



---

# Universidad de Valladolid

Facultad de Educación y Trabajo Social

TRABAJO DE FIN DE GRADO:

## **EL USO DE LA PROGRAMACIÓN INFORMÁTICA EN EL AULA DE LA LENGUA EXTRANJERA INGLÉS.**

Curso académico 2021/2022

Presentado por Álvaro Alejos Izquierdo para optar al Grado de

Educación Primaria

por la Universidad de Valladolid

Tutelado por Natalia Barranco Izquierdo

# **RESUMEN**

Este Trabajo de Fin de Grado recoge una revisión de la literatura actual sobre el uso de la programación informática en la Etapa de Educación Primaria, haciendo hincapié en su posible introducción en el aula de la lengua extranjera inglés. Para ello, se exponen los múltiples beneficios a nivel cognitivo del uso de la programación en el contexto educativo, entre las que destaca el pensamiento computacional, concepto en boga entre los académicos de la educación.

Por otra parte, se presenta una propuesta de intervención didáctica en forma de proyecto anual en la que se trabaja la programación en la lengua extranjera haciendo uso de la plataforma Code.org, web estadounidense diseñada en la lengua extranjera para el aprendizaje de dichas aptitudes informáticas en edades tempranas.

Palabras clave: adquisición de las lenguas extranjeras, programación informática, pensamiento computacional y efecto de transferencia.

# **ABSTRACT**

This End of Degree Project exposes a revision of the actual literature about computer programming in Primary Education, considering its possible introduction in the EFL classroom. Throughout the document, the cognitive benefits of programming as an educational tool are exposed, highlighting the development of computational thinking, a hotly debated educational issue at the moment.

In addition, a project design to bring computer programming to the EFL classroom is introduced, using the platform Code.org as the environment where the acquisition of the foreign language takes place. This American webpage has been designed with the aim of learning programming skills since an early age.

Key words: foreign language acquisition, computer programming, computational thinking and transfer effect.

# ÍNDICE

Introducción.....	4
Objetivos.....	5
Justificación e interés personal .....	6
fundamentación teórica.....	8
1. La programación informática .....	8
1.1 Definición y contextualización en el ámbito escolar.....	8
2. ¿Qué es el efecto de transferencia?.....	10
3. Resultados de metaanálisis realizados sobre la transferibilidad de la programación .....	12
4. La adquisición de lenguas extranjeras a través de la programación informática	15
Intervención didáctica.....	21
1. Contexto .....	21
2. Momento de implantación en el curso.....	22
3. Alumnado .....	22
4. Objetivos.....	22
5. Temporalización .....	22
6. Metodología.....	23
7. Contenidos.....	23
8. Criterios de evaluación .....	24
9. Competencias básicas .....	24
10. Desarrollo del proyecto.....	24
Conclusión.....	50
Anexos .....	51
Bibliografía.....	52

# INTRODUCCIÓN

El presente trabajo de investigación tiene como objeto principal establecer la relación entre la adquisición de la Lengua Extranjera Inglés y el ámbito de la programación informática. Aunque a primera vista estas áreas pueden parecer distantes, una vez se indaga en ellas podemos encontrar habilidades subyacentes comunes, siendo la principal la capacidad de resolución de problemas.

A partir de este momento se empieza a divisar el vínculo entre el pensamiento computacional, inherente a la programación, y el paradigma constructivista del aprendizaje de lenguas extranjeras. Se promueve la adquisición de la lengua extranjera por medio de un aprendizaje integrado de la lengua y contenidos, a través de la propuesta de problemas en las que el alumnado construye su conocimiento a partir del previo. Dado que el idioma en el que se programa el código fuente de programas informáticos y apps móviles es el inglés, el aprendizaje de las habilidades de programación y la lengua extranjera se puede producir de manera simultánea.

*Computer programming* es nuevo enfoque metodológico surgido en Estados Unidos, que introduce la programación como parte de la educación integral de los estudiantes con el fin de trabajar habilidades como la resolución de problemas y desarrollar el espíritu crítico. Tras haberse popularizado en su país de origen, algunos docentes en Europa ya están empezando a introducir la programación en sus aulas, tanto en la Educación Primaria como en la Educación Secundaria.

Diversas investigaciones realizadas por académicos de la educación como Scherer, Barr o Stephenson, han tratado de determinar cuál es el grado de influencia de este nuevo enfoque metodológico en el desarrollo de otras habilidades cognitivas relacionadas con la programación, tratando de medir el efecto de transferencia de la capacidad de manipular código a otras áreas del ámbito escolar y habilidades cognitivas como el pensamiento computacional.

# OBJETIVOS

El objetivo general de este trabajo es analizar la viabilidad de la introducción de la programación informática en el aula de la lengua extranjera para el fomento de su adquisición.

Como objetivos específicos, se han determinado los siguientes:

- Presentar investigaciones realizadas sobre la programación informática en ámbitos educativos.
- Indagar sobre el efecto de la programación informática en el desarrollo cognitivo del alumnado.
- Dar cuenta de la relación entre la programación informática y la adquisición de lenguas extranjeras.
- Diseñar una intervención didáctica que integre la adquisición de la lengua extranjera y la programación informática.

# JUSTIFICACIÓN E INTERÉS PERSONAL

La elección del tema del *computer programming* para este trabajo de fin de grado viene dada por la participación del autor en la conferencia “ProSocial Values Community International Conference. The Good Nutrients for Schools” que tuvo lugar en septiembre del 2020, donde tuvo contacto con una gran variedad de proyectos europeos de innovación educativa. Entre ellos, varios países mostraron sus proyectos de robótica y programación para el desarrollo de las habilidades sociales.

A partir de ese momento y al comentar sobre el tema con compañeros de la facultad, uno de ellos mencionó un proyecto que su padre, profesor de Educación Secundaria, estaba realizando con sus alumnos en torno a la creación de código. Tras varias conversaciones con dicho docente, me hizo entrega de múltiples recursos, entre los que se encontraba el sitio web de “CODE”, una organización educativa estadounidense que ha creado una cantidad ingente de recursos para la enseñanza de la programación e imparte cursos de formación para docentes en su país de origen.

Este recurso en línea proporciona un amplio repertorio de documentos académicos que fundamentan la acción de la organización y muestran los resultados de la introducción de sus cursos en centros educativos de distintas etapas. En general, CODE reporta un incremento en la continuación de los estudios del alumnado, aumentando los niveles de innovación en los centros, hecho que Partovi (2018) determina como relevante para los resultados académicos del alumnado.

Todos estos hechos condujeron al autor del trabajo a reflexionar sobre la posible aplicación de este interesante método en el aula de la lengua extranjera inglés. Considerando la teoría del constructivismo, la creación de código podría llegar a hacerse un lugar en el aula, siempre que su enseñanza se diese en la lengua extranjera y los materiales didácticos utilizados también estuviesen creados en inglés, teniendo lugar en este caso un aprendizaje combinado de la segunda lengua y las habilidades de codificación, en situaciones en el que el uso de la lengua inglesa y el lenguaje de programación sitúen al estudiante en la ZDP.

Para lograr esta meta, se podría optar por posibilidades como diseñar una unidad didáctica por proyectos en la que los alumnos, adecuando el producto final en función del nivel a la que la unidad se dirigiese, como podría ser la creación de una aplicación o

sitio web en cursos superiores o el diseño de una secuencia de comandos simples para los niveles inferiores. Sin embargo, dado que la información y resultados de los estudios revisados durante la realización del trabajo indicaban unos mayores beneficios de programar cuando esta tenía lugar periódicamente en periodos de tiempo más extendidos, sería más adecuado diseñar un proyecto transversal de centro que pueda tener lugar tanto en el área de, dedicando una sesión a la semana para el trabajo de la programación en el aula.

# FUNDAMENTACIÓN TEÓRICA

## 1. LA PROGRAMACIÓN INFORMÁTICA

### 1.1 DEFINICIÓN Y CONTEXTUALIZACIÓN EN EL ÁMBITO ESCOLAR

La programación puede ser considerada de múltiples maneras; un método para enseñar y aprender el pensamiento computacional (Flórez et al., 2017), una forma de asistir el pensamiento computacional (Grover, Pea, 2013) o una manera de exponer al alumnado al pensamiento computacional a través de la creación de artefactos computacionales, como pueden ser el código fuente o los programas informáticos (Lye, Koh, 2014).

Dentro del campo de la programación informática, Scherer et al. (2018) definen las habilidades de programación como “the skills to create, modify, and evaluate code and the conceptual and procedural knowledge needed to apply these skills” (p. 766)

“The processes involved in programming require problem-solving skills, such as decomposing problems, applying algorithms, abstracting, and automatizing, and ultimately aid the acquisition of computational thinking skills” (Scherer et al. 2018, p. 765)

Estas aseveraciones desvelan la relación entre la programación y la resolución de problemas, siendo la redacción y buen funcionamiento del código un problema en sí. Por este simple hecho, la programación ya podría ser considerada de interés educativo, dado que promueve la puesta en práctica y desarrollo de las habilidades de resolución de problemas de manera alternativa a la planteada habitualmente en el área de matemáticas, a parte del hecho de que el conocimiento sobre esta tiene una gran proyección laboral.

Si continuamos analizando el impacto de un buen desarrollo de la capacidad de resolución de problemas, encontramos que estos problemas no se restringen a los contenidos estrictamente científicos del currículo de Educación Primaria, si no que en el aula de la lengua extranjera también se propician situaciones en las que, para resolver un problema, el uso del lenguaje es la vía para llegar a la solución. Teorías como el constructivismo apoyan este tipo de ejercicios y los consideran ideales para la adquisición de la segunda lengua de manera que los estudiantes utilizan la utilizan con una finalidad real, la de comunicarse efectivamente, construyendo su conocimiento a



partir del previo, vinculando las situación problemáticas presentadas con otras anteriores para su resolución.

Múltiples autores como Román-González et al. (2017) o Wing (2006) muestran ser conscientes de la necesidad actual de incluir secuencias didácticas o proyectos en los centros educativos que expongan al alumnado al uso de las TICs y que introduzcan el pensamiento computacional

Actualmente, un aumento sustancial está teniendo lugar en el número de investigaciones relacionadas con la introducción de la programación en el entorno escolar. Para comprender la razón de ser de este incremento, debemos conocer el concepto de pensamiento computacional. Wing (2006) lo define como el conjunto de habilidades cognitivas que “involucra la resolución de problemas, diseño de sistemas y comprensión de comportamiento humano” (p. 33). Según Shut et al. (2017) se pueden diferenciar cinco procesos cognitivos derivados del pensamiento computacional:

- La reformulación de problemas, de manera que, reorganizando e introduciendo cambios en la información expuesta para su mejor comprensión.
- La recursión de problemas, proceso el cual consiste en la asimilación de un problema a través de la comparación a otro más sencillo. Repitiendo este proceso se puede llegar a la solución de un problema a través de la simplificación del mismo.
- La descomposición de problemas, en varias partes más pequeñas y fáciles de afrontar.
- La abstracción, la cual se refiere a la aislación de un cierto elemento del resto que lo envuelven, para así poder analizarlo.
- El ensayo sistemático, realizando pruebas con valores de forma ordenada para llegar a la solución.

El pensamiento computacional es un elemento clave dentro de la inteligencia lógico-matemática, definida por Gardner, ya que su función principal es la de desencadenar los procesos cognitivos recién mencionados para que un individuo sea capaz de resolver la situación problemática en la que se encuentra.

Barr y Stephenson (2011) estudiaron la relación entre el pensamiento computacional y la demanda cognitiva de la resolución de problemas STEM, concluyendo que las

habilidades de programación, el pensamiento computacional y la resolución de problemas están interrelacionadas.

Para determinar la influencia real de la programación en el desarrollo cognitivo de los estudiantes, los académicos han acuñado el término de *transfer effect*, o *efecto de transferencia* en castellano, para medir el nivel de dicha influencia.

## 2. ¿QUÉ ES EL EFECTO DE TRANSFERENCIA?

Perkins y Salomon (1992) definieron el efecto de transferencia, como una situación en la cual el aprendizaje en un contexto determinado, tiene un impacto directo en el aprendizaje y desempeño en un contexto distinto o nuevo.

Esto viene a decir que, la actividad cognitiva de un ser humano en un ámbito determinado, puede tener un efecto positivo o negativo en el ejercicio de otro diferente al primero. Para ilustrar este concepto con un ejemplo, podemos considerar las matemáticas y el ajedrez. Para poder participar en este juego de mesa, un requisito *sine qua non* es la comprensión del movimiento de las piezas dentro de un plano cartesiano. El ajedrez desencadena el pensamiento complejo de los jugadores, quienes una vez da comienzo la partida, deberán calcular coordenadas y vectores de movimiento para anticiparse a las jugadas de su adversario. En este caso, el efecto de transferencia sería real si el conocimiento adquirido a través del juego es aplicable a otros contextos, como pudiera ser el área de matemáticas, donde los estudiantes trabajan con coordenadas cartesianas, o incluso la orientación espacial.

Con el fin de confirmar las hipótesis respecto al efecto de transferencia, Woodworth y Thorndike (1901) propusieron la *teoría de los elementos comunes*, la cual asociaba el nivel de éxito de la transferencia al grado de coincidencia de los elementos que conforman los dos contextos estudiados, dando así a entender que, la transferencia de habilidades entre dos campos de conocimiento notablemente desiguales es, menos frecuente que la transferencia entre campos similares.

La *teoría de los elementos comunes* dio lugar a dos nuevos conceptos dentro de este paradigma: la *transferencia cercana* y la *transferencia lejana*. Estos términos acuñados por Perkins y Salomon (1992), pretendían acotar los dominios de transferencia del conocimiento, considerándose *transferencia cercana* a la ocurrida entre dominios estrechamente relacionados, requiriendo una serie de habilidades y estrategias

semejantes, mientras que la *transferencia lejana* se refiere a la extrapolación exitosa dada en contextos distintos, los cuales pueden requerir habilidades o estrategias dispares.

Schunk (2012) concluye que “la transferencia de habilidades depende en la similitud y superposición entre los contextos y los problemas en los que las habilidades fueron adquiridas y aquellos en los que se presentaron posteriormente”.

Brandsford y Schwartz (1999) también señalaron que “el conocimiento previo es un prerequisite adicional para una transferencia exitosa, en particular aquel conocimiento sobre la estructura del problema, la variables involucradas y las estrategias de resolución”.

Estas afirmaciones parecen revelar que la correcta interiorización del sistema de resolución de problemas favorece la transferencia del mismo a otras situaciones con problemas similares, lo que podría significar que la importancia del efecto de transferencia subyace en los procesos cognitivos que los contextos de aprendizaje involucran.

Múltiples investigaciones, como las realizadas por Sala y Gobet (2017) en los ámbitos del ajedrez, la música, la memoria y los videojuegos, muestran que el grado de éxito de la *transferencia lejana* depende en gran parte de los ámbitos a los que se opusiesen, siendo esta en ocasiones inexistente.

En resumen, la literatura actual sobre el efecto de transferencia establece las siguientes determinaciones:

- El nivel de éxito del *efecto de transferencia* depende de la semejanza de los contextos que se traten.
- El conocimiento previo, así como la adquisición de los procesos de resolución son esenciales para el éxito de la transferencia.
- La transferencia lejana no siempre es certera, la cual varía en gran medida en función de los contextos de aprendizaje seleccionados.

### **3. RESULTADOS DE METAANÁLISIS REALIZADOS SOBRE LA TRANSFERIBILIDAD DE LA PROGRAMACIÓN**

Para determinar si el uso del *computer programming* en el aula de la lengua extranjera tiene un beneficio cognitivo real para el alumnado, se ha recurrido a una serie de metaanálisis de distintos tipos para estudiar la transferibilidad de las habilidades de programación a otros dominios. Los resultados de dichos metaanálisis se muestran a continuación en orden cronológico:

- Liao and Bright (1991), a través de un modelo de análisis de efectos aleatorios, obtuvieron un tamaño de efecto  $d= 0,41$  en relación a la transferibilidad de las habilidades de programación, es decir, confirma un efecto de transferencia moderado de estas habilidades a otros ámbitos. Sin embargo, esta investigación no diferenció entre habilidades cognitivas, lo cual se ha determinado de gran importancia en el apartado anterior a la hora de determinar la transferibilidad de habilidades.
- En el metaanálisis realizado por Liao (2000), amplió los resultados de su anterior investigación, incluyendo esta vez una distinción entre efectos de transferencia cercana, los cuales mostraban altos niveles de transferibilidad ( $d = 2,48$ ), y transferencia lejana, mostrando efectos muy leves, en concreto cuando se relacionaba con situaciones de pensamiento creativo. Por otra parte, Liao también midió la transferencia en las habilidades del pensamiento crítico, la metacognición, la resolución de problemas y la inteligencia espacial, concluyendo en que existía un efecto moderado.
- Uttal et al. (2013) realizó siete estudios planificando intervenciones con el objetivo de mejorar las habilidades espaciales de los estudiantes. Los resultados mostraron que, seis de las siete intervenciones tuvieron un efecto significativo en el desarrollo de las habilidades espaciales, lo cual desvela una correlación positiva en la transferencia del aprendizaje de la programación a contextos en los que involucran el uso de habilidades espaciales.
- Scherer et al. (2018) diseñaron una investigación en la cual hicieron diferencia entre las habilidades cognitivas de razonamiento, pensamiento creativo, pensamiento crítico, orientación espacial y lógico-matemáticas, además de los

efectos de transferencia cercana y lejana. Gracias a los resultados de las pruebas que realizaron llegaron una serie de conclusiones interesantes.

Para comenzar, determinan que la realización de actividades de *computer programming* favorece el desarrollo de otras habilidades cognitivas, las cuales son aplicables en otros ámbitos. Al igual que otros investigadores, están de acuerdo en que hay una base teórica que revela la existencia un conjunto de habilidades cognitivas que muestran un cierto de grado de transversalidad.

Uno de los autores que apoyan este postulado es Román-González et al. (2017) quien afirma que las habilidades de programación requieren la activación de la inteligencia fluida y formas de razonamiento alternativas.

Los resultados de Scherer et al. muestran efectos de transferencia lejanos significativos para procesos cognitivos que comparten ciertas microhabilidades con la programación, siendo el más significativo el pensamiento creativo, el cual el mismo Scherer , en 2016, ya había calificado como relevante para la creación del código informático, sustentando sus afirmaciones con las propuestas de otros académicos:

“Programming is a creative process that produces computational artifacts”  
(Grover y Pea, 2013, p. 39)

El proceso de la programación requiere convertir problemas a modelos computacionales para encontrar soluciones originales (Clemens, 1995)

También fundamentan sus declaraciones sobre la existencia de cierto efecto de transferencia a otras áreas distintas del pensamiento creativo, haciendo uso de las aportaciones al paradigma de Shute et al. (2017) quienes

“revisaron las similitudes y diferencias entre la programación como parte del pensamiento computacional o del pensamiento matemático, concluyendo que habilidades como la resolución de problemas y la modelación requieren la abstracción de problemas del mundo real, su formulación como modelos computacionales, la aplicación de estrategias y algoritmos para solucionarlos y la interpretación de la solución. Estas

microhabilidades compartidas pueden explicar el gran nivel de transferencia en el ámbito de las matemáticas” (Scherer et al., 2018, p. 783)

Por otra parte, también encontramos a los procesos de razonamiento y metacognición envueltos en este contexto de la programación informática. Los autores del artículo referencian a McCauley et al. (2008) quienes revelan que actividades del tipo *debugging*, es decir, búsqueda y solución de errores en el código, y reflexión sobre el uso de estrategias de solución, activan procesos de metacognición en el sujeto. Mientras tanto según Shute et al. (2017) y Yadav et al. (2017), el razonamiento forma parte de la ecuación cuando se realizan actividades que involucran el pensamiento lógico y algorítmico, así como los ejercicios que implican cierto grado de razonamiento, ya sea formal o informal.

Dichas enunciaciones previas a la elaboración del metaanálisis de Scherer et al., generó en los investigadores la expectativa de cierto éxito en la transferencia de la programación a estas dos habilidades. Sus resultados fueron similares a los otros estudios, estableciendo un grado de transferencia moderada, siendo este inferior al encontrado en relación al pensamiento creativo y el matemático, lo cual justifican haciendo referencia a declaración de Greiff et al. (2014) sobre el gran número de dominios a los que estos procesos cognitivos son aplicables.

En la misma línea, Scherer et al. incluyen la competencia en la orientación espacial dentro del espectro de destrezas trabajadas a través de la creación de código, justificando su presencia con el razonamiento de Clements y Sarama (1997) explican esta relación debido a la presencia de conocimientos como la comprensión de movimientos en el plano y el trabajo con objetos geométricos.

Finalmente, partiendo de la aseveración de Sala y Gobet (2017) quienes, en su estudio sobre la transferencia de aptitudes de los ámbitos del ajedrez, la música y el trabajo de la memoria, encontraron que, una habilidad en concreto no se transfiere de la misma manera a un contexto que a otro. Es decir, el éxito de la transferencia depende del contexto de destino. Scherer et al. comentan al respecto lo siguiente:

“The more distinct the situations students are invited to transfer their skills to are from computer programming the more challenging the far transfer is (...) this evidence cannot be interpreted causally—alternative explanations for the existence of far transfer exist. This interpretation assumes that learning computer programming may stimulate the development of cognitive skills that are required to solve complex problems in other contexts and domains. In any case, the positive effect sizes at least suggest that learning programming “does not do harm” to other cognitive skills.” (p. 784)

Resumiendo los resultados de los múltiples estudios, los académicos consideran que la programación tiene un impacto positivo en el desarrollo cognitivo del ser humano, concibiendo el acto de programar como un ejercicio directo del pensamiento computacional. Por otra parte, se ha demostrado que efectivamente existe la transferencia lejana de las habilidades de programación, la cual se da en distintos grados en función de la habilidad cognitiva, siendo las más beneficiadas la resolución de problemas, el pensamiento lógico-matemático, la orientación espacial, el razonamiento y la metacognición.

#### **4. LA ADQUISICIÓN DE LENGUAS EXTRANJERAS A TRAVÉS DE LA PROGRAMACIÓN INFORMÁTICA**

En los apartados previos, se ha discutido la evidencia de los beneficios para el desarrollo de competencias cognitivas de la introducción de la creación de código fuente en el aula. En esta sección, se centra la atención en la influencia directa de la programación sobre el proceso de enseñanza-aprendizaje de la lengua extranjera inglés.

Debido a que los países pioneros en el ámbito de la codificación informática didáctica son angloparlantes, la literatura disponible sobre la enseñanza de lenguas utilizando la programación informática como recurso es bastante reducida.

Además, aunque ya hacía años que la importancia del conocimiento sobre la codificación comenzaba a tener eco en la sociedad, la realización de proyectos

educativos que involucrasen el *computer programming* tenían dos principales obstáculos para ser llevados a cabo:

- La complejidad técnica de los programas utilizados, lo cual dificultaba de gran manera la transposición didáctica.
- La estética de los programas de creación de código. Este aspecto puede parecer superficial, pero la sobriedad y el pobre estímulo visual de estas supondría una desmotivación para los niños.

Sin embargo, con la aparición de plataformas en línea como *Scratch* (fig. 1), la realización de dichos proyectos se hizo posible, ya que simplificó el proceso de codificación a través de la creación de bloques de comandos e incluyendo una estética que capta la atención del público infantil.

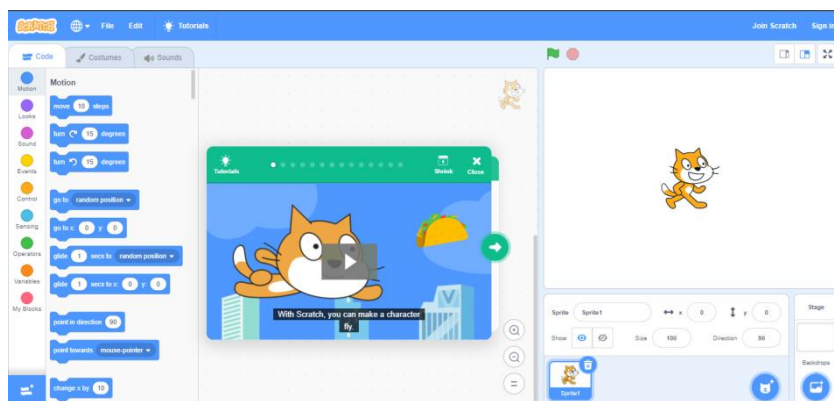


Figura 1. Interfaz de Scratch

Gracias a nuevos recursos, investigaciones como la llevada a cabo por Moreno y Robles (2015) fueron posibles. Estos académicos de Madrid diseñaron una unidad didáctica de 12 sesiones para el área de la lengua extranjera inglés, establecieron, entre los grupos de alumnos de 4º y 5º de Primaria, se creó para cada curso un grupo de control, al cual se le impartió la unidad de la manera habitual, sin ningún tipo de contacto con programas de redacción de código fuente, y uno experimental, el cuál haría si hizo uso de plataformas de programación, en concreto *Scratch*, durante el desarrollo de la unidad. Los docentes encargados de impartir la unidad didáctica fueron los mismos autores de la publicación. Ambos realizaron una breve formación inicial de 3 horas sin haber tenido contacto previo con la plataforma *Scratch*. Sin embargo, el conocimiento de uno de ellos tenía experiencia previa en el campo de la programación, ya que posee un grado universitario



en “Desarrollo de aplicaciones web”, lo que determinó un ritmo de aprendizaje y progreso sustancialmente mayor durante la formación.

Para medir la influencia de la programación informática en el desarrollo del lenguaje de los niños y niñas, se hizo una prueba sobre los contenidos de la unidad en dos momentos distintos, antes y después de la secuencia didáctica. Cada uno de los profesores fue asignado un curso, siendo responsables tanto de las sesiones del grupo de control, como de las del grupo experimental del curso asignado.

La investigación de Moreno y Robles (2015) mostró resultados positivos para la enseñanza de la lengua extranjera. Aunque en las calificaciones finales la diferencia no fue significativa, donde los resultados mostraron datos interesantes fue en la mejora respecto del primer test. El grupo experimental con el que trabajó el profesor con experiencia previa en la programación mejoró aproximadamente un 40% más de lo que lo hizo el grupo control.

Por otra parte, los docentes condujeron 6 encuestas distintas, con el objetivo de medir el impacto en otros ámbitos más vinculados con la dimensión socioafectiva y actitudinal del niño o niña en relación con la adquisición de la lengua extranjera, para lo cual debían indicar el grado de acuerdo con las siguientes afirmaciones:

- He disfrutado de las clases en las que he trabajado con Scratch
- Trabajar con Scratch me ha ayudado a aprender a trabajar en grupo
- Trabajar con Scratch me ha animado a aprender a buscar información de manera independiente
- Trabajar con Scratch me ha ayudado a aprender inglés
- Trabajar con Scratch me ha animado a aprender inglés
- Tras trabajar con Scratch, creo que la lengua inglesa es realmente importante si quiero aprender de manera independiente en Internet.

Las respuestas de las encuestas mostraban actitudes muy positivas en todos los apartados recién mencionados, suponiendo en todas ellas al menos un 80% los estudiantes que estaban de acuerdo o totalmente de acuerdo. Dichos porcentajes muestran un efecto positivo en la actitud hacia el aprendizaje de la lengua extranjera y la motivación para su adquisición.

Moreno y Robles (2015) indican que el conocimiento, naturalidad y manejo del docente con la materia expuesta en el aula, en este caso la *computer science*, es un factor diferencial, pudiendo determinar totalmente el aprovechamiento del alumnado de las sesiones. Este elemento también puede afectar al efecto de transferencia del conocimiento, sobre todo si nos centramos en la enseñanza de lenguas extranjeras, ya que el hecho de que el alumno o alumna sea consciente de la aplicabilidad de una habilidad a otro contexto.

Al respecto, la docente Laura Kirsop, docente y creadora de la organización Code Club de Reino Unido, hace la siguiente declaración cuando es entrevistada por Morrison (2013) “todavía queda un largo recorrido antes de que los maestros y maestras se sientan lo suficientemente seguros como para enseñar estas habilidades (de programación).”, en respuesta al comentario hecho por el entrevistador, quien puntualizaba que, para que la creación de código esté completamente integrada en el sistema educativo, será necesario un gran esfuerzo en materia de formación del profesorado.

Años más tarde, un gran número de organizaciones educativas internacionales comenzaron a cubrir esta falta de accesibilidad al dominio de la ciencia computacional, creando para ello plataformas online que pusieran a la disposición del profesorado un amplio abanico de actividades en el ámbito de la programación, así como cursos para la formación de profesores de todas las etapas educativas para su enseñanza. Stevens y Verschoor (2017) revisan una serie de recursos online al respecto, afirmando que

“In the context of ELT there are several ways to start integrating coding as a new kind of literacy. Most coding websites are easy to follow and they provide clear tutorials on how to get started, so neither students nor English language teachers need to have previous knowledge of coding.” (p. 9)

En el mismo artículo, Stevens y Verschoor exponen la transferibilidad de la programación informática a contextos lingüísticos, haciendo mención de Meredith Towne, profesora de teatro e inglés como lengua extranjera, quien en una entrevista realizada por Morais (2015) declara que “el lenguaje de *Scratch* es muy similar al lenguaje teatral”. Otros docentes también apoyan esta visión en la que la programación informática está vinculada con el lenguaje, ya sea en la lengua extranjera o en la lengua

materna, como Brown (2016), quien considera los procesos de redacción de textos y escritos y creación de código similares en muchos aspectos, destacando la fase de planificación necesaria en ambos casos, o Cohen (2017), quien utiliza la plataforma *Scratch* para realizar actividades en las que los alumnos deben codificar conversaciones breves, la cual libera la adquisición de la lengua de los obstáculos sintácticos y ortográficos que el alumnado pueda encontrar, ya que esta plataforma pone a la disposición de sus usuarios bloques predeterminados que no dan margen a error en este ámbito (fig. 2).

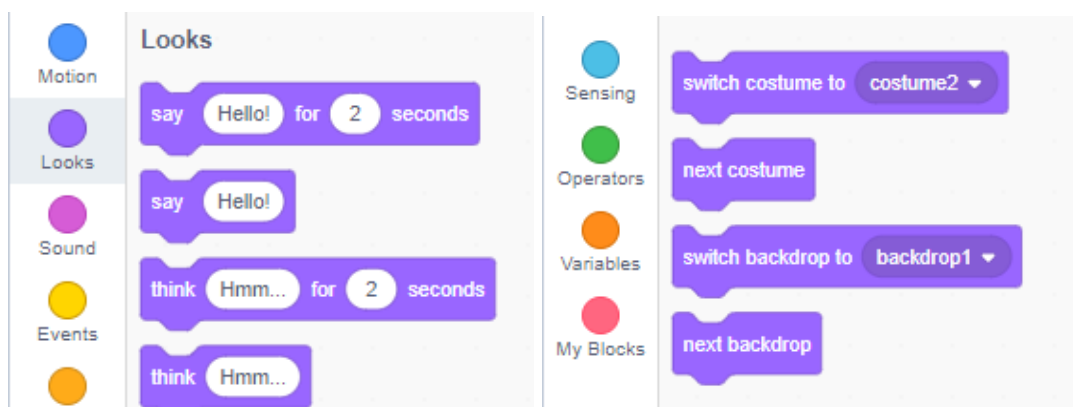


Figura 2. Ejemplos de comandos de Scratch

Por otra parte, teniendo en cuenta el paradigma de las habilidades de la lengua, Stevens y Verschoor consideran que, dado la tipología de actividades que se dan en estos sitios web, se desarrollan las habilidades de comprensión escrita, ya que en cada actividad se propone un problema en la lengua extranjera que deben comprender para poder solucionar, y la expresión escrita, a través de un proceso simplificado en el que sea crea un texto el cual, en vez de tener forma de redacción, tiene forma de código fuente.

En relación a las habilidades de la lengua, Weng y Wong (2017) diseñaron y pusieron en práctica, un proyecto de integración del pensamiento computacional en el aprendizaje de la interacción oral de la lengua extranjera inglés, haciendo uso de la plataforma didáctica de programación informática *Scratch*. El proyecto consistía en la simulación informática de situaciones comunicativas. Para ello, los alumnos, siguiendo las directrices del docente, diseñaban el diálogo y posteriormente se les pedía corregir los errores gramaticales que encontrasen. Los investigadores concluyen que la programación informática “proporciona la posibilidad para los estudiantes de aprender de forma efectiva la habilidad de comunicación mientras utilizan o mejoran su

pensamiento computacional” (p. 5) Además, Weng y Wong (2017) coinciden con Moreno y Robles (2015) en dos aspectos de considerable importancia. Por un parte, están de acuerdo en que, la adquisición de la lengua extranjera haciendo uso de lenguaje de programación gráfico, tiene un gran potencial como elemento motivador para el alumnado. Por otra parte, arrojan la misma luz sobre la influencia del nivel de formación del maestro o maestra en el éxito del proceso de enseñanza-aprendizaje, siendo este determinante al respecto.

# INTERVENCIÓN DIDÁCTICA

Como producto de la investigación realizada sobre el *computer programming*, se ha diseñado una propuesta de proyecto anual, llamado “Let’s code!”, en el cual se trabajaría la programación en el aula de informática como parte de las áreas de inglés y matemáticas.

## 1. CONTEXTO

El proyecto “Let’s code!” podría tener en cualquier centro con unas dotaciones mínimas de al menos una sala de informática. Sería más adecuado para centro con una certificación TIC CoDiCe de la Junta de Castilla y León de nivel 4 o superior, ya que al estar los alumnos y alumnas acostumbrados al hecho de utilizar los dispositivos electrónicos como herramienta de aprendizaje, la participación en este proyecto no supondría un “shock” para ellos. Sin embargo, esto no es indicativo de que el proyecto no pueda tener lugar en otro tipo de centros con una menor integración de las TICs en el aula, si no que sería muy beneficioso ya que podría ser el primer paso para la evolución del centro en esta dirección. Sí sería adecuado la realización de una fase preparatoria para casos en los que los alumnos y alumnas participantes no hayan tenido un contacto previo con un ordenador, ya que a pesar de ser considerados “nativos digitales”, los niños y niñas de estas generaciones muestran ciertas dificultades en el paso del smartphone o tablet al portátil o ordenador de sobremesa, debido principalmente al cambio de la pantalla táctil al ratón, llegándose a frustrar en ocasiones por esta razón, también enseñándoles el uso del navegador de Internet.

Para contextualizar la propuesta, se ha elegido el centro CEIP Kantic@ Arroyo de titularidad pública, el cual comenzó su actividad en el año 2008 como centro de línea tres. La enseñanza en este centro es bilingüe en inglés y la integración de las TIC está valorada con un nivel “4: Avanzado” en el baremo CoDiCe TIC. El centro cuenta con la dotación de unas tablets por parte de la Junta de Castilla y León para su uso en el centro escolar, las cuales permitirán la realización de este proyecto.

El CEIP Kantic@ Arroyo se encuentra en Sotoverde, una urbanización perteneciente al municipio de Arroyo de la Encomienda, situada dentro del alfoz de Valladolid y a 10 kilómetros del centro de la capital. El alumnado matriculado proviene tanto de la zona

en la que enclava como las circundantes: Monasterio del Prado, Aranzana, La Vega y Arroyo de la Encomienda.

Por lo general, el nivel económico de las familias es medio, siendo escasa la presencia de economías más restringidas. En cuanto al nivel cultural y de estudios, este se sitúa en una posición media-alta. Dentro de la comunidad educativa coexisten familias muy implicadas en la educación de sus hijos con un pequeño porcentaje de familias desinteresadas, matrimonios separados y desestructuración familiar.

## **2. MOMENTO DE IMPLANTACIÓN EN EL CURSO**

“Let’s code” está formado por 12 sesiones que se reparten a lo largo del curso, correspondiendo 4 de ellas a cada trimestre. Al ser un centro de línea tres, cada semana, el docente que haya recibido la formación sobre *computer programming* visitaría uno de los tres grupos para la impartición de las sesiones

## **3. ALUMNADO**

El proyecto “Let’s code” está adaptado al nivel de desarrollo cognitivo y destreza en la lengua extranjera de los alumnos y alumnas del 6º curso de la etapa Primaria del centro.

## **4. OBJETIVOS**

- Introducir la programación informática en el aula de Educación Primaria
- Posibilitar la adquisición la lengua extranjera a través de su uso como lenguaje de programación y lengua de comunicación durante las sesiones
- Favorecer el desarrollo de los procesos cognitivos del razonamiento, la metacognición, la resolución de problemas, la orientación espacial, el pensamiento creativo y el pensamiento computacional
- Motivar al alumnado a descubrir áreas de conocimiento alternativas a las habituales y a formarse en ellas
- Generar situaciones problemáticas diversas para su resolución
- Fomentar el uso responsable de las TICs

## **5. TEMPORALIZACIÓN**

“Let’s code!” está diseñado para durar en 12 sesiones de una hora aproximadamente cada una, las cuales tendrán lugar los viernes en la franja horaria de las 11:00 a las 12:00. Al ser un proyecto que globaliza las áreas de English y Matemáticas, a la hora de

situarlo en el horario, ocupará media hora de una sesión de English y media hora de una sesión de Matemáticas en el cómputo de horas semanales dedicadas a cada área.

## **6. METODOLOGÍA**

“Let’s code” es un proyecto que fuerza al uso de una metodología activa en el aula, situando al estudiante en el centro del aprendizaje. Las sesiones están diseñadas teniendo como base la teoría del constructivismo.

Cada lección aumenta la profundidad del conocimiento de la anterior, siendo los niños y niñas los principales agentes del proceso de aprendizaje, ya que son ellos quienes, tras el uso de los materiales proporcionados por el docente, quienes crean su propio conocimiento.

El maestro o maestra cumple la función de facilitador, seleccionando materiales de calidad que cumplan la función de andamiaje y enriquezcan el proceso de enseñanza-aprendizaje.

Por otra parte, todos los materiales incluidos son *realia*, ya que han sido diseñados en países angloparlantes, en concreto en Estados Unidos, para el aprendizaje de disciplinas no lingüísticas.

En conjunto, tenemos un proyecto que trata de llevar la programación informática, así como todos los beneficios cognitivos que esta aporta, al aula, utilizando la lengua extranjera inglés como el medio que envuelve todo el proyecto.

## **7. CONTENIDOS**

- Comprensión de fragmentos del lenguaje como comandos en inglés
- Comprensión de enunciados de problemas en la lengua extranjera inglés
- Programación en papel
- Comandos de programación básicos
- Detección de bugs
- Loops en la programación
- Diseño de escenarios
- Comandos de eventos
- Píxeles como medida
- Rotaciones con grados

## 8. CRITERIOS DE EVALUACIÓN

- Construye secuencias simples de comandos en papel
- Completa tareas de creación de código en inglés haciendo uso de una plataforma en línea
- Diseña creaciones propias en la lengua extranjera haciendo uso de una plataforma en línea

## 9. COMPETENCIAS BÁSICAS

Las competencias básicas a trabajar son las siguientes:

- Competencia en comunicación lingüística
- Competencia matemática y competencias básicas en ciencia y tecnología
- Competencia digital
- Competencia social y cívica

## 10. DESARROLLO DEL PROYECTO

Session 1	
<b>Aims</b> <ul style="list-style-type: none"><li>• Students should be able to understand the information presented on a video</li><li>• Students should be able to make a simple programming sequence on paper.</li></ul>	<b>Contents</b> <ul style="list-style-type: none"><li>• Comprehension of language fragments in form of commands in English</li><li>• Comprehension of problem statements in English</li><li>• Programming on paper</li></ul>
<b>Assessment criteria</b> <ul style="list-style-type: none"><li>• Students will be able to make a simple sequence of commands in paper</li></ul>	
<b>Activities</b>	<b>Timing</b>
Let's start our project!	10 minutes
My Robotic Friends	45 minutes



<b>Activity 1</b> <b>Session 1</b>		<b>Skills:</b> Listening and speaking
<b>Title:</b> Let's start our project!	<b>Type:</b> Introduction	<b>Timing:</b> 10 minutes
<b>Classroom arrangement:</b> Students sit individually in their seats	<b>Resources:</b> Video: <a href="#">Unplugged – My Robotic Friends</a> URL: <a href="https://youtu.be/xaW3PAzHxCU">https://youtu.be/xaW3PAzHxCU</a>	
<b>Aims:</b> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> </ul>		
<b>Input:</b> Good morning children! <i>(Waiting for their response)</i> How are you? <i>(Waiting for their response)</i> Today we are going to start our project “Let’s code!”! I am very excited! Are you excited too? <i>(Waiting for their response)</i> First of all, we are going to see this video about programming, which will show us the instructions of our first activity. Let’s go! <i>(Plays the video)</i>		

<b>Activity 2</b> <b>Session 1</b>		<b>Skills:</b> Listening and speaking
<b>Title:</b> My Robotic Friends	<b>Type:</b> Reinforcement	<b>Timing:</b> 45 minutes
<b>Classroom arrangement:</b> Students work in trios or groups of four	<b>Resources:</b> Annex I. Stacking Cups	
<b>Aims:</b> <ul style="list-style-type: none"> <li>• Students should be able to make a simple programming sequence on paper.</li> </ul>		
<b>Input:</b> Alright! Do you understand the activity? <i>(Waiting for their response)</i> OK! Now that we have seen the video, I am going to give you these papers with three <i>(showing three fingers)</i> images of different constructions <i>(showing the images)</i> , and		

these plastic cups (*pointing them*) for you to do the constructions.

In a piece of paper, you will have to draw arrows to help your partners build the constructions in the images. The builder can only see the arrows you have written.

They can't see the image of the constructions you use to make the instructions. OK?

(*Waiting for their response*)

Perfect! Let's see an example.

(*The teacher gives an example of a random construction, writing in the blackboard the arrows they should draw*)

Alright, now, get in groups of three or four people and I will give you the sheet with the constructions.

Session 2	
<p><b>Aims</b></p> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>	<p><b>Contents</b></p> <ul style="list-style-type: none"> <li>• Comprehension of language fragments in form of commands in English</li> <li>• Comprehension of problem statements in English</li> <li>• Basic programming commands</li> </ul>
<p><b>Assessment criteria</b></p> <ul style="list-style-type: none"> <li>• Students will be able to complete a simple coding task on an online platform in English</li> </ul>	
Activities	Timing
Introduction to Code Studio	10 minutes
Chasing the pigs!	45 minutes

<b>Activity 1</b> <b>Session 2</b>		<b>Skills:</b> Listening and speaking
<b>Title:</b> Introduction to Code Studio	<b>Type:</b> Introduction	<b>Timing:</b> 10 minutes
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Video: Maze Intro: Programming with Blocks (Course C-F) URL: <a href="https://youtu.be/uBZpd6zGVFI">https://youtu.be/uBZpd6zGVFI</a>
<b>Aims:</b> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> </ul>		
<b>Input:</b> Good morning children! <i>(Waiting for their response)</i> How are you? <i>(Waiting for their response)</i> Today we are going to play Angry Birds! Wow! How exciting! We are going to learn how to program using the tablets! OK! Before we start, let's see a video. In the video we will see how we can program in the Code.org web. Listen carefully, please. <i>(Plays the video)</i> Wow! It seems really entertaining... Do you want to start? <i>(Waiting for their response)</i>		

<b>Activity 2</b> <b>Session 2</b>		<b>Skills:</b> Listening, reading and writing
<b>Title:</b> Chasing the pigs!	<b>Type:</b> Reinforcement	<b>Timing:</b> 45 minutes
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Programming with Angry Birds Activity URL: <a href="https://studio.code.org/s/coursesec-2021/lessons/3/levels/2">https://studio.code.org/s/coursesec-2021/lessons/3/levels/2</a>
<b>Aims:</b> <ul style="list-style-type: none"> <li>• Students should be able to comprehend the basics of programming</li> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>		

**Input:**

OK! Now, take your tablets and open the link I have left on Teams.

There are ten activities for you to do. To do them you will need to put into practice what you have learnt on the video. Do them one by one reading carefully the statement at the top. When you finish an activity, raise your hand, I will go to check and then you can continue. If you have any problem, raise your hand too.

We start in 3, 2, 1... Go!

*(Students work on their own)*

Session 3	
<b>Aims</b> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> <li>• Students should be able to detect minor errors in a sample of code</li> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>	<b>Contents</b> <ul style="list-style-type: none"> <li>• Comprehension of language fragments in form of commands in English</li> <li>• Comprehension of problem statements in English</li> <li>• Basic programming commands</li> <li>• Bugs detection</li> </ul>
<b>Assessment criteria</b> <ul style="list-style-type: none"> <li>• Students will be able to complete a simple coding task on an online platform in English</li> </ul>	
Activities	Timing
Let's fix some bugs!	10 minutes
Debugging in Maze	45 minutes

<b>Activity 1</b> <b>Session 3</b>		<b>Skills:</b> Listening and speaking
<b>Title:</b> Let's fix some bugs!	<b>Type:</b> Introduction	<b>Timing:</b> 10 minutes
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Video: Debugging with the Step Button URL: <a href="https://youtu.be/RUZTRNakV9c">https://youtu.be/RUZTRNakV9c</a>

<p><b>Aims:</b></p> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> </ul>
<p><b>Input:</b></p> <p>Good morning children!</p> <p><i>(Waiting for their response)</i></p> <p>How are you?</p> <p><i>(Waiting for their response)</i></p> <p>Today we are going to fix bugs! What? Bugs? OK, let's start with this video, it explains everything you need to know about computer bugs. Listen carefully, please.</p> <p><i>(Plays the video)</i></p> <p>Alright! What new thing are we learning today?</p> <p><i>(Waiting for their response)</i></p> <p>Perfect! Bugs and debugging! Bugs are errors in our code and debugging is the action of correcting them. Excellent!</p>

<b>Activity 2</b>		<b>Session 3</b>		<b>Skills:</b> Listening, reading and writing	
<b>Title:</b> Debugging in Maze		<b>Type:</b> Reinforcement		<b>Timing:</b> 45 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats			<b>Resources:</b> Debugging in Maze Activity URL: <a href="https://studio.code.org/s/coursesec-2021/lessons/4/levels/2">https://studio.code.org/s/coursesec-2021/lessons/4/levels/2</a>		
<p><b>Aims:</b></p> <ul style="list-style-type: none"> <li>• Students should be able to detect minor errors in a sample of code</li> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>					
<p><b>Input:</b></p> <p>Alright! Any questions?</p> <p><i>(Waiting for their response)</i></p> <p>OK! Now, take your tablets and open the link I have sent in Teams. There are nine activities that you can do. To do them you will need to put into practice what you have learnt on the video, search the bugs in the code and debug them. Do them one by one reading carefully the statement at the top. When you finish an activity, raise your hand</p>					

and I will go to check and then you can continue. If you have any problem, raise your hand too.

We start in 3, 2, 1... Go!

*(Students work on their own)*

Session 4	
<p><b>Aims</b></p> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>	<p><b>Contents</b></p> <ul style="list-style-type: none"> <li>• Comprehension of language fragments in form of commands in English</li> <li>• Comprehension of problem statements in English</li> <li>• Basic programming commands</li> </ul>
<p><b>Assessment criteria</b></p> <ul style="list-style-type: none"> <li>• Students will be able to complete a simple coding task on an online platform in English</li> </ul>	
Activities	Timing
Let's meet the collector!	10 minutes
Collecting treasure with Laurel	45 minutes

<b>Activity 1</b>		<b>Session 4</b>		<b>Skills:</b> Listening and speaking	
<b>Title:</b> Let's meet the collector!		<b>Type:</b> Introduction		<b>Timing:</b> 10 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats			<b>Resources:</b> Video: The Collector URL: <a href="https://youtu.be/_hUFPk3qA78">https://youtu.be/_hUFPk3qA78</a>		
<b>Aims:</b>					
<ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> </ul>					

**Input:**  
 Good morning children!  
*(Waiting for their response)*  
 How are you?  
*(Waiting for their response)*  
 Today, in our programming lesson, we are going to meet a brave explorer! Wow! I am very excited! Are you excited?  
*(Waiting for their response)*  
 OK! Let’s meet Laurel in this video. Are you ready?  
*(Plays the video)*  
 Alright! What new command are we learning today?  
*(Waiting for their response)*  
 Well done! “Collect” is the new command to take objects.

<b>Activity 2</b> <b>Session 4</b>		<b>Skills:</b> Listening, reading and writing	
<b>Title:</b> Collecting treasure with Laurel	<b>Type:</b> Reinforcement	<b>Timing:</b> 45 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Collecting with Laurel Activity URL: <a href="https://studio.code.org/s/coursec-2021/lessons/5/levels/2">https://studio.code.org/s/coursec-2021/lessons/5/levels/2</a>	
<b>Aims:</b>			
<ul style="list-style-type: none"> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>			
<b>Input:</b>			
<p>Alright! Is everything clear?  <i>(Waiting for their response)</i>        OK! Now, take your tablets and open the link I have published in Teams. There are twelve activities that you can do. To do them you will need to put into practice what you have learnt on the video, putting the command “Collect” in your code. Do them one by one reading carefully the statement at the top. When you finish an activity, raise your hand and I will go to check and then you can continue. If you have any problem, raise your hand too.</p>			

We start in 3, 2, 1... Go!  
*(Students work on their own)*

Session 5	
<p><b>Aims</b></p> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> <li>• Students should be able to use simple coding commands to reach a goal</li> <li>• Students should be able draw simple shapes following instructions</li> </ul>	<p><b>Contents</b></p> <ul style="list-style-type: none"> <li>• Comprehension of language fragments in form of commands in English</li> <li>• Comprehension of problem statements in English</li> <li>• Basic programming commands</li> <li>• Pixels as a measure</li> <li>• Rotations with degrees</li> </ul>
<p><b>Assessment criteria</b></p> <ul style="list-style-type: none"> <li>• Students will be able to complete a simple coding task on an online platform in English</li> </ul>	
Activities	Timing
Let's become artists!	10 minutes
Creating art with Code!	45 minutes

<b>Activity 1</b>		<b>Session 5</b>		<b>Skills:</b> Listening and speaking	
<b>Title:</b> Let's become artists!		<b>Type:</b> Introduction		<b>Timing:</b> 10 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats			<b>Resources:</b> Video: Artist Intro with JR Hildebrand URL: <a href="https://youtu.be/MUavAHn-QUg">https://youtu.be/MUavAHn-QUg</a>		
<b>Aims:</b>					
<ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> </ul>					



**Input:**  
 Good morning children!  
*(Waiting for their response)*  
 How are you?  
*(Waiting for their response)*  
 Today, in our programming lesson, we are going to become artists! Wow! I am very excited! Are you excited?  
*(Waiting for their response)*  
 OK! In this video, we are going to meet JR Hildebrand, a race car driver! He is going to teach us how to become artists. Listen to him carefully, please.  
*(Plays the video)*  
 Alright! What new things are we learning today?  
*(Waiting for their response)*  
 Well done! We are introducing the use degrees and pixels in our code.

<b>Activity 2</b> <b>Session 5</b>		<b>Skills:</b> Listening, reading and writing	
<b>Title:</b> Creating art with Code!	<b>Type:</b> Reinforcement	<b>Timing:</b> 45 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Creating art with Code! Activity URL: <a href="https://studio.code.org/s/coursesec-2021/lessons/6/levels/2">https://studio.code.org/s/coursesec-2021/lessons/6/levels/2</a>	
<b>Aims:</b>			
<ul style="list-style-type: none"> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>			
<b>Input:</b>			
<p>Alright! Is everything clear?  <i>(Waiting for their response)</i></p> <p>OK! Now, take your tablets and open the link you have on Teams. There are nine activities that you can do. To do them you will need to put into practice what you have learnt on the video, you will use pixels and degrees to make drawings in the computer. Do them one by one reading carefully the statement at the top. When you finish an activity, raise your hand and I will go to check and then you can continue. If you have</p>			

any problem, raise your hand too.

We start in 3, 2, 1... Go!

*(Students work on their own)*

Session 6	
<p><b>Aims</b></p> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> <li>• Students should be able to make a simple programming sequence on paper.</li> <li>• Students should be able to use loops in their coding sequence to shorten the code</li> </ul>	<p><b>Contents</b></p> <ul style="list-style-type: none"> <li>• Comprehension of language fragments in form of commands in English</li> <li>• Comprehension of problem statements in English</li> <li>• Programming on paper</li> <li>• Loops in programming</li> </ul>
<p><b>Assessment criteria</b></p> <ul style="list-style-type: none"> <li>• Students will be able to make a simple sequence of commands in paper</li> </ul>	
Activities	Timing
What is a loop?	10 minutes
My Loopy Robotic Friends	45 minutes

<b>Activity 1</b>		<b>Session 6</b>	<b>Skills:</b> Listening and speaking
<b>Title:</b> What is a loop?	<b>Type:</b> Introduction		<b>Timing:</b> 10 minutes
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Video: Unplugged – My Robotic Friends URL: <a href="https://youtu.be/xaW3PAzHxCU">https://youtu.be/xaW3PAzHxCU</a>	
<b>Aims:</b>			
<ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> </ul>			

**Input:**  
 Good morning children!  
*(Waiting for their response)*  
 How are you?  
*(Waiting for their response)*  
 Today, in our programming lesson, we are going to become robots! Do you want to know how?  
*(Waiting for their response)*  
 OK! Do you remember our first lesson?  
*(Waiting for their response)*  
 Good! What we are doing is very similar. Let's see the video.  
*(Plays the video)*  
 Well done! We are learning to use degrees and pixels.

<b>Activity 2</b> <b>Session 6</b>		<b>Skills:</b> Listening, reading and writing	
<b>Title:</b> My Loopy Robotic Friends	<b>Type:</b> Reinforcement	<b>Timing:</b> 45 minutes	
<b>Classroom arrangement:</b> Students work in groups of four		<b>Resources:</b> Annex I. Stacking cups	
<b>Aims:</b> <ul style="list-style-type: none"> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>			
<p><b>Input:</b>          Alright! Do you understand it? Do you remember?  <i>(Waiting for their response)</i>          OK! Now, I will give you these papers with three (<i>showing three fingers</i>) images of different constructions (<i>showing the images</i>). As well, there are some plastic cups (<i>pointing them</i>) for you to do the constructions.          In a piece of paper, you will have to draw arrows to help your partners build the constructions in the images. The builder can only see the arrows you have written. They can't see the model image you use to make the constructions. OK?  <i>(Waiting for their response)</i>          Perfect! Let's see an example.</p>			

*(The teacher gives an example of a random construction, writing in the blackboard the arrows they should draw)*

Now, there is something new today. We are learning about loops! Do you want to know what a loop is?

*(Waiting for their response)*

Great! A loop is something that repeats. When we program, a loop is a repeated command. Now, you can write x4 and an arrow instead of four arrows. Let's see an example.

*(The teacher gives an example simplifying the "code" using loops in the place of repeated arrows)*

Alright, now, get in groups of three or four people and I will give you the sheet with the constructions.

Session 7	
<p><b>Aims</b></p> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> <li>• Students should be able to use simple coding commands to reach a goal</li> <li>• Students should be able to use loops in their coding sequence to shorten the code</li> </ul>	<p><b>Contents</b></p> <ul style="list-style-type: none"> <li>• Comprehension of language fragments in form of commands in English</li> <li>• Comprehension of problem statements in English</li> <li>• Basic programming commands</li> <li>• Loops in programming</li> </ul>
<p><b>Assessment criteria</b></p> <ul style="list-style-type: none"> <li>• Students will be able to complete a simple coding task on an online platform in English</li> </ul>	
<b>Activities</b>	<b>Timing</b>
New coding friends!	10 minutes

Coding with Rey and BB-8	10 minutes
BB-8 teaches us how to loop	5 minutes
Loops with Rey and BB-8	30 minutes

<b>Activity 1</b> <b>Session 7</b>		<b>Skills:</b> Listening and speaking
<b>Title:</b> New coding friends!	<b>Type:</b> Introduction	<b>Timing:</b> 10 minutes
<b>Classroom arrangement:</b> Students sit individually in their seats	<b>Resources:</b> Video: Programming with Rey and BB-8 URL: <a href="https://youtu.be/SJgje3OuTJY">https://youtu.be/SJgje3OuTJY</a>	
<b>Aims:</b> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> </ul>		
<b>Input:</b> <p>Good morning children!  <i>(Waiting for their response)</i></p> <p>How are you?  <i>(Waiting for their response)</i></p> <p>Today, in our programming lesson, we are meeting two space travellers! Wow! How exciting! Do you want to meet them?  <i>(Waiting for their response)</i></p> <p>OK! Let's go!  <i>(Plays the video)</i></p> <p>Wow! That is an exciting mission. Do you want to help?  <i>(Waiting for their response)</i></p> <p>Good to hear that!</p>		

<b>Activity 2</b> <b>Session 7</b>		<b>Skills:</b> Listening, reading and writing	
<b>Title:</b> Coding with Rey and BB-8	<b>Type:</b> Reinforcement	<b>Timing:</b> 10 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Loops with Rey and BB-8 Activities 2 - 3 URL: <a href="https://studio.code.org/s/coursec-2021/lessons/6/levels/2">https://studio.code.org/s/coursec-2021/lessons/6/levels/2</a>	
<b>Aims:</b>			
<ul style="list-style-type: none"> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>			
<b>Input:</b>			
<p>So, is everything clear? (Waiting for their response)</p> <p>OK! Now, take your tablets and open the link I have sent you through Teams. First, you will do exercises 2 and 3 to practice a bit, and then we will see another video which teaches us something new. But first, let's do those two exercises. As always, ask any questions if you need to.</p> <p>Ready? Steady? Go! (Students work on their own)</p>			

<b>Activity 3</b> <b>Session 7</b>		<b>Skills:</b> Listening and speaking	
<b>Title:</b> BB-8 teaches us how to loop	<b>Type:</b> Introduction	<b>Timing:</b> 5 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Video: Repeat Blocks with BB-8 URL: <a href="https://youtu.be/sDg32zIIHrM">https://youtu.be/sDg32zIIHrM</a>	
<b>Aims:</b>			
<ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> </ul>			
<b>Input:</b>			
<p>Alright! Let's now listen carefully to BB-8, he wants to tell us something! (Plays video)</p> <p>Mmm... Interesting. That is how we use loops in our code. Let's try to use them!</p>			

<b>Activity 4</b> <b>Session 7</b>		<b>Skills:</b> Listening, reading and writing	
<b>Title:</b> Loops with Rey and BB-8	<b>Type:</b> Reinforcement	<b>Timing:</b> 30 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Loops with Rey and BB-8 Activities: 5 to 14 URL: <a href="https://studio.code.org/s/coursesec-2021/lessons/8/levels/5">https://studio.code.org/s/coursesec-2021/lessons/8/levels/5</a>	
<b>Aims:</b>			
<ul style="list-style-type: none"> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>			
<b>Input:</b>			
<p>Alright! Is everything clear? (<i>Waiting for their response</i>)</p> <p>OK! Now, take your tablets and go back to Teams and follow the second link you have on Teams. There are nine activities that you can do. To do them you will need to put into practice what you have learnt on the video, you will use pixels and degrees to make drawings in the computer. Do them one by one reading carefully the statement at the top. When you finish an activity, raise your hand and I will go to check and then you can continue. If you have any problem, raise your hand too.</p> <p>We start in 3, 2, 1... Go! (<i>Students work on their own</i>)</p>			

<b>Session 8</b>	
<b>Aims</b> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> <li>• Students should be able to use simple coding commands to reach a goal</li> <li>• Students should be able to use loops</li> </ul>	<b>Contents</b> <ul style="list-style-type: none"> <li>• Comprehension of language fragments in form of commands in English</li> <li>• Comprehension of problem statements in English</li> <li>• Basic programming commands</li> </ul>

in their coding sequence to shorten the code	<ul style="list-style-type: none"> <li>Loops in programming</li> </ul>
<b>Assessment criteria</b> <ul style="list-style-type: none"> <li>Students will be able to complete a simple coding task on an online platform in English</li> </ul>	
<b>Activities</b>	<b>Timing</b>
Harvest time!	10 minutes
Harvesting crops with loops	45 minutes

<b>Activity 1</b> <b>Session 8</b>		<b>Skills:</b> Listening and speaking
<b>Title:</b> Harvest time!	<b>Type:</b> Introduction	<b>Timing:</b> 10 minutes
<b>Classroom arrangement:</b> Students sit individually in their seats	<b>Resources:</b> Video: The Harvester URL: <a href="https://youtu.be/Ns8GMU6Y_C8">https://youtu.be/Ns8GMU6Y_C8</a>	
<b>Aims:</b> <ul style="list-style-type: none"> <li>Students should be able to understand the information presented on a video</li> </ul>		
<b>Input:</b> <p>Good morning children! (Waiting for their response)</p> <p>How are you? (Waiting for their response)</p> <p>Today, we are continuing our “Let’s code” project. We are also meeting a new friend... a harvester! Do you want to meet her? (Waiting for their response)</p> <p>Alright! This video will introduce us the harvester and her new function! (Plays the video)</p> <p>Alright! What new command are we learning today? (Waiting for their response)</p> <p>Well done! “Pick” is the new command to take objects. It is very similar to collect, but it only works with and specific object. For example, if the harvester is on a pumpkin</p>		



and the command written on the code is “pick pumpkin”, she will pick it. But, if the command is “pick lettuce”, nothing will happen and there will be an error in the code. Don’t worry if it sounds strange, you will understand it when you do the activities.

<b>Activity 2</b> <b>Session 8</b>		<b>Skills:</b> Listening, reading and writing	
<b>Title:</b> Harvesting crops with loops	<b>Type:</b> Reinforcement	<b>Timing:</b> 45 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Harvesting crops with loops Activity URL: <a href="https://studio.code.org/s/coursec-2021/lessons/9/levels/2">https://studio.code.org/s/coursec-2021/lessons/9/levels/2</a>	
<b>Aims:</b>			
<ul style="list-style-type: none"> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>			
<b>Input:</b>			
<p>Alright! Do you have any doubts? (<i>Waiting for their response</i>)</p> <p>OK! Now, take your tablets and open the link you have on Teams. There are ten activities that you can do. To do them you will need to put into practice what you have learnt on the video, you will using the command “pick” and some loops to make your code shorter. Do them one by one reading carefully the statement at the top. When you finish an activity, raise your hand and I will go to check. Then, you can continue. If you have any problem, raise your hand too.</p> <p>We start in 3, 2, 1... Go! (<i>Students work on their own</i>)</p>			

<b>Session 9</b>	
<b>Aims</b>	<b>Contents</b>
<ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> </ul>	<ul style="list-style-type: none"> <li>• Comprehension of language fragments in form of commands in English</li> </ul>

<ul style="list-style-type: none"> <li>• Students should be able to use simple coding commands to reach a goal</li> <li>• Students should be able to use loops in their coding sequence to shorten the code</li> </ul>	<ul style="list-style-type: none"> <li>• Comprehension of problem statements in English</li> <li>• Basic programming commands</li> <li>• Loops in programming</li> <li>• Pixels as a measure</li> <li>• Rotations with degrees</li> </ul>
<b>Assessment criteria</b> <ul style="list-style-type: none"> <li>• Students will be able to make their own creations in the foreign language on a coding online platform</li> </ul>	
<b>Activities</b>	<b>Timing</b>
Presenting our new project: Sticker Art	5 minutes
Hands on with our project	50 minutes

<b>Activity 1</b>		<b>Session 9</b>		<b>Skills:</b> Listening and speaking	
<b>Title:</b> Presenting our new project: Sticker Art		<b>Type:</b> Introduction		<b>Timing:</b> 5 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats			<b>Resources:</b> None		
<b>Aims:</b> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> </ul>					
<b>Input:</b> <p>Good morning children!  <i>(Waiting for their response)</i></p> <p>How are you?  <i>(Waiting for their response)</i></p> <p>Today, we are starting a new mini-project! Hurray! Are you excited?</p> <p>Our project is called “Sticker Art” and you will learn how to make images using different stickers and create whatever you want.</p> <p>Does it sound interesting?  <i>(Waiting for their response)</i></p> <p>Alright! So let’s start!</p>					

<b>Activity 2</b> <b>Session 9</b>		<b>Skills:</b> Listening, reading and writing	
<b>Title:</b> Hands on with our project	<b>Type:</b> Reinforcement	<b>Timing:</b> 50 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Mini-project: Sticker Art Activity URL: <a href="https://studio.code.org/s/coursec-2021/lessons/10/levels/1">https://studio.code.org/s/coursec-2021/lessons/10/levels/1</a>	
<b>Aims:</b> <ul style="list-style-type: none"> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>			
<b>Input:</b> <p>Alright! Do you have any doubts? (<i>Waiting for their response</i>)</p> <p>OK! Now, take your tablets and open the link I have published on Teams. To do the project, you will go through several phases to make a final creation. Make sure to read carefully all the statements in each phases. When you finish a phase, raise your hand and I will go to check what you have done. Then, you can continue. If you have any problem, raise your hand too.</p> <p>We start in 3, 2, 1... Go! (<i>Students work on their own</i>)</p> <p>Well done! Let's share your creations! (<i>Students move freely around the class showing their classmates their final products</i>)</p>			

<b>Session 10</b>	
<b>Aims</b> <ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> <li>• Students should be able to use simple coding commands to reach a goal</li> <li>• Students should be able to use loops</li> </ul>	<b>Contents</b> <ul style="list-style-type: none"> <li>• Comprehension of language fragments in form of commands in English</li> <li>• Comprehension of problem statements in English</li> <li>• Basic programming commands</li> </ul>

in their coding sequence to shorten the code	<ul style="list-style-type: none"> <li>Loops in programming</li> <li>Designing settings</li> <li>Event commands</li> </ul>
<b>Assessment criteria</b>	
<ul style="list-style-type: none"> <li>Students will be able to complete a simple coding task on an online platform in English</li> </ul>	
<b>Activities</b>	<b>Timing</b>
Becoming a game designer!	10 minutes
Hands on with our project	45 minutes

<b>Activity 1</b>		<b>Session 10</b>	<b>Skills:</b> Listening and speaking
<b>Title:</b> Becoming a game designer!	<b>Type:</b> Introduction		<b>Timing:</b> 10 minutes
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Video: Code your own Flappy Game URL: <a href="https://youtu.be/VQ4lo6Huylc">https://youtu.be/VQ4lo6Huylc</a>	
<b>Aims:</b>			
<ul style="list-style-type: none"> <li>Students should be able to understand the information presented on a video</li> </ul>			
<b>Input:</b>			
<p>Good morning children! (Waiting for their response)</p> <p>How are you? (Waiting for their response)</p> <p>Today, we are going to create our own game! Wow! How exciting! I am very excited! Are you excited? (Waiting for their response)</p> <p>Alright! This video will show us how! (Plays the video)</p> <p>Alright! What new command are we learning today? (Waiting for their response)</p> <p>Well done! We are learning about events. What is an event in programming? (Waiting for their response)</p>			

Great! An event is a response to an action made by the user of a program!

<b>Activity 2</b> <b>Session 10</b>		<b>Skills:</b> Listening, reading and writing
<b>Title:</b> Hands on with our project	<b>Type:</b> Reinforcement	<b>Timing:</b> 45 minutes
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Build a Flappy Game Activity URL: <a href="https://studio.code.org/s/coursesec-2021/lessons/12/levels/2">https://studio.code.org/s/coursesec-2021/lessons/12/levels/2</a>
<b>Aims:</b>		
<ul style="list-style-type: none"> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>		
<b>Input:</b>		
<p>Alright! Is everything clear? (<i>Waiting for their response</i>)</p> <p>OK! Now, take your tablets and open the link you have on Teams. There are ten activities that you can do. To do them you will need to put into practice what you have learnt on the video, you will use some event commands you will create your own game. You will also use some commands to choose a background for the game. When you finish an activity, raise your hand and I will go to check. Then, you can continue. If you have any problem, raise your hand too.</p> <p>We start in 3, 2, 1... Go! (<i>Students work on their own</i>)</p>		

<b>Session 11</b>	
<b>Aims</b>	<b>Contents</b>
<ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> <li>• Students should be able to use simple coding commands to reach a goal</li> </ul>	<ul style="list-style-type: none"> <li>• Comprehension of language fragments in form of commands in English</li> <li>• Comprehension of problem statements in English</li> <li>• Basic programming commands</li> </ul>

<ul style="list-style-type: none"> <li>Students should be able to use loops in their coding sequence to shorten the code</li> </ul>	<ul style="list-style-type: none"> <li>Loops in programming</li> <li>Designing settings</li> <li>Event commands</li> <li>Pixels as a measure</li> <li>Rotations with degrees</li> </ul>
<b>Assessment criteria</b> <ul style="list-style-type: none"> <li>Students will be able to make their own creations in the foreign language on a coding online platform</li> </ul>	
<b>Activities</b>	<b>Timing</b>
Introduction to Play Lab	10 minutes
Let's create a chase Game	45 minutes

<b>Activity 1</b> <b>Session 11</b>		<b>Skills:</b> Listening and speaking
<b>Title:</b> Introduction to Play Lab	<b>Type:</b> Introduction	<b>Timing:</b> 10 minutes
<b>Classroom arrangement:</b> Students sit individually in their seats	<b>Resources:</b> Video: Introduction to Play Lab URL: <a href="https://youtu.be/FVL6Yo7fOIU">https://youtu.be/FVL6Yo7fOIU</a>	
<b>Aims:</b> <ul style="list-style-type: none"> <li>Students should be able to understand the information presented on a video</li> </ul>		
<b>Input:</b> Good morning children! <i>(Waiting for their response)</i> How are you? <i>(Waiting for their response)</i> Today, we are going to discover a new place to code! It is called Play Lab! Does it sound exciting? <i>(Waiting for their response)</i> Alright! This video will show us how this place works! <i>(Plays the video)</i>		

Great! It is very similar to what we have used before. Tell me some examples of creations that we can make in the Play Lab.

*(Waiting for their response)*

Excellent! There are a lot of possibilities! Do you want to start to create?

*(Waiting for their response)*

<b>Activity 2</b> <b>Session 11</b>		<b>Skills:</b> Listening, reading and writing
<b>Title:</b> Let's create a chase Game	<b>Type:</b> Reinforcement	<b>Timing:</b> 45 minutes
<b>Classroom arrangement:</b> Students sit individually in their seats		<b>Resources:</b> Mini-Project: Chase Game Activity URL: <a href="https://studio.code.org/s/coursec-2021/lessons/13/levels/2">https://studio.code.org/s/coursec-2021/lessons/13/levels/2</a>
<b>Aims:</b>		
<ul style="list-style-type: none"> <li>• Students should be able to use code to create a simple game or conversation</li> </ul>		
<b>Input:</b>		
<p>Great! Now, take your tablets and open the link I have sent you on Teams. There are twelve activities that you can do. To do them you will need to put into practice what has been shown in the video. You can use all kinds of commands we have used other days. When you finish an activity, raise your hand and I will go to check. Then, you can continue. If you have any problem, raise your hand too.</p> <p>We start in 3, 2, 1... Go!</p> <p><i>(Students work on their own)</i></p>		

<b>Session 12</b>	
<b>Aims</b>	<b>Contents</b>
<ul style="list-style-type: none"> <li>• Students should be able to understand the information presented on a video</li> <li>• Students should be able to use simple coding commands to reach a</li> </ul>	<ul style="list-style-type: none"> <li>• Comprehension of language fragments in form of commands in English</li> <li>• Comprehension of problem statements in English</li> </ul>

<p>goal</p> <ul style="list-style-type: none"> <li>Students should be able to use loops in their coding sequence to shorten the code</li> </ul>	<ul style="list-style-type: none"> <li>Basic programming commands</li> <li>Loops in programming</li> <li>Designing settings</li> <li>Event commands</li> <li>Pixels as a measure</li> <li>Rotations with degrees</li> </ul>
<p><b>Assessment criteria</b></p> <ul style="list-style-type: none"> <li>Students will be able to make their own creations in the foreign language on a coding online platform</li> </ul>	
<b>Activities</b>	<b>Timing</b>
Our final project	55 minutes

<b>Activity 1</b>		<b>Session 12</b>		<b>Skills:</b> Listening, speaking, reading and writing
<b>Title:</b> Our final project	<b>Type:</b> Project		<b>Timing:</b> 55 minutes	
<b>Classroom arrangement:</b> Students sit individually in their seats			<b>Resources:</b> Mini-Project: Chase Game Activity URL: <a href="https://studio.code.org/s/coursec-2021/lessons/16/levels/1">https://studio.code.org/s/coursec-2021/lessons/16/levels/1</a>	
<b>Aims:</b> <ul style="list-style-type: none"> <li>Students should be able to use code to create a simple game or conversation</li> </ul>				
<b>Input:</b> Good morning children! <i>(Waiting for their response)</i> How are you? <i>(Waiting for their response)</i> Today, is our last day on our programming project. Today you will go to the link I have sent you on Teams, which will take you to our last activity. What you will have to do in this activity make your own creation. It can be a game or				



a story. You can choose what you prefer.

To do it, you will have an example of what you can do and some instructions to help you. You have all the time of the lesson to do this and then you will send me your projects.

As always, if you have any question, raise your hand and I will go.

Are you ready?

*(Waiting for their response)*

Great! Start!

# CONCLUSIÓN

El pensamiento computacional conforma un valor en alza en el paradigma de la educación, el cual muchos académicos aseguran con seguridad que será considerado como una habilidad cognitiva primordial en la educación del futuro.

No cabe duda de que esta metodología, al igual que todas aquellas que implican el uso de las TICs en el aula, como las actividades gamificadas o la robótica, se encuentran en un estadio evolutivo muy primitivo si lo comparamos con el que estarán dentro de cinco o diez años. Se trata de una corriente educativa innovadora con un gran potencial, la cual está dando sus primeros pasos.

A nivel científico, se ha comprobado que la introducción de la programación informática en el aula reporta grandes beneficios a nivel cognitivo para las habilidades del pensamiento computacional, pensamiento creativo, pensamiento lógico-matemático, metacognición, pensamiento crítico y razonamiento, es necesario un periodo de tiempo más extendido y la realización de proyectos a más largo plazo para verificar el impacto positivo en la adquisición de las lenguas extranjeras, en especial en el caso del idioma inglés, ya que los lenguajes de programación que se utilizan a nivel profesional utilizan como lengua vehicular el inglés.

Debido a la precocidad de esta, todavía quedan muchas propuestas que presenciar al respecto y una gran cantidad de investigación que realizar para discriminar dichas propuestas, estableciendo criterios de calidad y diferenciando las habilidades cognitivas que se pretenden desarrollar.

Tampoco es descabellado considerar que la programación informática pudiese llegar a formar parte de los elementos transversales, o bien que el pensamiento computacional fuese incluido entre las competencias básicas, al igual que lo han logrado otras disciplinas como la educación ambiental. Son las necesidades sociales aquellas que determinan mayormente el rumbo de la educación.

En palabras de Steve Jobs:

“Todo el mundo debería saber programar, porque te enseña a pensar”

# ANEXOS

## Anexo I. Stacking Cups



# BIBLIOGRAFÍA

- Barr, V., & Stephenson, C. (2011). *Bringing computational thinking to K–12: What is involved and what is the role of the computer science education community?* ACM Inroads, 2, pp. 48–54. <http://dx.doi.org/10.1145/1929887.1929905>
- Bransford, J. D., & Schwartz, D. L. (1999). *Rethinking transfer: A simple proposal with multiple implications*. Review of Research in Education, 24, pp. 61–100. <http://dx.doi.org/10.3102/0091732X024001061>
- Brown, J. (2016). *Coding in English class – Perfect pair*. Current and Cool. <http://currentandcool.blogspot.ae/2016/01/coding-in-english-class-perfect-pair.html>
- Century, J., Ferris, y K., Zuo, H. (s. f.) *Finding Time for Computer Science in the Elementary Day*. Time for CS. <https://outlier.uchicago.edu/TimeforCS/findings/>
- Clements, D. H. (1995). *Teaching creativity with computers*. Educational Psychology Review, 7, pp. 141–161. <http://dx.doi.org/10.1007/BF02212491>
- Clements, D. H., y Sarama, J. (1997). *Research on Logo*. Computers in the Schools, 14(1–2), p. 9–46. [http://dx.doi.org/10.1300/J025v14n01\\_02](http://dx.doi.org/10.1300/J025v14n01_02)
- Cohen, A. (2017). *Teaching ESL through coding*. LinkedIn Pulse. <https://www.linkedin.com/pulse/teaching-esl-through-coding-alan-cohen>
- Flórez, F. B., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., y Danies, G. (2017). *Changing a generation's way of thinkin: Teaching computational thinking through programming*. Review of Educational Research, 87, pp. 834-860. <http://dx.doi.org/10.3102/0034654317710096>
- Greiff, S., Wüstenberg, S., Csapó, B., Demetriou, A., Hautamäki, J., Graesser, A. C., y Martin, R. (2014). *Domain-general problem solving skills and education in the 21st century*. Educational Research Review, 13, pp. 74–83. <http://dx.doi.org/10.1016/j.edurev.2014.10.002>

- Grover, S., & Pea, R. (2013). *Computational thinking in K–12: A review of the state of the field*. *Educational Researcher*, 42, pp. 38–43. <http://dx.doi.org/10.3102/0013189X12463051>
- McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). *Debugging: A review of the literature from an educational perspective*. *Computer Science Education*, 18, pp. 67–92. <http://dx.doi.org/10.1080/08993400802114581>
- Morais, B. (2015). *Can an English teacher learn to code?* *New Yorker*. <http://www.newyorker.com/tech/elements/can-an-english-teacher-learn-to-code>
- Moreno-León, J. Y Robles, R. (2015) *Computer programming as an educational tool in the English classroom a preliminary study*. 2015 IEEE Global Engineering Education Conference (EDUCON), 2015, pp. 961-966, doi: <http://dx.doi.org/10.1109/EDUCON.2015.7096089>
- Morrison, N. (2013). *Teach kids how to code and you give them a skill for life*. *Forbes*. <https://www.forbes.com/sites/nickmorrison/2013/12/27/teach-kids-how-to-code-and-you-give-them-a-skill-for-life/>
- Liao, Y.-C. (2000, Junio). *A meta-analysis of computer programming on cognitive outcomes: An updated synthesis*. Association for the Advancement of Computing in Education Canada. <https://www.learntechlib.org/primary/p/16132/>
- Liao, Y.-K. C., y Bright, G. W. (1991). *Effects of computer programming on cognitive outcomes: A meta-analysis*. *Journal of Educational Computing Research*, 7, p. 251–268. <http://dx.doi.org/10.2190/E53G-HH8KAJRR-K69M>
- Lye, S. Y., & Koh, J. H. L. (2014). *Review on teaching and learning of computational thinking through programming: What is next for K-12?* *Computers in Human Behavior*, 41, pp. 51–61. <http://dx.doi.org/10.1016/j.chb.2014.09.012>
- Partovi, H. (2018, Abril 17). *Code.org + resourceful teachers = higher student achievement!* *Code.org* <https://codeorg.medium.com/code-org-resourceful-teachers-higher-student-achievement-8be1efdec06e>

- Perkins, D. N., & Salomon, G. (1992). *Transfer of learning*. T. N. Postlethwaite & T. Husen (Eds.), *International encyclopedia of education*, 2, pp. 6452–6457. Oxford, UK: Pergamon Press.
- Román-González, M., Pérez-González, J.-C., y Jiménez-Fernández, C. (2017). *Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test*. *Computers in Human Behavior*, 72, p. 678–691. <http://dx.doi.org/10.1016/j.chb.2016.08.047>
- Sala, G., y Gobet, F. (2017). *Does far transfer exist? Negative evidence from chess, music, and working memory training*. *Current Directions in Psychological Science*, 26, pp. 515–520. <http://dx.doi.org/10.1177/0963721417712760>
- Scherer, R. (2016). *Learning from the past—The need for empirical evidence on the transfer effects of computer programming skills*. *Frontiers in Psychology*, 7, p. 1390. <http://dx.doi.org/10.3389/fpsyg.2016.01390>
- Scherer, R., Siddiq, F. y Sánchez, B. (2018, Octubre 25). *The Cognitive Benefits of Learning Computer Programming: A Meta-Analysis of Transfer Effects*. *Journal of Educational Psychology*, 111, p. 764-792.
- Schunk, D. H. (2012). *Learning theories: An educational perspective*. Boston, MA: Pearson Education
- Shute, V. J., Sun, C., y Asbell-Clarke, J. (2017). *Demystifying computational thinking*. *Educational Research Review*, 22, pp. 142–158. <http://dx.doi.org/10.1016/j.edurev.2017.09.003>
- Stevens, H. y Verschoor, J. (2017, Agosto). *Coding and English Language Teaching*. *The Electronic Journal for English as a Second Language (TESL-EJ)*, 21. <https://eric.ed.gov/?id=EJ1153835>
- Uttal, D. H., Meadow, N. G., Tipton, E., Hand, L. L., Alden, A. R., Warren, C., & Newcombe, N. S. (2013). *The malleability of spatial skills: A meta-analysis of training studies*. *Psychological Bulletin*, 139, pp. 352–402. <http://dx.doi.org/10.1037/a0028446>

- Weng, K. y Wong, G. K. W. (2017). *Integrating computational thinking into english dialogue learning through graphical programming tool*. IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE), 2017, (pp. 320-325). <http://dx.doi.org/10.1109/TALE.2017.8252356>
- Wing, J. M. (2006). *Computational thinking*. Communications of the ACM, 49, pp. 33–35. <http://dx.doi.org/10.1145/1118178.1118215>
- Woodworth, R. S., & Thorndike, E. L. (1901). *The influence of improvement in one mental function upon the efficiency of other functions I*. Psychological Review, 8, pp. 247–261. <http://dx.doi.org/10.1037/h0074898>
- Yadav, A., Good, J., Voogt, J., y Fisser, P. (2017). *Computational thinking as an emerging competence domain*. Competence based vocational and professional education: Bridging the worlds of work and education (pp. 1051–1067). [http://dx.doi.org/10.1007/978-3-319-41713-4\\_49](http://dx.doi.org/10.1007/978-3-319-41713-4_49)