando seguridad y motivación a nuestros alumnos tanto con el trabajo en el editor de código de Micro:bit como con el uso y posibilidades del hardware y la realización del proyecto STEAM. En esta ocasión, crearemos un termómetro que nos indicará en cifras y con un emoticono la temperatura ambiente.

Objetivos específicos del proyecto:

- Aprender a crear y manejar variables.
- Aprender a crear y manejar condicionales.
- Aprender a crear una comparación.

Contenidos específicos del proyecto:

- Variables: creación y uso.
- Condicionales: creación y uso.
- Comparaciones: creación y uso.

Temporalización: 1 hora (incluimos el tiempo necesario para terminar el proyecto anterior si fuera necesario).

Materiales específicos del proyecto:

- Tijeras.
- Celo/Cinta americana.
- Pegamento.
- Lana
- Porexpán.
- Pintura y pinceles.

Desarrollo:

- Los alumnos acceden al editor de código de Micro:bit y pulsan en “NEW PROJECT” para acceder al apartado donde crearán el código.
- En gran grupo:

- Explicación práctica a los alumnos sobre cómo se usan y para qué sirven los nuevos bloques que utilizaremos en este proyecto.
- Explicación práctica de qué son las variables y cómo y para qué se utilizan.
- Explicación práctica de qué son los condicionales y cómo y para qué se utilizan.
- Explicación práctica de qué son las comparaciones y cómo y para qué se utilizan.
- Explicación práctica y creación, paso a paso, del código necesario para crear el termómetro (ver **ANEXO X**).

Proyecto STEAM:

Una vez creado el código, los alumnos tendrán que crear una “caja” para colgar su Micro:bit y su fuente de alimentación de una pared (del mismo modo que haríamos con un termómetro real).

Nosotros les propondremos el siguiente *ejemplo*:

- Recortamos dos planchas de porexpán de 8x7 centímetros.
- Una de las planchas será la base del termómetro.
- Con la otra plancha tendremos que hacer un marco que aloje la placa de nuestro Micro:bit. Para ello, vaciaremos el interior de la plancha, dejando un marco exterior de 1.5 centímetros por cada lado.
- Colocamos la placa del Micro:bit en el hueco formado por las dos planchas.
- Hacemos un agujero en la parte superior de la plancha de debajo (a la altura del conector de la batería).
- Colocamos la fuente de alimentación en la parte trasera de la plancha de debajo pegada con celo o cinta americana.
- Pasamos el conector por el agujero de la plancha de modo que podamos conectar la placa con la fuente de alimentación.
- Adherimos la placa de Micro:bit a la plancha de debajo con celo o cinta americana.
- Pegamos la plancha de arriba (en forma de marco) con la plancha de abajo con pegamento.
- Realizamos dos agujeros en la parte superior del conjunto de modo que lo atraviesen.
- Pasamos un hilo de lana de una longitud aproximada de unos 15 o 20 centímetros por ambos agujeros.
- Los aseguramos con nudos de modo que el hilo sujete el conjunto.
- Comprobamos que el conjunto queda sujeto y sin riesgo de caída.

- Colgamos de una pared nuestro termómetro.

Los alumnos podrán crear este dispositivo o elaborar cualquier otro de su invención.

4. Proyecto 3: creamos un cuenta pasos. Trabajo por parejas.

El código de esta actividad está basado en una actividad ya creada en la propia web de Micro:bit (Step counter):

<https://makecode.microbit.org/projects/step-counter>

Además de tratarse de una actividad muy sencilla y atractiva para los alumnos, nos permitirá abordar e introducir uno de los conceptos básicos en programación: las funciones.

Objetivos específicos del proyecto:

- Crear y manejar variables.
- Aprender a crear y manejar funciones.

Contenidos específicos del proyecto:

- Variables: creación y uso.
- Funciones: creación y uso.

Temporalización: 1 hora (incluimos el tiempo necesario para terminar el proyecto anterior si fuera necesario).

Materiales específicos del proyecto:

- Tijeras.
- Celo.
- Carpetas clasificadoras de plástico.
- Grapadora y grapas.
- Cartulina o tela.
- ...

Desarrollo:

- Los alumnos acceden al editor de código de Micro:bit y pulsan en “NEW PROJECT” para acceder al apartado donde crearán el código.
- En gran grupo:

- Explicación práctica a los alumnos sobre cómo se usan y para qué sirven los nuevos bloques que utilizaremos en este proyecto.
- Recordamos qué son las variables y cómo y para qué se utilizan.
- Explicación práctica de qué son las funciones y cómo y para qué se utilizan.
- Explicación práctica y creación, paso a paso, del código necesario para crear el cuenta pasos (ver **ANEXO X**).

Proyecto STEAM (por parejas):

Una vez creado el código, tendremos que encontrar una manera de crear un dispositivo para utilizar el programa creado. Para ello, los alumnos tendrán que idear un dispositivo que les permita llevar el Micro:bit y su fuente de alimentación en uno de sus tobillos.

Nosotros les propondremos el siguiente *ejemplo*:

- Recortamos una carpeta clasificadora de plástico de tal modo que creamos una especie de bolsa para contener el Micro:bit y su fuente de alimentación.
- Metemos el Micro:bit y la fuente de alimentación dentro de la bolsa.
- Mediante grapas, cerramos la bolsa para evitar que se caiga el hardware.
- Recortamos una tira de cartulina o tela de unos 7 centímetros de grosor y 15 centímetros de longitud.
- Mediante grapas, sujetamos la bolsa con el hardware hacia el centro de la tira de cartulina o tela.
- Colocamos la tira de cartulina o tela con la bolsa alrededor de uno de nuestros tobillos.
- Cerramos la tira mediante grapas y cortamos la cartulina o la tela sobrante.
- Comprobamos que el conjunto queda sujeto y sin riesgo de caída.

Los alumnos podrán crear este dispositivo o elaborar cualquier otro de su invención.

Para evitar que la actividad se alargue mucho en el tiempo, los alumnos tendrán que usar materiales que se encuentren en el aula.

5. Proyecto 4: bola mágica de respuestas. Trabajo individual.

El código de esta actividad está inspirado en una idea vista en:

<https://www.101computing.net/bbc-microbit-magic-8-ball/>

Objetivos específicos del proyecto:

- Crear y manejar variables.
- Crear y manejar funciones.
- Crear y manejar condicionales.
- Crear una comparación.

Contenidos específicos del proyecto:

- Variables: creación y uso.
- Funciones: creación y uso.
- Condicionales: creación y uso.
- Comparaciones: creación y uso.

Temporalización: 1 hora y 15 minutos.

Materiales específicos del proyecto:

- Bola de porexpán partida a la mitad y vaciada.
- Pintura negra, blanca y pinceles.
- Cúter.
- Periódicos viejos.

Desarrollo:

- Los alumnos acceden al editor de código de Micro:bit y pulsan en “NEW PROJECT” para acceder al apartado donde crearán el código.
- En gran grupo:
 - Explicación práctica a los alumnos sobre cómo se usan y para qué sirven los nuevos bloques que utilizaremos en este proyecto.
 - Recordamos a los alumnos qué son las variables y cómo y para qué se utilizan.
 - Recordamos qué es una comparación en programación y cómo y para qué se utilizan.
 - Recordamos qué es un condicional y cómo y para qué se utilizan.
 - Explicación práctica y creación, paso a paso, del código necesario para el sistema de respuestas aleatorias basado en el juguete “Magic 8 ball / Bola Mágica de Respuestas” (ver **ANEXO X**).
- Respuestas a incluir en el programa: ver **ANEXO X**.

- Animaremos a los alumnos a que utilicen nuestras respuestas o a que creen las suyas propias.

Proyecto STEAM:

A continuación crearemos la bola en la que meteremos el Micro:bit y su fuente de alimentación para que el proyecto se parezca lo más posible al juguete real.

- Cogemos la bola de porexpán y la pintamos de negro (actividad a realizar con anterioridad en la clase de Plástica).
- En una de las mitades, pintamos un número 8 de color blanco (actividad a realizar con anterioridad en la clase de Plástica).
- Hacemos una ventana en el lado de la bola que no tiene el número 8 de modo que los leds de la placa sean visibles pero que la placa no pueda caerse.
- Rellenamos con papeles de periódico ambas mitades de la bola de tal modo que la presión de los papeles impida que el Micro:bit se mueva.
- Cerramos la bola.
- Comprobamos que el conjunto queda sujeto y sin riesgo de caída del Micro:bit o de la fuente de alimentación.

Si no diera tiempo a acabar esta parte se terminaría en la siguiente sesión.

6. Proyecto 5: ¿Cómo andas de pulso? Trabajo por parejas.

El código y la idea de esta actividad se pueden encontrar en:

<https://codeclubprojects.org/en-GB/microbit/frustration/>

Objetivos específicos del proyecto:

- Crear y manejar variables.
- Conocer la función y el manejo de los pines de entrada y salida de la placa.
- Conocer la función y el manejo del pin de tierra de la placa

Contenidos específicos del proyecto:

- Variables: creación y uso.
- Pines de entrada y salida de la placa.
- Pin de tierra de la placa.

Temporalización: 1 hora y 15 minutos.

Materiales específicos del proyecto:

- Alambre.
- Alicates.
- Plastilina.
- Cinta aislante.
- Pinzas de cocodrilo.

Desarrollo:

- Los alumnos acceden al editor de código de Micro:bit y pulsan en “NEW PROJECT” para acceder al apartado donde crearán el código.
- En gran grupo:
 - Recordamos a los alumnos qué son las variables y cómo se utilizan.
 - Explicación práctica a los alumnos sobre cómo se usan y para qué sirven los nuevos bloques que utilizaremos en este proyecto.
 - Explicación práctica sobre qué son y para que se pueden utilizar los pines de entrada y salida y el pin de tierra de la placa.
 - Explicación práctica y creación, paso a paso, del código necesario para elaborar nuestro sistema de detección y control de errores (ver **ANEXO X**).

Proyecto STEAM:

Realizado el código, crearemos un juego con el que podamos comprobar nuestro programa para detectar errores. Básicamente, se trata de un camino hecho de alambre por el que haremos pasar un lazo también de alambre. Cada vez que ambos alambres entren en contacto, nuestro programa contabilizará un “fallo”. El número total de fallos será visible en los leds de Micro:bit al final de cada partida.

- Cogemos un trozo de alambre de unos 20 centímetros de longitud y lo doblamos hasta que tenga la forma de un lazo.
- Cogemos un trozo de alambre de unos 40 centímetros de longitud y lo doblamos creando el camino por el que tendrá que ir el lazo. Cuantas más vueltas tenga este camino, más difícil será el juego.
- Enrollamos un poco de cinta aislante al final de cada extremo del camino pero sin tapar el final del mismo: hay que dejar unos 2 centímetros de alambre expuesto en un extremo y unos 6 centímetros en otro.

- Introducimos el lazo en el camino.
- Hacemos 2 pelotas grandes con la plastilina.
- Introducimos los extremos del camino en los bloques de plastilina: un extremo en cada bola.
 - o El extremo con 2 centímetros de alambre expuesto en su final deberá ser colocado dentro de la bola de plastilina sin que asome su final.
 - o El extremo con 6 centímetros de alambre expuesto en su final deberá ser colocado en la plastilina de tal forma que uno o dos centímetros de alambre sobresalgan de la plastilina.
- Conectamos una pinza de cocodrilo del pin de tierra al extremo que sobresale de la plastilina del extremo de 6 centímetros de alambre.
- Conectamos una pinza de cocodrilo del pin 0 al lazo de alambre.
- El juego está terminado. Para jugar:
 - o Pulsamos el botón A.
 - o Cada vez que toquemos el camino con nuestro lazo, el programa contabilizará 1 fallo y en los leds aparecerá una X durante 1 segundo.
 - o El número total de fallos se podrá ver en los leds al final de cada juego.

Animaremos a los alumnos a realizar versiones distintas del juguete: caminos más largos, lazos más pequeños, caminos realizados con alambre doblado en varias direcciones,...

Para aumentar la motivación y la implicación por parte del alumnado, podríamos hacer un pequeño concurso de uso del “juguete” con varias variables: alumno más rápido, alumno con menos fallos totales en varios juegos,...

7. Proyecto 6: creamos una brújula. Trabajo por parejas.

El código y la idea de esta actividad se pueden encontrar en:

<https://www.101computing.net/bbc-microbit-digital-compass/>

Objetivos específicos del proyecto:

- Crear y manejar variables.
- Aprender a crear y manejar variable booleanas.
- Crear y manejar comparaciones.
- Repasar los grados contenidos en una circunferencia y su localización.

Contenidos específicos del proyecto:

- Variables: creación y uso.
- Variables booleanas: creación y uso.
- Comparaciones: creación y uso.
- Circunferencia: grados contenidos y localización de los mismos.

Temporalización: 1 hora y 15 minutos.

Materiales específicos del proyecto:

- Los propios de un aula de Ed. Primaria: cartulina, celo, pinturas, plastilina, grapadora, cartón,...

Desarrollo:

- Los alumnos acceden al editor de código de Micro:bit y pulsan en “NEW PROJECT” para acceder al apartado donde crearán el código.
- En gran grupo:
 - Recordamos a los alumnos qué son las comparaciones y cómo y para qué se utilizan.
 - Recordamos qué son las variables y cómo y para qué se utilizan.
 - Explicación práctica sobre qué son las variables booleanas y cómo y para qué se utilizan.
 - Explicación práctica a los alumnos sobre cómo se usan y para qué sirven los nuevos bloques que utilizaremos en este proyecto.
 - Recordamos los grados contenidos en una circunferencia y su localización en la misma.
 - Explicación práctica y creación, paso a paso, del código necesario para elaborar una brújula (ver **ANEXO X**).

Proyecto STEAM:

Una vez creado el código, los alumnos construirán un dispositivo el que puedan instalar el Micro:bit y su fuente de alimentación para usarlo como brújula.

Esta vez, y ante la relativa sencillez de este proyecto STEAM y la práctica acumulada por parte de los alumnos, no les daremos ninguna indicación de creación del dispositivo requerido, si no que serán los propios alumnos los que deberán crear, mediante su ingenio y los materiales accesibles en cualquier aula de Ed. Primaria, el dispositivo en cuestión.

Antes de usar cualquier brújula, se deberá comprobar que el conjunto queda sujeto y sin riesgo de caída del Micro:bit o de la fuente de alimentación.

Evaluación:

El docente evaluará la brújula creada mediante una rúbrica (ver **ANEXO V**).

D.EVALUACIÓN

Como se ha visto en la dinámica de actividades propuesta, la evaluación será de dos tipos: por observación directa y por rúbrica.

Por observación directa

Dado el carácter de las actividades y de los objetivos perseguidos, la mayoría de las actividades propuestas se evaluarán de forma directa por parte del docente, ya que creo que es el mejor sistema para este tipo de actividades y objetivos. Eso sí, el docente deberá llevar un escrupuloso registro de las evaluaciones realizadas a cada alumno, ya sea mediante medios digitales (por ejemplo, utilizando Excel) o medios analógicos (por ejemplo, un diario de aula).

Por rúbrica

Dadas sus características, algunas actividades deberán ser evaluadas mediante una rúbrica:

- **Las tres actividades propuestas en el apartado “Robots humanos”**. En estas actividades la evaluación directa por parte del docente sería difícil de llevar a cabo, ya que todos los tríos realizarán el ejercicio al mismo tiempo. Es mucho más sencillo, práctico y viable que sean los tres alumnos implicados en cada actividad los que se evalúen entre sí.
- **El proyecto 6 (“Creamos una brújula”) del apartado “Micro:bit”**. Este proyecto, al dejar vía libre a los alumnos en cuanto al diseño del dispositivo para contener la brújula, necesitará ser evaluado mediante una rúbrica para, además de guiar y motivar al alumnado indicándoles los elementos y aspectos más importantes que su dispositivo debe reflejar, controlar y comprobar el producto final.

Las rúbricas utilizadas se encuentran en el **ANEXO V**.

7. CONCLUSIONES

Como ya he comentado con anterioridad, debido a mi situación laboral actual como asesor de formación del profesorado en el CRFPTIC no he podido llevar a la práctica la presente propuesta de proyecto. Pero en mi opinión se trata una propuesta de relativa facilidad de implementación en el aula por varias razones:

- Las herramientas y el software utilizados son gratuitos, aspecto muy a tener en cuenta debido a la realidad económica de los centros educativos en la actualidad. El único coste que implica este proyecto es la compra de los Micro:bit, aunque su reducido coste, 25/30 euros por unidad, lo hacen muy accesible para la práctica totalidad de los centros.
- Las actividades y herramientas propuestas son muy atractivas para los alumnos por dos motivos:
 - Las que se encuentran dentro del apartado “desenchufadas” permiten a los alumnos su modificación adaptándolas a sus propios gustos o necesidades, aspecto muy ventajoso en cuanto a motivación e implicación se refiere.
 - Las que se encuentran dentro del apartado “enchufadas” son de uso sencillo, visualmente muy atractivas, contienen feedback inmediato y se adaptan fácilmente a las características particulares de cada centro y de sus alumnos.
- Las herramientas trabajadas, aún con cierto grado de complejidad, no resultan difíciles de manejar por parte de los docentes, pues incluyen manuales, vídeos explicativos, soluciones a los ejercicios, soporte,...
- Todas las herramientas utilizadas tienen la posibilidad de poder ser trabajadas en el propio hogar de los alumnos, lo que propicia la implicación de los mismos y de sus familias.
- La gamificación junto con la web en la que se reflejarán distintos aspectos de la propuesta del proyecto (insignias, batallas, actividades realizadas, vídeos e imágenes de los alumnos,...) aportan un plus de motivación tanto a los alumnos como a sus familias.
- Las metodologías utilizadas son atractivas y eficaces a la hora de involucrar a los alumnos y de lograr los objetivos pretendidos.

Por último, comentar que una cuestión muy importante a la hora de llevar a cabo con éxito un proyecto educativo es la de lograr involucrar al mayor número posible de docentes de un centro.

Estoy convencido de que la transversalidad que proporciona la programación nos permitirá “enganchar” a otros docentes en el mismo cuando comprueben que pueden trabajar de una forma diferente, práctica y ajustada a la realidad los contenidos y conceptos de sus asignaturas.

8. REFERENCIAS BIBLIOGRÁFICAS

- 101 Computing. (2016). *BBC micro:bit – Digital Compass*. Recuperado de <https://www.101computing.net/bbc-microbit-digital-compass/>
- 101 Computing. (2016). *BBC micro:bit - Magic 8 Ball*. Recuperado de <https://www.101computing.net/bbc-microbit-magic-8-ball/>
- Ananiadou, K. y Claro, M. (2009). *21st Century skills and competences for new millennium learners in OECD Countries*. OECD Education Working Papers.
- Binkley, M. et al. (2012). *Defining Twenty-First Century Skills*. Dordrecht, Países Bajos: Springer.
- Brennan, K. y Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Annual American Educational Research Association meeting*. Vancouver, Canadá: BC.
- Burke, R.J. y Mattis, M.C. (2007). *Women and Minorities in Science, Technology, Engineering and Mathematics*. Cheltenham, UK: Edward Elgar Publishing.
- Clements, D.H. y Swaminathan S. (1995) *Technology and School Change New Lamps for Old?* *Childhood Education*, 71:5, 275-281.
- Code.org. (2019). *What will you create?* Recuperado de <https://code.org/>
- Creatividad*. (s.f.). En *Diccionario de la Lengua Española* (23.^a ed., versión 23.2 en línea). Recuperado de <https://dle.rae.es/?id=BD3eZdM>
- Cuny, J., Snyder, L. y Wing, J.M. (2010). *Demystifying computational thinking for non-computer scientists*. Work in progress.
- Davidson, C. N. (2011). *Now you see it: How the brain science of attention will transform the way we live, work, and learn*. Nueva York, Estados Unidos: Viking Press.

De la Vega, L. F. (2012). Currículum y objetivos fundamentales transversales en Chile: resultados y proyecciones. *Akademía revista digital Universidad UCINF*, 3.

Decreto 89/2014, de 24 de julio (2014, 24 de julio).

Decreto 119/2015, de 23 de junio (2015, 23 de junio).

Decreto 26/2016, de 21 de julio (2016, 21 de julio).

Decreto Foral 60/2014, de 16 de julio (2014, 6 de julio).

EUR-Lex.europa.eu. (s.f.). *Comunicación de la Comisión al Parlamento Europeo, al Consejo, al Comité Económico y Social Europeo y al Comité de las Regiones: Un nuevo concepto de educación: invertir en las competencias para lograr mejores resultados socioeconómicos*. Recuperado de <https://eur-lex.europa.eu/legal-content/ES/TXT/HTML/?uri=CELEX:52012DC0669&from=EN>

European Commission. (2018). *Digital Skills & Jobs*. Recuperado de <https://ec.europa.eu/digital-single-market/en/policies/digital-skills>

European Commission. (s.f.) *Employment, Social Affairs & Inclusion*. Recuperado de <https://ec.europa.eu/social/main.jsp?catId=1223>

European Schoolnet. (2015). *Computing our future*. Recuperado de http://fcl.eun.org/documents/10180/14689/Computing_our_future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0

European Schoolnet. (s.f.). *Serious games can affect positively students' knowledge and behavioural change*. Recuperado de <http://www.eun.org/news/detail?articleId=2529054>

Ferrán, T.A. (2014). *Gamificación: Fundamentos y Aplicaciones*. Barcelona, España: Editorial UOC.

Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. Nueva York, Estados Unidos: Palgrave Macmillan.

Giang, V. (2013, 18 de septiembre). "Gamification" Techniques Increase Your Employees' Ability To Learn By 40%. *Business insider*. Recuperado de <https://www.businessinsider.com/gamification-techniques-increase-your-employees-ability-to-learn-by-40-2013-9?IR=T>

- Hakulinen, L., Auvinen, T. y Korhonen, A. (2015). The Effect of Achievement Badges on Students' Behavior: An Empirical Study in a University-Level Computer Science Course. *International Journal Of Emerging Technologies In Learning (IJET)*, 10(1), 18-29. Recuperado de https://www.researchgate.net/publication/276415762_The_Effect_of_Achievement_Badges_on_Students'_Behavior_An_Empirical_Study_in_a_University-Level_Computer_Science_Course
- INTEF. (2018). *Cody & Roby*. Recuperado de <http://code.intef.es/cody-roby/>
- INTEF. (s.f.). *La Escuela de Pensamiento Computacional*. Recuperado de <https://intef.es/tecnologia-educativa/pensamiento-computacional/>
- Ioannidou, A., Bennett, V., Repenning, A., Koh, K. H., y Basawapatna, A. (2011). Computational thinking pattern. *Annual American Educational Research Association meeting*. Recuperado de <https://eric.ed.gov/?id=ED520742>
- Jenkins, H. (2006). *Confronting the challenges of participatory culture: Media education for the 21st century*. Chicago, Estados Unidos: MacArthur.
- Junta de Castilla y León. (s.f.). *Escuela de verano 2019 - Castilla y León*. Recuperado de <https://www.educa.jcyl.es/profesorado/es/formacion-profesorado/actualidad-formacion-profesorado/escuela-verano-2019-castilla-leon>
- Junta de Castilla y León. (s.f.). *Proyecto 'Creando código' - Curso 2018-2019*. Recuperado de <https://www.educa.jcyl.es/profesorado/es/formacion-profesorado/convocatorias-proyectos-relacionados-formacion-permanente-p/competencia-digital/convocatorias/proyecto-creando-codigo-curso-2018-2019>
- Junta de Castilla y León. (s.f.). *Proyecto de innovación 'TIC STEAM' - Curso 2016-2017*. Recuperado de <https://www.educa.jcyl.es/profesorado/en/formacion-profesorado/convocatorias-proyectos-relacionados-formacion-permanente-p/competencia-digital/convocatorias/proyecto-innovacion-tic-steam-curso-2016-2017>
- Kafai, Y. y Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan*. Recuperado de https://www.edweek.org/ew/articles/2013/09/01/kappan_kafai.html

- Kazakoff, E.R, Sullivan, A. y Bers, M. (2012). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Springer Science*. Recuperado de <https://link.springer.com/article/10.1007/s10643-012-0554-5>
- Laboda, X., Galimany, J., Pena, R.M. y Gual, A. (1985). *Software*. Barcelona, España: Ediciones Océano-Éxito.
- Lazzaro, N. (2004). *Why we Play Games: Four Keys to More Emotion without Story*. Oakland, Estados Unidos: XEODesign Inc. Recuperado de http://www.ergonomia.ca/doc/xeodesign_whyweplaygames_motivationaldimensionofdecisionmaking.pdf
- Lee, J. y Hammer, J. (2011). Gamification in education: What, how, why bother? *Academic Exchange Quarterly*, 15(2), 1-5.
- Lu, J. J., y Fletcher, G. H. L. (2009). Thinking about computational thinking. En S. Fitzgerald, S. M. Guzdial, G. Lewandowski, y S. A. Wolfman (Eds.). *Proceedings of the 40th SIGCSE Technical Symposium on Computer Science Education* (pp. 260-264). Nueva York, Estados Unidos: Association for Computing Machinery, Inc.
- Madill, H., Campbell, R.G., Cullen, D.M., Armour, M. A., Einsiedel, A. A., Ciccocioppo, A. L., et al. (2007). Developing career commitment in STEM-related fields: myth versus reality. En R. Burke, M. Mattis y E. Elgar (Eds.). *Women and minorities in science, technology, engineering and mathematics: Upping the numbers*. Northampton, Estados Unidos: Edward Elgar Publishing.
- Markert, L. R. (1996). Gender related to success in science and technology. *The Journal of Technology Studies*, 22.
- Meeth, L.R. (1978). Interdisciplinary studies: a matter of definition. *Change*, 10(7), 10.
- Metz, S. S. (2007). Attracting the engineering of 2020 today. En Burke, R. Mattis, M. y Elgar, E. (Eds.). *Women and minorities in science, technology, engineering and mathematics: Upping the numbers*. Northampton, Estados Unidos: Edward Elgar Publishing.
- Microes.org. (s.f.). *Características y Funcionalidades*. Recuperado de <http://microes.org/caracteristicas.php>

- Microsoft. (2019). *Step Counter*. Recuperado de <https://makecode.microbit.org/projects/step-counter>
- Microsoft. (2019). *Temperature blocks lesson*. Recuperado de <https://makecode.microbit.org/lessons/temperature>
- Micro:bit Educational Foundation. (s.f.). *Micro:bit Educational Foundation | micro:bit*. Recuperado de <https://microbit.org/>
- Ministerio de Educación y Formación Profesional. (2018). *Programación, robótica y pensamiento computacional en el aula*. Recuperado de <http://code.intef.es/wp-content/uploads/2018/10/Ponencia-sobre-Pensamiento-Computacional.-Informe-Final.pdf>
- Mioduser, D., Levy, S., y Talis, V. (2009). Episodes to scripts to rules: Concrete- abstractions in kindergarten children's explanations of a robot's behaviors. *International Journal of Technology and Design Education*, 19(1), 15–36.
- Mioduser, D., y Levy, S. (2010). Making sense by building sense: Kindergarten children's construction and understanding of adaptive robot behaviors. *International Journal of Computers for Mathematical Learning*, 15(2), 99–127.
- Muntean, C. I. (2011). Raising engagement in e-learning through gamification. En *Proc. 6th International Conference on Virtual Learning ICVL* (pp. 323-329).
- Orden ECI/3857/2007, de 27 de diciembre (2007, 27 de diciembre).
- Orden EDU/763/2017, de 31 de agosto (2017, 31 de agosto).
- Pandey, A. (2018). *Top 6 Benefits of Gamification in eLearning*. Recuperado de <https://elearningindustry.com/top-6-benefits-of-gamification-in-elearning>
- Political Intelligence (2014). *European Commissioners call on EU Education Ministers to promote coding in primary school*. Recuperado de <https://www.political-intelligence.com/european-commissioners-call-on-eu-education-ministers-to-promote-coding-in-primary-school/>
- Raspberry Pi Foundation (s.f.). *Frustration*. Recuperado de <https://codeclubprojects.org/en-GB/microbit/frustration/>

- Real Decreto 126/2014, de 28 de febrero. (2014, 28 de febrero).
- Resnick, M. (2006). Computer as paintbrush: Technology, play, and the creative society. En Singer D., Golikoff, R. y Hirsh-Pasek K. (Eds.). *Play = learning: How play motivates and enhances children's cognitive and social-emotional growth*. Nuevo York, Estados Unidos: Oxford University Press.
- Resnick, M. (2018). Learn to Code, Code to Learn. *EdSurge*. Recuperado de <https://www.edsurge.com/news/2013-05-08-learn-to-code-code-to-learn>
- Resnick, M., Flanagan, M., Kelleher, C., MacLaurin, M., Ohsima, Y., Perlin, K. y Torres, R. (2009). Growing up programming: Democratizing the creation of dynamic interactive media. En *Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems, CHI 2009*. Nueva York, Estados Unidos: ACM.
- Ross, P. y Howe, J. (1981). Teaching mathematics through programming: ten years on. En R. Lewis y D. Tagg (Eds.). *Computers in Education*. Amsterdam: North-Holland.
- Rushkoff, D. (2010). *Program or be programmed: Ten commands for a digital age*. New York, Estados Unidos: O/R Books.
- Scherer R. (2016). Learning from the Past-The Need for Empirical Evidence on the Transfer Effects of Computer Programming Skills. *Frontiers in psychology*, 7, 1390.
- Sengupta, P., Kinnebrew, J., Basu, S., Biswas, G., y Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18(2), 351-380.
- Simões, J., Redondo, R. D. y Vilas, A. F. (2013). A social gamification framework for a K-6 learning platform. *Computers in Human Behavior*, 29(2), 345-353.
- SparkFun Electronics (s.f.). *Getting Started with the micro:bit*. Recuperado de <https://learn.sparkfun.com/tutorials/getting-started-with-the-microbit/all>
- Steele, C. M. (1997). A threat in the air: How stereotypes shape intellectual identity and performance. *American Psychologist*, 52(6), 613-629.

Universidad de Valladolid (2010). *Graduado/a en Educación Primaria*. Valladolid, España: Universidad de Valladolid.

Wartella, E., y Jennings, N. (2000). Children and Computers: New Technology. Old Concerns. *The Future of Children*, 10(2), 31-43.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*. Nueva York, Estados Unidos: ACM.

Wing J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 366, 3717-3725.

ANEXO I – CONTEXTO DEL CENTRO

SITUACIÓN

El colegio se encuentra en una zona central de la capital. Cuenta con 2 entradas, la principal, sita en la calle Padre Claret, y la secundaria, en la avenida Cardenal Cisneros.

Existen 2 parques cercanos, el Salón y la Huerta Guadián, además de un complejo deportivo, el Campo de la Juventud.

También cercano se encuentra otro centro educativo de titularidad concertada, el Colegio Santo Domingo de Guzmán.

PLANTILLA DOCENTE

Actualmente, el centro cuenta con 2 líneas de Ed. Infantil y Ed. Primaria. El número total de docentes es de 7 en Ed. Infantil y de 12 de Ed. Primaria, 2 docentes de Ed. Física, 5 docentes en la sección Bilingüe, 2 docentes de Lengua Inglesa, 1 docente de Lengua Alemana, 1 docente de Música, 1 docente de Audición y Lenguaje, 1 docente de Pedagogía Terapéutica, 2 docentes con doble perfil de Audición y Lenguaje y Pedagogía Terapéutica, 1 docente de Religión y 3 miembros del equipo directivo (1 Director, 1 Jefe de Estudios y 1 Secretaria).

ALUMNADO Y FAMILIAS

En total hay 445 alumnos, siendo 148 de Ed. Infantil y 297 de Ed. Primaria.

El centro se encuentra situado en un barrio que podemos denominar de clase media o media-baja, lo que, junto con la posibilidad de libre elección de centro (Distrito Único), hace que la variedad en cuanto a alumnado y familias sea muy amplia.

El nivel socioeconómico de la mayoría de las familias es medio o medio-bajo, aunque también contamos con familias con una renta baja. Comparados con otros centros educativos públicos de Palencia, tenemos pocos alumnos inmigrantes o en riesgo de exclusión social.

INSTALACIONES DEL CENTRO

Además de las instalaciones típicas de un centro educativo, contamos con dos gimnasios (uno para trabajar la psicomotricidad en Ed. Infantil y otro para trabajar la Ed. Física en Ed. Primaria), un aula de PT y otra de AL, un laboratorio de idiomas y un antiguo laboratorio que se suele utilizar como aula de desdoble.

SERVICIOS DEL CENTRO

Entre los muchos servicios que ofrece centro destacan el programa madrugadores, la sección bilingüe, la revista escolar y un auxiliar de conversación en Lengua Inglesa.

PROGRAMAS Y PLANES DEL CENTRO

Entre otros, contamos con los siguientes programas y planes en nuestro centro:

- Iniciación en el aprendizaje de la Lengua Inglesa y a la música desde Ed. Infantil.
- Sección bilingüe (iniciada el curso 2008/2009).
- Plan de Lectura: se realizan diversas actividades en el aula, en la Biblioteca del centro, se celebran conmemoraciones importantes, el día del libro, cuentacuentos, jornadas culturales, visitas a la Biblioteca de Miguel de Unamuno, utilizamos la plataforma Leemos, realizamos talleres de poesía creativa...
- Programa THAO sobre prevención de la obesidad infantil.
- Programa de impartición de Lengua Alemana en 5º y 6º de Ed. Primaria: 1 hora semanal.
- Escuelas para la Sostenibilidad: programa de Ed. Ambiental que promueve la participación escolar en la resolución de los problemas socio ambientales a través de una propuesta de trabajo basado en una eco auditoría escolar.
- Plan de Trabajo Cooperativo: formación de centro sobre Trabajo Cooperativo iniciada el curso 2016/2017 para su paulatina implantación en las aulas. Una formación de centro es un plan de formación sobre un aspecto concreto en el que se embarca docentes de un centro educativo y que está recogido en la Programación General Anual.
- Programa Aulas de la Naturaleza.
- Cantania: proyecto musical en el que alumnos de Ed. Primaria de Castilla y León, tras preparar en su aula diversas canciones y coreografías, finalmente las interpretan en el auditorio Miguel Delibes de Valladolid junto a músicos profesionales.

LÍNEAS SEGUIDAS

En el CEIP Padre Claret queremos:

- Potenciar la igualdad, el respeto y la resolución de conflictos mediante el diálogo.
- Que las relaciones personales entre padres, maestros y alumnos sean respetuosas, aceptando las diferencias personales, culturales y étnicas.

- Ser un centro abierto a la Comunidad.
- Favorecer un buen clima de trabajo y esfuerzo en el aula.
- Que los alumnos adquieran hábitos de trabajo y estrategias de estudio.

ANEXO II - PROPUESTA DE GAMIFICACIÓN (HACKERS DEL CLARET)

La gamificación que propongo a continuación se llevará a cabo mientras dure el proyecto de introducción a la programación. La temática escogida ha sido la del mundo hacker, pues creo que ese mundo, lleno de claroscuros y con un cierto carácter de “prohibido”, gustará mucho a los alumnos de 6º de Ed. Primaria.

Las características más importantes de la gamificación propuesta son las siguientes:

INTRODUCCIÓN DE LA GAMIFICACIÓN EN EL AULA

Comenzaremos explicando a los alumnos qué es un hacker y los diferentes tipos de hackers que existen; White hats, Grey hats y Black hats.

- Los White hats son también llamados hackers éticos, pues ponen sus conocimientos al servicio de las empresas para detectar y mejorar sus vulnerabilidades en cuanto a seguridad informática se refiere.
- Los Grey hats son hackers que actúan de forma ética o no dependiendo de sus intereses personales. Son un híbrido entre los White hats y los Black hats.
- Los Black hats reciben también el nombre de crackers, pues su cometido es el de romper las barreras de seguridad de sistemas, redes o equipos informáticos para robar información, datos, contraseñas... es decir, cometer actos delictivos.

Explicaremos a los alumnos que, a través de la resolución de ciertos ejercicios y pruebas relacionadas con la programación informática que van a llevar a cabo en el aula, aprenderán una serie de habilidades que les convertirán en White hats.

AGRUPAMIENTOS

La clase se dividirá en 5 grupos de 5 alumnos. Yo seré el que elija los miembros de cada equipo para asegurarme cierta igualdad de nivel entre los grupos. Cada grupo elegirá un nombre. Cada alumno también deberá elegir un nombre distinto del suyo. Ambos nombres, constarán en la ficha de cada jugador.

INSIGNIAS

Cuando un alumno termine un apartado del proyecto (7 en total), ganará una insignia que podrá poner en su ficha personal. Todas las insignias son diferentes entre sí y hacen referencia a la temática y a los contenidos trabajados. Si un alumno obtiene todas las insignias, ganará la insignia “final” de White hat.

Las insignias que los alumnos pueden ganar son las siguientes:

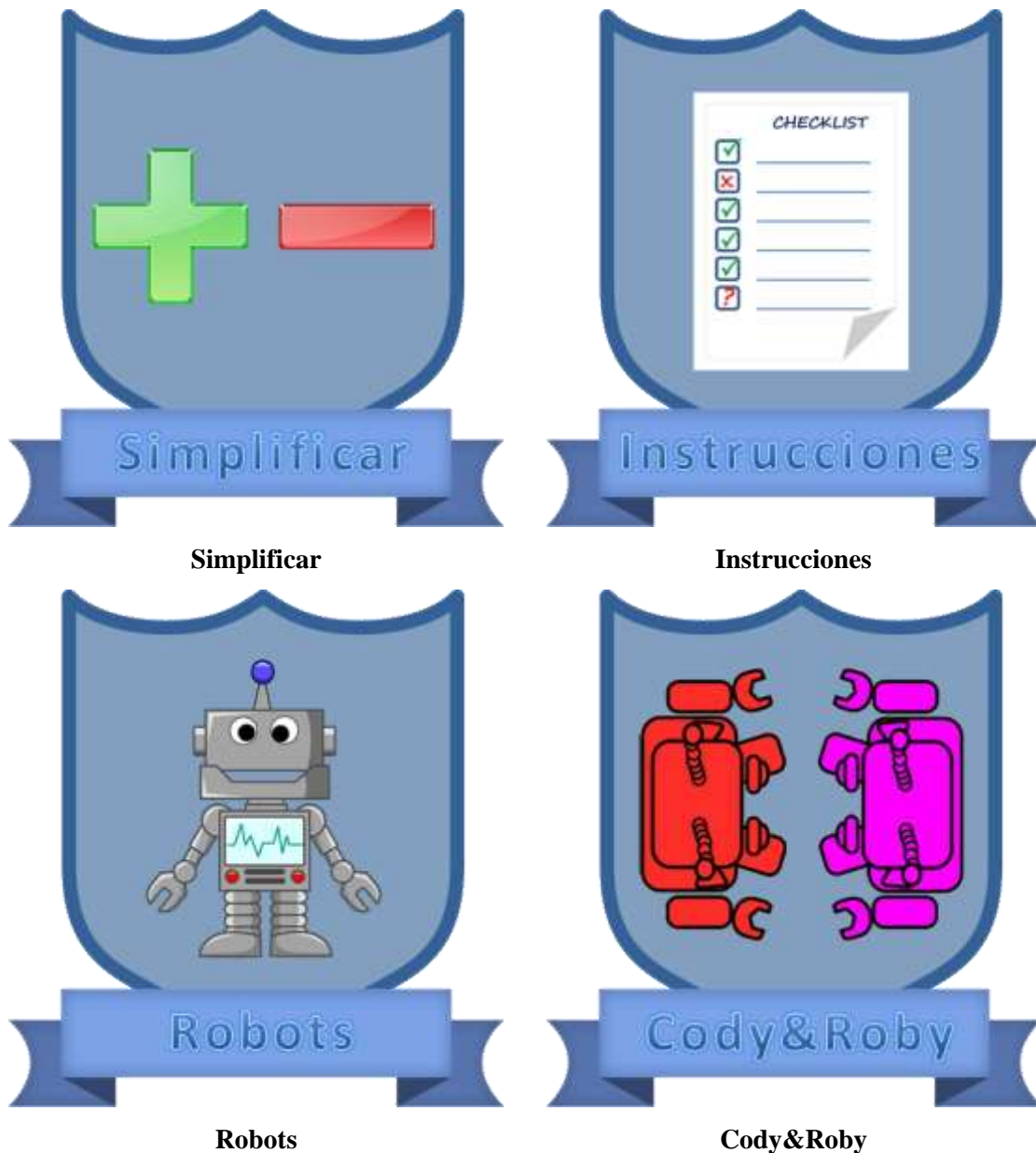


Tabla 1 - Insignias: Simplificar, Instrucciones, Robots, Cody&Roby.

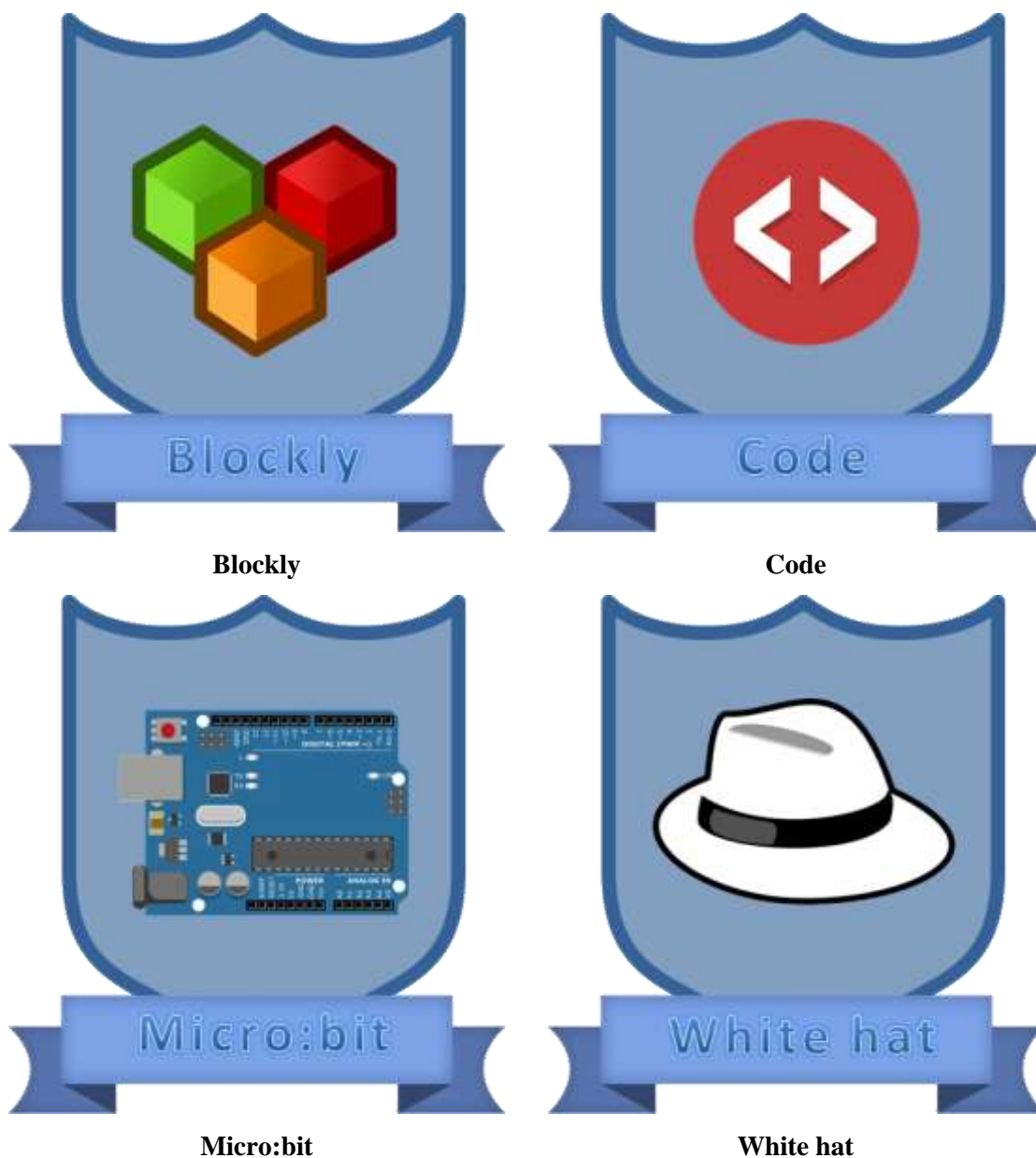


Tabla 2 - Insignias: Blockly, Code, Micro:bit, White hat

PUNTOS

A la vez que al final de cada tema damos una insignia, también pondré en marcha un juego de rol con puntos. Mi experiencia personal me indica que incluir cierto grado de competitividad sana añade un punto de emoción a la gamificación y hace que los alumnos se tomen más en serio la actividad, al mismo tiempo que se implican más en ella.

Como se ha visto en la dinámica de actividades propuesta, los alumnos recibirán puntos por cada juego/actividad/proyecto que resuelvan satisfactoriamente dentro del proyecto global de

programación. Cada alumno dispondrá de un “espacio personal” (papel con u nombre) situado en un papel continuo en la pared en el que el docente irá reflejando los puntos conseguidos por los alumnos.

Para incentivar la participación en el aula, también se otorgarán puntos por participar y tener iniciativa en cualquier clase de cualquier asignatura. De esta forma, los docentes irán “premiando” a los alumnos más participativos en cada clase, independientemente de la asignatura en la que nos encontremos. De esta forma, nos aseguramos la participación y el interés del alumnado no sólo mientras éstos están trabajando la programación, si no en todas las asignaturas. Para asegurarnos de esta circunstancia, todos los maestros que impartan docencia en el grupo, tendrán que conocer y participar en el proyecto de gamificación.

BATALLAS

Los puntos conseguidos nos servirán para celebrar “batallas”. Esto se realizará una o dos veces al mes. Los alumnos, antes de las batallas, reflejarán los puntos de su espacio personal en su ficha personal (que estarán a su vez dispuestas en otro papel continuo situado en la pared), aumentado, de esta forma, sus habilidades de hackers (ver parte ficha personal del alumno en la parte final del anexo). Esto se realizará coloreando un número de celdas determinado (dependiendo de los puntos que tengan para repartir: dos puntos, colorean dos celdas).

Las habilidades de hacker que los alumnos podrán conseguir o aumentar son:

- Acceso a webs.
- Descriptación de contraseñas.
- Control de sistemas.
- Usurpación de identidad.
- Control de dispositivos móviles.
- Asalto a redes.

Las batallas se podrán celebrar de forma individual (todos contra todos) o por equipos (cada equipo contra el resto de equipos). Las batallas serán de dos tipos:

- Batallas en las que se tendrán en cuenta los puntos de las 6 habilidades.
- Batallas en las que sólo se tendrán en cuenta los puntos solamente de algunas de las 6 habilidades.

De esta forma, al no saber nunca los alumnos qué habilidades necesitarán para ganar la siguiente batalla, nos aseguramos de que repartan de forma equitativa los puntos entre las diferentes habilidades.

Cada batalla tendrá una temática y un enunciado diferente para aumentar la motivación entre el alumnado. A modo de ejemplo:

El presidente del gobierno ha llamado al colegio y nos ha explicado que un grupo de Black hats quiere crackear el sistema informático del Banco Central para así robar el dinero que en él se encuentra. Necesitamos vuestra ayuda para reclutar al mejor equipo posible para luchar contra este grupo. Las habilidades que debe tener este equipo son: descriptación de contraseñas, control de sistemas y asalto a redes (es decir, en la batalla sólo contarán los puntos que tengáis en esas 3 habilidades).

PÁGINA WEB

<https://hackersdelclaret.weebly.com/>

En la gamificación incluyo una web de soporte, apoyo e información del estado del proyecto. A esta web podrán acceder tanto los alumnos como las familias. Por mi experiencia personal, considero este recurso como muy valioso, pues motiva especialmente a los alumnos a la vez que implica a las familias.

En esta web, se podrá encontrar la siguiente información:

- Una explicación de los objetivos de los distintos apartados del proyecto.
- Una explicación de la gamificación (con la posibilidad de descargar un PDF con una explicación más exhaustiva).
- Los progresos, los logros y los resultados de los alumnos, al incluir imágenes y vídeos de las actividades realizadas por los alumnos.
- Los mejores programas y actividades creadas por los alumnos
- Enlaces a las webs originales de donde se han sacado algunas de las ideas y de las actividades (motivando, de esta forma, la posibilidad de ampliación de contenidos).
- Las insignias que se pueden lograr.
- Imágenes, vídeos y resultados de las diferentes batallas.

Además de para motivar a los alumnos y de implicar a las familias, incluir cierto tipo de información (ficha personal del alumno, explicación del proyecto, explicación más exhaustiva del proyecto de gamificación y ejemplos de los programas creados por los alumnos), me parece muy interesante, ya que puede ofrecer a otros docentes un ejemplo y una motivación para tratar de implantar tanto el proyecto de introducción a la programación como la gamificación en sus centros y/o aulas. Por experiencia personal, la mejor motivación para que los docentes se animen a introducir nuevas experiencias en sus aulas es a través del ejemplo visto en otros docentes.

CUESTIONES A TENER EN CUENTA

- ¿Cómo pueden los alumnos conseguir insignias que durante el desarrollo normal del proyecto no han logrado conseguir?

Para aquellos alumnos que no consigan superar todos los objetivos marcados, voy a plantear un recurso que me gusta utilizar mucho en mi aula: alumnos mentores. Se trata de alumnos que, voluntariamente, trabajan ayudando a compañeros que van más rezagados o que tienen problemas para entender cierto tipo de contenidos o conceptos, explicando y trabajando con ellos dichos contenidos o conceptos.

Por propia experiencia, tengo que decir que este recurso funciona excepcionalmente bien, siempre y cuando se trate de alumnos que voluntariamente accedan a ello. La labor del docente, en este caso, será la de guiar al alumno mentor cuando lo necesite y procurar un clima adecuado para que esta actividad se desarrolle correctamente.

- ¿Reciben “premios” el ganador o ganadores de las batallas?

No soy muy partidario de dar premios materiales a los alumnos, pues suelen provocar un clima de competitividad malsana en el aula: aportan más perjuicios que beneficios. Pero también me parece justo y motivante para los alumnos que reciban algo a cambio del esfuerzo por conseguir puntos (cuantos más puntos tengan, más posibilidades tendrán de ganar una batalla). Por lo tanto, el ganador o ganadores de las batallas recibirán una insignia especial en forma de corona que podrán poner en su ficha personal (ver el final del anexo).



Figura 1 - Insignia de Batallas

- ¿Existe un premio final para los alumnos con mayor número de batallas ganadas?

Siguiendo la línea de no dar premios materiales, el premio final será elegir, al final del proyecto, el visionado de una película de su elección (previa aprobación por parte del docente) o la realización de una actividad de aula específica. Otra vez tengo que decir que, por propia experiencia, esto, que parece un premio trivial, gusta mucho a los alumnos.

FICHA PERSONAL DEL ALUMNO

NOMBRE REAL:					LOGO EQUIPO		
NOMBRE WHITE HAT:							
NOMBRE EQUIPO:							
INSIGNIAS CONSEGUIDAS							
HABILIDADES							PUNTOS
ACCESO A WEBS							
DESENCRIPTACIÓN DE CONTRASEÑAS							
CONTROL DE SISTEMAS							
USURPACION DE IDENTIDAD							
CONTROL DE DISPOSITIVOS MÓVILES							
ASALTO A REDES							
BATALLAS						PUNTOS TOTALES	

Figura 2 - Ficha personal del alumno

ANEXO III – ÍNDICE DE ACTIVIDADES

ACTIVIDADES DESENFUFADAS

ii. Simplificar comandos / dar y seguir explicaciones

1. Consideraciones previas
2. Actividades iniciales: simplificando comandos
3. Actividades de dar/seguir instrucciones

iii. Robots humanos

1. Consideraciones previas
2. Actividad 1: nivel fácil
3. Actividad 2: nivel medio
4. Actividad 3: nivel difícil
5. Actividades extra

iv. Cody&Roby

1. Consideraciones previas
2. Actividad 1: nivel fácil (tablero sin islas)
3. Actividad 2: nivel medio (tablero con islas)
4. Actividad 3: nivel difícil (ejercicios competitivos y de colaboración)

Juego “La Carrera”

Juego “La Serpiente”

Juego “El Duelo”

Juego “Rellenar el Tablero”

ACTIVIDADES ENCHUFADAS

i. Introducción a la programación basada en bloques – Blockly games

1. Consideraciones previas
2. Sesión 1: *Rompecabezas y Laberinto*
3. Sesión 2: *Pájaro, Tortuga y Música*
4. Sesión 3

ii. Trabajando la programación basada en bloques - Code.org

1. Consideraciones previas
2. Sesión 1 – Trabajo individual

3. Sesión 2 – trabajo individual
4. Sesión 1 – Trabajo individual
5. Sesión 2 – Trabajo individual
6. Sesión 3 – Trabajo en parejas
7. Sesión 4 – Trabajo en parejas
8. Sesión 5 – Trabajo en parejas
9. Sesión 6 – Trabajo individual
10. Sesión 7 – Trabajo en parejas
11. Sesión 8 – Trabajo en parejas
12. Sesión 9 – Trabajo individual
13. Sesión 10 – Trabajo en parejas
14. Sesión 11 – Trabajo en parejas

iii. Micro:bit

1. Consideraciones previas
2. Proyecto 1: Creamos un saludo
3. Proyecto 2: Creamos un termómetro
4. Proyecto 3: Creamos un cuenta pasos
5. Proyecto 4: Bola mágica de respuestas
6. Proyecto 5: ¿Cómo andas de pulso?
7. Proyecto 6: Creamos una brújula

ANEXO IV – ROBOTS HUMANOS

ACTIVIDAD 1 - FICHAS DE NÚMEROS ROMANOS

Ejemplo de serie de números del 1 al 100: recortar por la línea de puntos

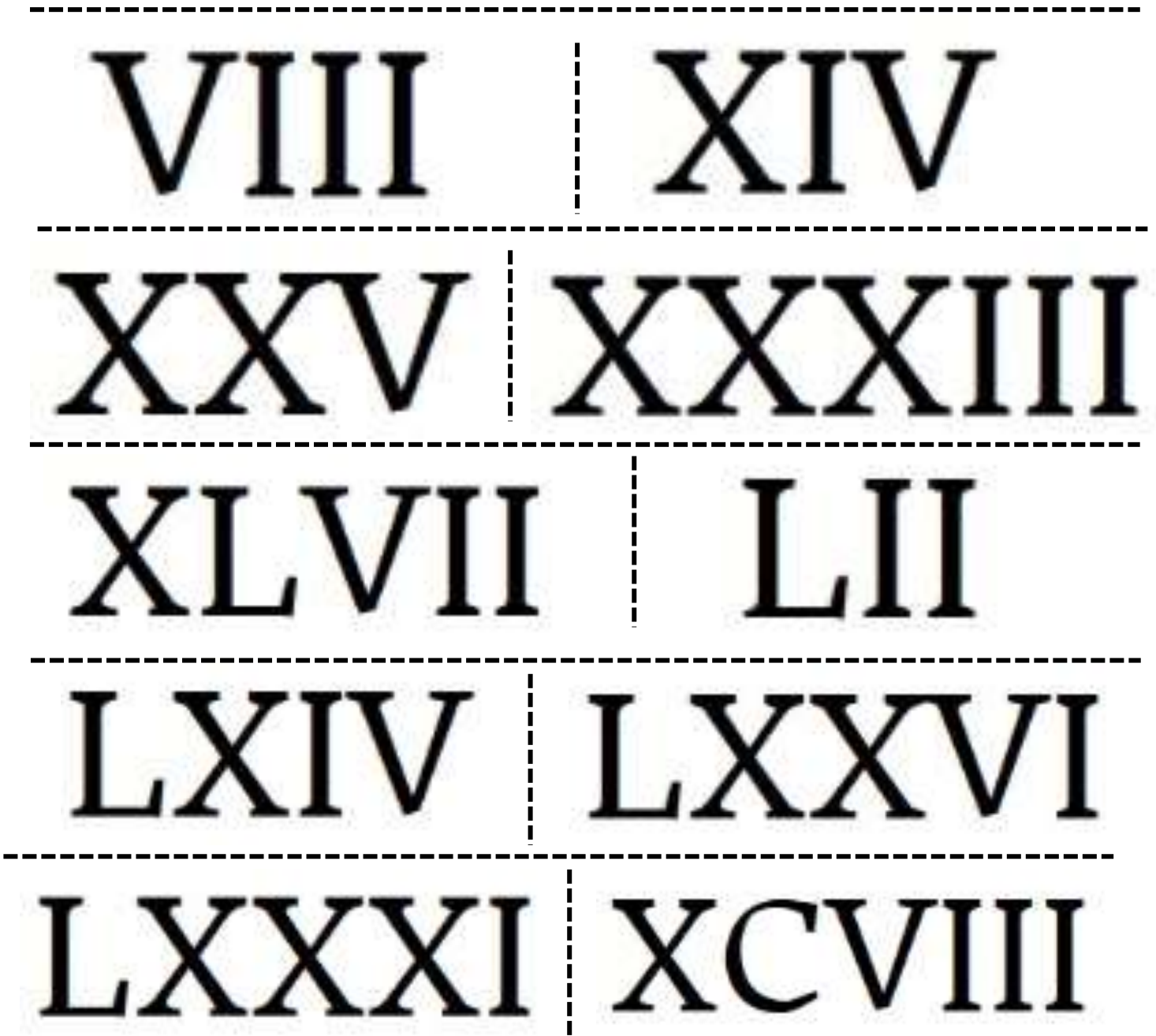


Figura 6 - Ejemplo de serie de números del 1 al 100

Ejemplo de grid (números romanos)

LII					LXIV		
		VIII					
	LXXXI					XXV	
			XXXIII				
					LXXVI		
		XIV					
	XLVII						
			XCVIII				START

Tabla 3 - Ejemplo de grid (números romanos)

ACTIVIDAD 2 – FICHAS DE OPERACIONES

Ejemplo de banco de operaciones sencillas: recortar por la línea de puntos

$17 + 4$

$33 + 18$

$49 + 22$

$37 + 29$

$78 + 54$

$45 - 14$

$79 - 27$

$28 - 3$

$88 - 33$

$114 - 79$

7×6

9×5

10×8

12×3

29×7

$8 / 2$

$15 / 3$

$72 / 8$

$100 / 2$

$120 / 4$

Ejemplo de grid (operaciones sencillas)

		31		80		132	
	25					5	
203			21				42
					52		9
START		4		45			
			35				51
				36			
50	71		30		66		55

Tabla 4 - Ejemplo de grid (operaciones sencillas)

ACTIVIDAD 3 – FICHAS DE VERBOS EN LENGUA INGLESA

Ejemplo de banco de verbos en lengua inglesa en presente y sus equivalencias en forma pasada regular o irregular (una de las dos formas es incorrecta: la forma verbal del medio es siempre la correcta): recortar por la línea de puntos

LIVE	LIVED	LIVEN
LIKE	LIKED	LIKETH
ANSWER	ANSWERED	ANSWERENTH
ASK	ASKED	ASKGH
CLOSE	CLOSED	CLOSEN
DANCE	DANCED	DANCUN

Figura 8 – Ejemplo de banco de verbos en Lengua Inglesa (formas regulares)

CUT	CUT	CUTTED
DO	DID	DOED
DRINK	DRANK	DRINKED
FIND	FOUND	FINDED
EAT	ATE	EATE
GIVE	GAVE	GIVED
HAVE	HAD	HAVED

Figura 9 – Ejemplo de banco de verbos en Lengua Inglesa (formas irregulares)

Ejemplo de grid (verbos en inglés)

CUT	HAD		ASKGHT			DOED	
				HAVED	ATE		START
ASKED			FOUND				
	CLOSEN			LIVED			DANCED
CUTTED					DRINKED		
FINDED		LIKED		DANCUN		GAVE	
	GIVED		DID		LIKETH		DRANK
	ANSWERED		EATED		CLOSED		LIVEN

Tabla 5 - Ejemplo de grid (verbos en inglés)

ANEXO V - RÚBRICAS

ROBOTS HUMANOS – ACTIVIDAD 1 – NIVEL FÁCIL

Rúbrica para el alumno que da las indicaciones

Nombre del alumno evaluado: _____

Fecha: _____

Instrucciones:

- Marca con una X el logro obtenido por tu compañero.
- Para evaluar el resto de ítems, el primer ítem (DINÁMICA) tiene que tener, al menos, una calificación de REGULAR.
- Suma la puntuación total. El resultado total tiene que ser de 9 puntos como mínimo.

	MUY MAL – 0	MAL – 1	REGULAR – 2	BIEN – 3	MUY BIEN - 4
DINÁMICA (Ítem indispensable)	No conoce la dinámica del juego.	Tiene muchas dudas sobre la dinámica del juego.	Conoce y aplica la dinámica del juego de forma regular (2 dudas).	Conoce y aplica la dinámica del juego de forma correcta (1 duda).	Conoce y aplica la dinámica del juego perfectamente y sin errores.
CONTENIDO (NÚMEROS ROMANOS)	No reconoce la mitad o más de los números romanos utilizados.	Reconoce algunos de los números romanos utilizados (4/5 errores).	Reconoce muchos de los números romanos utilizados en el juego (2/3 errores).	Reconoce la mayoría de los números romanos utilizados en el juego (1 error).	Reconoce todos los números romanos utilizados en el juego.
PRECISIÓN	No sabe cómo utilizar las órdenes (+ 5 errores).	Utiliza de forma regular las órdenes (3/4 errores).	Utiliza las órdenes de forma casi precisa (2 errores).	Utiliza las órdenes de forma precisa (1 error).	Utiliza las órdenes de forma precisa sin fallo.
IMPLICACIÓN	Su implicación es inexistente.	Se distrae con facilidad.	Se implica de forma regular en el juego.	Se implica en el juego.	Muestra una implicación total en el juego.

Tabla 6 – Actividad 1 - Rúbrica para el alumno que da las indicaciones

ROBOTS HUMANOS – ACTIVIDAD 1 – NIVEL FÁCIL

Rúbrica para el alumno que sigue las indicaciones

Nombre del alumno evaluado: _____

Fecha: _____

Instrucciones:

- Marca con una X el logro obtenido por tu compañero.
- Para evaluar el resto de ítems, el primer ítem (DINÁMICA) tiene que tener, al menos, una calificación de REGULAR.
- Suma la puntuación total. El resultado total tiene que ser de 9 puntos como mínimo.

	MUY MAL – 0	MAL – 1	REGULAR – 2	BIEN – 3	MUY BIEN - 4
DINÁMICA (Ítem indispensable)	No conoce la dinámica del juego.	Tiene muchas dudas sobre la dinámica del juego.	Conoce y aplica la dinámica del juego de forma regular (2 dudas).	Conoce y aplica la dinámica del juego de forma correcta (1 duda).	Conoce y aplica la dinámica del juego perfectamente y sin errores.
CONTENIDO (NÚMEROS ROMANOS)	No reconoce la mitad o más de los números romanos utilizados.	Reconoce algunos de los números romanos utilizados (4/5 errores).	Reconoce muchos de los números romanos utilizados en el juego (2/3 errores).	Reconoce la mayoría de los números romanos utilizados en el juego (1 error).	Reconoce todos los números romanos utilizados en el juego.
PRECISIÓN	No sabe cómo seguir las órdenes (+ 5 errores).	Sigue de forma regular las órdenes (3/4 errores).	Sigue las órdenes de forma casi precisa (2 errores).	Sigue las órdenes de forma precisa (1 error).	Sigue las órdenes de forma precisa sin fallo.
IMPLICACIÓN	Su implicación es inexistente.	Se distrae con facilidad.	Se implica de forma regular en el juego.	Se implica en el juego.	Muestra una implicación total en el juego.

Tabla 7 - Actividad 1 - Rúbrica para el alumno que sigue las indicaciones

ROBOTS HUMANOS – ACTIVIDAD 2 – NIVEL MEDIO

Rúbrica del alumno que da las indicaciones

Nombre del alumno evaluado: _____

Fecha: _____

Instrucciones:

- Marca con una X el logro obtenido por tu compañero.
- Para evaluar el resto de ítems, el primer ítem (DINÁMICA) tiene que tener, al menos, una calificación de BIEN.
- Suma la puntuación total. El resultado total tiene que ser de 10 puntos como mínimo.

	MUY MAL – 0	MAL – 1	REGULAR – 2	BIEN – 3	MUY BIEN - 4
DINÁMICA (Ítem indispensable)	No conoce la dinámica del juego.	Tiene muchas dudas sobre la dinámica del juego.	Conoce y aplica la dinámica del juego de forma regular (2 dudas).	Conoce y aplica la dinámica del juego de forma correcta (1 duda).	Conoce y aplica la dinámica del juego perfectamente y sin errores.
CONTENIDO (OPERACIONES MATEMÁTICAS BÁSICAS)	No resuelve satisfactoriamente la mitad o más de las operaciones propuestas.	Resuelve satisfactoriamente muchas de las operaciones propuestas (3/4 errores).	Resuelve satisfactoriamente la mayor parte de las operaciones propuestas (2 errores).	Resuelve satisfactoriamente la mayoría de las operaciones propuestas (1 error).	Resuelve satisfactoriamente todas las operaciones propuestas.
PRECISIÓN	No sabe cómo utilizar las órdenes (+ 5 errores).	Utiliza de forma regular las órdenes (3/4 errores).	Utiliza las órdenes de forma casi precisa (2 errores).	Utiliza las órdenes de forma precisa (1 error).	Utiliza las órdenes de forma precisa y correcta sin fallo.
IMPLICACIÓN	Su implicación es inexistente.	Se distrae con facilidad.	Se implica de forma regular en el juego.	Se implica en el juego.	Muestra una implicación total en el juego.

Tabla 8 - Actividad 2 - Rúbrica para el alumno que da las indicaciones

ROBOTS HUMANOS – ACTIVIDAD 2 – NIVEL MEDIO

Rúbrica del alumno que sigue las indicaciones

Nombre del alumno evaluado: _____

Fecha: _____

Instrucciones:

- Marca con una X el logro obtenido por tu compañero.
- Para evaluar el resto de ítems, el primer ítem (DINÁMICA) tiene que tener, al menos, una calificación de BIEN.
- Suma la puntuación total. El resultado total tiene que ser de 10 puntos como mínimo.

	MUY MAL – 0	MAL – 1	REGULAR – 2	BIEN – 3	MUY BIEN - 4
DINÁMICA (Ítem indispensable)	No conoce la dinámica del juego.	Tiene muchas dudas sobre la dinámica del juego.	Conoce y aplica la dinámica del juego de forma regular (2 dudas).	Conoce y aplica la dinámica del juego de forma correcta (1 duda).	Conoce y aplica la dinámica del juego perfectamente y sin errores.
CONTENIDO (OPERACIONES MATEMÁTICAS BÁSICAS)	No resuelve satisfactoriamente la mitad o más de las operaciones propuestas.	Resuelve satisfactoriamente muchas de las operaciones propuestas (3/4 errores).	Resuelve satisfactoriamente la mayor parte de las operaciones propuestas (2 errores).	Resuelve satisfactoriamente la mayoría de las operaciones propuestas (1 error).	Resuelve satisfactoriamente todas las operaciones propuestas.
PRECISIÓN	No sabe cómo seguir las órdenes (+ 5 errores).	Sigue de forma regular las órdenes (3/4 errores).	Sigue las órdenes de forma casi precisa (2 errores).	Sigue las órdenes de forma precisa (1 error).	Sigue las órdenes de forma precisa y correcta sin fallo.
IMPLICACIÓN	Su implicación es inexistente.	Se distrae con facilidad.	Se implica de forma regular en el juego.	Se implica en el juego.	Muestra una implicación total en el juego.

Tabla 9 - Actividad 2 - Rúbrica para el alumno que sigue las indicaciones

ROBOTS HUMANOS – ACTIVIDAD 3 – NIVEL DIFÍCIL

Rúbrica del alumno que da las indicaciones

Nombre del alumno evaluado: _____

Fecha: _____

Instrucciones:

- Marca con una X el logro obtenido por tu compañero/a y suma su puntuación.
- El resultado total tiene que ser de 11 puntos como mínimo.
- Para evaluar el resto de logros/ítems, el primer ítem (DINÁMICA) tiene que tener, al menos, una calificación de BIEN.

	MUY MAL – 0	MAL – 1	REGULAR – 2	BIEN – 3	MUY BIEN - 4
DINÁMICA (Ítem indispensable)	No conoce la dinámica del juego.	Tiene muchas dudas sobre la dinámica del juego.	Conoce y aplica la dinámica del juego de forma regular (2 dudas).	Conoce y aplica la dinámica del juego de forma correcta (1 duda).	Conoce y aplica la dinámica del juego perfectamente y sin errores.
CONTENIDO (PASADO VERBOS REGULARES - IRREGULARES EN INGLÉS)	No conoce la forma correcta de la mitad o más de los verbos propuestos.	Conoce muchas de las formas correctas de los verbos propuestos (3/4 errores).	Conoce la mayor parte de las formas correctas de los verbos propuestos (2 errores).	Conoce la mayoría de las formas correctas de los verbos propuestos (1 error).	Conoce todas las formas correctas de los verbos propuestos.
PRECISIÓN	No sabe cómo utilizar las órdenes (+ 5 errores).	Utiliza de forma regular las órdenes (3/4 errores).	Utiliza las órdenes de forma casi precisa (2 errores).	Utiliza las órdenes de forma precisa (1 error).	Utiliza las órdenes de forma precisa y correcta sin fallo.
IMPLICACIÓN	Su implicación es inexistente.	Se distrae con facilidad.	Se implica de forma regular en el juego.	Se implica en el juego.	Muestra una implicación total en el juego.

Tabla 10 - Actividad 3 - Rúbrica para el alumno que da las indicaciones

ROBOTS HUMANOS – ACTIVIDAD 3 – NIVEL DIFÍCIL

Rúbrica del alumno que sigue las indicaciones

Nombre del alumno evaluado: _____

Fecha: _____

Instrucciones:

- Marca con una X el logro obtenido por tu compañero/a y suma su puntuación.
- El resultado total tiene que ser de 11 puntos como mínimo.
- Para evaluar el resto de logros/ítems, el primer ítem (DINÁMICA) tiene que tener, al menos, una calificación de BIEN.

	MUY MAL – 0	MAL – 1	REGULAR – 2	BIEN – 3	MUY BIEN - 4
DINÁMICA (Ítem indispensable)	No conoce la dinámica del juego.	Tiene muchas dudas sobre la dinámica del juego.	Conoce y aplica la dinámica del juego de forma regular (2 dudas).	Conoce y aplica la dinámica del juego de forma correcta (1 duda).	Conoce y aplica la dinámica del juego perfectamente y sin errores.
CONTENIDO (PASADO VERBOS REGULARES - IRREGULARES EN INGLÉS)	No conoce la forma correcta de la mitad o más de los verbos propuestos.	Conoce muchas de las formas correctas de los verbos propuestos (3/4 errores).	Conoce la mayor parte de las formas correctas de los verbos propuestos (2 errores).	Conoce la mayoría de las formas correctas de los verbos propuestos (1 error).	Conoce todas las formas correctas de los verbos propuestos.
PRECISIÓN	No sabe cómo seguir las órdenes (+ 5 errores).	Sigue de forma regular las órdenes (3/4 errores).	Sigue las órdenes de forma casi precisa (2 errores).	Sigue las órdenes de forma precisa (1 error).	Sigue las órdenes de forma precisa y correcta sin fallo.
IMPLICACIÓN	Su implicación es inexistente.	Se distrae con facilidad.	Se implica de forma regular en el juego.	Se implica en el juego.	Muestra una implicación total en el juego.

Tabla 11 - Actividad 3 - Rúbrica para el alumno que sigue las indicaciones

MICRO:BIT – PROYECTO 6: CREAMOS UNA BRÚJULA

Fecha: _____

Nombre del alumno evaluado: _____

Instrucciones:

- Para evaluar el resto de ítems, el primer ítem (EDITOR DE CÓDIGO) tiene que tener, al menos, una calificación de BIEN.
- El resultado total tiene que ser de 20 puntos como mínimo.

	MUY MAL – 0	MAL – 1	REGULAR – 2	BIEN – 3	MUY BIEN - 4
EDITOR DE CÓDIGO	No conoce el uso del editor de código.	Tiene muchas dudas sobre el uso del editor de código.	Utiliza de forma casi correcta el editor de código (2 dudas).	Utiliza de forma correcta el editor de código (1 duda).	Domina la dinámica de uso del editor de código.
BLOQUES ESPECÍFICOS (VARIABLES, VARIABLES BOOLEANAS Y COMPARACIONES)	No conoce el uso de los bloques específicos utilizados en este proyecto.	3 errores en la creación o manejo de los bloques específicos utilizados en este proyecto.	2 errores en la creación o manejo de los bloques específicos utilizados en este proyecto.	1 error en la creación o manejo de los bloques específicos utilizados en este proyecto.	No comete errores en el uso de los bloques específicos utilizados en este proyecto.
CONTENIDOS (CIRCUNFERENCIA)	No conoce los grados contenidos en una circunferencia y su localización.	Muchas dudas sobre los grados contenidos en una circunferencia y su localización (más de 3 dudas).	Dudas sobre los grados contenidos en una circunferencia y su localización (2 dudas).	Pocas dudas sobre los grados contenidos en una circunferencia y su localización (1 duda).	Conoce sin error los grados contenidos en una circunferencia y su localización.
PROGRAMA	Tiene varios errores graves o más de 2 pequeños.	Tiene 1 error grave.	Tiene 2 errores pequeños.	Tiene 1 error pequeño.	El programa funciona sin errores.
STEAM	El dispositivo creado es inviable.	Hay errores graves en la aplicación práctica del dispositivo creado.	Hay errores en la aplicación práctica del dispositivo creado.	Hay errores leves en la aplicación práctica del dispositivo creado.	La aplicación práctica del dispositivo creado no presenta errores.
IMPLICACIÓN	Su implicación es inexistente.	Se distrae con facilidad.	Se implica de forma regular en el proyecto.	Se implica en el proyecto.	Muestra una implicación total en el proyecto.

Tabla 12 - Micro:bit - Proyecto 6 - Rúbrica para el alumno

ANEXO VI – CARACTERÍSTICAS DEL MATERIAL/SOFTWARE/HARDWARE UTILIZADO

A.CODY&ROBY

Se trata de un juego de mesa descargable y gratuito que permite a dos jugadores iniciarse en la programación de una forma sencilla y lúdica. El juego fue creado por Alessandro Bogliolo en noviembre de 2014 para ser utilizado durante la Semana Europea de la Robótica. Cuenta con una licencia Creative Commons Internacional 4.0 Atribución-NoComercial-SinDerivadas (CC BY-NC-ND 4.0).

El juego, para su versión en Ed. Primaria, incluye en su descarga el tablero, las fichas-robot, la caja para guardar el juego y las fichas de movimiento (adelante, derecha e izquierda). Estas últimas pueden descargarse en color o en blanco y negro (aspecto interesante para, antes de iniciar el juego, trabajar la implicación y el interés del alumnado por el mismo). El número total de cartas de movimiento que se necesitarán, dependerá del juego a desarrollar.

La web del proyecto cuenta con vídeos explicativos de cómo jugar a los 4 juegos propuestos en la misma: rellenar todo, duelo, carrera y serpiente.

La mecánica del juego es muy sencilla:

- Se juega en parejas (un miembro de la pareja es el “programador” y el otro el “robot”).
- Se elige un juego (con un objetivo final).
- El programador, mediante el uso de las cartas de movimiento, da instrucciones al robot para que se mueva por el tablero de tal forma que logre realizar con éxito el objetivo final pretendido.
- Los papeles de ambos alumnos se pueden intercambiar.

B.BLOCKLY

Blockly es una web creada por Google cuyo principal objetivo es enseñar los rudimentos de la programación por bloques. Es un proyecto de código abierto y gratuito. La web cuenta con

varios juegos y cada juego propone varios ejercicios, ordenados de menor a mayor complejidad, que habrá que resolver para ir avanzando en el conocimiento de los principales aspectos y conceptos de la programación: bucles, condicionales y funciones. Cada juego permite trabajar un concepto distinto aunque éstos se van reforzando al tiempo que se va avanzando en la web.

Si bien la propia página guarda de forma automática los avances conseguidos en los distintos juegos, en la web de desarrollo (GitHub) se puede, entre otras cosas, descargar el programa para poder jugar offline sin depender de esta forma de la conexión a Internet (opción que es mi preferida). Considero esta opción más útil que la online, además de para no colapsar la red del cole evitando los fallos de conexión, para que no haya problemas de pérdida de logros (por si falla el guardado online automático o los alumnos cambian de mini portátil).

Existe la posibilidad de jugar o descargar la aplicación en varios idiomas. Valoraremos, junto con el docente de inglés, la posibilidad de descargarla en inglés para trabajar conceptos de lengua inglesa a la vez que trabajamos los conceptos de programación. Yo, en mi secuencia de actividades, y por tratarse de las primeras actividades de trabajo con la programación por bloques, he considerado que sería mejor trabajar directamente en lengua castellana, para facilitar la tarea a los alumnos y que se concentren solamente en los conceptos de programación.

La interfaz de la web es sencilla pero atractiva. Cada juego consta de:

- Selector de nivel: muestra en qué nivel estás, los niveles que hay en cada juego y permite seleccionar otro nivel (opción muy útil si te quedas atascado en algún nivel en concreto).
- Selector de idioma.
- Icono para compartir tus avances (muy útil para realizar la evaluación: los propios alumnos nos pueden enviar en un mail un enlace que refleje en qué nivel se llegan).
- Icono para cambiar el personaje que realiza las acciones: cambia el personaje y el fondo en el que se encuentra.
- Panel con un personaje que nos indica qué tenemos que realizar en cada ejercicio propuesto y además de obedecer las acciones que hemos programado. Cuenta con un icono para “ejecutar el programa” que hemos creado para comprobar, de forma visual, si hemos realizado el ejercicio correctamente o no.
- Programador por bloques. Se divide en dos partes: listado de bloques que puedes utilizar y panel para colocar los bloques que resolverán el ejercicio propuesto.

C.CODE.ORG

Se trata de una webapp con juegos para trabajar a distintos niveles (principiante, medio y avanzado) los conceptos más importantes relacionados con la programación. Se trata de una web que trabaja conceptos más avanzados que los vistos en la web de Blockly.

La webapp está claramente enfocada al mundo educativo. Así, permite a los docentes crear una cuenta de aula y crear cuentas a sus alumnos. Aunque también se puede jugar sin autenticarte, esta opción no guarda los avances conseguidos, por lo que nuestra mejor opción será la de crearnos una cuenta de profesor para aprovechar todas las ventajas que la web nos proporciona.

La web cuenta con multitud de puzzles organizados por lecciones. De hecho, al principio, el gran número de recursos a tu disposición puede ser un poco apabullante.

La **dinámica** de funcionamiento de Code.org es sencilla:

- Después de elegir una lección o un curso, todos los ejercicios propuestos en éste cuentan con un ESPACIO DE TRABAJO (lugar en el que se tienen que ir colocando los bloques) y que en muchos casos te indica el número de bloques que deberías usar para resolver dicho puzzle.
- Cada vez que resolvemos un ejercicio de forma correcta (es decir, utilizando el número de bloques que nos propone el programa), el punto que representa esta lección en el “VISOR DE AVANCES” que está en la parte superior de la pantalla, se pondrá de color verde oscuro. Esto nos indicará los ejercicios que tenemos resueltos (para más información sobre este punto, ver **ANEXO VI**).
- Si resolvemos un ejercicio utilizando un número mayor de bloques, el sistema nos dejará pasar al siguiente ejercicio, pero no considerará del todo correcta nuestra respuesta, por lo que en el “visor de avances” que hay en la parte superior no se coloreará de verde oscuro. Con este sistema se evita que los alumnos se queden “atascados” en un ejercicio determinado, dejando que avancen, pero les ofrece feedback inmediato de los ejercicios que todavía no han resuelto y que deben resolver (para más información sobre este punto, ver **ANEXO VI**).
- Niveles especiales: dependiendo de la lección la web nos permite el acceso a niveles especiales. Estos niveles especiales tienen ciertas características:
 - Algunos de estos niveles no nos permiten modificar los bloques que nos ofrecen. Tenemos que resolver el ejercicio reorganizando los bloques existentes en el espacio de trabajo.

- En algunos de estos niveles nos dan una serie de bloques que no podemos eliminar. Tenemos que resolver el ejercicio añadiendo más bloques y/o modificando los que están en el espacio de trabajo.
- En algunos niveles accedemos a ejercicios especiales (llamados desafíos).
- En algunos niveles se nos hace una pregunta sobre lo que pasará con los bloques que tenemos (no nos deja modificar los mismos). La resolución del ejercicio se basa en responder de modo correcto a la pregunta planteada.

El sistema es tan flexible que incluso nos permite “esconder” las lecciones que no queramos que los alumnos realicen para adaptarnos todavía más a nuestra realidad de aula.

Como hemos dicho con anterioridad, la web nos permite crear una **cuenta de docente** para facilitar el proceso de seguimiento de los avances de los alumnos. Para llevar a cabo este seguimiento, crearemos una “SECCIÓN” en nuestro “SALÓN DE CLASE”, es decir, un aula virtual donde iremos agregando a los alumnos. Esto nos permitirá ver información muy completa sobre cada alumno: los avances en cada lección; las lecciones y los ejercicios que les faltan por completar y los niveles completados y las líneas de código que han creado (para más información sobre este punto, ver **ANEXO VI**).

D.MICRO:BIT

Micro:bit es básicamente un “ordenador de bolsillo” con el que se puede programar, de una forma muy sencilla e intuitiva, multitud de programas utilizando un lenguaje de programación basado en bloques. Además, y gracias a su pequeño tamaño, también te permite crear proyectos basados en la metodología STEAM utilizando materiales fáciles de conseguir.

Cada Micro:bit dispone de:

- 25 LEDs: pueden mostrar texto, números e iconos simples.
- 2 botones programables que, al ser pulsados, mandan información a la placa para ejecutar una acción determinada.
- Pines de entrada y salida: la placa dispone de 25 pines de entrada y salida para conectar motores, sensores, leds,...
- Sensor de luz: los leds actúan como sensores de entrada y recogen información sobre la luz ambiente.
- Sensor de temperatura: el microprocesador de la placa actúa como sensor de temperatura.

- Acelerómetro: el sensor puede medir la aceleración de la placa cuando ésta se mueve.
- Brújula: el sensor puede detectar el campo magnético terrestre por lo que sabe cómo está orientado el Micro:bit.
- Comunicación inalámbrica vía Radio: un Micro:bit se puede conectar con otro u otros Micro:bit a través de su sensor de radio.
- Comunicación inalámbrica vía Bluetooth: un Micro:bit puede mandar y recibir información vía Bluetooth de un PC, Tablet o Smartphone.
- USB: el Micro:bit se conecta a un PC vía USB para recibir el archivo con la programación. También se puede alimentar utilizando dicho puerto.
- Conector para batería externa: el Micro:bit, cuando no está conectado a un PC vía USB, recibe la energía que necesita para funcionar de una batería externa (una caja con 2 pilas conectadas que se conectan a la placa mediante un conector situado en la parte trasera).

Micro:bit es un proyecto de la BBC lanzado en 2016. Su objetivo fundamental era y es acercar el mundo digital y la programación a los alumnos desde una perspectiva creativa.

En su página web hay multitud de recursos y actividades ya creadas y que se pueden utilizar. De hecho, animan a todo el mundo a compartir sus proyectos.

El lenguaje utilizado es el de bloques (muy parecido al popular Scratch), aunque también se puede trabajar directamente con lenguaje JavaScript (para alumnos muy avanzados, nunca para alumnos de Ed. Primaria).

El editor se encuentra online y su dinámica de uso, básicamente, es la siguiente:

- Abrimos el editor online.
- Creamos un programa utilizando los bloques a nuestra disposición.
- Guardamos el programa: necesitamos ponerle un nombre.
- Descargamos el programa creado a nuestro ordenador (los programas se guardan automáticamente en la plataforma).
- Copiamos/Movemos el archivo del programa del ordenador a nuestro Micro:bit.

Como Scratch, el editor de Micro:bit cuenta con multitud de bloques diferentes que ofrecen multitud de opciones de creación diferentes. Además, en la parte superior izquierda de la pantalla del editor, se puede ver un Micro:bit virtual en el que se puede comprobar de forma directa el resultado de nuestra programación, lo que ofrece una evaluación inmediata de la misma.

La importancia que Micro:bit está tomando en el mundo educativo es tal, que incluso Scratch ha incluido en su última actualización un plug-in para poder trabajar directamente con él.

ANEXO VII – CODY&ROBY

MATERIALES DE LA ACTIVIDAD CODY&ROBY

Tablero

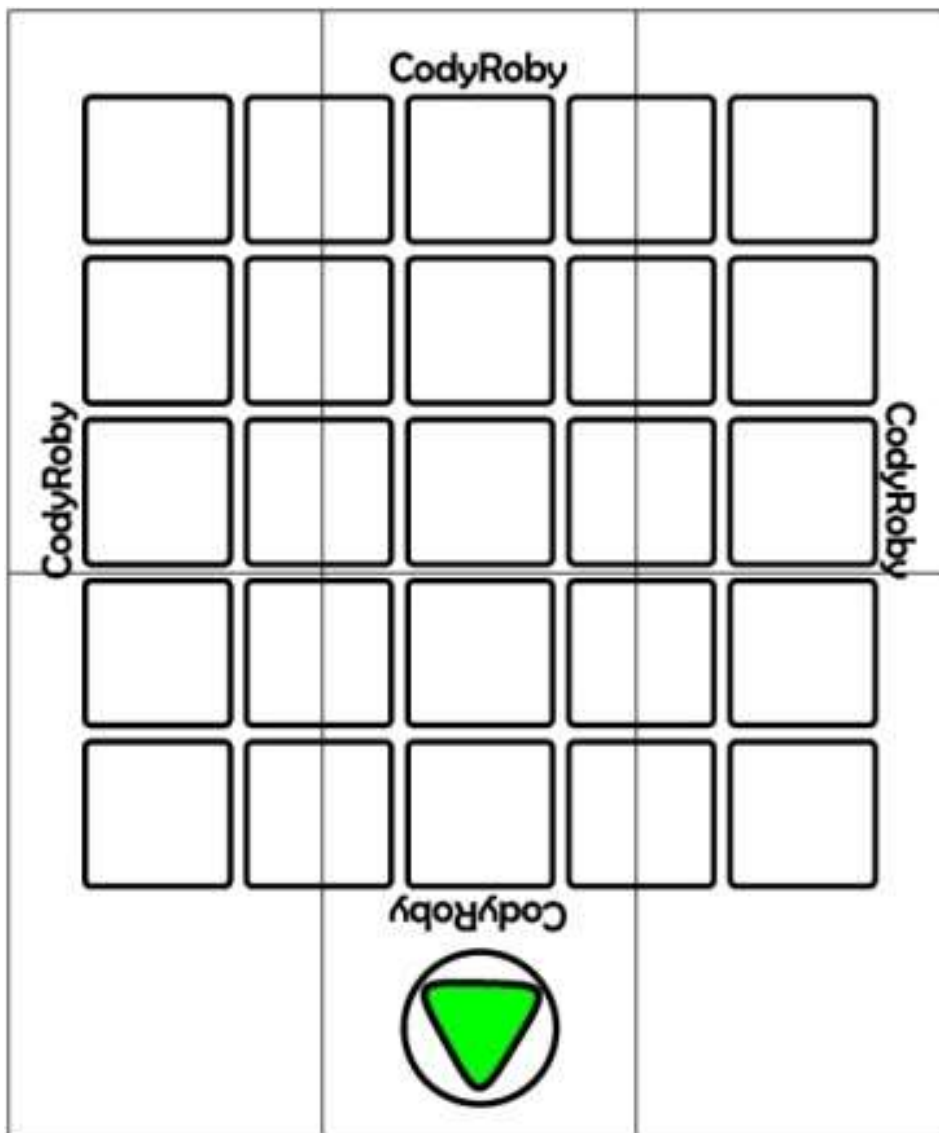


Figura 10 - Tablero

Islas

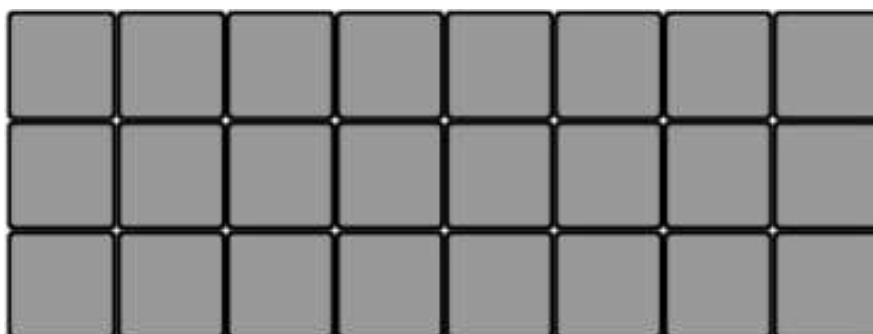


Figura 11 - Islas

Cartas de movimiento

AVANZAR

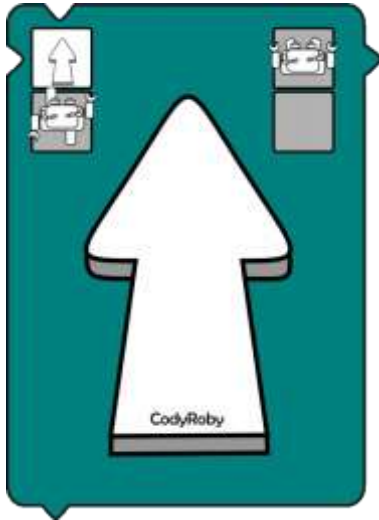


Figura 12 - Carta avanzar

GIRAR IZQUIERDA

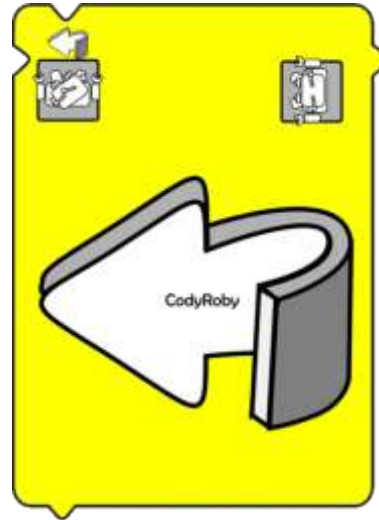


Figura 13 - Carta girar izquierda

GIRAR DERECHA

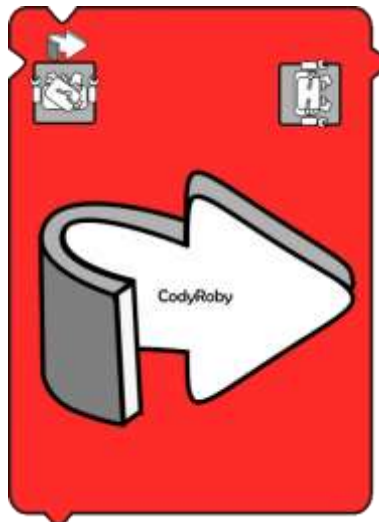


Figura 14 - Carta girar derecha

BUCLE

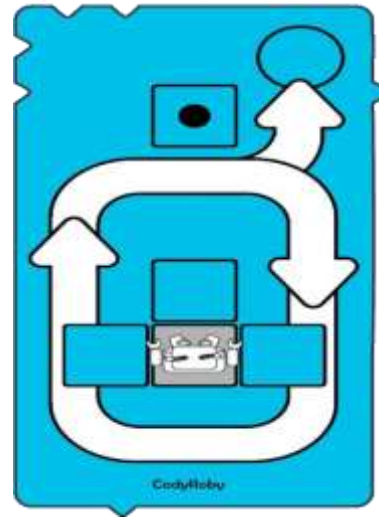


Figura 15 - Carta bucle

Fichas robot

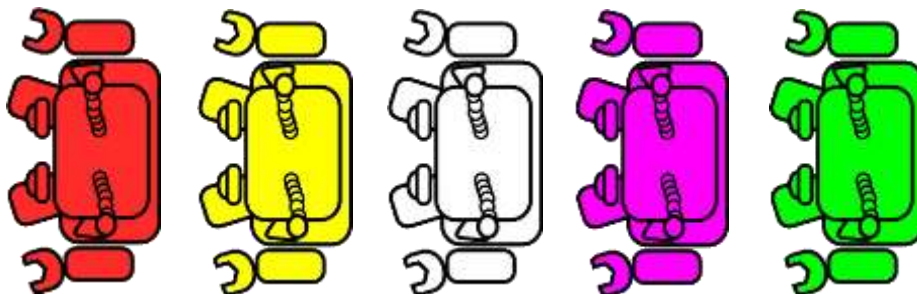


Figura 16 - Fichas robot

ACTIVIDAD 1 – NIVEL FÁCIL – EJEMPLOS DE ACTIVIDADES

Ejemplo 1

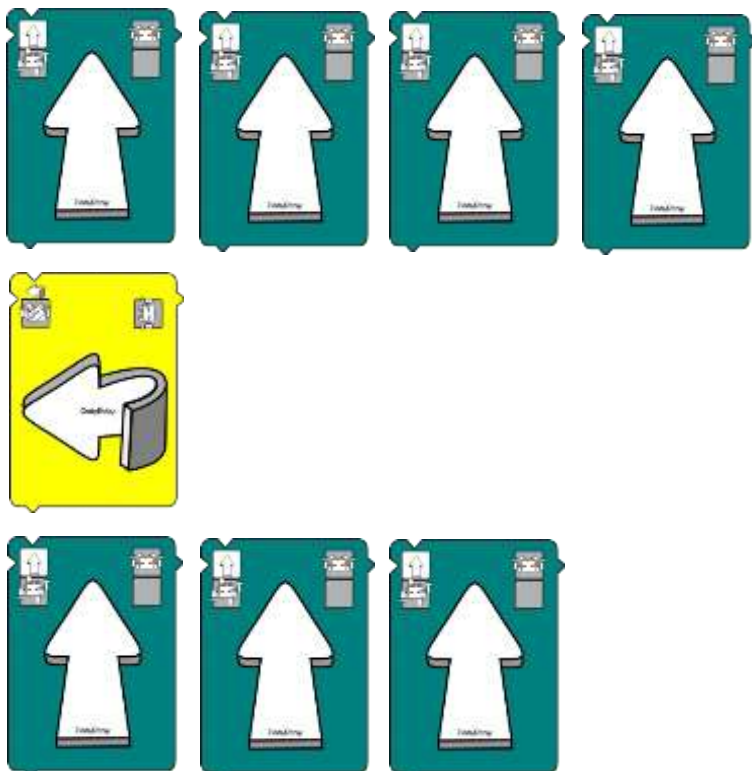
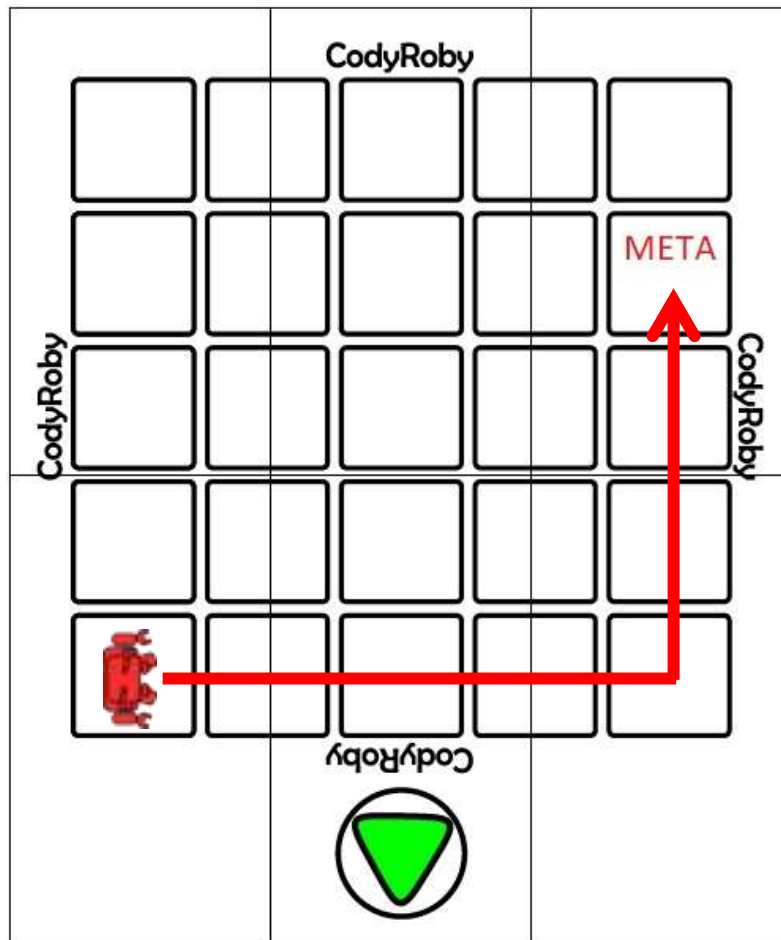
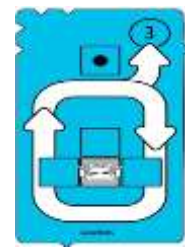
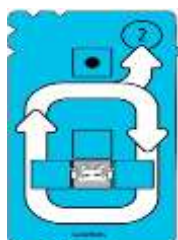
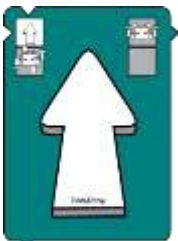
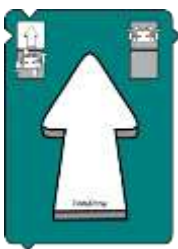
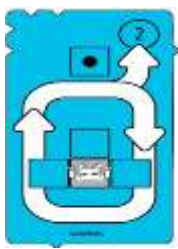
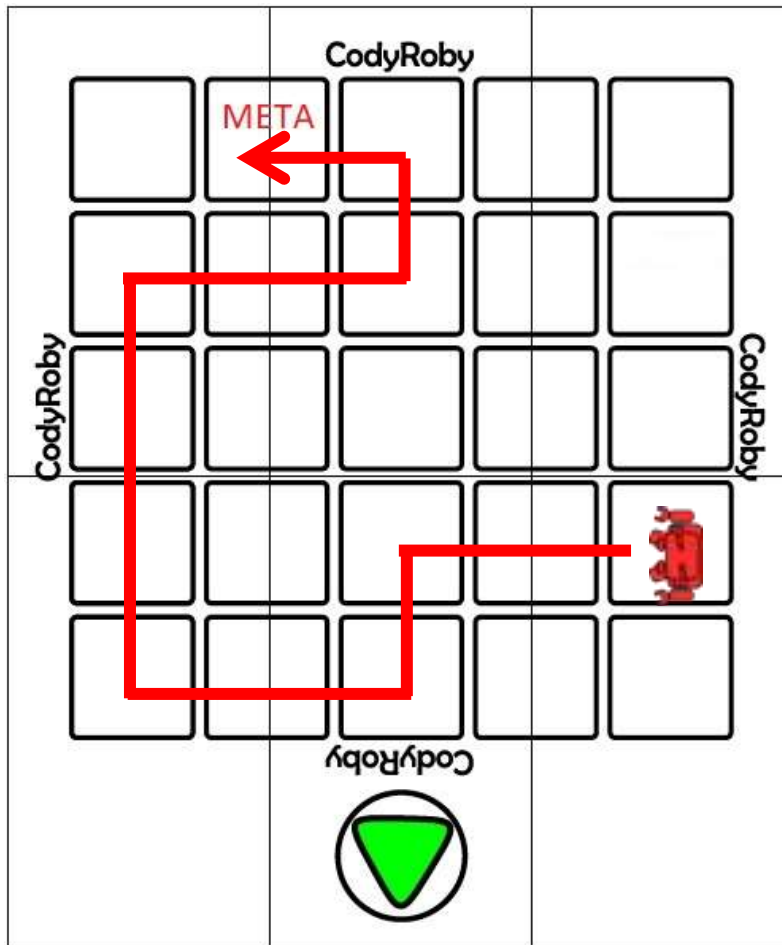


Figura 17 – Ejemplo 1 (nivel fácil)

Ejemplo 2



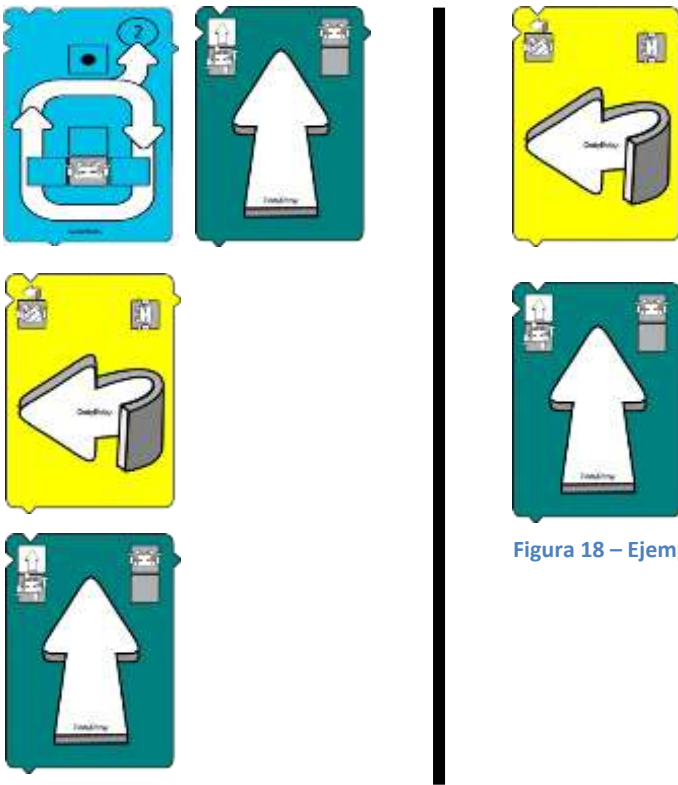


Figura 18 – Ejemplo 2 (nivel fácil)

ACTIVIDAD 2 – NIVEL MEDIO – EJEMPLOS DE ACTIVIDADES

Ejemplo 1

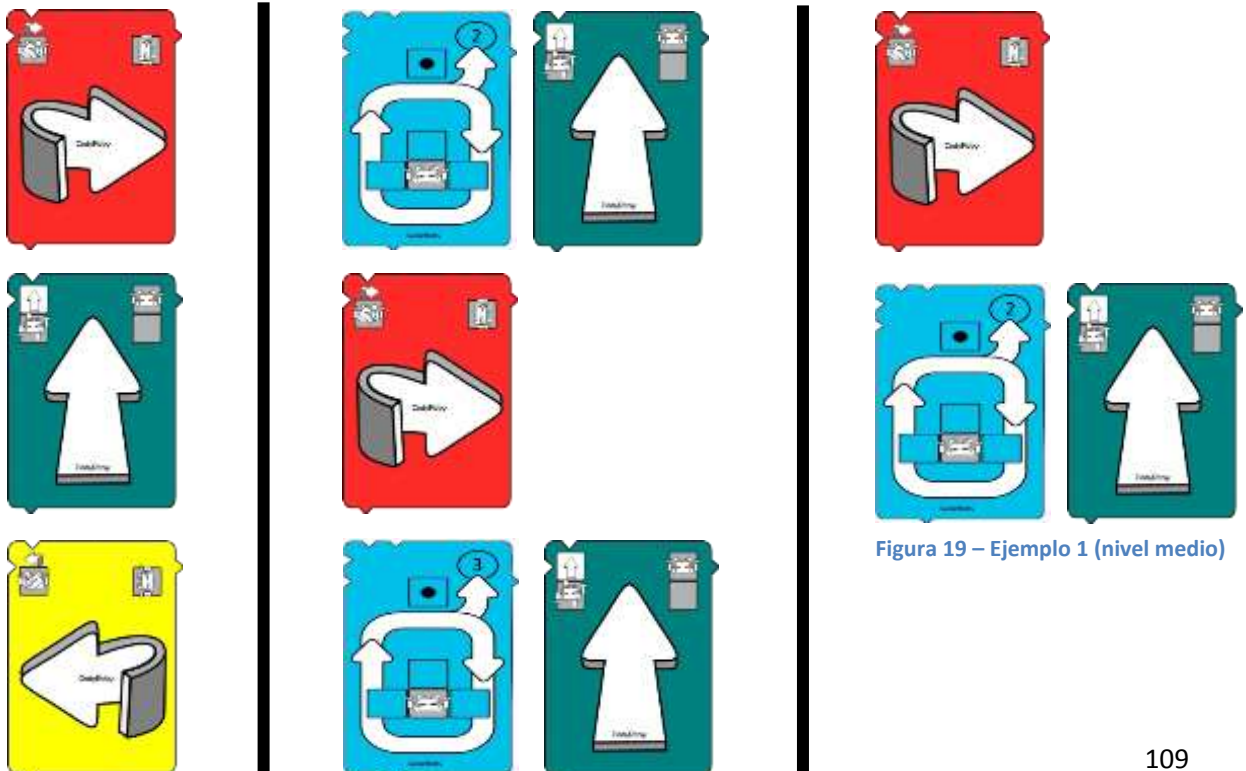
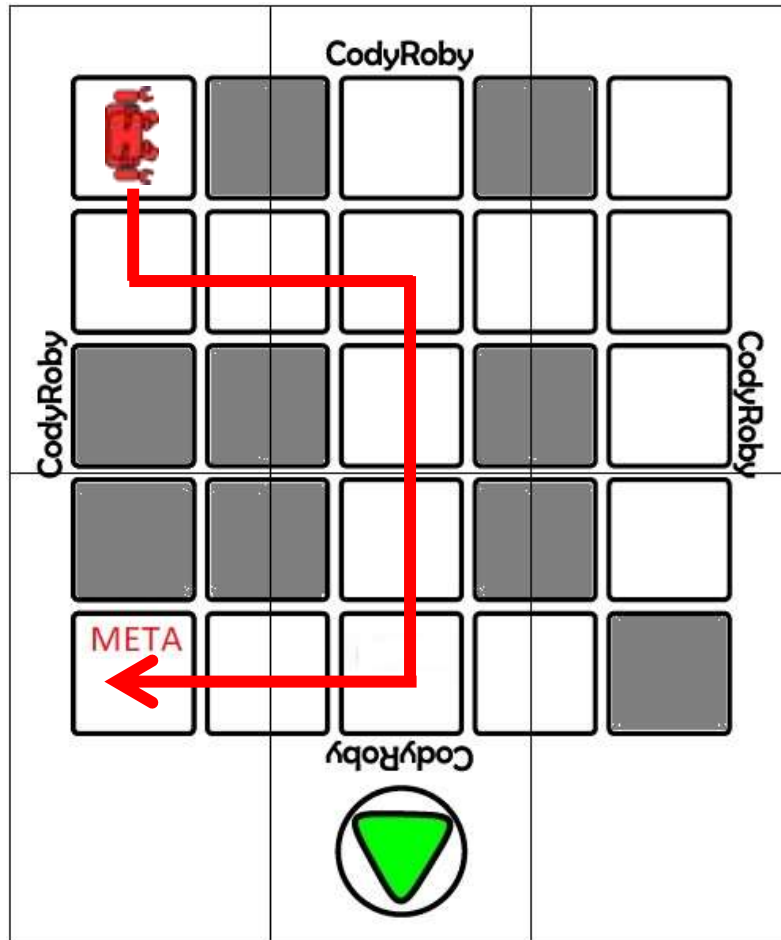


Figura 19 – Ejemplo 1 (nivel medio)

Ejemplo 2

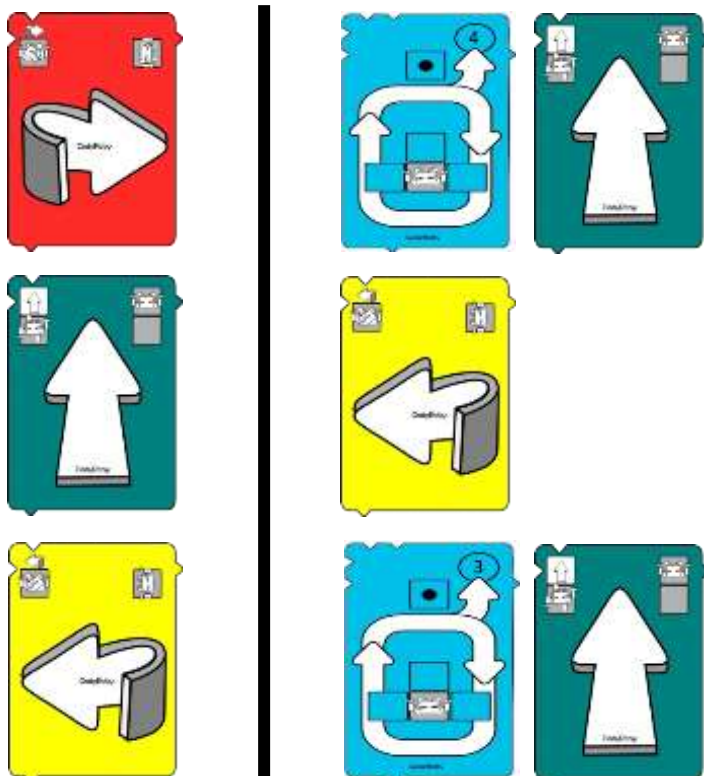
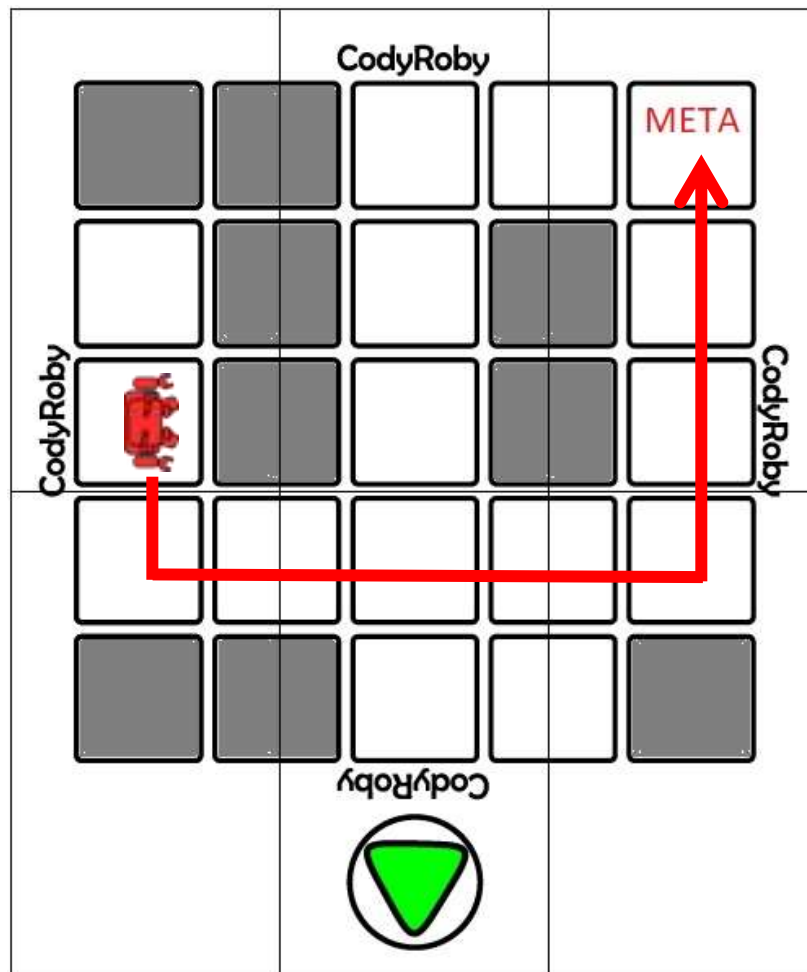


Figura 20 – Ejemplo 2 (nivel medio)

ACTIVIDAD 3 – NIVEL DIFÍCIL – EJEMPLOS DE ACTIVIDADES

Juego “La Carrera”

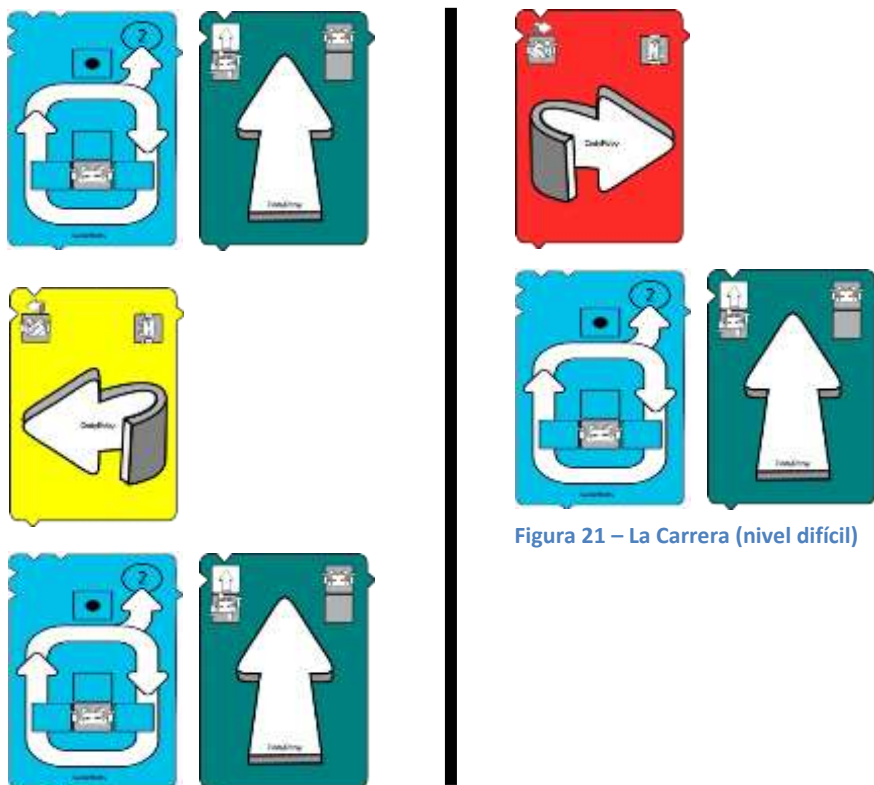
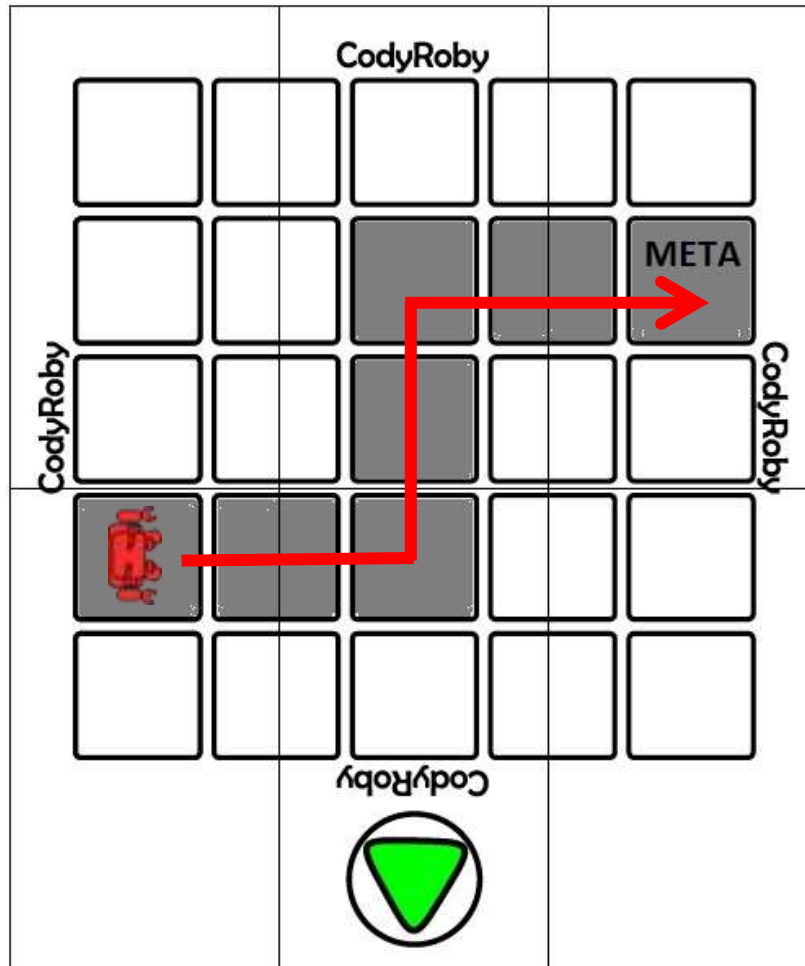
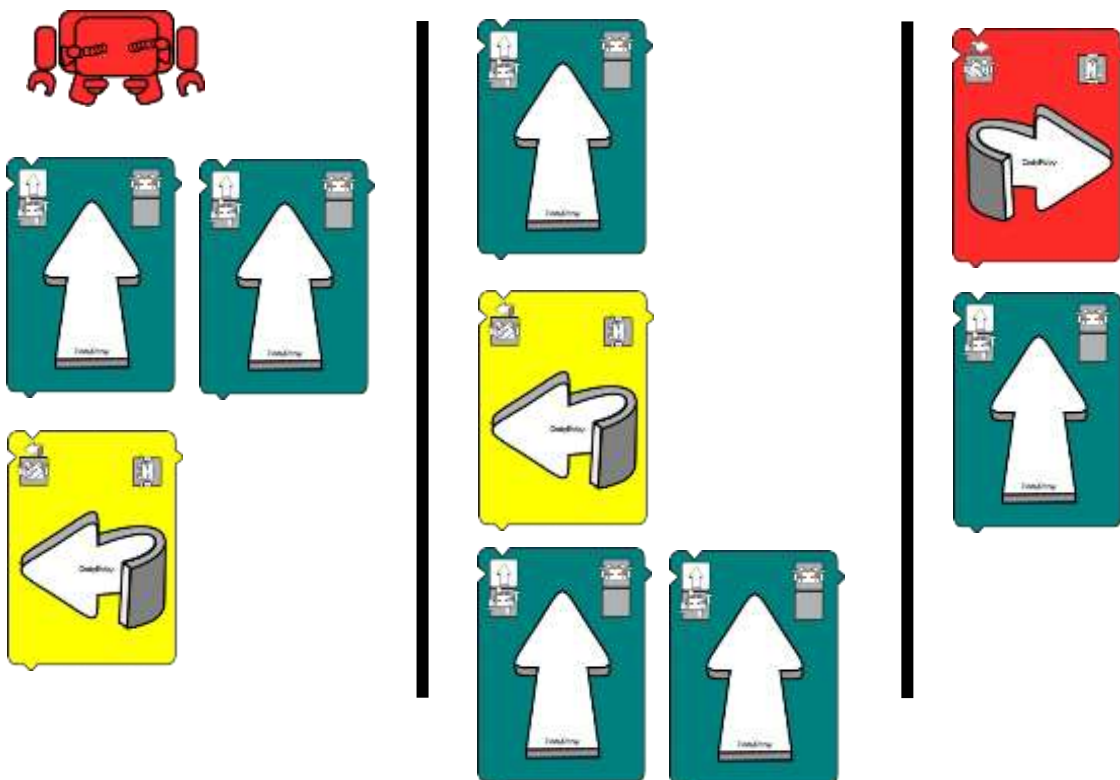
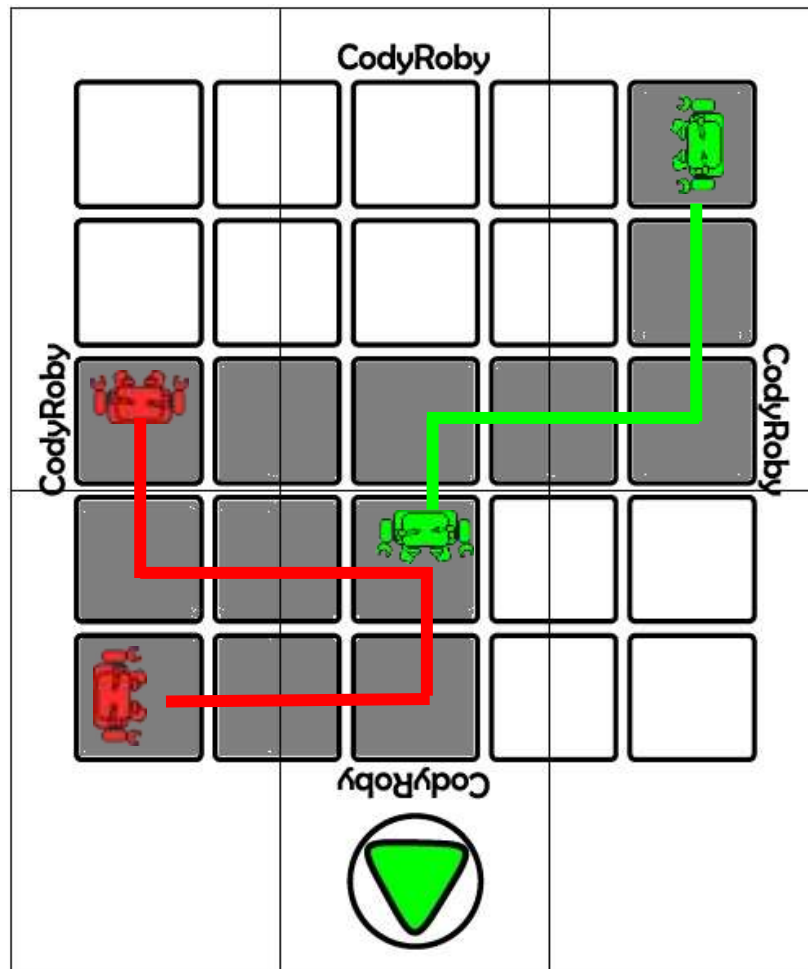


Figura 21 – La Carrera (nivel difícil)

Juego “La Serpiente”

Comienza la partida el jugador con el robot rojo, por lo que el jugador con el robot verde, ganador de la partida, acaba con una carta utilizada menos.



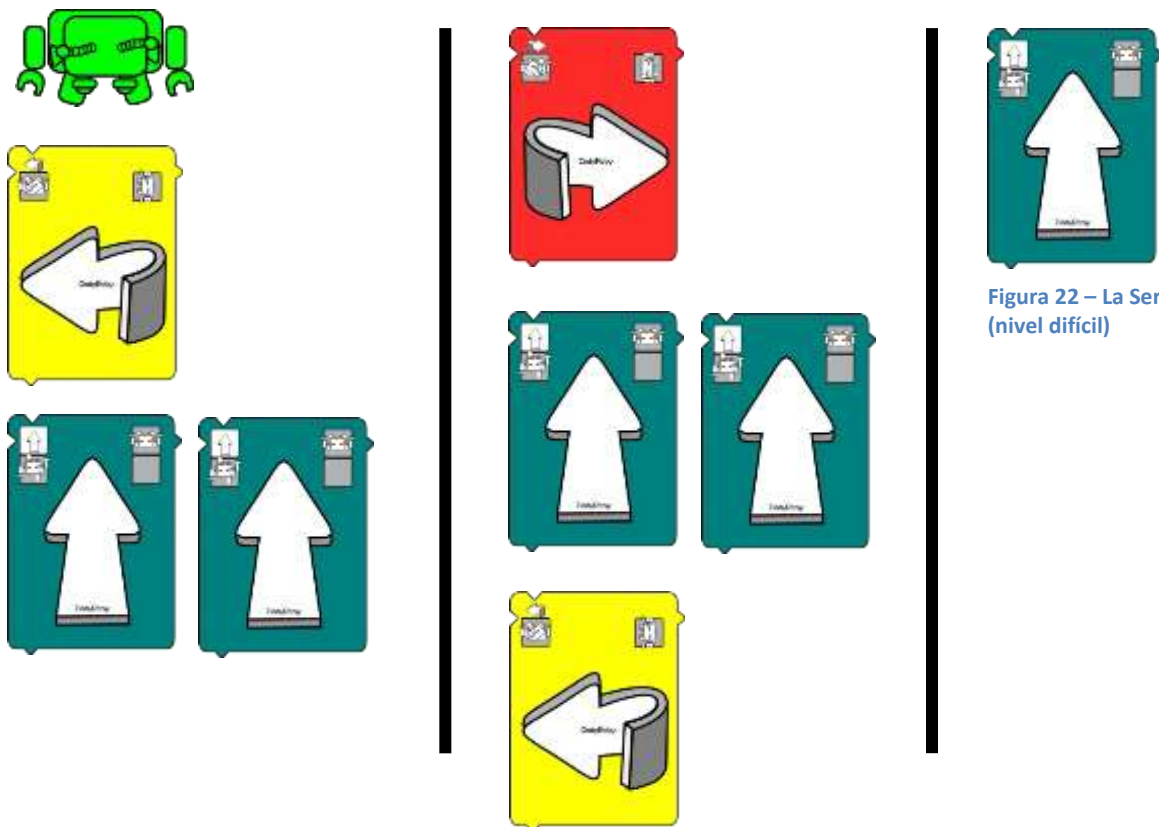
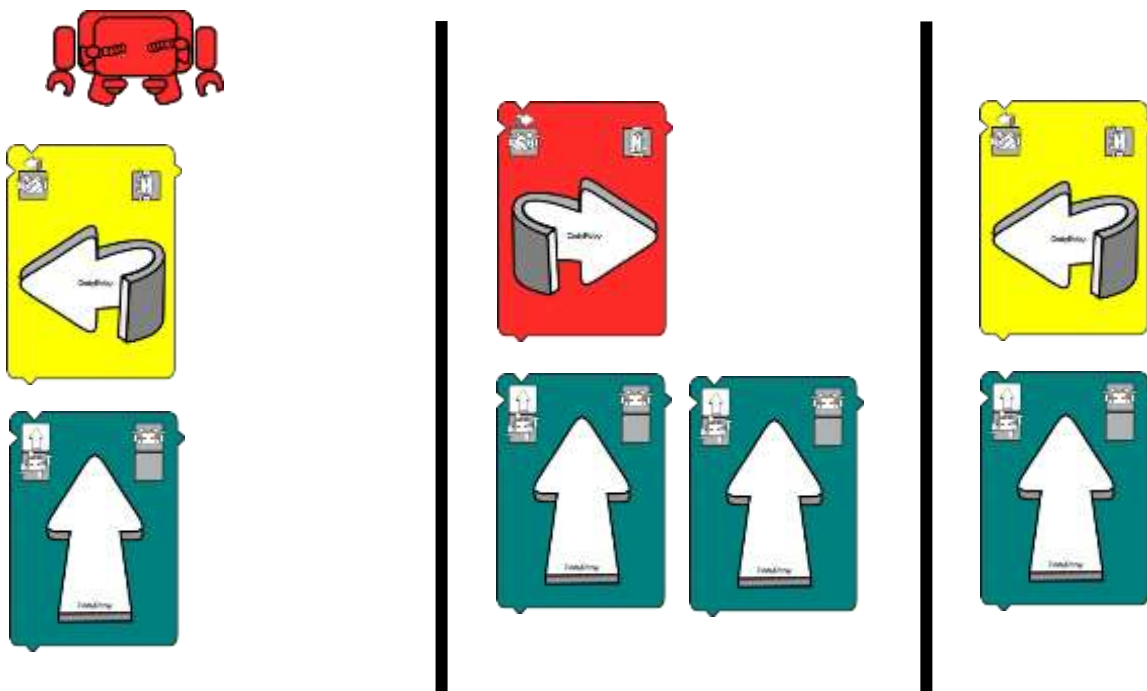
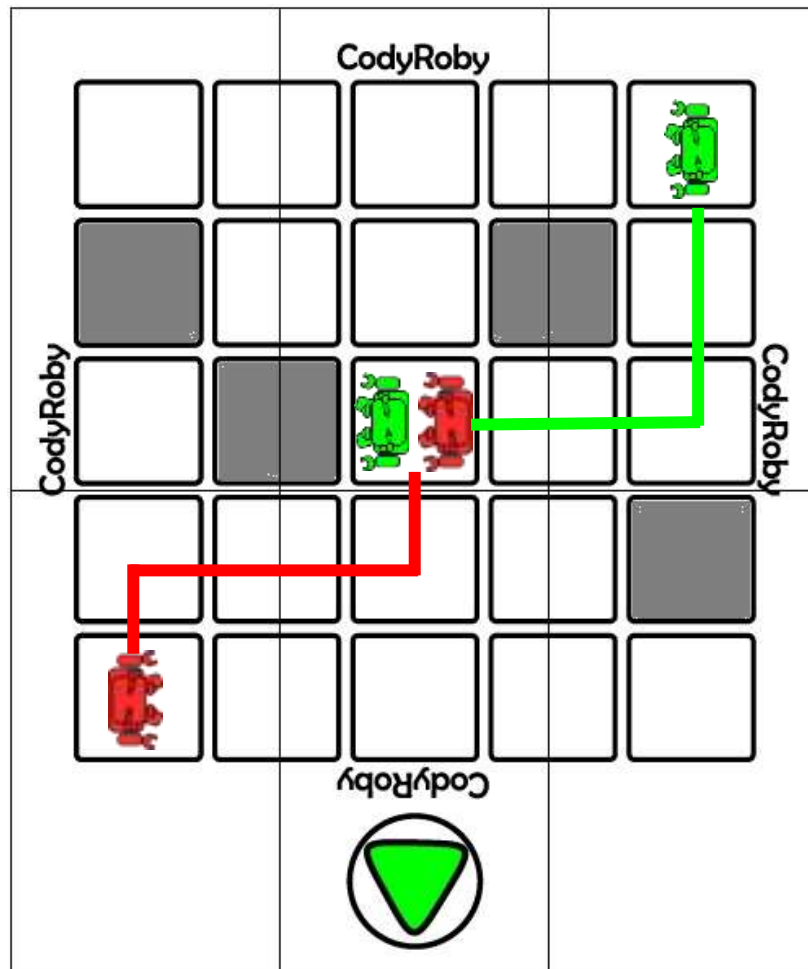


Figura 22 – La Serpiente (nivel difícil)

Juego “El Duelo”

Comienza la partida el jugador con el robot rojo que consigue atrapar al robot verde. Por lo tanto, usará una carta más que este.



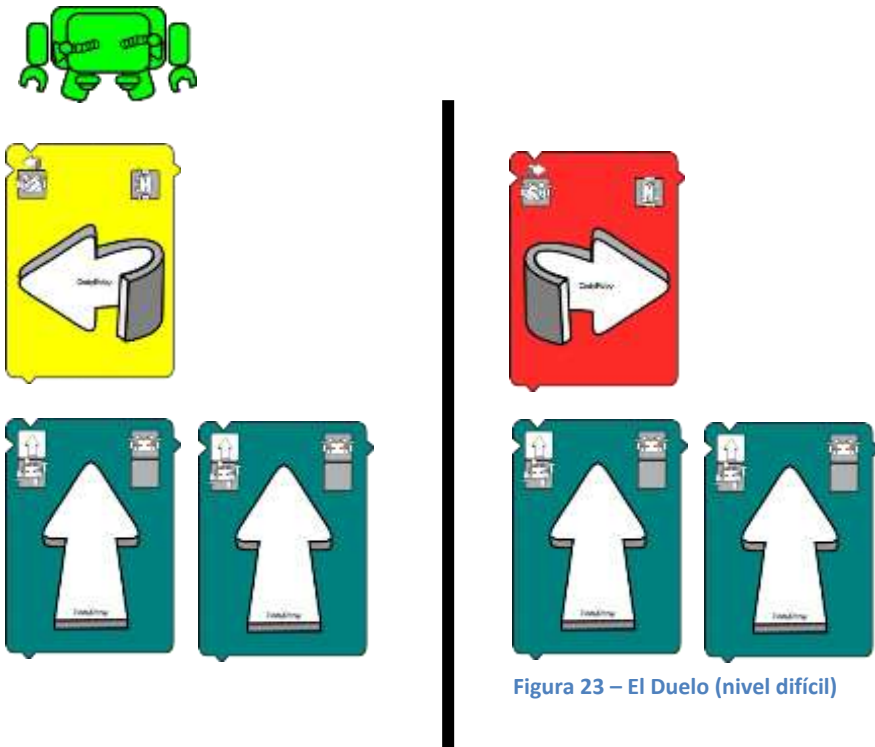
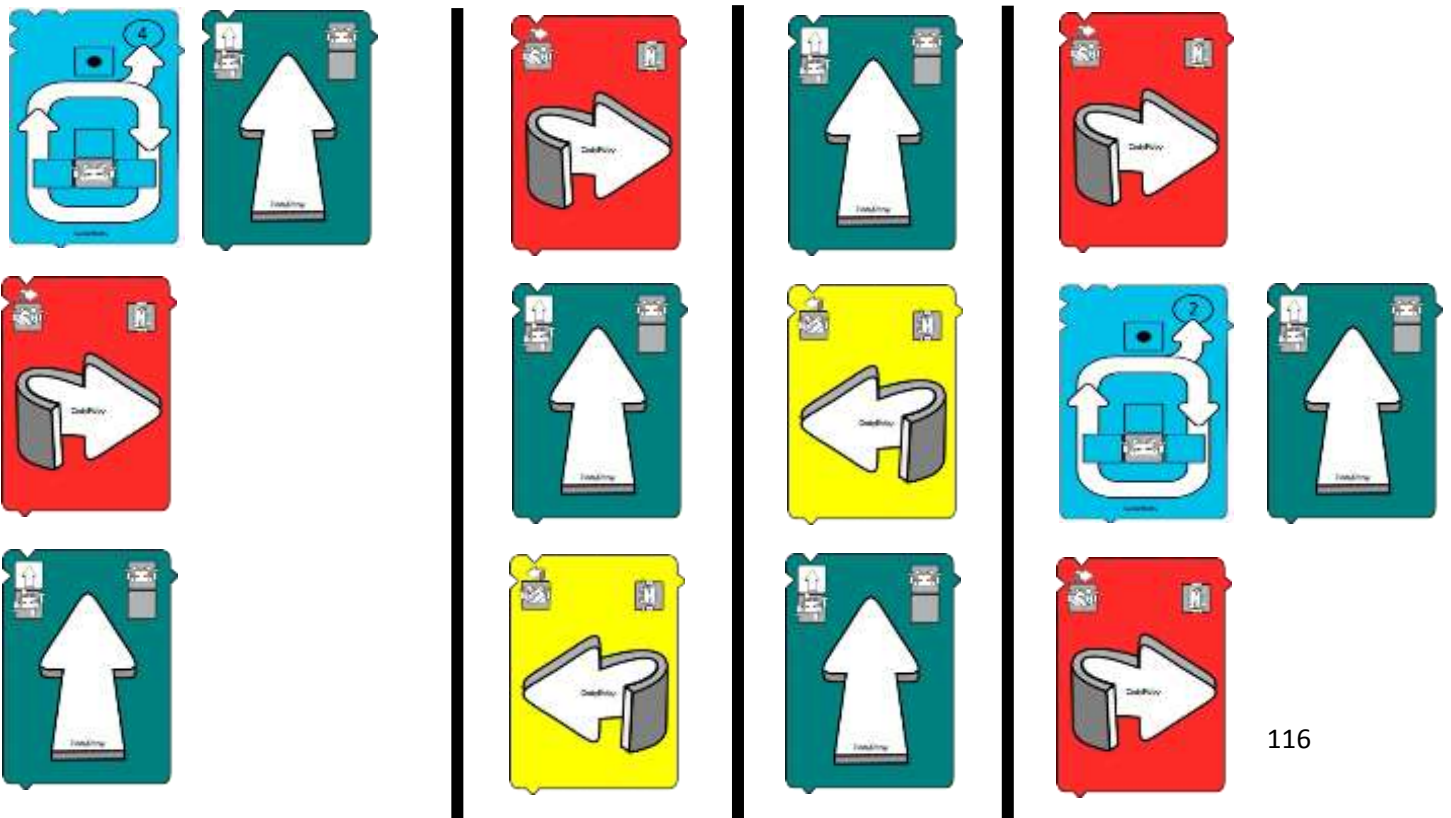
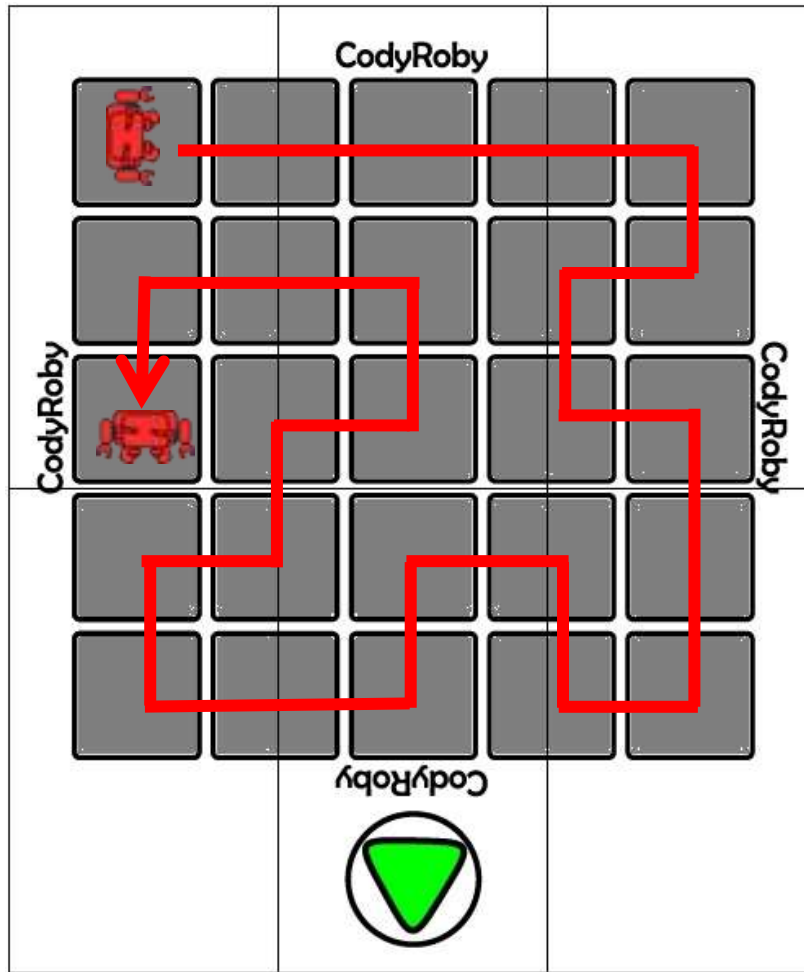


Figura 23 – El Duelo (nivel difícil)

Juego "Rellener el Tablero"



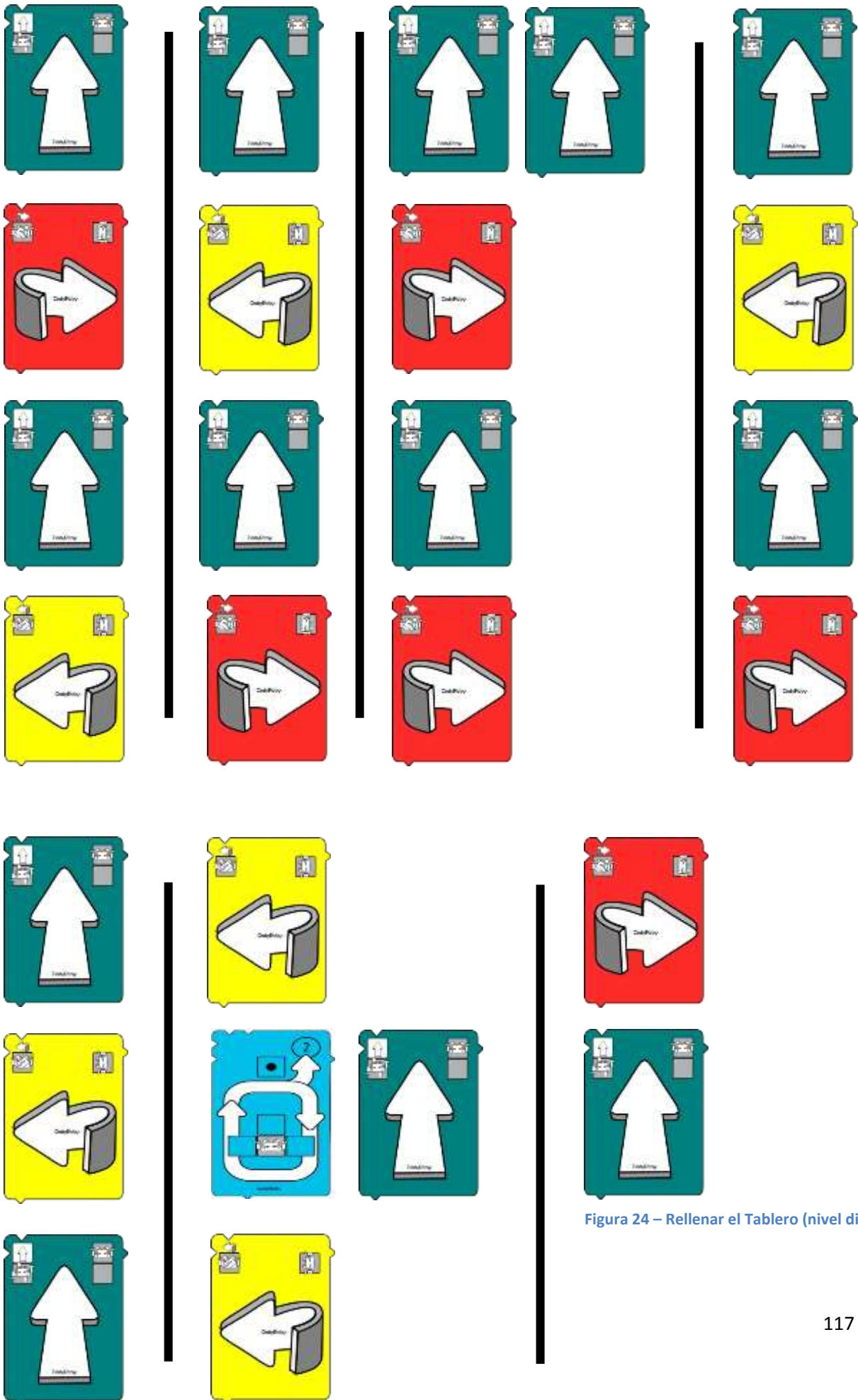


Figura 24 – Rellenar el Tablero (nivel difícil)

ANEXO VIII - BLOCKLY

ROMPECABEZAS - SOLUCIÓN

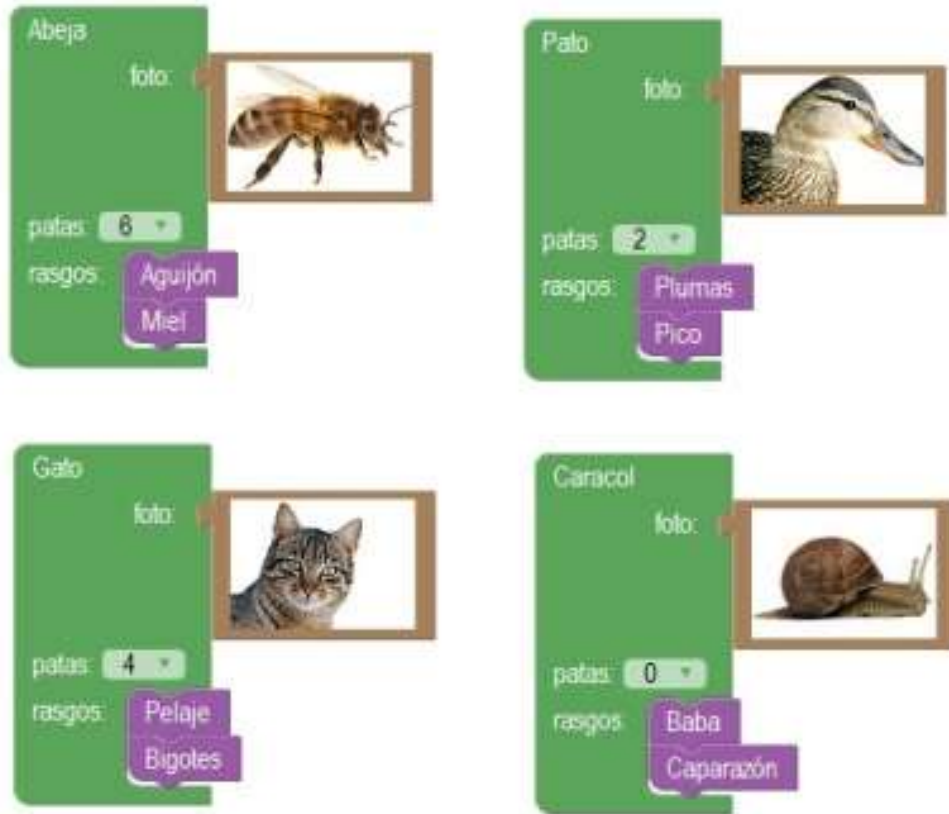


Figura 25 – Solución Rompecabezas

LABERINTO - SOLUCIONES

Juego 1

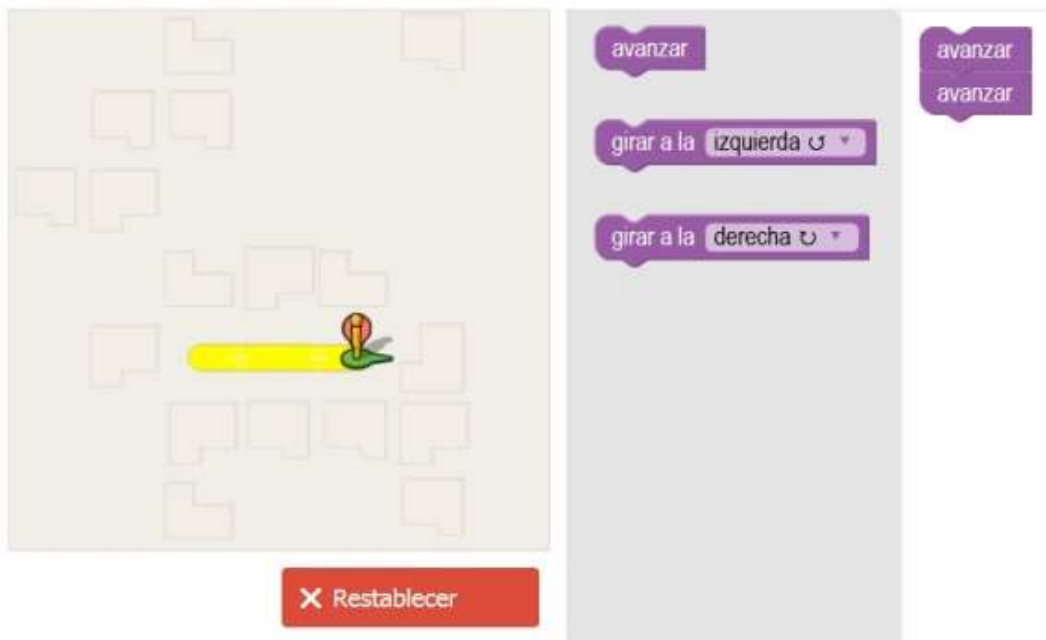


Figura 26 – Solución Juego 1 Laberinto

Juego 2

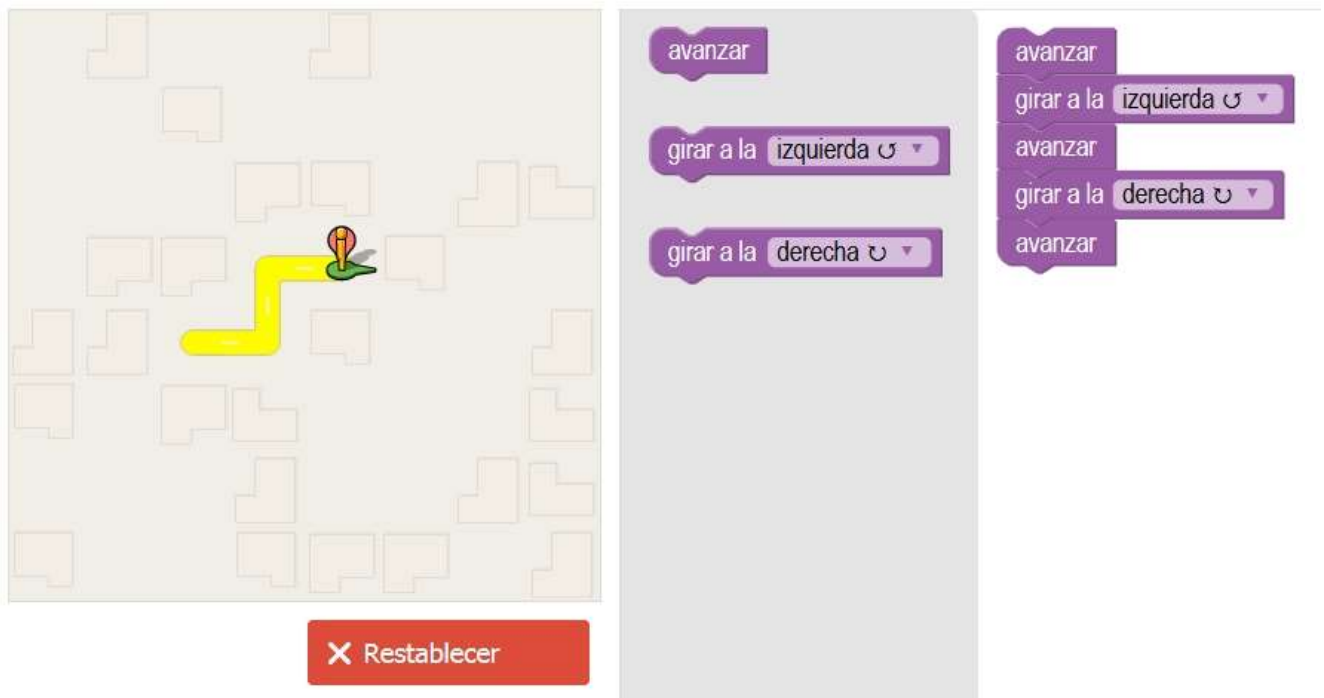


Figura 27 – Solución Juego 2 Laberinto

Juego 3



Figura 28 – Solución Juego 3 Laberinto

Juego 6



Figura 31 – Solución Juego 6 Laberinto

Juego 7



Figura 32 – Solución Juego 7 Laberinto

Juego 8



Te queda 1 bloque.


Restablecer

```

    avanzar
    girar a la izquierda
    girar a la derecha
    repetir hasta
    hacer
    si hay camino enfrente
    hacer
    repetir hasta
    hacer
    si hay camino enfrente
    hacer
    si hay camino enfrente
    hacer
    hacer
    si hay camino enfrente
    hacer
    si hay camino a la izquierda
    hacer
    girar a la izquierda
    avanzar
    si hay camino a la derecha
    hacer
    girar a la derecha
    avanzar
  
```

Figura 33 – Solución Juego 8 Laberinto

Juego 9



Te quedan 2 bloques.

Restablecer

```

    avanzar
    girar a la izquierda
    girar a la derecha
    repetir hasta
    hacer
    si hay camino enfrente
    hacer
    si hay camino enfrente
    hacer
    si hay camino enfrente
    hacer
    sino
    girar a la izquierda
    avanzar
  
```

Figura 34 – Solución Juego 9 Laberinto

PÁJARO - SOLUCIONES

Juego 1

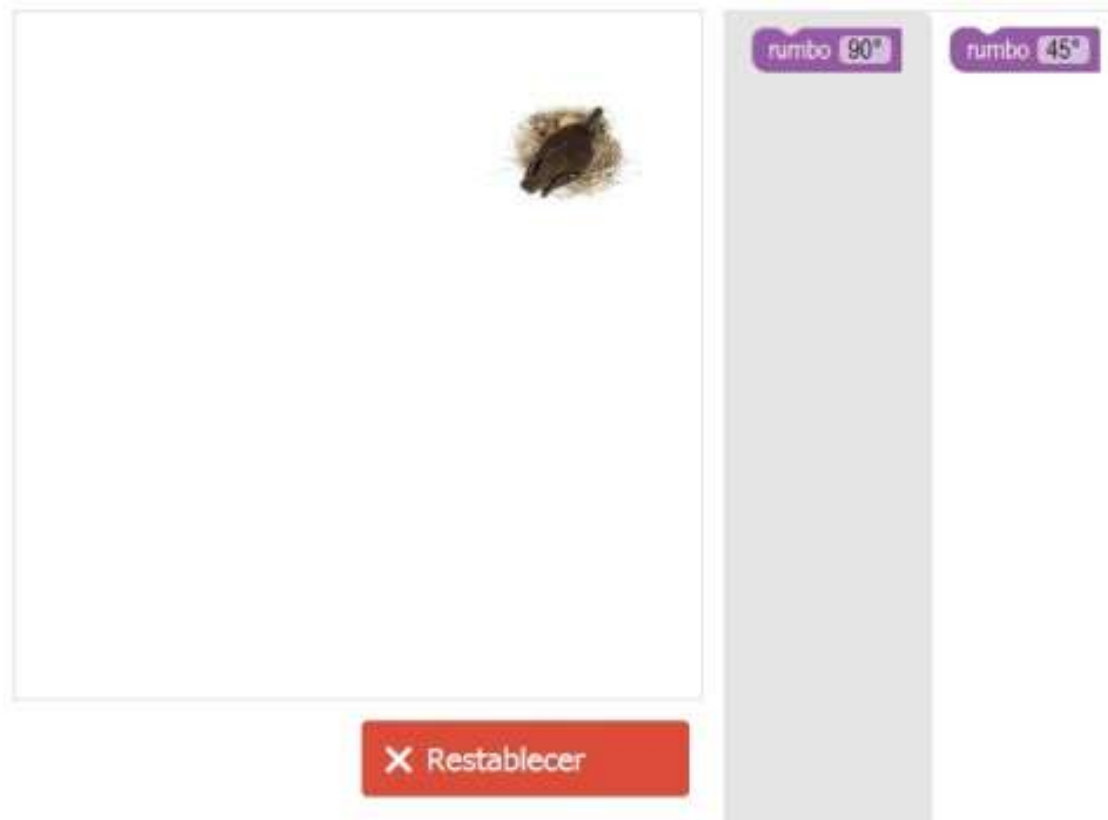


Figura 35 – Solución Juego 1 Pájaro

Juego 2

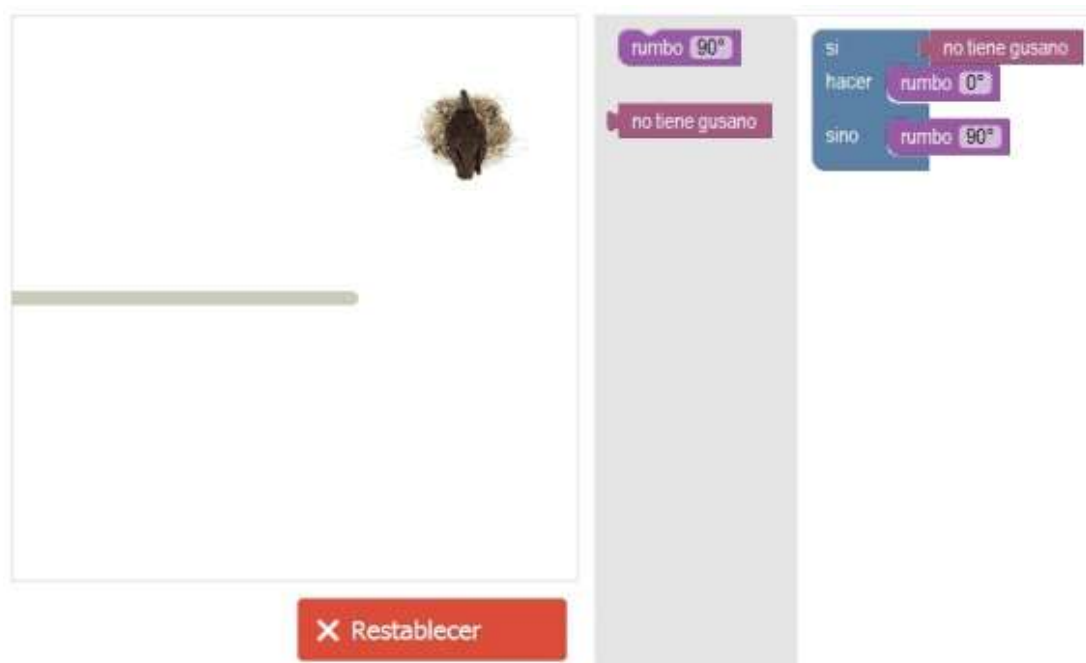


Figura 36 – Solución Juego 2 Pájaro

Juego 3

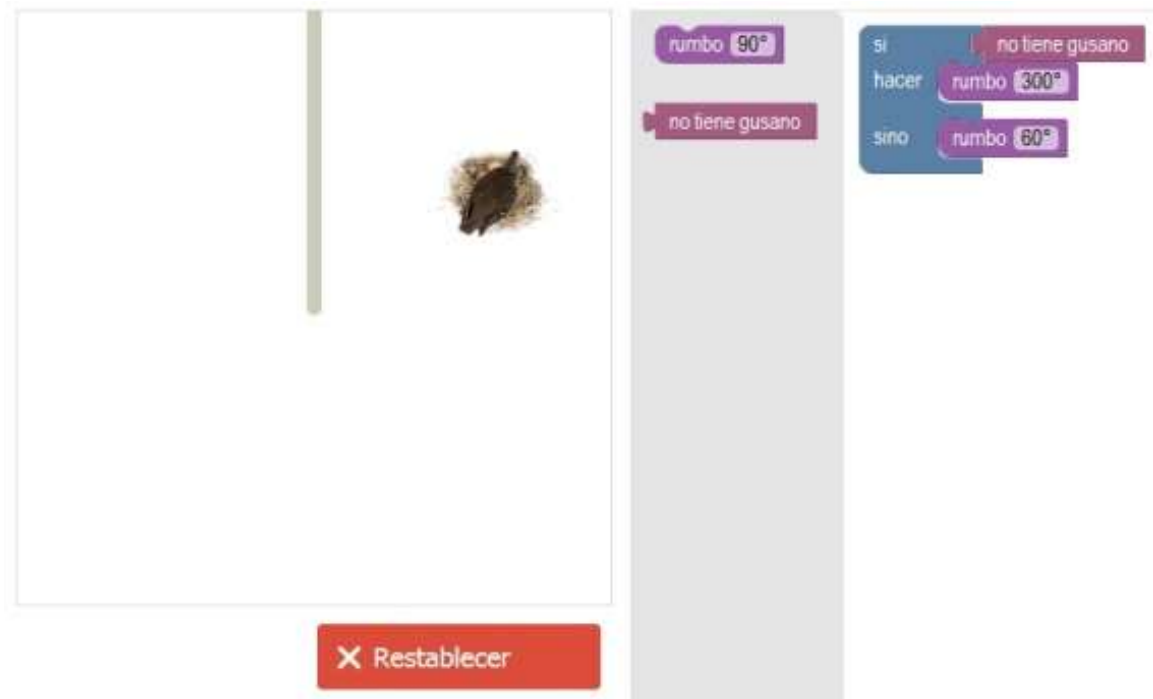


Figura 37 – Solución Juego 3 Pájaro

Juego 4

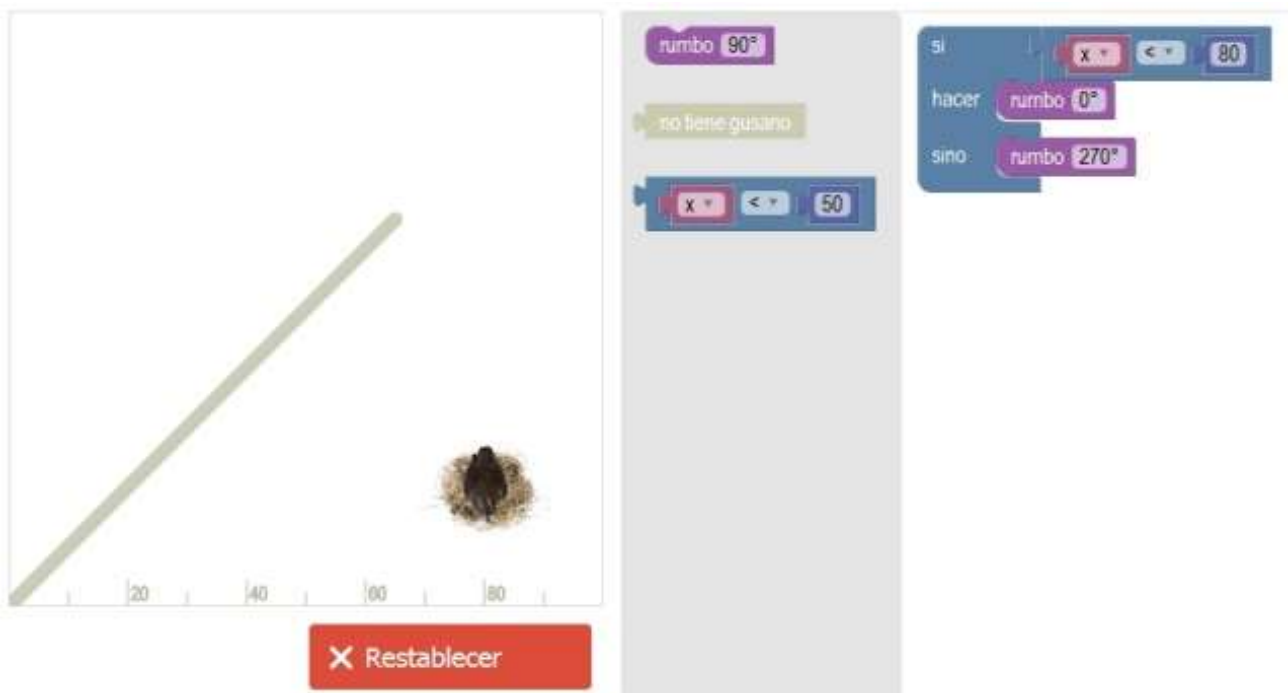


Figura 38 – Solución Juego 4 Pájaro

Juego 5

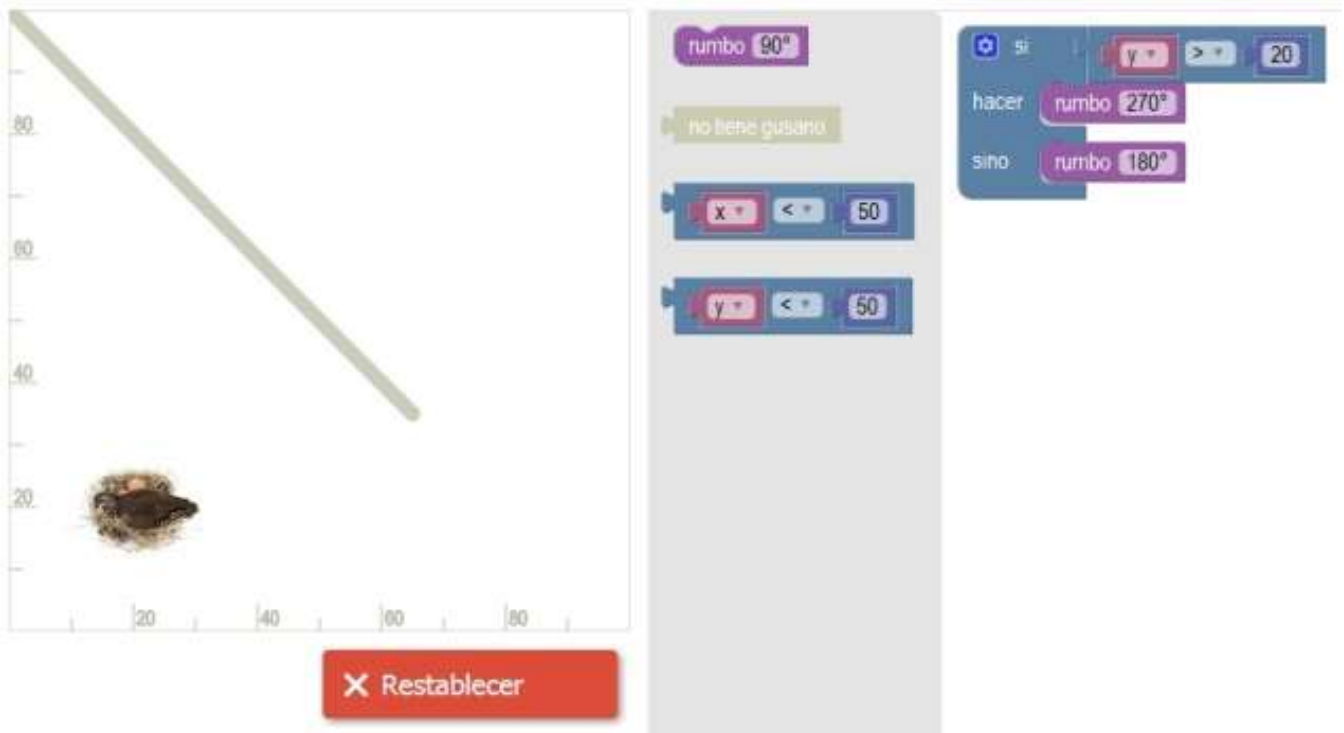
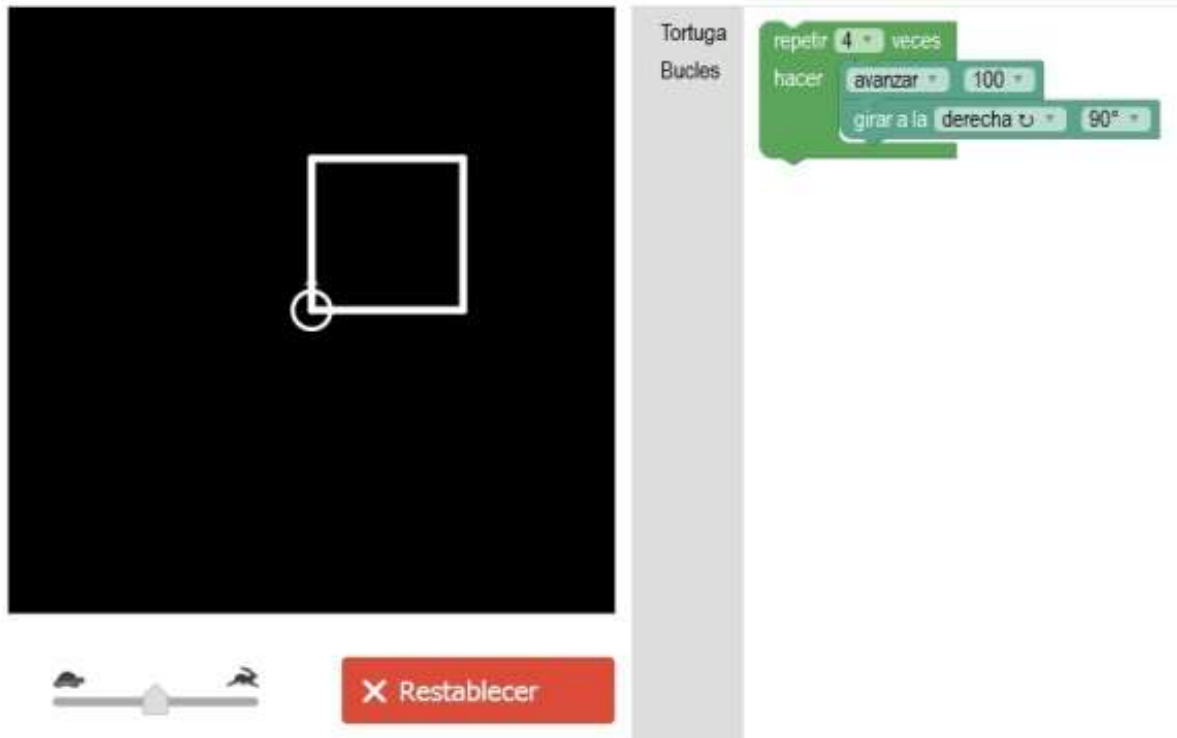


Figura 39 – Solución Juego 5 Pájaro

TORTUGA - SOLUCIONES

Juego 1



The screenshot shows the Scratch 'Tortuga' game interface. On the left, a black stage displays a white square with a small white circle at its bottom-left corner, representing the turtle's starting position. Below the stage are navigation arrows and a red 'Restablecer' button. On the right, the 'Tortuga' and 'Bucles' (Loops) categories are visible. The code consists of a 'repetir 4 veces' (repeat 4 times) block containing an 'hacer' (do) block with three steps: 'avanzar 100' (move 100), 'girar a la derecha 90' (turn right 90), and another 'girar a la derecha 90' (turn right 90).

Figura 40 – Solución Juego 1 Tortuga

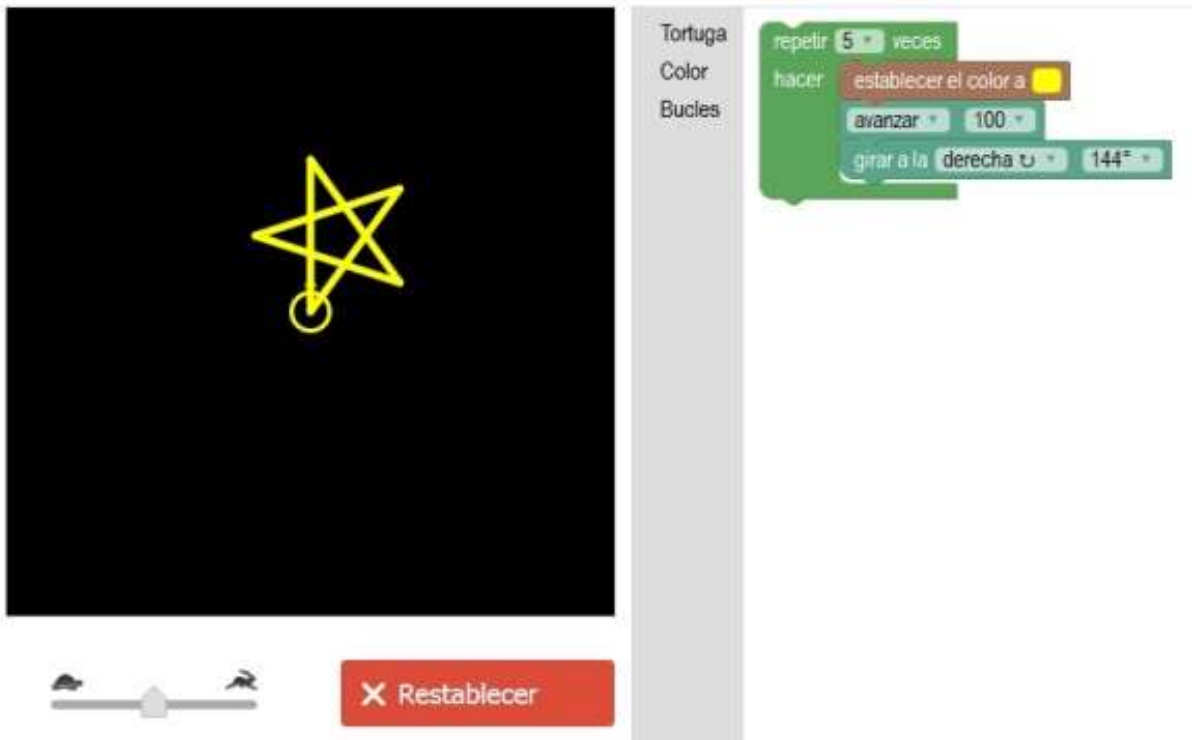
Juego 2



The screenshot shows the Scratch 'Tortuga' game interface. On the left, a black stage displays a white pentagon with a small white circle at its bottom-left corner, representing the turtle's starting position. Below the stage are navigation arrows and a red 'Restablecer' button. On the right, the 'Tortuga' and 'Bucles' (Loops) categories are visible. The code consists of a 'repetir 5 veces' (repeat 5 times) block containing an 'hacer' (do) block with three steps: 'avanzar 100' (move 100), 'girar a la derecha 72' (turn right 72), and another 'girar a la derecha 72' (turn right 72).

Figura 41 – Solución Juego 2 Tortuga

Juego 3



Tortuga
Color
Bucles

```
repetir 5 veces  
  hacer  
    establecer el color a [yellow]  
    avanzar 100  
    girar a la derecha 144
```

Restablecer

Figura 42 – Solución Juego 3 Tortuga

Juego 4



Tortuga
Color
Bucles

```
repetir 5 veces  
  hacer  
    establecer el color a [yellow]  
    avanzar 50  
    girar a la derecha 144  
  levantar el boligrato  
  avanzar 150  
  bajar el boligrato  
  establecer el color a [yellow]  
  avanzar 20
```

Restablecer

Figura 43 – Solución Juego 4 Tortuga

MÚSICA - SOLUCIONES

Juego 1

The screenshot shows a Scratch music player interface. On the left, a musical staff displays a sequence of four notes: C4, D4, E4, and C4. Below the staff is a progress bar and a volume slider. A red button labeled "Restablecer" is visible. On the right, the "Música" palette contains a script starting with "when clicked" followed by four "reproducir nota" blocks with notes C4, D4, E4, and C4.

Figura 44 – Solución Juego 1 Música

Juego 2

The screenshot shows a Scratch music player interface. On the left, a musical staff displays a sequence of four notes: C4, D4, E4, and C4. Below the staff is a progress bar and a volume slider. A red button labeled "Restablecer" is visible. On the right, the "Música" palette contains a script starting with "when clicked" followed by two "primera parte" blocks, and a "para" loop containing four "reproducir nota" blocks with notes C4, D4, E4, and C4.

Figura 45 – Solución Juego 2 Música

Juego 3

The screenshot shows a music game interface. On the left, a musical staff displays two notes: C4 and D4. Below the staff is a volume slider and a red button labeled 'Restablecer'. On the right, the 'Música Funciones' panel contains the following code blocks:

- when clicked** event block.
- para primera parte** loop containing:
 - reproducir nota C4 por
 - reproducir nota D4 por
 - reproducir nota E4 por
 - reproducir nota C4 por
- para segundaparte** loop containing:
 - reproducir nota E4 por
 - reproducir nota F4 por
 - reproducir nota G4 por

Figura 46 – Solución Juego 3 Música

Juego 4

The screenshot shows a music game interface. On the left, a musical staff displays four notes: E4, F4, G4, and C4. Below the staff is a volume slider and a red button labeled 'Restablecer'. On the right, the 'Música Funciones' panel contains the following code blocks:

- when clicked** event block.
- para primera parte** loop containing:
 - reproducir nota C4 por
 - reproducir nota D4 por
 - reproducir nota E4 por
 - reproducir nota C4 por
- para segundaparte** loop containing:
 - reproducir nota E4 por
 - reproducir nota F4 por
 - reproducir nota G4 por
- para tercera parte** loop containing:
 - reproducir nota G4 por
 - reproducir nota A4 por
 - reproducir nota G4 por
 - reproducir nota F4 por
 - reproducir nota E4 por
 - reproducir nota C4 por

Figura 47 – Solución Juego 4 Música

Juego 5

Música Funciones

when clicked

primera parte

primera parte

segunda parte

segunda parte

tercera parte

tercera parte

cuarta parte

cuarta parte

para primera parte

reproducir nota C4 por

reproducir nota D4 por

reproducir nota E4 por

reproducir nota C4 por

para segunda parte

reproducir nota E4 por

reproducir nota F4 por

reproducir nota G4 por

para tercera parte

reproducir nota G4 por

reproducir nota A4 por

reproducir nota G4 por

reproducir nota F4 por

reproducir nota E4 por

reproducir nota C4 por

para cuarta parte

reproducir nota C4 por

reproducir nota G3 por

reproducir nota C4 por

Restablecer

Figura 48 – Solución Juego 5 Música

Juego 6

Música Funciones

when clicked

definir el instrumento como violín

primera parte

primera parte

segunda parte

segunda parte

tercera parte

tercera parte

cuarta parte

cuarta parte

para primera parte

reproducir nota C4 por

reproducir nota D4 por

reproducir nota E4 por

reproducir nota C4 por

para segunda parte

reproducir nota E4 por

reproducir nota F4 por

reproducir nota G4 por

para tercera parte

reproducir nota G4 por

reproducir nota A4 por

reproducir nota G4 por

reproducir nota F4 por

reproducir nota E4 por

reproducir nota C4 por

para cuarta parte

reproducir nota C4 por

reproducir nota G3 por

reproducir nota C4 por

Restablecer

Figura 49 – Solución Juego 6 Música

ANEXO IX – CODE

VISOR DE AVANCES

Ejercicio resuelto correctamente

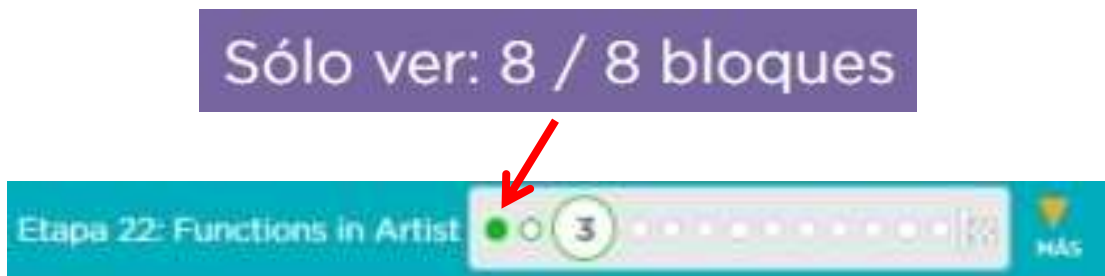


Figura 50 – Avance (ejercicio resuelto correctamente)

Ejercicio resuelto con un número mayor de bloques del necesario

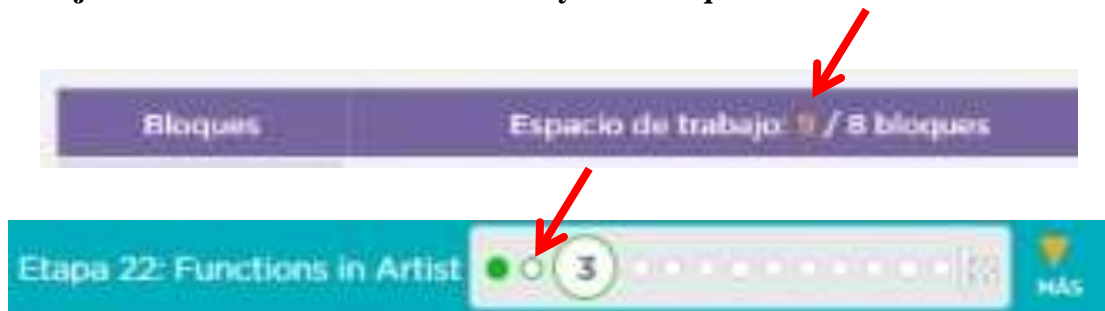


Figura 51 – Avance (ejercicio resuelto incorrectamente)

CUENTA DE DOCENTE

Avances de un alumno

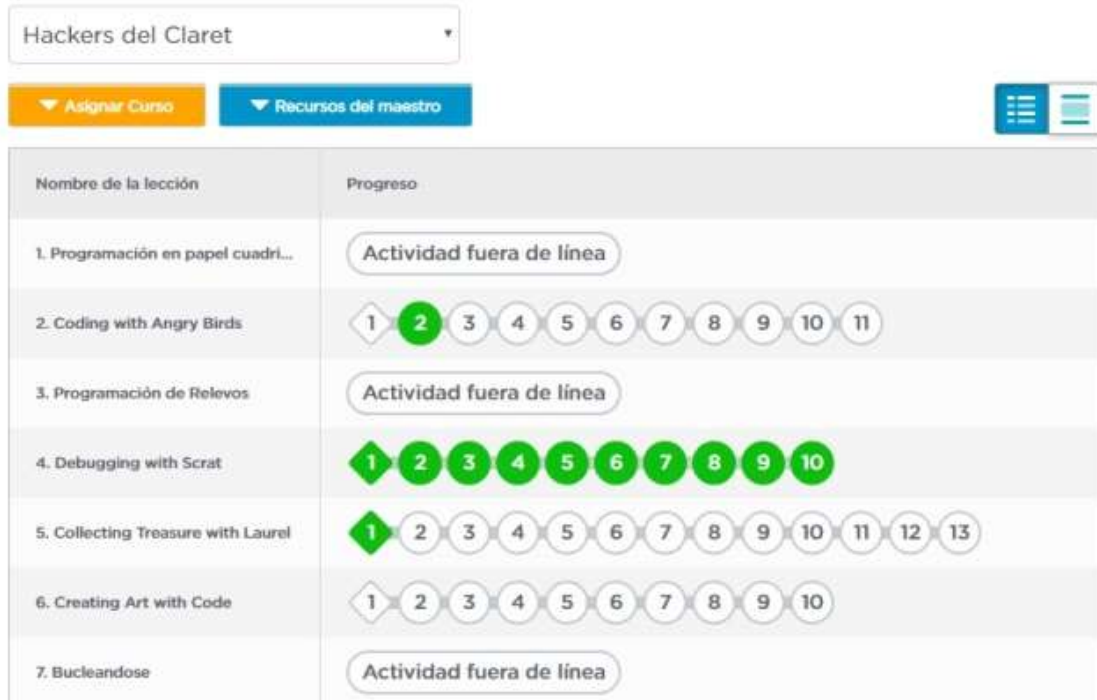


Figura 52 – Avances totales de un alumno en el curso

Niveles completados y líneas de código de un alumno



Figura 53 – Niveles completados y líneas de código programadas por un alumno en el curso

Gráfico simple de la actividad de un alumno



Figura 54 – Gráfico simple de la actividad de un alumno en el curso

Gráfico avanzado de la actividad de un alumno

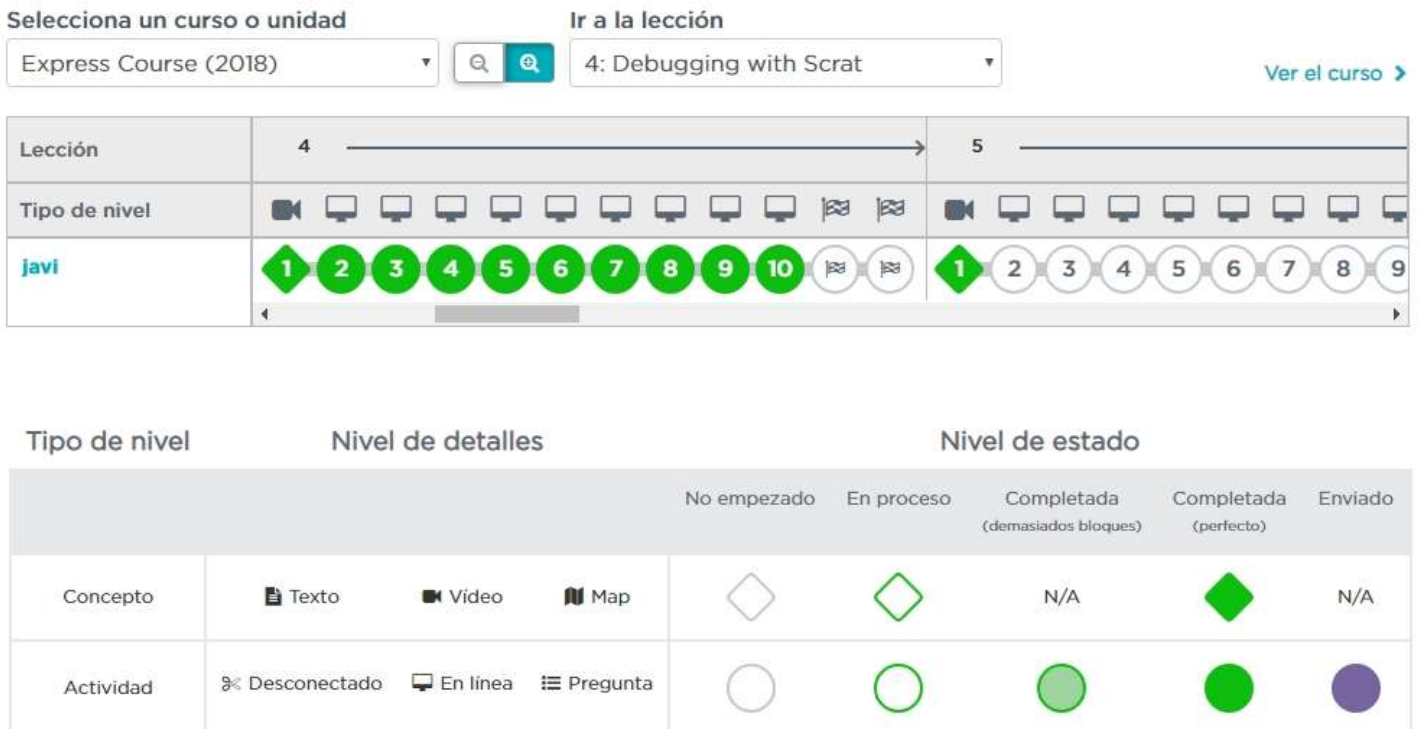


Figura 55 – Gráfico avanzado de la actividad de un alumno en el curso

ANEXO X – MICRO:BIT

PROYECTO 1 – CREAMOS UN SALUDO

Saludo basado en texto



Figura 56 – Código necesario para crear un saludo basado en texto

Saludo basado en un icono predeterminado del editor

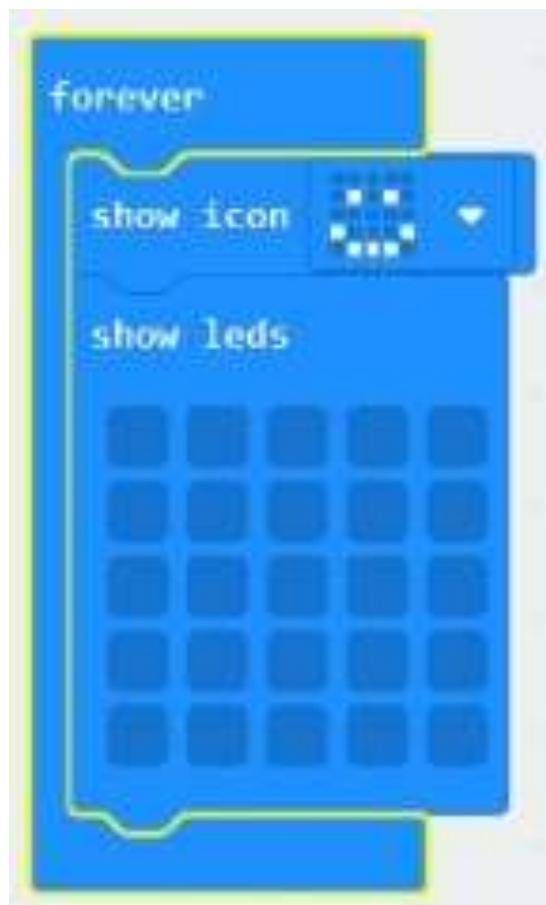


Figura 57 – Código necesario para crear un saludo basado en un icono predeterminado del editor

Saludo basado en icono creado por nosotros

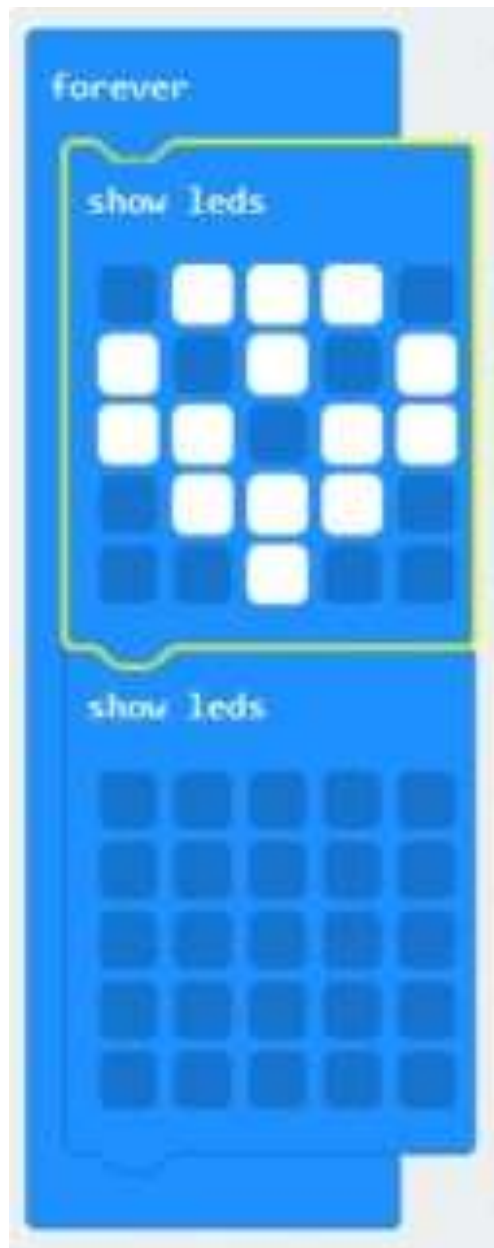


Figura 58 – Código necesario para crear un saludo basado en un icono creado por nosotros

PROYECTO 2 – CREAMOS UN TERMÓMETRO

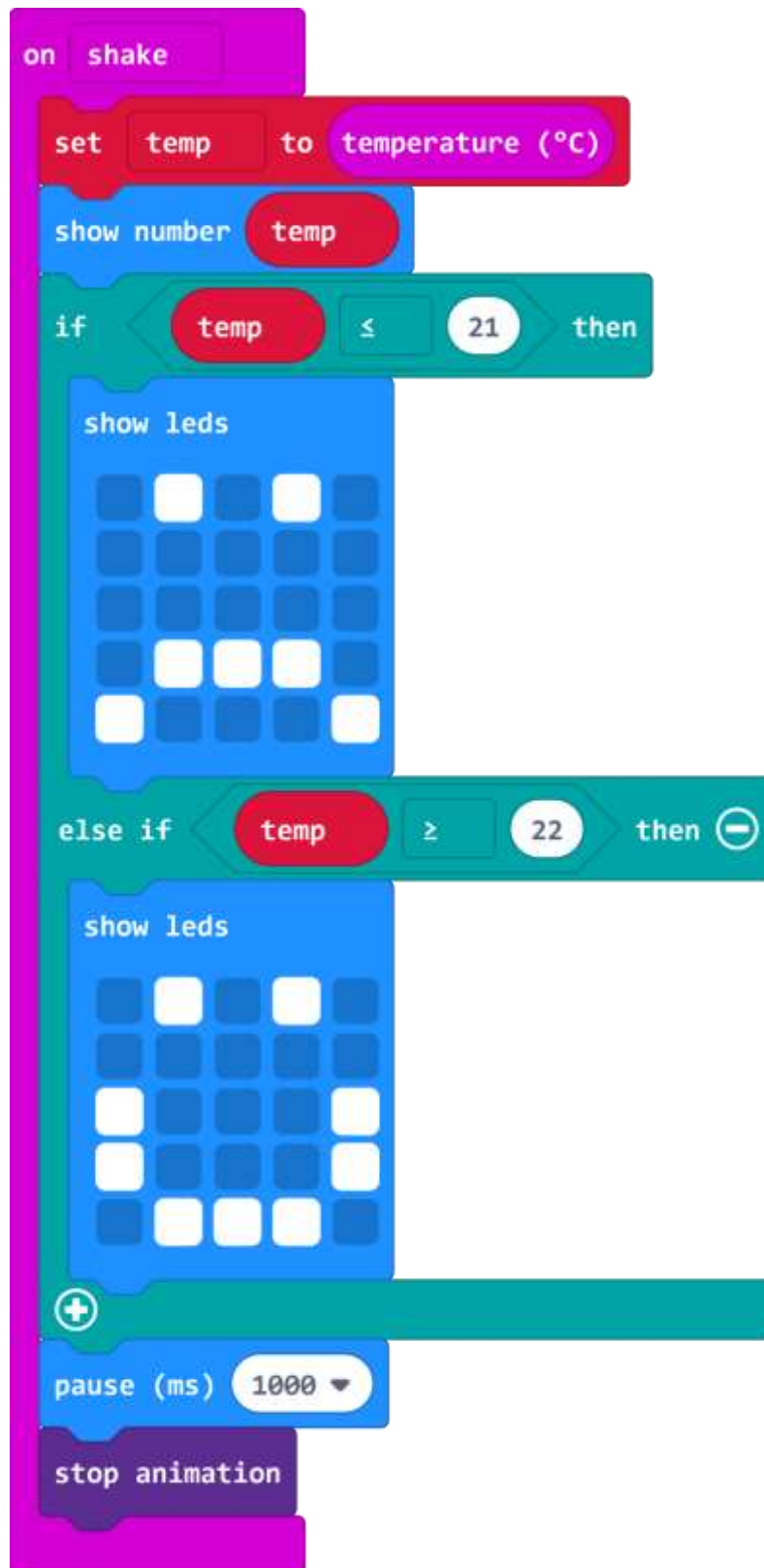


Figura 59 – Código necesario para crear un termómetro

PROYECTO 3 – CREAMOS UN PODÓMETRO

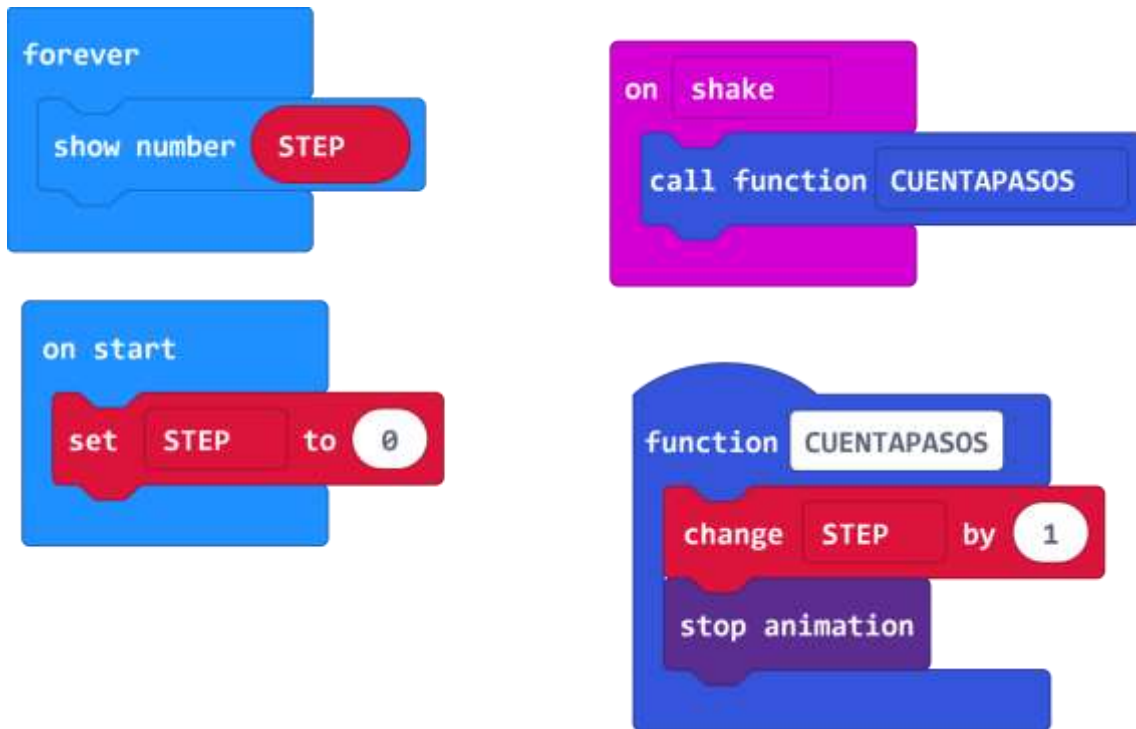


Figura 60 – Código necesario para crear un podómetro

PROYECTO 4 – CREAMOS UNA BOLA MÁGICA DE RESPUESTAS

The image displays a Scratch script for a magic ball of answers. The script is organized into several sections:

- on start:** A blue block containing a 'show string' block with the text 'PREGUNTA Y AGITANE'.
- on shake:** A purple block containing a 'call function' block named 'RESPUESTAS'.
- on button A pressed:** A purple block containing a 'show string' block with the text 'PREGUNTA Y AGITANE'.
- on button B pressed:** A purple block containing a 'show string' block with the text 'PREGUNTA Y AGITANE'.
- function RESPUESTAS:** A large blue block containing a 'set' block for 'RESPUESTA' to 'pick random' from '1' to '15'. This is followed by a series of 15 'if-then' blocks, each corresponding to a number from 1 to 15. Each 'if' block contains a 'show string' block with a specific response: 1: SEGURO, 2: SIN DUDA ALGUNA, 3: ESTOY SEGURO DE ELLO, 4: DEFINITIVAMENTE, 5: POR SUPUESTO, 6: QUIZAS, 7: A LO MEJOR, 8: PUEDE SER, 9: NO ESTOY SEGURO DE ELLO, 10: TENGO MIS DUDAS, 11: NO, 12: IMPOSIBLE, 13: ES MUY DIFÍCIL, 14: ES MUY DUDOSO, 15: NO CUENTES CON ELLO.

Figura 61 – Código necesario para crear una bola mágica de respuestas

Respuestas incluidas en el programa: verde (afirmativo), naranja (dudoso) y rojo (negativo).

SEGURO	QUIZAS	NO
SIN DUDA ALGUNA	A LO MEJOR	IMPOSIBLE
ESTOY SEGURO DE ELLO	PUEDE SER	ES MUY DIFÍCIL
DEFINITIVAMENTE	NO ESTOY SEGURO DE ELLO	ES MUY DUDOSO
POR SUPUESTO	TENGO MIS DUDAS	NO CUENTES CON ELLO

Tabla 13 – Respuestas aleatorias incluidas en la bola mágica de respuestas

PROYECTO 5 - ¿Cómo andas de pulso?

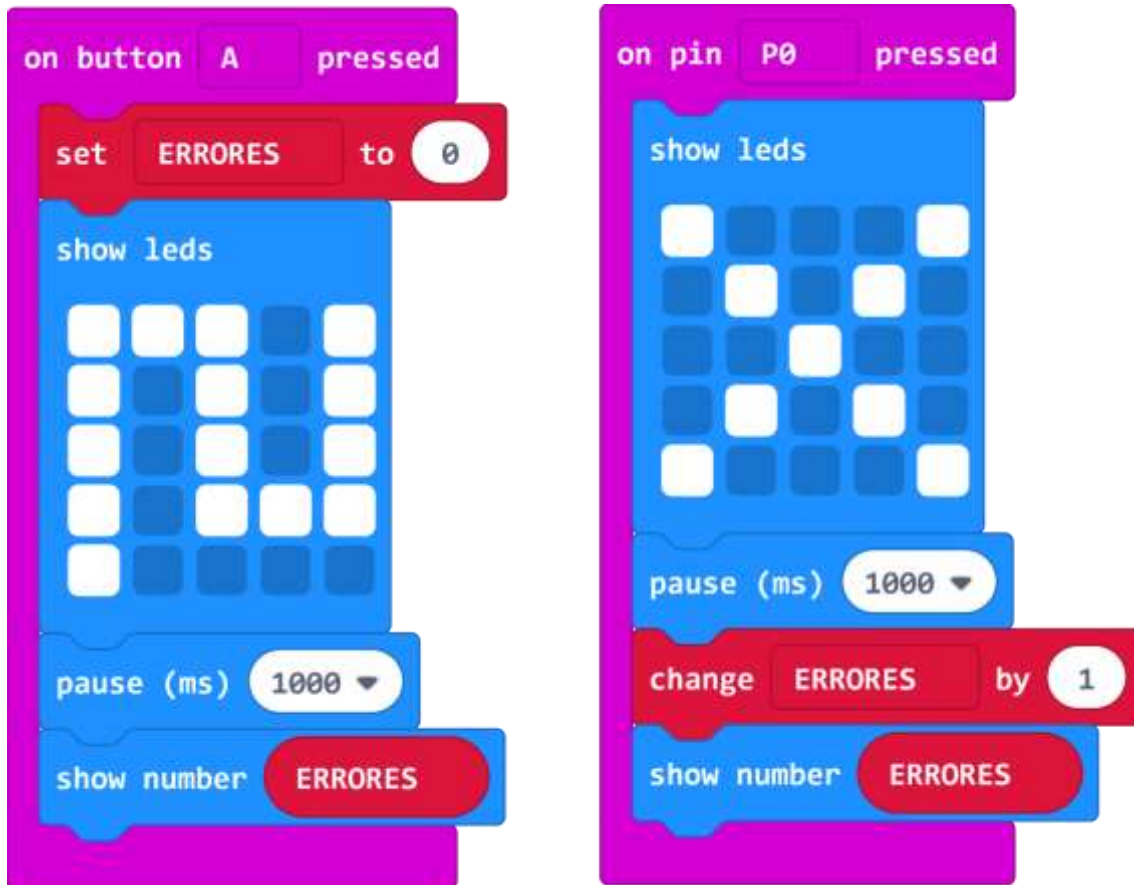


Figura 62 – Código necesario para crear un dispositivo que detecte cuando tocamos un alambre ("error") y cuente estos "errores"

PROYECTO 6: Creamos una brújula.

```
forever
  set ANGLE to compass heading (*)
  if ANGLE > 315 and ANGLE ≤ 360 or ANGLE ≥ 0 and ANGLE ≤ 45 then
    show string N
  else if ANGLE > 45 and ANGLE ≤ 135 then
    show string E
  else if ANGLE > 135 and ANGLE ≤ 225 then
    show string S
  else if ANGLE > 225 and ANGLE ≤ 315 then
    show string W
```

Figura 63 – Código necesario para crear una brújula