



---

**Universidad de Valladolid**

**TRABAJO FIN DE MÁSTER**

**MÁSTER EN PROFESOR DE EDUCACIÓN  
SECUNDARIA OBLIGATORIA Y  
BACHILLERATO, FORMACIÓN PROFESIONAL  
Y ENSEÑANZAS DE IDIOMAS**

Especialidad de Tecnología e Informática

**Diseño de actividades con  
Scratch, fomentando el  
pensamiento computacional**

Autor:

**Daniel Merino Sanz**

Tutora:

**Carmen Hernández Díez**

*Valladolid, 17 de Julio de 2017*

## **Dedicatoria**

A los alumnos de Tecnologías de la Información y la Comunicación del I.E.S Zorrilla de Valladolid, con los que tuve la oportunidad de tener mi primera experiencia como docente.

## **Agradecimientos**

A mi tutora del Trabajo Fin de Máster, Carmen Hernández Díaz.

A todos los profesores que imparten docencia en el Máster en profesor de educación secundaria obligatoria y bachillerato, formación profesional y enseñanzas de idiomas.

# Índice

<b>Resumen .....</b>	<b>6</b>
<b>Abstract.....</b>	<b>7</b>
<b>Parte I – Introducción y Contexto .....</b>	<b>8</b>
<b>Capítulo 1. Introducción .....</b>	<b>9</b>
1.1 Motivación.....	9
1.2 Objetivos.....	9
<b>Capítulo 2. Contexto.....</b>	<b>11</b>
<b>Parte II – El pensamiento computacional y como favorecer su desarrollo</b>	<b>13</b>
<b>Capítulo 3. El pensamiento computacional .....</b>	<b>14</b>
3.1 ¿Qué es el pensamiento computacional? .....	14
3.2 El pensamiento computacional en el currículo .....	14
3.3 La informática como ciencia integrada en el aprendizaje de otras asignaturas. .....	16
3.4 ¿Qué metodologías emplear para el desarrollo del pensamiento computacional? .....	19
<b>Parte III – Actividades para el desarrollo del pensamiento computacional .....</b>	<b>22</b>
<b>Capítulo 4. Actividades .....</b>	<b>23</b>
4.1 Moviendo el gato .....	23
4.2 El camino .....	25
4.3 El piano.....	28
4.4 Provincias y comunidades.....	31
4.5 Pacman .....	35
<b>Parte III – Conclusiones .....</b>	<b>40</b>
<b>Conclusiones .....</b>	<b>41</b>
<b>Referencias .....</b>	<b>42</b>
<b>Parte IV – Anexos .....</b>	<b>43</b>
<b>APENDICE A: Contenido del CD .....</b>	<b>44</b>



## Índice de Ilustraciones

Ilustración 1: Entorno de Scratch.....	11
Ilustración 2: Espiral de pensamiento creativo. ....	12
Ilustración 3: Mejora entre la prueba inicial y la final. Los grupos de control en color azul; los grupos experimentales en verde. Extraído de [7].....	17
Ilustración 4: Mejora entre la prueba inicial y la final. Los grupos de control en color azul; los grupos experimentales en verde. Extraído de [7].....	18
Ilustración 5: Mejora entre la prueba inicial y la final. Los grupos de control en color azul; los grupos experimentales en verde. Extraído de [7].....	19
Ilustración 6: Aprendizaje basado en problemas. Extraído de [9].....	20
Ilustración 7: Aprendizaje basado en proyectos. Extraído de [11].....	21
Ilustración 8: Script Actividad 1.....	23
Ilustración 9: Escenario Actividad 1.....	24
Ilustración 10: Escenario Actividad 2.....	26
Ilustración 11: Script Actividad 2.....	26
Ilustración 12: Escenario actividad el piano.....	29
Ilustración 13: Script actividad el piano.....	29
Ilustración 14: Escenario inicial actividad provincias y comunidades.....	32
Ilustración 15: Escenario provincias.....	32
Ilustración 16: Escenario comunidades.....	32
Ilustración 17: Script provincias.....	33
Ilustración 18: Script comunidades.....	33
Ilustración 19: Escenario actividad Pacman.....	35
Ilustración 20: Script Pacman.....	36
Ilustración 21: Script bolas.....	36
Ilustración 22: Script fantasma.....	37

# Resumen

El siguiente trabajo tiene como objetivo exponer una serie de actividades con *Scratch* para alumnos de la E.S.O con las que favorecer el desarrollo del pensamiento computacional desde la educación secundaria. Las actividades que se plantean están orientadas a que los alumnos sean capaces de resolver diferentes problemas utilizando un ordenador, analizando la información aportada para implementar la solución más adecuada según convenga.

# **Abstract**

The following work aims to expose a series of activities with Scratch for students of the US with which to favor the development of computational thinking from secondary education. The activities are aimed at students being able to solve different problems using a computer, analyzing the information provided to implement the most appropriate solution as appropriate.

## **Parte I – Introducción y Contexto**



# Capítulo 1. Introducción

## 1.1 Motivación

Actualmente en el sistema educativo Español, la informática, la programación, y el pensamiento computacional en general han quedado relegados a un segundo plano y no se les da la importancia que debería. En una sociedad en la que las nuevas tecnologías marcan el futuro, es necesario formar a nuestros ciudadanos para que independientemente de su profesión adquieran unas habilidades que los capaciten para ser capaces de vivir en una sociedad tecnológica siendo capaces de resolver problemas utilizando conocimientos algoritmos y matemáticos.

Este documento es el resultado de la asignatura *Trabajo Fin de Master* (TFM en adelante) del Master en Profesor de Educación Secundaria Obligatoria y Bachillerato, Formación Profesional y Enseñanzas de Idiomas, especialidad de Tecnología e Informática de la Facultad de Educación y Trabajo Social de la Universidad de Valladolid.

Este TFM consiste en el análisis y estudio de investigaciones relacionadas con el pensamiento computacional y el posterior desarrollo de una serie de actividades con Scratch para alumnos de la E.S.O con las que se pretende conseguir aportar una base sobre la que trabajen los conceptos básicos del pensamiento computacional y la programación.

Toda la documentación generada, así como el código Scratch desarrollado se pueden encontrar en el CD adjunto, cuya estructura se describe en el **APENDICE A: Contenido del CD**.

## 1.2 Objetivos

La finalidad del TFM debe estar de acuerdo con los objetivos generales del Master de Profesorado de Enseñanza Secundaria, Bachillerato, Formación Profesional y Enseñanzas de Idiomas.

Los objetivos específicos del TFM son:

- Mostrar, mediante una presentación oral pública ante un tribunal, que ha adquirido el conjunto de competencias del Máster y que le capacitan para iniciar su actuación como docente de su especialidad en un centro de educación no universitaria.
- Permitir evaluar los conocimientos y capacidades adquiridos por el alumno teniendo en cuenta el carácter especializado del Máster y su orientación a la especialización profesional.

Específicamente este TFM tiene como objetivos:

- Realizar un estudio en profundidad de las ideas y propuestas que se hacen en el monográfico [5] realizado por la Asociación de Enseñantes Universitarios de la

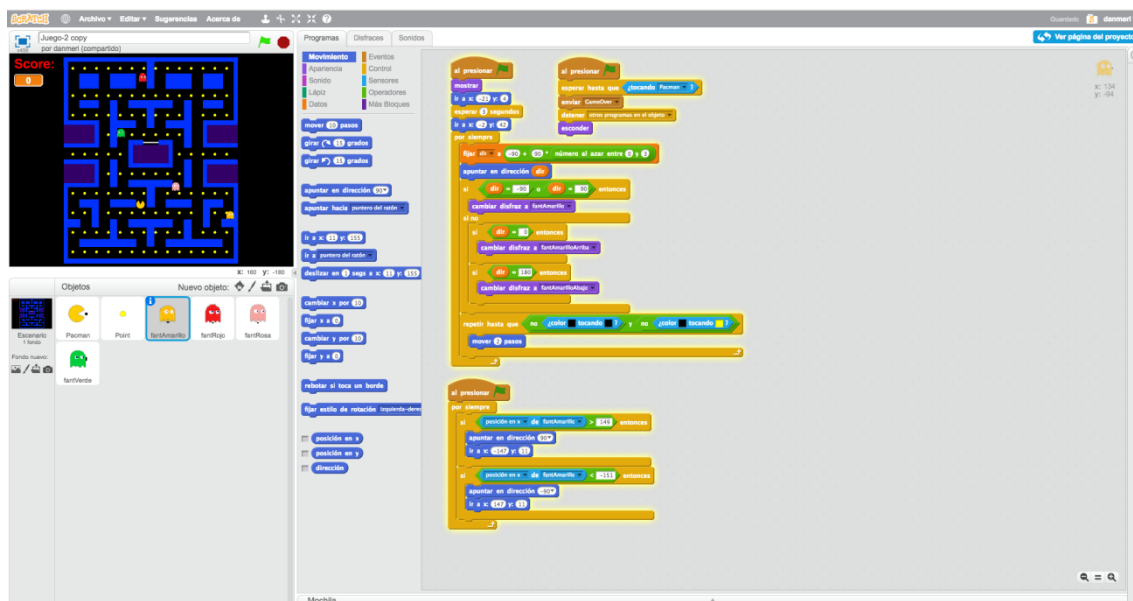
informática, donde se trata el tema de la educación en informática para alumnos menores de 18 años.

- Proponer una serie de actividades que se pueden realizar en cualquier nivel de secundaria, encaminadas a fomentar el pensamiento computacional entre los estudiantes.

## Capítulo 2. Contexto

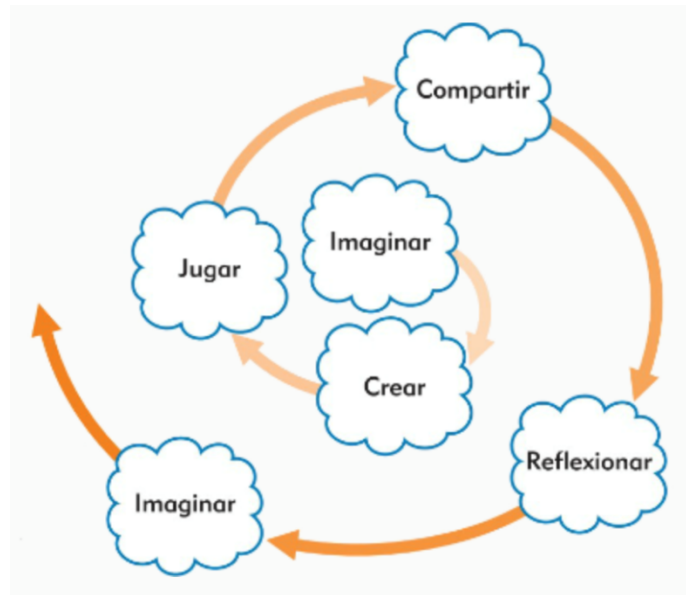
En este TFM se va a detallar qué es el pensamiento computacional y qué factores pueden influir en su desarrollo temprano, además de exponer una serie de actividades para facilitar su desarrollo entre los estudiantes, pero es importante detallar con qué entorno desarrollaremos estas actividades.

Scratch es un lenguaje de programación visual creado por el MIT (Instituto tecnológico de Massachusetts) [12]. Su principal virtud reside en que no es necesario escribir código para hacer nuestros programas, sino que tendremos que ir “encajando” una serie de piezas que representan las instrucciones. Esto hace que sea el lenguaje más adecuado para iniciar en la programación a los jóvenes.



### Ilustración 1: Entorno de Scratch

Mediante su utilización se pretende desarrollar un pensamiento creativo a partir del juego. Pero es muy importante que los profesores que decidan dar el paso de utilizarlo en sus aulas se formen adecuadamente para ello.



**Ilustración 2: Espiral de pensamiento creativo.**

Su instalación es muy sencilla, pero además podremos crear, guardar y compartir nuestros proyectos online. Algunas de las ventajas de este lenguaje de programación son las siguientes:

- Es gratuito
- Es software libre
- Está disponible para varios sistemas operativos.
- Es muy adecuado para personas que deseen iniciarse en la programación.
- Fomenta la educación libre y gratuita.
- Se puede utilizar en multitud de disciplinas, no solo en informática.
- Motiva a los usuarios.

## **Parte II – El pensamiento computacional y cómo favorecer su desarrollo**

## Capítulo 3. El pensamiento computacional

### 3.1 ¿Qué es el pensamiento computacional?

Jeannette Wing, la principal precursora de este concepto, en el artículo *Computational thinking* [1] publicado en Marzo de 2016 define al Pensamiento Computacional *como los procesos de pensamiento implicados en la formulación de problemas y representación de sus soluciones, de manera que dichas soluciones puedan ser efectivamente ejecutadas por un agente de procesamiento de información (humano, computadora o combinaciones de humanos y computadoras)*. Según Wing, esta habilidad será empleada por todas las personas a mediados del siglo XXI.

Por su parte Alfred V. Aho, profesor de computación en la universidad de Columbia, en su artículo *Computation and Computational thinking* [2] define este concepto como *el proceso que permite formular problemas de forma que sus soluciones pueden ser representadas como secuencias de instrucciones y algoritmos*.

La Royal Society, una de las sociedades científicas más antiguas del mundo, en el artículo *Shut down or restart? The way forward for computing in UK schools* [3], entiende el pensamiento computacional como *el proceso de reconocimiento de aspectos de la informática en el mundo que nos rodea, y aplicar herramientas y técnicas de la informática para comprender y razonar sobre los sistemas y procesos tanto naturales como artificiales*.

Para la Sociedad Internacional de la Tecnología en la Educación y la Asociación de Profesores de Informática el pensamiento computacional es un proceso de resolución de problemas con las siguientes características:

- Enunciar problemas de forma que puedan ser resueltos mediante la utilización de un ordenador.
- Organizar y analizar la información.
- Utilización de abstracciones, modelos y simulaciones.
- Elaborar soluciones para un problema mediante el uso de algoritmos (secuencias ordenadas y finitas de instrucciones).
- Analizar las posibles soluciones para aplicar la más eficiente.
- Generalizar soluciones para poder resolver problemas similares.

### 3.2 El pensamiento computacional en el currículo

Actualmente en España el uso de las nuevas tecnologías y la informática está considerado como una mera herramienta de apoyo para el aprendizaje en otras disciplinas, en contadas ocasiones nos encontramos con una consideración más profunda, ya que son muy pocas las comunidades autónomas que incluyen asignaturas que traten a la informática realmente como una ciencia.

Tanto es así que la AENUI (Asociación de Enseñantes Universitarios de la Informática) publicó en Mayo de 2017 un monográfico [5] en el que diversos estudios señalan que el aprendizaje temprano en técnicas como la programación, pueden afectar

significativamente a la adquisición de distintas habilidades, como el pensamiento computacional, pero también puede favorecer una mejor comprensión de conceptos de otras asignaturas como las matemáticas, la física, o aunque parezca difícil la lengua.

Este monográfico se ha convertido en el eje vertebrador de este capítulo, pues considero que la aproximación que se hace en el mismo es muy actual y presenta un panorama de lo que se considera qué es y cómo se fomenta el pensamiento computacional en la educación española muy completo y pertinente.

Hoy en día la programación, una de las disciplinas que más puede favorecer el desarrollo temprano del pensamiento computacional, está incluida en el currículo de la asignatura de Tecnología en la E.S.O de manera muy superficial y de manera algo más profunda en la asignatura de Tecnologías de la Información y la Comunicación en los dos cursos del Bachillerato, siendo estas asignaturas opcionales [4]. Aunque también es cierto que ciertas comunidades autónomas están incluyendo en el currículo asignaturas opcionales como Programación y Robótica donde la formación en estos conceptos es mayor.

La Comisión Europea desarrolló en el año 2013 un marco para el desarrollo y la comprensión de la competencia digital [6], en el que se especifican las siguientes áreas de esta competencia:

- *Información:*
  - Identificar, localizar, recuperar, almacenar, organizar y analizar la información digital.
- *Comunicación:*
  - Comunicarse en entornos digitales.
  - Compartir recursos a través de Internet.
  - Colaboración a través de herramientas digitales.
  - Interactuar y participar en comunidades y redes.
- *Creación de contenidos:*
  - Crear y editar nuevos contenidos (texto, imágenes, video).
  - Integrar y reelaborar conocimiento previo y contenido.
  - Tratar y aplicar la propiedad intelectual, los derechos y licencias.
- *Seguridad:*
  - Protección personal.
  - Protección de datos.
  - Protección de la identidad digital.
  - Las medidas de seguridad, uso seguro y sostenible.
- *Resolución de problemas*
  - Identificar las necesidades y los recursos digitales.
  - Tomar decisiones sobre cuál es la herramienta más apropiada en función del propósito o la necesidad.
  - Solucionar problemas conceptuales a partir de medios digitales.
  - Resolver problemas técnicos.

Atendiendo a las recomendaciones de la Comisión Europea, algunos países europeos como Reino Unido o Eslovaquia ya han incluido una asignatura obligatoria de Informática desde la educación primaria. Otros países han optado por usar las habilidades y los conocimientos informáticos como un instrumento para el aprendizaje de otras asignaturas.

Es evidente que es necesario un replanteamiento del currículo en nuestro país ya que no considerar a la informática como una ciencia puede estar ejerciendo una influencia en la sociedad, que desde mi punto de vista no tiene todavía claro lo que realmente es la informática y las ramas de la misma.

Dentro del monográfico [5] hay una serie de estudios y propuestas relacionadas con el tema que nos ocupa que voy a presentar de manera resumida en las siguientes secciones, porque enmarcan de manera adecuada la propuesta docente que presento en el capítulo 4.

### **3.3 La informática como ciencia integrada en el aprendizaje de otras asignaturas.**

En el artículo *Programar para aprender en Educación Primaria y Secundaria: ¿qué indica la evidencia empírica sobre este enfoque?* [7] se trata la integración de la programación con otras asignaturas y se hace un estudio, que a continuación detallaré, sobre los factores que favorecen esta integración.

Que la programación puede ser una potente herramienta para favorecer el aprendizaje en otras asignaturas no es una idea novedosa, Seymour Papert en su libro *Mindstorms* [8], afirmaba que “al programar el ordenador, el niño adquiere un sentido de maestría frente a uno de los tipos de tecnología más potente y novedosa, y establece un contacto íntimo con algunas de las ideas más profundas de las ciencias, las matemáticas y la construcción de modelos intelectuales”.

Como se menciona en este artículo, esta idea ha sido tratada en multitud de estudios durante el siglo XX, si bien casi todos ellos mantenían esa hipótesis algunos de los mismos no observaron ningún tipo de beneficio.

Las investigaciones más recientes realizadas en nuestro país indican que la transferencia de conocimientos a través del uso de la programación en otras asignaturas depende fundamentalmente de tres factores:

- En función de la asignatura en la que se integra la programación.
- En función de la edad de los alumnos.
- En función de la formación de los docentes.

#### **3.3.1 Transferencia de conocimientos en función de la asignatura**

Según se muestra en [7], a partir de un estudio llevado a cabo durante el curso 2014/2015, se han podido obtener importantes conclusiones con respecto a la influencia que la programación y el pensamiento computacional parecen tener según la asignatura donde estas disciplinas se integran.

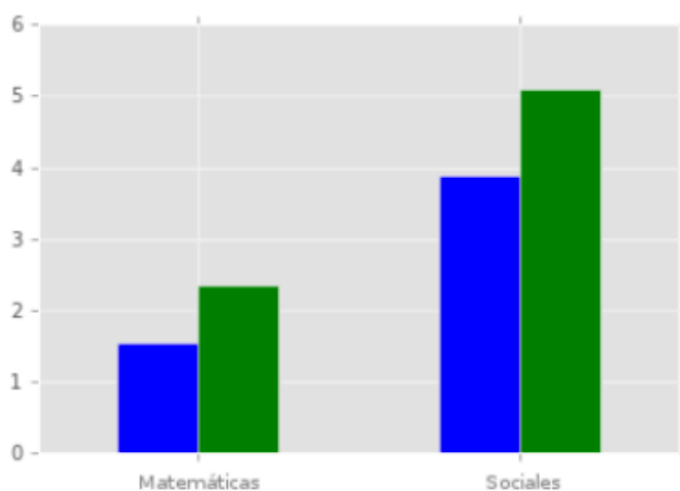
El estudio realizado utilizó dos grupos, uno experimental y otro de control. Al inicio de la investigación, ambos grupos realizaron un cuestionario inicial para valorar los conocimientos iniciales sobre los contenidos que se iban a tratar en la unidad. Posteriormente los alumnos del grupo experimental trabajaron la unidad



mediante ejercicios y actividades basadas en la programación, mientras que en el grupo de control el desarrollo del tema fue llevado a cabo de manera tradicional.

Por último y una vez terminada la unidad, ambos grupos se sometieron a una evaluación final sobre los contenidos desarrollados. Los resultados muestran que la curva de aprendizaje aumentó significativamente en los alumnos que trataron el tema mediante el uso de la programación y el pensamiento computacional.

Hay que señalar que este estudio fue llevado a cabo en dos asignaturas distintas de 6° curso de Primaria, en Matemáticas en una unidad relacionada con los ángulos y en Ciencias Sociales en un tema sobre la Unión Europea. Los resultados señalan que el aceleramiento de la curva de aprendizaje se duplicó en la asignatura de Ciencias Sociales en comparación con las Matemáticas. Esto puede ser debido en gran parte porque la integración de la programación con las matemáticas es bastante más ardua conceptualmente.



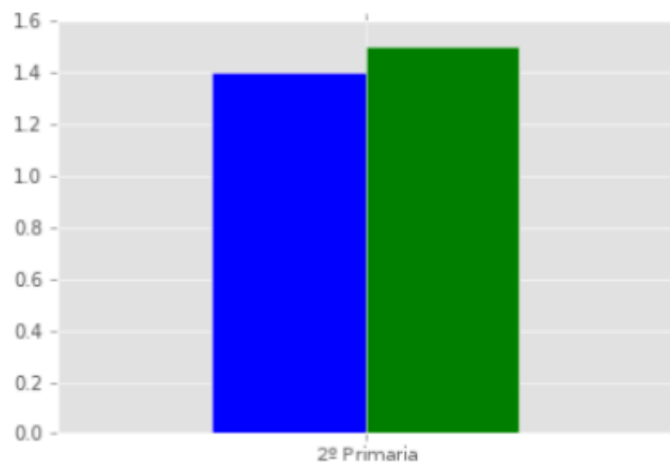
**Ilustración 3: Mejora entre la prueba inicial y la final. Los grupos de control en color azul; los grupos experimentales en verde. Extraído de [7]**

A partir de este estudio, como docentes podemos aplicar mejor un tipo de actividades u otras dependiendo de la asignatura en la que vayan a ser integradas, sabiendo que si esta integración se produce en temas con una dificultad conceptual más elevada el efecto beneficioso del uso de la programación puede ser menor que empleadas en otro tipo de temas.

### **3.3.2 Transferencia de conocimientos en función de la edad de los alumnos**

El mismo estudio comentado en el apartado anterior fue realizado en otro centro educativo, en 2° curso de primaria en la asignatura de Lengua, en un tema en el que se trataba la estructura narrativa. A diferencia de lo ocurrido cuando se trabaja con alumnos de 6° de primaria donde la curva de aprendizaje se vio aumentada de manera importante, con estos alumnos de inferiores edades no se encontraron diferencias de aprendizaje significativas entre el grupo de control y el experimental.

Estos resultados se ven respaldados por otros estudios que muestran que el poder de transferencia del aprendizaje mediante el uso de la programación no tiene un efecto realmente importante hasta los últimos cursos de la educación primaria.



**Ilustración 4: Mejora entre la prueba inicial y la final. Los grupos de control en color azul; los grupos experimentales en verde. Extraído de [7]**

Es importante señalar que aunque el uso de la programación en otras asignaturas pueda no tener un efecto directo en el aprendizaje de los alumnos, el uso de la misma a edades tempranas puede favorecer el desarrollo de capacidades como el pensamiento computacional con mayor facilidad, sin repercutir negativamente en sus resultados académicos.

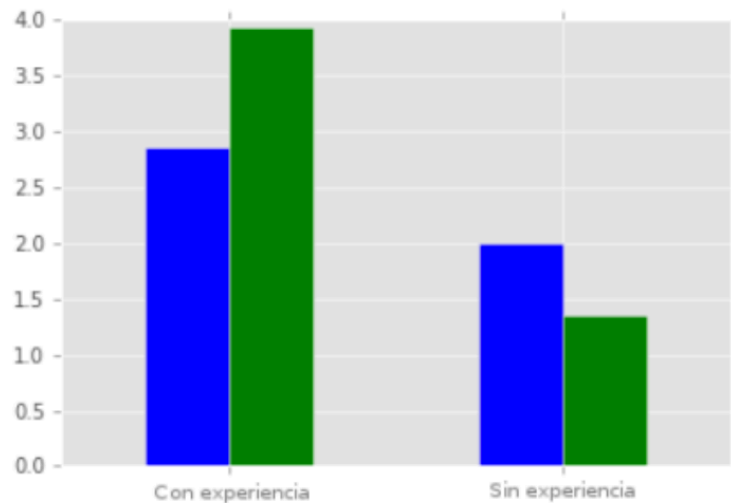
### **3.3.3 Transferencia de conocimientos en función de la formación de los docentes**

Es evidente que la formación del profesorado en aspectos como la programación y el pensamiento computacional es fundamental para que estos puedan desarrollar y fomentar un aprendizaje significativo en sus alumnos. No basta con realizar y supervisar actividades en las que se requiere el uso de la programación, sino que es necesario que los docentes estén adecuadamente formados en relación con estos conceptos, contando con unas bases sólidas que puedan ayudar a los alumnos en la comprensión de los distintos temas.

Una investigación similar a las anteriores fue realizada en grupos del segundo ciclo de educación primaria en clase de Inglés. En este estudio participaron dos profesores, uno que tenía experiencia en programación y otro que no, pero que recibió una formación previa de unas horas.

Los resultados muestran que los alumnos que siguieron el desarrollo de la unidad con el docente que tenía experiencia y trabajaron con actividades en Scratch, obtuvieron unos resultados mucho mejores que los alumnos de control. Por otra parte, los alumnos que realizaron las mismas actividades en Scratch con

el profesor sin experiencia obtuvieron peores resultados que en el grupo de control.



**Ilustración 5: Mejora entre la prueba inicial y la final. Los grupos de control en color azul; los grupos experimentales en verde. Extraído de [7]**

A la vista de estos resultados se puede observar la necesidad de formación del profesorado, ya que de no ser de esta manera se puede producir un efecto contrario al deseado, entorpeciendo el aprendizaje del alumnado. No es conveniente que el sistema educativo dé el paso de intentar formar a los alumnos en el pensamiento computacional sin antes asegurar una correcta formación de los docentes. Con respecto a esta formación, es vital señalar que no solo los profesores del área de Informática deberían estar formados, sino que debería ser indispensable para cualquier docente contar con unas competencias básicas en pensamiento computacional.

Por otra parte, y pensando ahora en la formación no reglada, se han llevado a cabo proyectos [10] en nuestro país en los que estudiantes de Ingeniería Informática, que cuentan con una sobrada preparación sobre pensamiento computacional y programación, se convertían en maestros de alumnos de educación secundaria, fomentando así un aprendizaje en cascada. Esto está basado en la metodología *aprender enseñando*: los alumnos que cursaban estudios superiores se encargaban de realizar actividades para proponer a estudiantes de niveles inferiores y luego serían ellos mismos quienes las explicarían y supervisarían.

Los resultados fueron bastante satisfactorios y casi todos los alumnos dijeron haber sentido que la informática les gustaba.

### **3.4 ¿Qué metodologías emplear para el desarrollo del pensamiento computacional?**

Una vez comprendida la importancia del pensamiento computacional y su interés para el alumnado es importante tratar con detenimiento la forma en el que estos conceptos van a ser transmitidos a los estudiantes. Si queremos despertar el interés de los alumnos

por la programación y la informática en general, deberemos conocer cuáles son las técnicas que pueden aportar un aprendizaje significativo y despertar curiosidad por el tema en los estudiantes.

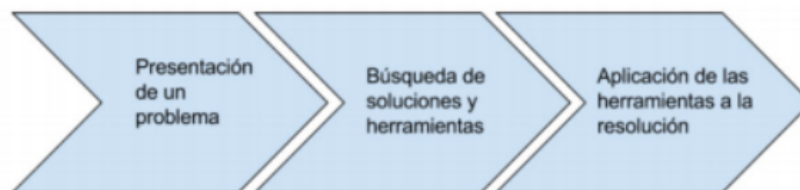
A raíz de los artículos publicados en el monográfico [5] dos de las metodologías que pueden favorecer este tipo de aprendizaje son:

#### **3.4.1 Aprendizaje basado en problemas (APL)**

En este método de enseñanza, el punto de partida lo establece un problema planteado por el profesor, que los alumnos deben resolver para desarrollar algunas competencias anteriormente especificadas.

El proceso se compone de cuatro etapas claramente diferenciadas:

1. El profesor presenta el problema y forma grupos de trabajo con distintos roles.
2. Los alumnos deberán saber de qué conocimientos carecen para poder afrontar la resolución del problema.
3. Los estudiantes deberán buscar información para completar sus conocimientos.
4. Los alumnos resuelven el problema y presentan la solución.



**Ilustración 6: Aprendizaje basado en problemas. Extraído de [9]**

Algunos estudios señalan que este método de enseñanza tiene múltiples virtudes para la enseñanza de la programación, entre ellas destacan:

- Autodescubrimiento: algunos conceptos sencillos surgen de manera natural con una buena aplicación de esta técnica y debidamente guiados por el profesor.
- Se fomenta el trabajo grupal.
- Se trabajan competencias complejas relacionadas a la resolución de problemas y la toma de decisiones.

#### **3.4.2 Aprendizaje basado en proyectos**

Los alumnos desarrollan un proyecto en un tiempo previamente establecido para resolver un problema mediante la planificación, diseño y realización de una serie de actividades. Esta forma de trabajo, sitúa a los alumnos en el centro del proceso de aprendizaje gracias a un planteamiento mucho más motivador en el que es necesario el intercambio de ideas, la creatividad y la colaboración.

Los proyectos deben tratar sobre temas reales, donde las posibles soluciones no estén cerradas y existan multitud de posibilidades para las mismas.

1. Información: los alumnos deberán recopilar toda la información necesaria para la resolución de la tarea.
2. Planificación: elaboración del plan de trabajo, elección de las posibles estrategias de resolución a utilizar, planificación de los instrumentos y herramientas necesarias.
3. Realización: acción experimental e investigadora, ejercitándose la creatividad, autonomía y responsabilidad.
4. Evaluación: los estudiantes presentan los resultados obtenidos y los discuten con el profesor.

- Presentar el proyecto.
- Aportar indicaciones básicas sobre la metodología.
- Revisar el plan de trabajo de todos los grupos.
- Reunirse con los grupos para orientar sobre el avance del proyecto.
- Realizar la evaluación teniendo en cuenta los resultados presentados.

- Formar los grupos.
- Preguntar al profesor ante posibles dudas.
- Definir el plan de trabajo.
- Buscar y recoger información, proponer soluciones, etc.
- Desarrollar el proyecto.
- Presentar los resultados obtenidos y los aprendizajes logrados.



## **Parte III – Actividades para el desarrollo del pensamiento computacional**

## Capítulo 4. Actividades

A continuación, voy a proponer diversas actividades sencillas para su utilización en la educación secundaria. Hay que tener en cuenta que en todas ellas, es necesaria una correcta formación del profesorado que las va a dirigir, pues como hemos visto en el apartado anterior, de no ser así, se podría conseguir el efecto contrario al deseado, dificultando el aprendizaje de los alumnos.

### 4.1 Moviendo el gato

El objetivo de esta actividad es que los alumnos se familiaricen con el entorno de desarrollo de Scratch. En este ejercicio los alumnos trabajarán con los bloques de movimiento, apariencia y control.

#### Nombre de la actividad

Moviendo el gato

#### Descripción

Los alumnos deberán crear un proyecto Scratch con las siguientes características:

- El fondo debe ser distinto al original.
- El gato debe en todo momento situarse en la posición del ratón.
- Cuando el gato toque uno de los bordes de la pantalla deberá mostrar un bocadillo de texto.

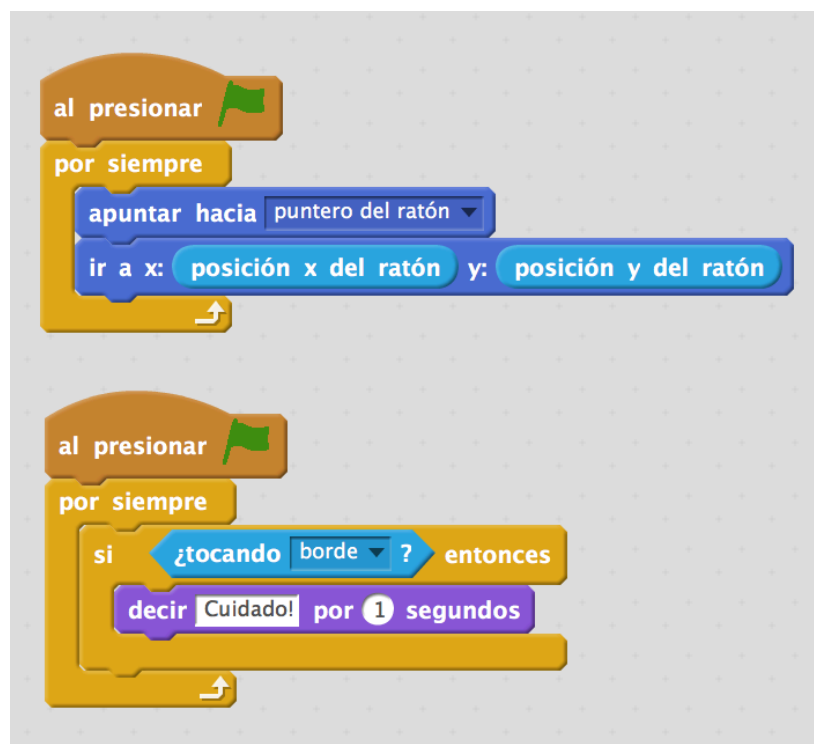


Ilustración 8: Script Actividad 1



**Ilustración 9: Escenario Actividad 1**

### **¿A quiénes va dirigida?**

Teniendo en cuenta la sencillez de este ejercicio, podemos considerar esta actividad muy útil para su utilización con alumnos que no tengan ningún tipo de experiencia con Scratch, ya sea en educación primaria o secundaria.

### **Contexto**

Esta actividad podría emplearse en las asignaturas que incluyen en su currículo aspectos relacionados con la programación como Tecnología o Tecnologías de la Información y la Comunicación, pero también se podría emplear en Matemáticas para abordar la comprensión de las coordenadas cartesianas.

Como hemos visto en el punto anterior, la integración de la programación y las matemáticas no produce un aumento muy significativo en la curva de aprendizaje de los alumnos, pero tratándose de conceptos de poca complejidad, como pueden ser las coordenadas, esta actividad podría ser muy adecuada para favorecer la motivación por el tema.

### **Duración**

Se trata de uno de los primeros ejercicios, pero su realización es muy sencilla por lo que los alumnos dispondrán de un máximo de veinte minutos para su realización.

### **Objetivos**

- Comprender el funcionamiento y los principales secciones de la ventana de Scratch (área de programas, escenario, lista de objetos y escenarios y la paleta de bloques).
- Utilizar los bloques más sencillos de Movimiento, Control, Apariencia y Eventos.
- Comprender de forma práctica las coordenadas cartesianas.
- Motivar a los alumnos para que desarrollen interés por este lenguaje de programación.



### **Desarrollo**

- El profesor realizará una breve introducción sobre Scratch y su funcionamiento básico, introducirá el concepto de bloque y realizará un pequeño ejemplo.
- El profesor planteará el ejercicio y los resultados esperados.
- Los alumnos se distribuirán individualmente y deberán resolver el problema.
- El profesor dará paso a un turno de preguntas y resolverá las dudas que hayan podido surgir durante el desarrollo de la actividad.

### **Competencias trabajadas**

- Competencia matemática y competencias básicas en ciencia y tecnología.
- Competencia para aprender a aprender.
- Competencia digital.

### **Evaluación**

Esta actividad no será evaluada puesto que se trata de una actividad de introducción a la materia. Si se desea aportar un *feedback* al alumno se utilizarán los siguientes criterios:

- Utilización de los bloques más adecuados al enunciado.
- Correcto funcionamiento del programa.

### **Técnica didáctica**

- Trabajo individual: Esta actividad se realizará preferentemente individualmente pero si los recursos disponibles (ordenadores) no lo permitieran se podría realizar por parejas.

### **Documentación didáctica**

- Permite un primer acercamiento de los alumnos con el lenguaje de programación Scratch.
- Recomendación: Si se realiza por parejas se deberá prestar especial atención para que los dos miembros de la pareja trabajen equitativamente y ambos adquieran los conceptos básicos que esta actividad pretende introducir.
- No es necesario que los alumnos tengan conocimientos previos en programación, bastará con una breve explicación introductoria para que los estudiantes puedan realizar la actividad sin ningún problema.

## **4.2 El camino**

El objetivo de esta actividad es que los alumnos afiancen los conceptos básicos de Scratch y profundicen en el uso de bloques y estructuras más avanzadas.

### **Nombre de la actividad**

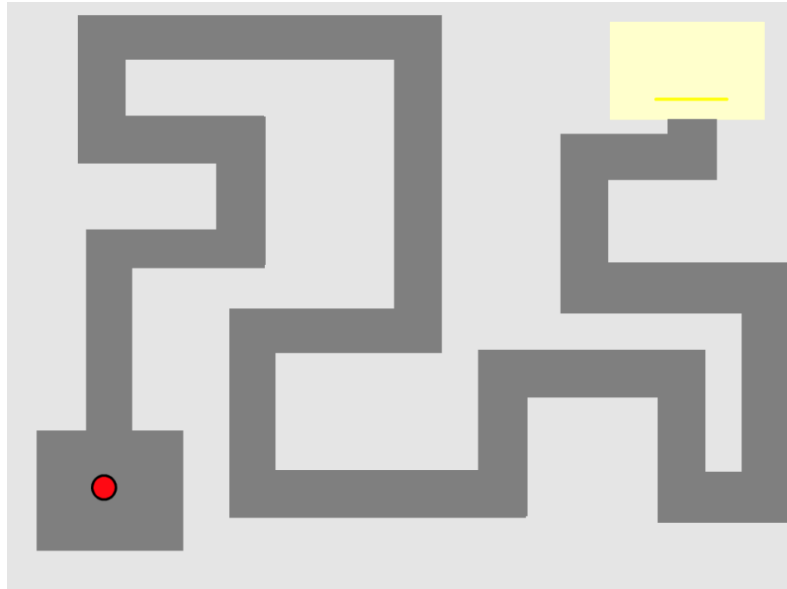
El camino

### **Descripción**

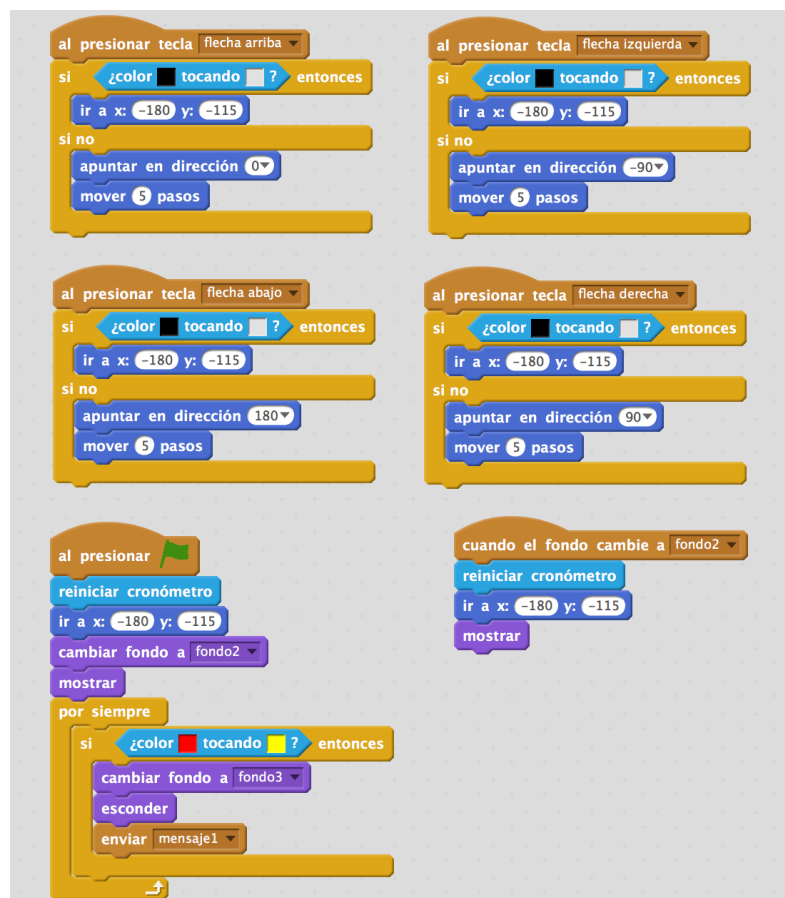
Los alumnos deberán crear un proyecto Scratch con las siguientes características:

- Deben utilizar el fondo llamado *camino.png* aportado por el profesor, que contiene un pequeño camino que deberá ser recorrido por una pequeña bola que deberán crear los alumnos.
- La bola partirá inicialmente de la casilla de salida.

- El objetivo del juego es que la bola alcance la casilla final sin salirse del camino marcado por el fondo.
- Si la bola se sale del camino debe volver a la posición inicial.
- La bola puede desplazarse hacia arriba, hacia abajo y de derecha a izquierda.
- Al lograr que la bola llegue a su destino se debe mostrar una pantalla de finalización.



**Ilustración 10: Escenario Actividad 2**



**Ilustración 11: Script Actividad 2**

### **¿A quiénes va dirigida?**

Al tratarse de una actividad sencilla, la podríamos aplicar en cualquier nivel educativo, siempre teniendo en cuenta que los alumnos deberán tener un mínimo conocimiento sobre Scratch, siendo recomendable que estén familiarizados con el entorno.

### **Contexto**

Al igual que la actividad anterior podríamos utilizar esta actividad en asignaturas específicas de informática o tecnología, pero teniendo en cuenta que para su resolución se necesitan instrucciones que utilizan ángulos y coordenadas cartesianas, también podría ser empleada en la asignatura de Matemáticas.

### **Duración**

La realización de esta actividad tomará algo más de tiempo a los estudiantes ya que su dificultad es algo mayor que la anterior. La duración estimada de esta actividad será por tanto de dos horas.

### **Objetivos**

- Afianzar los conocimientos adquiridos en la anterior actividad sobre los bloques de movimiento, control, apariencia y eventos.
- Introducir a los alumnos en el uso de sensores y eventos.
- Desarrollar un pequeño juego con funcionalidad completa.
- Compresión de las coordenadas cartesianas.
- Compresión y utilización práctica de ángulos.

### **Desarrollo**

- El profesor realizará una breve explicación sobre los nuevos conceptos que se necesitan para resolver el problema de manera adecuada.
- El profesor planteará el ejercicio y los resultados esperados.
- Los alumnos se distribuirán en grupos y deberán resolver el problema.
- El profesor dará paso a un turno de preguntas y resolverá las dudas que hayan podido surgir durante el desarrollo de la actividad.

### **Competencias trabajadas**

- Competencia matemática y competencias básicas en ciencia y tecnología.
- Competencia para aprender a aprender.
- Competencia digital.
- Competencias sociales y cívicas.

### **Evaluación**

Para evaluar esta actividad se utilizarán los siguientes criterios:

- El juego debe funcionar de manera adecuada.
- Los bloques utilizados deberán ser los más adecuados para resolver el problema.
- Se valorarán funciones adicionales introducidas por el alumno.

### **Técnica didáctica**

- Trabajo individual
- Aprendizaje basado en problemas.

### Documentación didáctica

- Actividad muy útil para alumnos con un mínimo conocimiento del funcionamiento de Scratch.
- Es recomendable dejar libertad a los alumnos para que añadan funcionalidad al juego.

## 4.3 El piano

El objetivo de esta actividad es que los alumnos se familiaricen con todas las notas de un piano y sean capaces de desarrollar un programa Scratch utilizando números aleatorios y sonidos.

### Nombre de la actividad

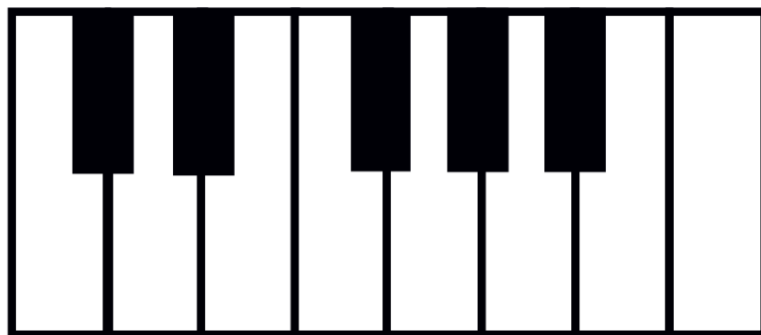
El piano

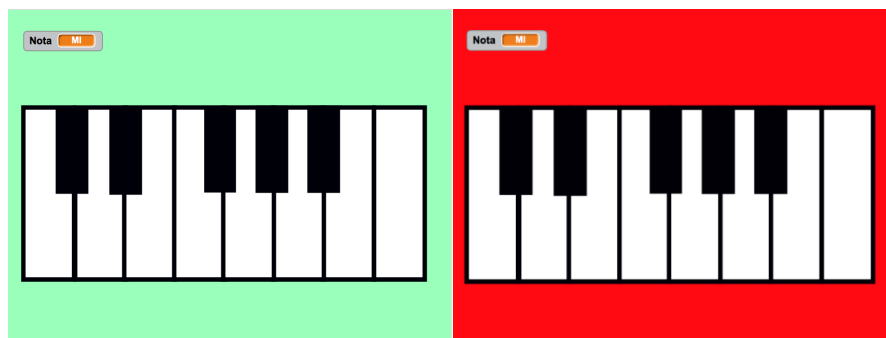
### Descripción

Los alumnos deberán crear un proyecto Scratch con las siguientes características:

- La pantalla del programa deberá contener el teclado de un piano con al menos una escala completa.
- Continuamente se deberá mostrar en la pantalla el nombre de la nota que tiene que ser tocada a continuación, siendo esta nota generada aleatoriamente.
- Si la nota tocada corresponde con la nota que se muestra en pantalla, el fondo deberá cambiar a color verde durante un periodo corto de tiempo.
- Si la nota es incorrecta el fondo deberá ponerse en color rojo hasta esta se pulse correctamente.

Nota SOL





**Ilustración 12: Escenario actividad el piano**



**Ilustración 13: Script actividad el piano**

### ¿A quiénes va dirigida?

Sería recomendable que los alumnos ya hubieran realizado previamente alguna otra actividad con el lenguaje Scratch y que cuenten con algunas nociones musicales básicas.

### Contexto

Esta actividad está especialmente recomendada para su utilización en la asignatura de música, siempre y cuando el profesor que la imparta cuente con los conocimientos necesarios en Scratch.

En el apartado anterior vimos como la integración de la programación con asignaturas en las que los conceptos no sean tan arduos puede ser muy adecuada, y además es en este tipo de asignaturas donde la curva de aprendizaje se ve claramente aumentada con respecto a los alumnos que siguen el tema mediante el uso de las metodologías tradicionales.

### **Duración**

La realización de esta actividad conllevará aproximadamente unas 5 horas, teniendo en cuenta que la metodología que se utilizará será la del aprendizaje basado en problemas, donde los alumnos deberán buscar ellos mismos la solución al problema con el apoyo del profesor, siendo la función de este orientada a guiar el aprendizaje.

### **Objetivos**

- Conocer la disposición tonal de un piano.
- Introducir el concepto de número aleatorio.
- Introducir el concepto de sentencia condicional.

### **Desarrollo**

- El profesor explicará cual es el objetivo de la actividad y que funcionalidad debe tener el proyecto Scratch que los alumnos deben realizar y además formará los grupos de trabajo.
- Los alumnos en sus respectivos grupos, deberán en una hoja plantear cual son los conocimientos que ellos creen que deben conocer para poder resolver el problema.
- Los alumnos buscarán información sobre aquellos conceptos que no conocen y que necesitan para resolver el problema.
- Por último, cada uno de los grupos deberá, una vez resuelto el problema, presentar la solución ante sus compañeros para después abrir un breve debate sobre la solución empleada y las posibles mejoras.

\* Durante el desarrollo de la actividad la función del docente será:

- Hacer un seguimiento del trabajo en grupo.
- Evaluar el progreso de cada uno de los grupos.
- Organizar las presentaciones de las actividades desarrolladas y moderar el posterior debate.

### **Competencias trabajadas**

- Competencia matemática y competencias básicas en ciencia y tecnología.
- Competencia para aprender a aprender.
- Competencia en comunicación lingüística.
- Sentido de la iniciativa y espíritu emprendedor.
- Competencia digital.
- Competencias sociales y cívicas.

### **Evaluación**

Para evaluar esta actividad se utilizarán los siguientes criterios:

- El piano emite el sonido de todas las notas de manera adecuada.
- Las notas generadas son aleatorias.
- El programa muestra el fondo verde al tocar la nota correcta.
- El programa muestra el fondo rojo al tocar una nota incorrecta.

- La exposición muestra de manera correcta el proceso de resolución de la actividad.

#### **Técnica didáctica**

- Aprendizaje basado en problemas.

#### **Documentación didáctica**

- Es recomendable dejar libertad a los alumnos para que añadan funcionalidad al juego.
- Los roles que desempeñaran los alumnos dentro de cada grupo deben ser explicados con anterioridad a la formación de los mismos.
- Para una correcta evaluación es recomendable seguir el progreso de los grupos en intervalos regulares de tiempo.

### **4.4 Provincias y comunidades**

El objetivo de esta actividad es ayudar a los alumnos en el aprendizaje de las provincias y comunidades autónomas españolas. Además su desarrollo contribuye a afianzar conceptos básicos en Scratch como son los disfraces, los fondos, las variables y las instrucciones de control.

#### **Nombre de la actividad**

Provincias y comunidades

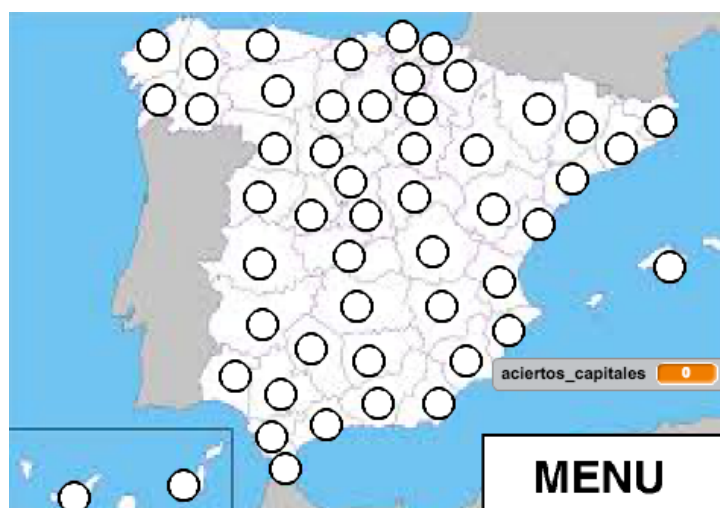
#### **Descripción**

Los alumnos deberán crear un proyecto Scratch con las siguientes características:

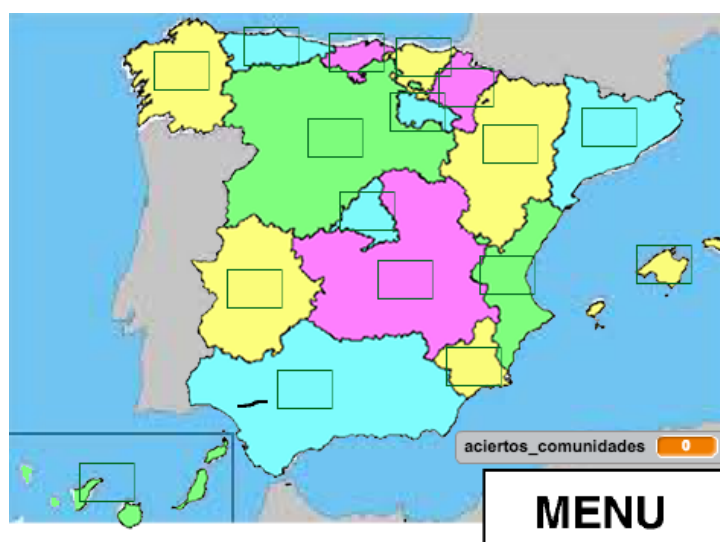
- La pantalla inicial deberá mostrar dos botones, “Comunidades” y “Provincias” para elegir en qué modo jugar.
- Si se elige el modo “Provincias” el fondo debe ser un mapa de España dividido por provincias con un botón en cada una de ellas, de tal forma que cuando este sea pulsado, pregunte el nombre de la provincia.
- Si se elige el modo “Comunidades” el fondo será un mapa de España dividido por comunidades autónomas con un botón en cada una de ellas y al igual que en el caso anterior, cuando este sea pulsado, se deberá preguntar el nombre de la comunidad.
- En ambos modos, cuando se acierte el nombre de la comunidad o la provincia el botón deberá establecerse de color verde.
- Cuando se falle el nombre de la comunidad o provincia el botón deberá ponerse rojo.
- Es necesario que el programa incorpore un contador de aciertos para provincias y comunidades.



**Ilustración 14: Escenario inicial actividad provincias y comunidades.**



**Ilustración 15: Escenario provincias.**



**Ilustración 16: Escenario comunidades**



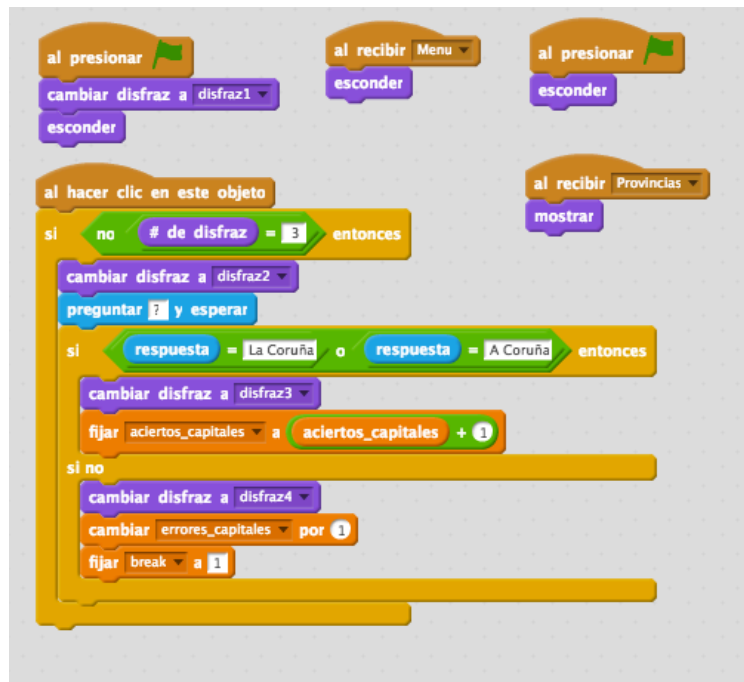


Ilustración 17: Script provincias

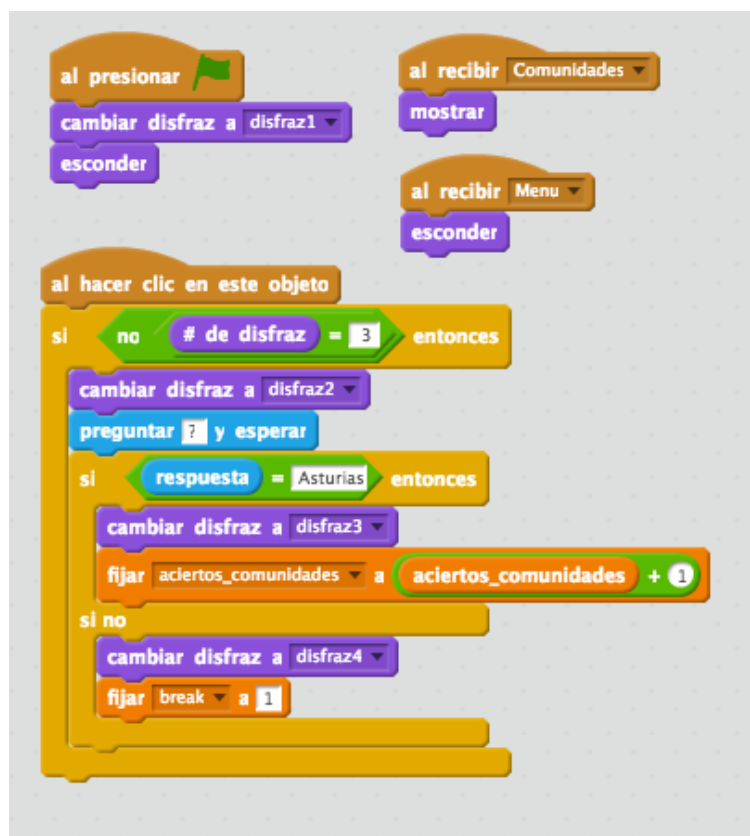


Ilustración 18: Script comunidades

### **¿A quiénes va dirigida?**

Es una actividad apropiada para alumnos de secundaria con algún mínimo conocimiento de Scratch. También podría ser aplicada en niveles educativos inferiores siempre y cuando el docente que la lleve a cabo cuente con una formación adecuada.

### **Contexto**

Debido a la temática de la misma, esta actividad está especialmente pensada para la asignatura de Geografía e Historia, ya que, como vimos en el capítulo 3, es en este tipo de asignaturas donde se consigue un aumento significativo de la curva de aprendizaje al utilizar la programación.

### **Duración**

La duración estimada de esta actividad será de 3 horas.

### **Objetivos**

- Aprender o repasar las provincias y comunidades autónomas españolas.
- Utilizar bloques de control, apariencia y eventos.

### **Desarrollo**

- El profesor expondrá los objetivos de la actividad, cuáles son los resultados esperados y formará los grupos de trabajo.
- Los alumnos deberán estudiar el problema y analizar los conocimientos que les faltan para poder llevar a cabo la solución.
- Los alumnos buscarán información sobre esos conceptos nuevos que necesitan conocer.
- Para finalizar, cada grupo, una vez resuelto el problema deberá presentar la solución ante sus compañeros, dando paso después a un breve debate sobre las posibles mejoras o alternativas, moderado por el docente.

\*El profesor deberá en todo momento hacer un correcto seguimiento del avance de los grupos.

### **Competencias trabajadas**

- Competencia matemática y competencias básicas en ciencia y tecnología.
- Competencia para aprender a aprender.
- Competencia en comunicación lingüística.
- Competencia digital.
- Sentido de la iniciativa y espíritu emprendedor.
- Competencias sociales y cívicas.

### **Evaluación**

Para evaluar esta actividad se utilizarán los siguientes criterios:

- Los mapas deben contar con un botón para todas las provincias y comunidades.
- Cuando una provincia o comunidad se acierta no debe poder volverse a pulsar.
- Se valorarán funciones adicionales introducidas por el alumno.

### **Técnica didáctica**

- Aprendizaje basado en problemas.

### Documentación didáctica

- Es imprescindible que el docente que vaya a llevar a cabo esta actividad cuente con los conocimientos necesarios sobre computación, programación y el lenguaje de programación Scratch.

## 4.5 Pacman

El objetivo de esta actividad es que los alumnos desarrollen un juego completo, donde se usen casi todos los bloques de Scratch, como son los de control, eventos, sensores, datos, apariencia y movimiento.

### Nombre de la actividad

Pacman

### Descripción

Los alumnos deberán crear un proyecto Scratch con las siguientes características:

- El escenario debe ser el laberinto típico del juego Pacman.
- El personaje principal debe ser un círculo amarillo.
- Debe haber cuatro fantasmas repartidos por el laberinto, que deberán moverse de manera aleatoria (No hay que hacer algoritmos específicos para recorrer todo el mapa).
- Si un fantasma toca al personaje principal el juego debe acabar.
- El laberinto debe estar lleno de pequeñas bolas amarillas que el personaje principal debe irse comiendo.
- Es necesario mostrar la puntuación en todo momento, teniendo en cuenta que cada pequeña bolita comida serán 10 puntos.

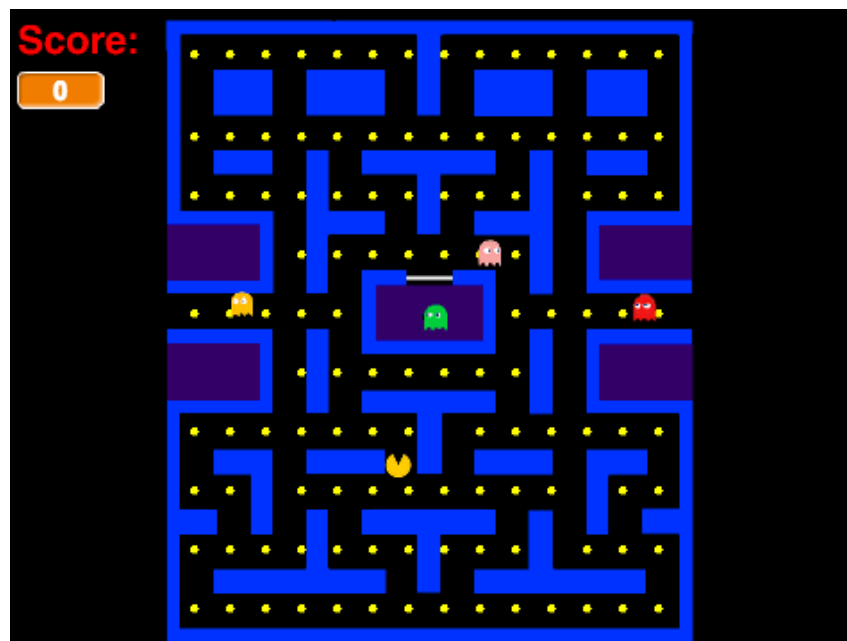


Ilustración 19: Escenario actividad Pacman

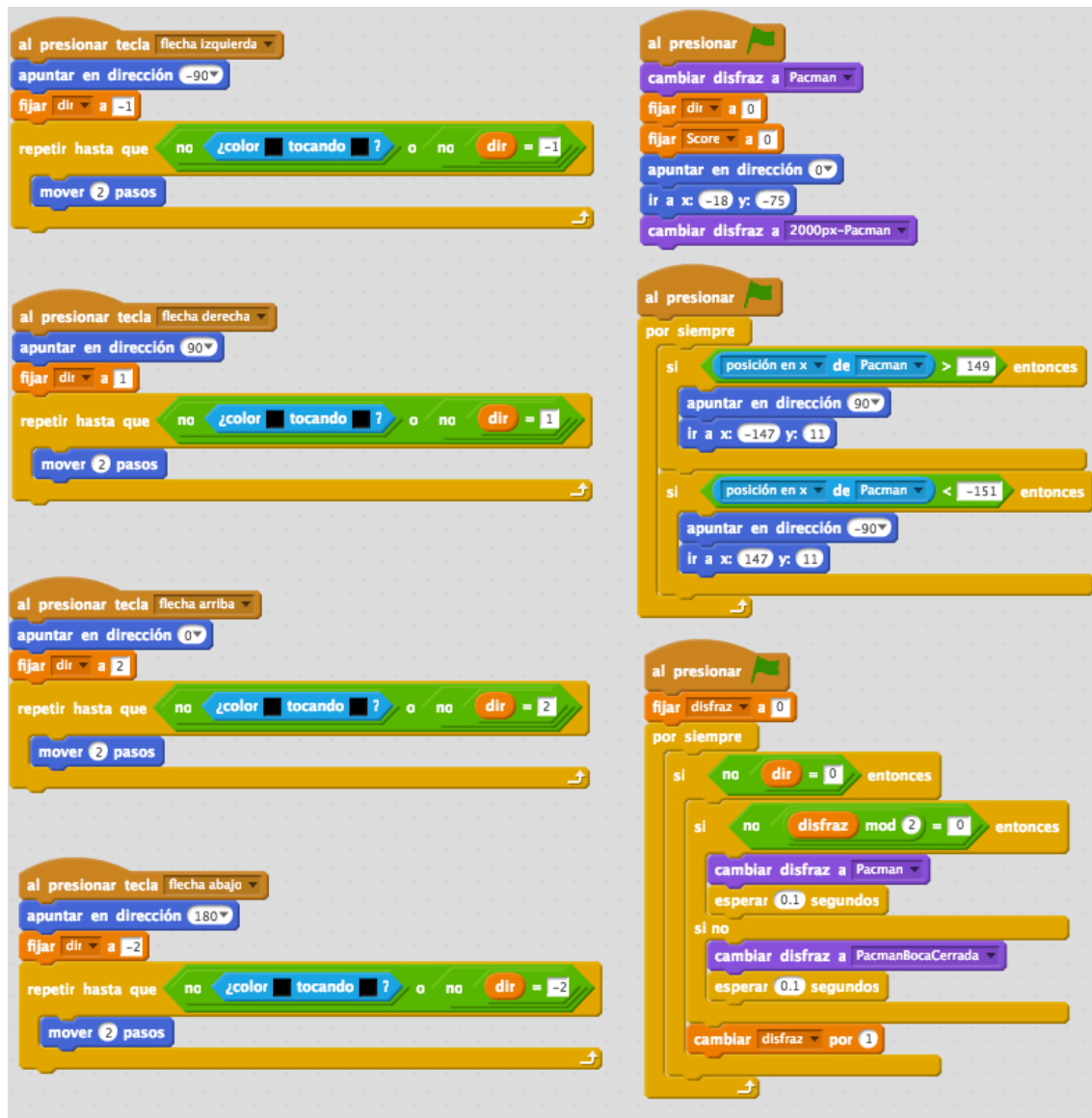


Ilustración 20: Script Pacman

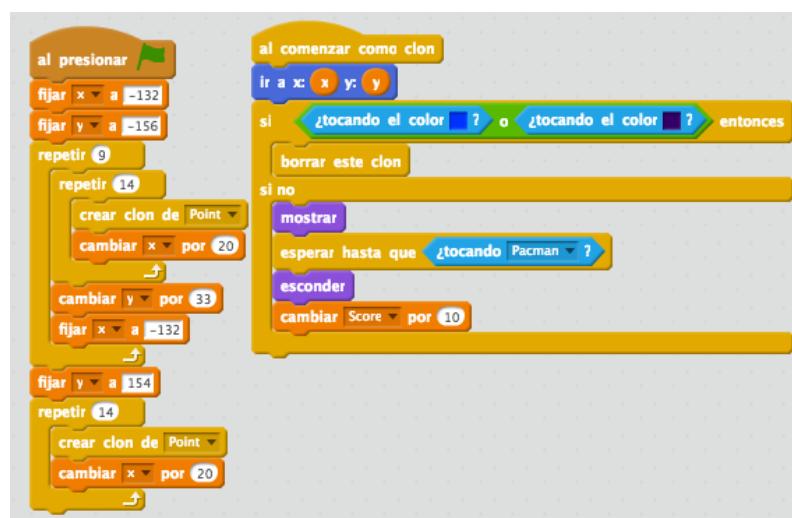


Ilustración 21: Script bolas

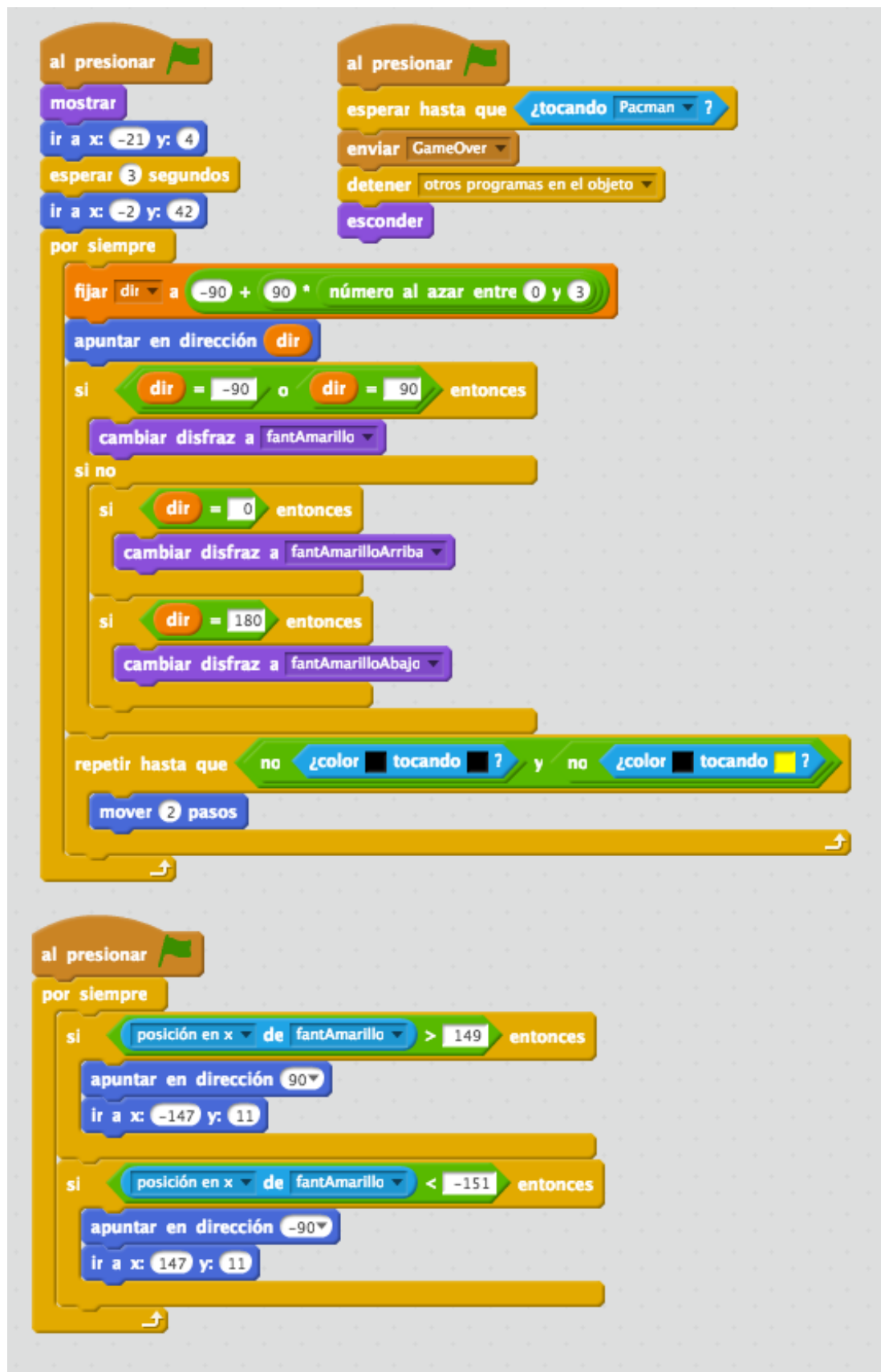


Ilustración 22: Script fantasma

### **¿A quiénes va dirigida?**

Esta es una actividad con un mayor nivel de dificultad, pero es perfectamente adecuada para alumnos de secundaria que ya hayan trabajado mínimamente con algunos conceptos básicos como las iteraciones, las instrucciones de control, etc.

### **Contexto**

Esta actividad no está pensada para ser integrada en asignaturas no tecnológicas, por lo que su desarrollo debería ser llevado a cabo en la asignatura de Tecnología o Programación y Robótica en la E.S.O y Tecnologías de la Información y la Comunicación en Bachillerato.

### **Duración**

Debido a su mayor complejidad esta actividad requerirá un tiempo estimado de 15 horas.

### **Objetivos**

- Utilizar bloques de control, apariencia y eventos.
- Utilizar sensores.
- Afianzar los conceptos básicos sobre algoritmos.
- Desarrollar un juego funcional completo.

### **Desarrollo**

- El profesor expondrá brevemente el funcionamiento del juego, enseñando un ejemplo si es necesario.
- El profesor debe plantear a los alumnos que piensen en los conceptos que ya saben y pueden emplear para desarrollar el juego.
- Los alumnos deberán formar los grupos de trabajo.
- Los alumnos deberán recopilar la información necesaria para desarrollar el juego.
- Los alumnos elaborarán el plan de trabajo y analizarán las posibles soluciones.
- Los alumnos con el apoyo del profesor comenzarán el desarrollo del juego.
- Los alumnos presentarán los resultados obtenidos.

\* El profesor deberá revisar el plan de trabajo de todos los grupos, aportar indicaciones sobre la metodología y se reunirá periódicamente con ellos para orientarlos sobre el avance del proyecto.

### **Competencias trabajadas**

- Competencia matemática y competencias básicas en ciencia y tecnología.
- Competencia para aprender a aprender.
- Competencia digital.
- Sentido de la iniciativa y espíritu emprendedor.
- Competencias sociales y cívicas.

### **Evaluación**

Para evaluar esta actividad se utilizarán los siguientes criterios:

- El juego debe funcionar de manera correcta.
- Se valorará positivamente que el personaje principal (Pacman) mueva la boca mientras se mueve.
- Se valorará positivamente que se incluyan sonidos.

**Técnica didáctica**

- Aprendizaje basado en proyectos.

**Documentación didáctica**

- El docente deberá supervisar la formación de los grupos para evitar desigualdades en los mismos.
- Es imprescindible que el docente que vaya a llevar a cabo esta actividad cuente con los conocimientos necesarios sobre computación, programación y el lenguaje de programación Scratch.

## **Parte III – Conclusiones**



## Conclusiones

En este trabajo se analiza el concepto de pensamiento computacional y su situación en la educación de nuestro país. Hemos comprobado que la programación es una herramienta muy útil para ser integrada con otras materias, pudiendo aumentar significativamente la curva de aprendizaje y fomentando un aprendizaje significativo en los alumnos.

También es importante destacar que necesitamos formar a los docentes, sea cual sea el área en el que imparten clase, para que puedan utilizar herramientas como Scratch adecuadamente, ya que una mala formación del profesorado que va a integrar la programación en su asignatura puede ser fatal, llegando incluso a poder dificultar el aprendizaje.

Se han propuesto una serie de actividades orientadas a desarrollar el pensamiento computacional en los alumnos, a la vez que se pretende motivarlos para favorecer el aprendizaje autónomo y significativo. En el catálogo de actividades propuestas se incluyen algunas que pueden ser utilizadas en asignaturas no tecnológicas como música o ciencias sociales.

Hay que señalar que casi todas las actividades incluidas siguen la metodología del aprendizaje basado en problemas, ya que como hemos podido comprobar a raíz del estudio en el que se basa el capítulo 3, esta metodología es muy adecuada para el aprendizaje de la programación, en el que son los propios alumnos los que van descubriendo los conocimientos con la ayuda del profesor, que ejerce de guía del aprendizaje.

Se incluye también una actividad algo más larga que puede ser trabajada mediante el aprendizaje basado en proyectos, otra de las metodologías adecuada para iniciar a los alumnos en la programación.

# Referencias

- [1] Jeannette M.Wing. Computational thinking. Communications of the acm. March 2006/Vol 49, Nº 3.
- [2] Alfred V. Aho; Computation and Computational Thinking. Comput J 2012; 55 (7): 832-835. doi: 10.1093/comjnl/bxs074
- [3] The Royal Society. *Shut down or restart? The way forward for computing in UK schools. Enero 2012*
- [4] Real Decreto 1105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y del Bachillerato.
- [5] Revista de Investigación en Docencia Universitaria de la Informática, vol. 10, núm. 2. Mayo 2017, ISSN 1989-1199.
- [6] DIGCOMP: A Framework for Developing and Understanding Digital Competence in Europe (<http://ftp.jrc.es/EURdoc/JRC83167.pdf>)
- [7] Jesús Moreno-León, Gregorio Robles, Marcos Román-González. Programar para aprender en Educación Primaria y Secundaria: ¿qué indica la evidencia empírica sobre este enfoque?. REVISIÓN. vol. 10. núm. 2. Mayo 2017. ISSN 1989-1199.
- [8] Seymour Papert, Mindstorms: Children, Computers, And Powerful Ideas, ISBN 0465046746.
- [9] Pablo Espeso, Carmen Hernández, Belén Palop. Didáctica de la computación a través de la programación de ordenadores: un nuevo enfoque. REVISIÓN. vol. 10. núm. 2. Mayo 2017. ISSN 1989-1199.
- [10] Edurne Larraza Mediluze, Olatz Arbelaitz Gallego, Ana Arruate Lasa, Nestor Garay Vitoria, Montse Maritxalar Anglada, Jose Ignacio Martín Aramburu, Txelo Ruiz Vázquez, Jose Francisco Lukas Mujika, Desde la universidad hasta primaria: proyecto piloto de innovación docente en Informática. REVISIÓN. vol. 10. núm. 2. Mayo 2017. ISSN 1989-1199.
- [11] <http://www.aulaplaneta.com/2015/02/04/recursos-tic/como-aplicar-el-aprendizaje-basado-en-proyectos-en-diez-pasos/> Consultado el 2/07/2017
- [12] <http://web.mit.edu> Consultado el 14/07/2017

## **Parte IV – Anexos**

## **APENDICE A: Contenido del CD**

El contenido del CD adjunto a este TFG es el siguiente:

- doc: Directorio que contiene la memoria del TFM.
  - o memoria.pdf: Fichero de la memoria del TFM.
  
- src: Directorio que contiene los ficheros de los proyectos Scratch desarrollados.
  - o gato.sb2: proyecto Scratch de la actividad moviendo el gato.
  - o camino.sb2: proyecto Scratch de la actividad el camino.
  - o piano.sb2: proyecto Scratch de la actividad el piano.
  - o provincias.sb2: proyecto Scratch de la actividad provincias y comunidades.
  - o pacman.sb2: proyecto Scratch de la actividad Pacman.

