Experiencias docentes

Pensamiento Matemático Avanzado y Scratch: El Caso del Máximo Común Divisor

Advanced Mathematical Thinking and Scratch: The Greatest Common Divisor

Miguel Ángel Baeza Alba, Francisco Javier Claros Mellado,
Mª Teresa Sánchez Compaña, Mónica Arnal Palacián

Revista de Investigación

Volumen VII, Número 2, pp. 043–064, ISSN 2174-0410
Recepción: 2 Abr'17; Aceptación: 2 Sep'17

1 de octubre de 2017

Resumen

En este artículo llevamos a cabo una propuesta didáctica, con los alumnos del Máster de Formación del Profesorado en Enseñanza Secundaria y Bachillerato de la Universidad Complutense de Madrid, de la especialidad de Matemáticas. Dicha propuesta consiste en la programación en Scratch del Algoritmo de Euclides para el Máximo Común Divisor. Esta metodología de trabajo, que mezcla matemáticas y programación, permitirá trabajar con los alumnos elementos propios del Pensamiento Matemático Avanzado como son la abstracción, la formalización y la generalización, entre otros.

Palabras Clave: Matemáticas, programación, algoritmo, didáctica.

Abstract

In this article we carry out a didactic proposal with the students of the Master which name is "Teacher training in E.S.O and Bachillerato" at Complutense University in Madrid, in the Maths speciality. This proposal consists in the Scratch programming of Euclides Algorithm for the Greatest Common Divisor. This methodology which mixes Maths and programming, let the students work with the main elements of Advanced Mathematical Thinking such as abstraction, formalization and generalization.

Keywords: Mathematics, programming, algorithm, didactics.

1. Introducción

La introducción de las TIC en la escuela abre un amplio campo de posibilidades que intenta responder a las necesidades que plantea esta nueva sociedad de la información. Este hecho va a llevar emparejado una serie de cambios; Pérez (1998) sugirió reconceptualizar el alcance de lo educativo, reformular el currículo e innovar en las estrategias educativas.

El uso de las nuevas tecnologías ha supuesto un esfuerzo en formación por parte del profesorado. Pascual (1998) advierte que el uso del ordenador no supone una mejora si no va acompañado de un adecuado planteamiento metodológico; el uso que hagamos del ordenador en el aula, tiene que formar parte de una actividad que haya sido diseñada por el profesor.

Carvajal, Font y Giménez (2014) hacen alusión a que, pese a que los alumnos del Máster de Secundaria en Matemáticas poseen una sólida formación en competencia digital, pocos de ellos utilizan las TIC en su periodo de prácticas en el centro de Secundaria. La principal razón que argumentan los estudiantes es la falta de infraestructura y recursos generales en el centro.

Ricoy y Couto (2012) señalan los beneficios, controversias y cambios o mejoras que puede aportar la introducción de las TIC en el aula. Entre las dificultades que puede suponer su utilización señalan, además de la desmotivación de ciertos docentes ante su uso, la escasez de medios tecnológicos en algunos centros. Por el contrario señalan como beneficios, el desarrollo de la comunicación bidireccional y la atención a la diversidad.

En el caso de las matemáticas, son muchos los programas que han surgido en los últimos años y que pretenden ser una herramienta útil en la enseñanza de las mismas. Fernández y Muñoz (2007) realizan un barrido sobre los programas matemáticos que más suelen emplearse en un aula de matemáticas. En la selección que proponen encontramos: Wiris, Geogebra, Cabri y Derive, así como páginas que aportan información y ejercicios sobre cómo trabajar contenidos matemáticos. Se presentan actividades para trabajar con estos programas, aunque son bastantes mecánicas y no promueven una reflexión sobre los contenidos que se están trabajando. De hecho, no encontramos entre las actividades propuestas, aquellas que permiten la programación de aplicaciones.

Scratch es un software libre desarrollado por Lifelong Kindergarten Group de los Laboratorios Media-Lab en MIT. Se trata de un lenguaje gráfico que permite programar uniendo bloques predefinidos. Esta facilidad hace que pueda iniciarse en él a edades muy tempranas. Son muchas las funcionalidades que pueden aplicarse a Scratch y que son objeto de investigación en la actualidad. Carralero (2011) señala como funcionalidad importante el hecho de manejar la programación implícita en Scratch para trabajar un contenido de primaria y secundaria. Es decir, propone no trabajar la programación directamente, sino elegir un concepto y diseñar qué elementos de la programación son necesarios para llegar a definir este concepto. Asimismo señala que para abordar determinados elementos de la programación (por ejemplo los bucles), es necesario que los alumnos se encuentren al menos en 3º o 4º de la ESO, ya que a esa edad los alumnos empiezan a poseer un pensamiento lógico-abstracto. En la construcción del programa a desarrollar, puede ser necesaria la resolución de algoritmos, elementos importantes en el desarrollo de las matemáticas.

Adoptamos la siguiente definición de algoritmo:

> *"Un algoritmo es una sucesión finita de reglas elementales, regidas por una prescripción precisa y uniforme, que permite efectuar paso a paso, en un encadenamiento estricto y riguroso, ciertas operaciones de tipo ejecutable, con vistas a la resolución de los problemas pertenecientes a una misma clase" (Ifrah, 2008, p. 1616)*

En la misma línea se manifiestan Gairín y Sancho (2002); es claro que si la Matemática tiene como objetivo prioritario resolver problemas y encontrar soluciones a cuestiones cada vez más difíciles, parece que la necesidad de utilizar algoritmos está totalmente justificada.

Por otro lado, el progreso en matemáticas exige la automatización de los procesos elementales para concentrar la atención en las nuevas ideas, las cuales, a su vez, necesitarán transformarse en automáticas para poder abordar otras más complejas y así sucesivamente (Skemp, 1993).

La consecuencia más natural que se extrae de estos argumentos es la necesidad de enseñar a los alumnos algoritmos en la escuela. Usiskin (1998), por su parte, enumera hasta nueve razones diferentes por las que es útil saber y enseñar algoritmos matemáticos: son eficaces, fiables, precisos, rápidos, proporcionan un registro escrito, establecen una imagen mental, son instructivos, pueden ser utilizados en otros algoritmos y pueden ser objetos de estudio.

En este documento llevaremos a cabo una propuesta didáctica que pretende trabajar el Pensamiento Matemático Avanzado, en el sentido de Tall (1991), con los alumnos del Máster de Secundaria y Bachillerato de la Universidad Complutense de Madrid (especialidad de Matemáticas) a través del desarrollo de situaciones didácticas. Para ello se trabajará el Algoritmo de Euclides para el cálculo del Máximo Común Divisor, utilizando la herramienta Scratch.

El documento se organiza en torno a seis apartados. Un primer apartado de introducción, un segundo en el que señalamos los objetivos principales y secundarios de la investigación. En el tercero, denominado marco teórico, describiremos dos teorías para el aprendizaje y enseñanza de las matemáticas, que determinan los principios didácticos sobre los que basaremos nuestra metodología: el Pensamiento Matemático Avanzado y la Teoría de las Situaciones Didácticas. En el cuarto apartado describimos la metodología que vamos a utilizar. En el quinto mostramos los resultados obtenidos y por último, en el sexto, señalamos las principales conclusiones obtenidas, así como las perspectivas futuras que abordaremos.

2. Objetivos

En esta investigación pretendemos introducir a los alumnos en el Pensamiento Matemático Avanzado dando respuesta a la siguiente cuestión: "Diseñar e implementar un algoritmo en Scratch que permita calcular el M.C.D de dos números naturales". Los objetivos que pretendemos son:

O1. Trabajar con los alumnos del Máster el razonamiento lógico-matemático a través del diseño del algoritmo de Euclides para el cálculo del M.C.D. Queremos observar qué dificultades surgen durante el diseño de dicho algoritmo y cómo éstas son resueltas. Una vez comprendido el algoritmo por parte de los alumnos, estos tendrán que diseñar el diagrama de flujo del mismo para estructurar cuáles son las instrucciones que deberán implementar posteriormente en Scratch. El uso del lenguaje Scratch para potenciar el desarrollo del pensamiento algorítmico en estudiantes fue trabajado también por Vidal, Cabezas, Parra y López (2015) quienes señalaron las virtudes de esta nueva metodología de enseñanza que usa la programación para trabajar conceptos matemáticos. Entre estás señalaban: la motivación del alumno, la participación activa de los alumnos en la búsqueda de soluciones y el análisis de la solución obtenida, la cual podrían probar y mejorar.

O2. Potenciar en los alumnos el uso de la abstracción a través de la traducción a Scratch del algoritmo diseñado. Es decir, se potencia la abstracción a través de la implementación del algoritmo para el M.C.D en Scratch. Una vez diseñado el diagrama de flujo para el algoritmo de Euclides, los alumnos deberán implementarlo en Scratch. La idea de trabajar la programación con Scratch en alumnos que serán futuros profesores de matemáticas, a partir de los algoritmos diseñados previamente, constituye una herramienta muy útil que permite resolver cualquier problema matemático que se presenta en secundaria (véase Barrera, 2013).

O3. Formular hipótesis que permitan ser contrastadas a través del algoritmo diseñado e implementado. Una vez que el algoritmo haya sido programado, los alumnos tendrán que formular hipótesis para comprobar si el algoritmo llega a resultados correctos. En este momento usarán el método tradicional para calcular el M.C.D de dos números y también la aplicación diseñada. Los resultados en ambos casos deben ser iguales; si no fuera así, el algoritmo no estaría bien diseñado o implementado y tendrían que buscar una solución al problema.

O4. Trabajar el concepto de generalización a partir de la comprobación de casos particulares en la aplicación creada. En la comprobación del algoritmo diseñado, es importante empezar por la realización de casos particulares con números pequeños para probar después con números grandes. Esto permitirá una aproximación intuitiva al concepto de generalización, consiguiendo con ello que el alumno adquiera un cierto grado de certeza sobre la aplicación construida. Kidron y Dreyfus (2014) denominan a este proceso "imagen demostración", que emerge en la construcción de la demostración de una afirmación o problema. En nuestro caso, los alumnos tendrán la imagen demostración de que la aplicación construida funciona para cualquier par de números, ya que serán capaces de aproximarse a la generalización de los resultados parciales obtenidos.

O5. Valorar el concepto de modelo matemático. Se incentivará a los alumnos a que conciban el programa diseñado como un modelo matemático que permite calcular el M.C.D y el M.C.M de dos números naturales y se incidirá en que dicho modelo puede ser mejorado. Con este objetivo pretendemos que los alumnos valoren el concepto de modelo matemático como un elemento que tiene una función local, pero que puede ser ampliado, para que permita resolver nuevos problemas. Por ejemplo, se pueden añadir instrucciones al programa del M.C.D para que también calcule el M.C.M (utilizando la propiedad que asegura que el producto de dos números naturales coincide con el de su M.C.D por su M.C.M).

3. Marco Teórico

Nuestro marco teórico se sustenta en dos pilares: el Pensamiento Matemático Avanzado (PMA) y la Teoría de Situaciones Didácticas (TSD).

La primera teoría se usará para determinar los procesos implicados en el diseño e implementación del algoritmo de Euclides para calcular el M.C.D con Scratch. Nos basaremos en ella para elegir el contenido y la secuenciación de las actividades. A su vez, éstas estarán diseñadas siguiendo la Teoría de Situaciones Didácticas.

3.1. Pensamiento Matemático Avanzado (PMA)

En 1985, en el seno del grupo Psicologist Mathematic Education (PME), se creó un grupo de trabajo cuyo objetivo era estudiar la naturaleza del "Pensamiento Matemático Avanzado". Dicho grupo pretendía profundizar en los procesos cognitivos de enseñanza y aprendizaje de temas relacionados con el cálculo infinitesimal (Dreyfus 1990 y Tall, 1991).

A partir de este momento, el interés en didáctica de la matemática se empieza a centrar en la problemática del aprendizaje en términos de procesos cognitivos y no como una simple adquisición de competencias y habilidades. También se produce una evolución en las investigaciones, que empiezan a ocuparse de tópicos que por su naturaleza y complejidad se situarían dentro de una matemática superior (el límite y la derivada, entre otros). Entre los procesos involucrados en el Pensamiento Matemático Avanzado citamos en primer lugar, por su importancia, la abstracción y la generalización.

La abstracción es definida como un proceso de construcción de objetos mentales a partir de objetos matemáticos (Dreyfus, 1991). Para este autor, la generalización es definida como la derivación o inducción de particulares, para identificar generalidades y expandir los dominios de validez.

Robert y Swarzenberger (1991) señalaron una serie de diferencias entre el pensamiento elemental y el avanzado: (1) En el Pensamiento Matemático Avanzado los alumnos tienen que aprender más conceptos en menos tiempo y además estos son presentados de manera formal. (2) Los conceptos enseñados llevan asociadas las siguientes propiedades: generalización, abstracción y formalización; propiedades que pueden entrar en conflicto con el conocimiento anterior que se tenía sobre el concepto. (3) Los alumnos se enfrentan a una amplia gama de problemas que nacen de una variedad de contextos, los cuales no pueden ser discutidos en todo detalle.

Estas diferencias entre el pensamiento elemental y el pensamiento avanzado propuestas por Tall (1991) y Robert y Swarzenberger (1991) son rebatidas por Edwards, Dubinsky y McDonald (2005); los cuales, además de proponer una definición alternativa de Pensamiento Matemático Avanzado, señalan que un concepto se considerará dentro del Pensamiento Matemático Avanzado dependiendo de los aspectos que se traten.

Edwards, Dubinsky y McDonald (2005; pp.17-18) proponen la siguiente definición de Pensamiento Matemático Avanzado:

> *"Pensamiento que requiere deductivo y riguroso razonamiento acerca de nociones matemáticas que no nos son enteramente accesibles a través de los cinco sentidos".*

Teniendo en cuenta todo lo anterior, si se realiza un tratamiento procedimental de un concepto, dicho concepto quedará fuera del Pensamiento Matemático Avanzado, a pesar de que sea uno que por su naturaleza debiera formar parte de él.

Todo esto nos induce a pensar que el concepto de M.C.D de dos números naturales, abordado a través del diseño e implementación de un algoritmo en Scratch, puede permitir la aparición de procesos cognitivos que forman parte del Pensamiento Matemático Avanzado (véase Tall (1991) y Robert y Swarzenberger (1991)). Estos procesos, como anteriormente se ha explicitado son: la abstracción, la generalización, la formulación de hipótesis, la verificación de dichas hipótesis y la formulación formal del concepto, expresada ésta a través de la aplicación creada.

3.2. Teoría de las Situaciones Didácticas (TSD)

La teoría de las situaciones didácticas aparece en 1970. Nació como un método simple de descripción y de interrogación matemática de los dispositivos psicológicos y didácticos. Dicha teoría se fundamenta en varias ideas:

- "[...] los conocimientos se manifiestan esencialmente como instrumentos de control de las situaciones" (Brousseau, 1997, p.3). Es decir, el conocimiento se adquiere mediante una adaptación al medio. El sujeto se enfrenta con un medio que le plantea una dificultad que tiene que superar. Si lleva a cabo esta tarea con éxito, el resultado es un nuevo conocimiento.
- Asociado a cada conocimiento existe una situación característica tal que el sujeto, al actuar sobre ella e intentar controlarla, adquiere el conocimiento con el que se corresponde.

Según Brousseau (2011), la teoría de las situaciones didácticas modeliza las condiciones bajo las cuales los seres humanos producen, comunican y asimilan los conocimientos matemáticos. Estas condiciones son modelizadas por sistemas llamados "situaciones", que conducen a agentes en interacción con ellas a manifestar este conocimiento. Son pues específicas del conocimiento en juego.

Algunas situaciones requieren de un conocimiento previo para poder tener éxito en alcanzar el nuevo conocimiento, pero hay otras situaciones en las que el sujeto puede construir por sí mismo el conocimiento requerido sin recurrir a ningún conocimiento anterior.

Brousseau (1998) distingue, en el campo de la enseñanza de las matemáticas, dos tipos de situaciones: las didácticas y las a-didácticas (generalmente incluidas como una fase, dentro de las situaciones didácticas).

Una situación didáctica es una situación que contiene intrínsecamente la intención de que alguien aprenda algo. Esta intención no desaparece en la situación o fase a-didáctica; la no intencionalidad contenida en este concepto se refiere a que el alumno debe relacionarse con el problema respondiendo al mismo en base a sus conocimientos, motivado por el problema y no por satisfacer un deseo del docente, y sin que el docente intervenga directamente ayudándolo a encontrar una solución.

En las situaciones didácticas es el profesor el que transmite el saber al alumno ignorando las relaciones entre los otros elementos de la situación. En las situaciones a-didácticas la intencionalidad del profesor queda oculta. El profesor no interviene didácticamente, sino que diseña un medio en el que el alumno tiene que actuar, eligiendo sus acciones libremente, sin la dirección del profesor. Son estas acciones que realiza el alumno, las que determinarán si es capaz de obtener el nuevo conocimiento. Esta situación debe ser diseñada de forma que el alumno no sepa qué saber va a adquirir. Se dirá que el alumno tiene éxito en dicha situación cuando ha adquirido el conocimiento que el profesor pretendía.

Se distinguen tres tipos de situaciones: de acción, de formulación y de validación que pueden darse en el proceso de enseñanza-aprendizaje (véase Brousseau, 1997).

Al finalizar cualquier situación didáctica o a-didáctica, es necesaria una institucionalización del saber alcanzado. En esta institucionalización se devuelve la responsabilidad al profesor, el cual define las relaciones que pueden tener los comportamientos o las producciones "libres" del alumno, con el saber cultural o científico que se ha trabajado y con el proyecto didáctico. El profesor realiza una revisión de las actividades realizadas y le da un estatus oficial. En ese momento el contenido matemático pasa a formar parte del saber de la clase.

4. Metodología

El trabajo se organiza en torno a 7 fichas que se llevan a cabo con los alumnos del Máster de Formación del Profesorado en Enseñanza Secundaria y Bachillerato de la Universidad Complutense de Madrid (especialidad de Matemáticas). El trabajo se desarrolló en la asignatura "Innovación Docente e Iniciación a la Investigación en Educación Matemática". Cada ficha de trabajo es una situación didáctica que pretende mostrar los elementos del Pensamiento Matemático Avanzado. Antes de comenzar con las fichas de trabajo, se dividió a los dieciocho alumnos en nueve grupos de dos integrantes cada uno; no se utilizó ningún criterio para la división en grupos, simplemente la disposición usual de los alumnos en clase. De los dieciocho alumnos, catorce son Licenciados en Matemáticas, uno Doctor en Matemáticas, uno Ingeniero en Informática, uno Licenciado en Físicas y un último licenciado en Administración y Dirección de Empresas.

Las fichas de los diferentes grupos con las tareas propuestas realizadas, son recogidas por el profesor al acabar el tiempo destinado a cada una.

La ficha 0 tiene como objetivo familiarizar a los alumnos con el lenguaje de programación Scratch. Para ello se llevó a cabo una sesión introductoria a Scratch en el aula de informática de la Facultad de Educación. Ninguno de los alumnos conocía previamente la herramienta, por lo que esta sesión introductoria fue esencial. Durante los 40 primeros minutos, la sesión fue conducida por el profesor, explicando nociones generales del programa y realizando ejemplos que todos pudieron visualizar con ayuda del proyector. Posteriormente se dedicaron 50 minutos para que los alumnos realizasen las tareas propuestas en la ficha 0.

La ficha 1 tiene como objetivo trabajar con los alumnos la noción de algoritmo, señalando la importancia que estos tienen en las matemáticas que se trabajan en la Enseñanza Primaria y Secundaria (algoritmo de la división, multiplicación, etc.). La ficha se inicia con la explicación del algoritmo de Euclides para el máximo común divisor. Se facilitan dos ejemplos resueltos sobre cómo llevarlo a la práctica y a continuación se pide a los alumnos que realicen 4 ejercicios de aplicación del mismo. La duración asignada a esta ficha fue de 30 minutos.

La ficha 2 va encaminada a que los alumnos sean capaces de diseñar diagramas de flujo a partir de un determinado algoritmo. Estos diagramas son un instrumento útil y previo a la programación de una aplicación. La ficha comienza presentando tres ejemplos concretos sobre cómo realizar el diagrama de flujo para tres algoritmos que se muestran. A continuación se propone que cada grupo realice el diagrama de flujo correspondiente al algoritmo de Euclides. La duración asignada a esta ficha fue de 30 minutos.

En la ficha 3 los alumnos trabajan con la aplicación Scratch. La ficha comienza mostrando el código en Scratch de los tres algoritmos que se utilizaron en la ficha anterior para ejemplificar los diagramas de flujo. Los alumnos deberán introducirlos en Scratch y validar su funcionamiento ejecutándolos varias veces. Esta labor, junto con la realizada en la ficha 0, intentará conseguir que los alumnos se familiaricen con el lenguaje del programa. A continuación se les pide que intenten programar en Scratch el algoritmo de Euclides. Para ello, deberán hacer uso del diagrama de flujo que diseñaron en la ficha 2. La duración asignada a esta ficha fue de 70 minutos (30 minutos para los ejercicios iniciales y 40 minutos para la implementación del algoritmo de Euclides).

La ficha 4 supone la institucionalización del diseño del algoritmo de Euclides a través del diagrama de flujo. Los alumnos tienen que comparar el diagrama de flujo que se les entrega en esta ficha con el que realizaron en la ficha 2 para subsanar los posibles errores que cometieran. El objetivo de esta ficha es intentar conseguir que todos los alumnos sean capaces de implementar bien el programa en Scratch. La duración asignada a esta ficha fue de 10 minutos.

La ficha 5 supone la institucionalización del programa en Scratch que permite calcular el M.C.D de dos números. En dicha ficha se hace entrega del código que permite programar el algoritmo de Euclides. Los alumnos tienen que compararlo con el suyo y corregir los posibles errores que tuvieran. A continuación se presentan tres ejercicios en los que se les pide calcular el M.C.D de varios pares de números; además, también se solicita que contesten a cuestiones relativas sobre los resultados que se esperan. La duración asignada a esta ficha fue de 10 minutos.

La ficha 6 supone una mejora de la aplicación creada en la ficha 5, ya que permite calcular tanto el M.C.D como el M.C.M. Se les pide que diseñen el diagrama de flujo que permite calcular el M.C.M y también el código del programa que realizará en Scratch el cálculo del M.C.M de dos números. La ficha acaba con varios ejercicios para que los alumnos prueben la aplicación creada. La duración asignada a esta ficha fue de 50 minutos.

5. Resultados

A lo largo de los siguientes párrafos, se hará alusión a los resultados obtenidos en las diferentes fichas señaladas en el apartado "metodología".

Respecto a la ficha 0 señalamos que aunque en algunos grupos se observó mayor facilidad que en otros para comprender el lenguaje y realizar los algoritmos que se pedían, todos finalizaron con éxito dicha ficha.

La siguiente sesión de clase (2ª sesión) se dedicó a la realización de las fichas 1 y 2. La ficha 1 (manejo del algoritmo) resultó muy sencilla para ellos. Los alumnos leyeron y comprendieron rápidamente el algoritmo y no tuvieron ninguna dificultad para realizarla. Tras los 30 minutos establecidos, todos los grupos tenían la ficha finalizada correctamente; dicha ficha fue recogida por el profesor para su análisis. No se encontró ningún error en la misma. El profesor creyó conveniente, antes del inicio de la ficha 2, entablar una conversación con los alumnos para analizar la ficha 1. En dicha conversación, los alumnos detectaron por ellos mismos que los pares de números elegidos en cada uno de los ejercicios de la ficha 1, constituían una variable didáctica, ya que cada par encerraba una situación distinta de la anterior.

Una vez resuelta la ficha 1 y acabada la conversación antes descrita, se repartió a los alumnos la ficha 2. En esta ficha debían comprender qué es un diagrama de flujo y representar el diagrama de flujo para el algoritmo de Euclides. Esta ficha supuso muchas dificultades para algunos grupos, mientras que para otros resultó algo muy sencillo. Para analizar los resultados de la ficha 2, se diseñó una tabla de categorías (Tabla 1) que pensamos, permite clasificar cada una de las respuestas que se obtuvieron.

Tabla 1. Tabla de categorías para el análisis de la ficha 2

Categorías	Subcategorías
C1-. Identificación de las representaciones del diagrama de flujo	C1.1-. Los polígonos del diagrama de flujo no se corresponden en absoluto con la instrucción que deben representar. C1.2-. Falta algún polígono o alguno de ellos no es el que debiese ser (por ejemplo: aparece un cuadrado donde debiese ser un rombo o un romboide). C1.3-. Todos los polígonos del diagrama de flujo se corresponden con el tipo de función que desempeñan (Inicio/Fin, Entrada/Salida de datos, Proceso o Decisión). Además, aparecen todas las líneas de flujo.
C2-. Realmacenamiento de las variables si N1<N2	C2.1-. No realiza este paso en el diagrama de flujo. C2.2-. Realmacena las variables, pero lo hace de forma incorrecta. C2.3-. Realmacena las variables de forma correcta.
C3-. Identificación de la condición de parada	C3.1-. No aparece en el diagrama de flujo ninguna decisión que se corresponda con una condición de parada. C3.2-. Identifica que existe un bucle y establece una decisión de parada, pero esta es incorrecta. C3.3-. Identifica correctamente la existencia de un bucle y su condición de parada correspondiente.
C4-. Reasignación de las variables dentro del bucle	C4.1-. No existe reasignación de variables dentro del bucle. C4.2-. Se reasignan variables dentro del bucle, pero esta reasignación es incorrecta. C4.3-. Se reasignan las variables correctamente dentro del bucle.
C5-. Devolución del MCD	C5.1-. El programa no devuelve correctamente el valor del MCD. C5.2-. El programa devuelve correctamente el valor del MCD.
C6-. Nivel de Eficiencia	C6.1-. El programa no funciona. Por tanto, no tiene sentido hablar de eficiencia. C6.2-. El programa funciona, pero se crean más variables de las necesarias. C6.3-. El programa funciona con el número mínimo de variables posibles.

Una vez establecidas las categorías, procedimos a analizar los resultados de cada grupo. Se ha asignado una subcategoría a cada grupo para cada una de las categorías. Cada categoría lleva asignada una puntuación para cada grupo. Así, la categoría C1 puede tomar los valores 1, 2 o 3 dependiendo de que la respuesta se clasifique como C1.1, C1.2 o C1.3. Entendemos que 1 es la puntuación más baja y 3 la puntuación máxima. Las demás categorías funcionan de igual forma, a excepción de la categoría C5, que solo puede tomar los valores 1 o 2 dependiendo si se le asigna C5.1 o C5.2, respectivamente. Las tablas que contienen los resultados obtenidos se muestran a continuación:

Tabla 2. Tabla de resultados por grupos de la ficha 2

Categoría	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 6	Grupo 7	Grupo 8	Grupo 9
C1-. Identifica correctamente las representaciones del diagrama de flujo	C1.3	C1.2	C1.2	C1.3	C1.3	C1.3	C1.3	C1.3	C1.3
C2-. Realmacena correctamente las variables si N1<N2	C2.2	C2.1	C2.3	C2.3	C2.3	C2.3	C2.3	C2.3	C2.3
C3-. Identifica correctamente la condición de parada	C3.3	C3.3	C3.3	C3.3	C3.3	C3.3	C3.3	C3.3	C3.3
C4-. Reasigna correctamente las variables dentro del bucle	C4.2	C4.2	C4.3	C4.2	C4.2	C4.3	C4.3	C4.3	C4.3
C5-. Devuelve correctamente el valor del MCD	C5.1	C5.1	C5.2	C5.1	C5.1	C5.2	C5.1	C5.2	C5.2
C6-. Nivel de Eficiencia	C6.1	C6.1	C6.3	C6.1	C6.1	C6.3	C6.1	C6.2	C6.3

Tabla 3. Tabla de resultados globales de la ficha 2

Grupo	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 6	Grupo 7	Grupo 8	Grupo 9
Puntuación Global	12	10	16	13	13	17	14	16	17

Como muestra de las respuestas, se adjuntan las de los grupos con mayor y menor puntuación (tabla 4 y Figura 1 respectivamente).

Tabla 4. Resultados de la ficha 2 de los grupos con mayor puntuación

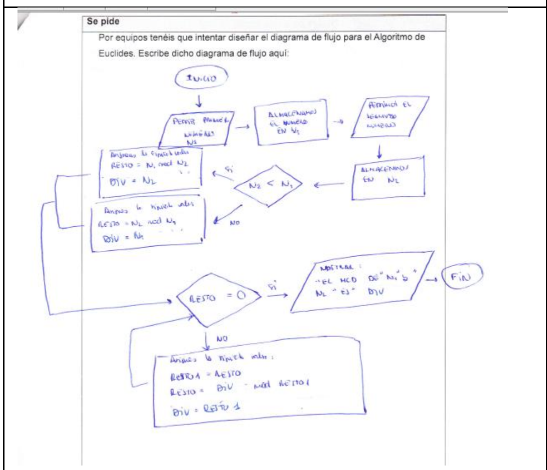
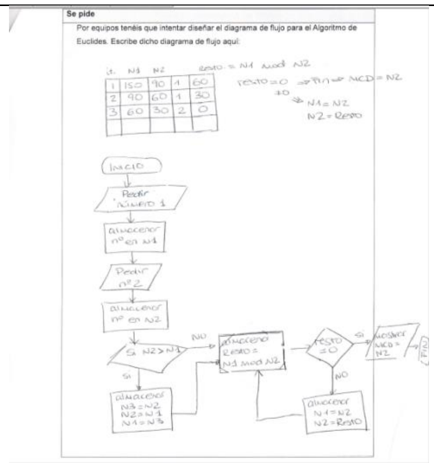
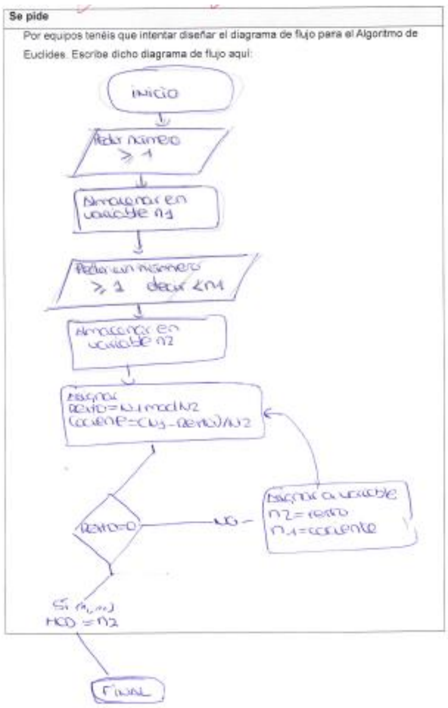
Grupo 6	Grupo 9
	

Figura 1. Resultados de la ficha 2 del grupo con menor puntuación



En la siguiente sesión de clase (3ª sesión) se realizó la ficha 3. Los 30 primeros minutos de la sesión se dedicaron a que los alumnos realizasen los tres primeros ejercicios de la ficha. Los alumnos resolvieron y enviaron correctamente los mencionados ejercicios en el tiempo pactado; los 40 minutos restantes se dedicaron a la realización del ejercicio 4 (traducir al lenguaje de Scratch el diagrama de flujo del algoritmo de Euclides que cada uno de los grupos había creado en la sesión anterior). Se indicó a los grupos que si detectaban algún error en el diagrama de flujo, tras introducir el algoritmo en Scratch, podían intentar solucionarlo. Tal y como se ha indicado en la tabla de resultados anterior, los algoritmos (en forma de diagrama de flujo) de los grupos 1, 2, 4, 5 y 7 no proporcionaban un valor correcto del máximo común divisor. Los resultados que se obtuvieron fueron:

El grupo 1 no consiguió hacer funcionar el algoritmo. En la segunda rama del condicional, no reasignaron el valor del resto dentro del bucle. Por tanto, el algoritmo ingresa en un proceso infinito y no devuelve una respuesta. El diagrama de flujo de este grupo no fue correcto.

El grupo 2 tampoco consiguió hacer funcionar el algoritmo. En este caso, una mala reasignación de la variable divisor antes de iniciar el bucle, hace que el programa no devuelva correctamente el valor del MCD. El diagrama de flujo de este grupo tampoco fue correcto.

El grupo 3 consiguió que su algoritmo funcionase. Su diagrama de flujo fue correcto.

El grupo 4 consiguió que su algoritmo funcionase, pese a que su diagrama de flujo no era correcto en un principio.

El grupo 5 tampoco consiguió hacer funcionar el algoritmo. En este caso, las variables que hay que intercambiar dentro del bucle, las intercambian fuera de este, dentro del condicional. Además, se intercambian de forma incorrecta. El diagrama de flujo de este grupo tampoco fue correcto.

El grupo 6 consiguió que su algoritmo funcionase. Su diagrama de flujo fue correcto.

El grupo 7, al igual que el grupo 4, consiguió que su algoritmo funcionase, pese a que su diagrama de flujo no era correcto en un principio. Cabe destacar que los errores en el diagrama de flujo de este grupo fueron el devolver por pantalla el valor del dividendo en lugar del divisor, en el caso en el que el resto sea 0 (por lo que puntuaron 1 en la categoría C5) además de introducir un condicional, en el caso en que el resto sea distinto de cero, cuya condición es que el dividendo sea mayor que el resto, algo que siempre ocurre (por lo que puntuaron 1 en la categoría C6). Estos errores fueron subsanados en la elaboración del algoritmo en Scratch.

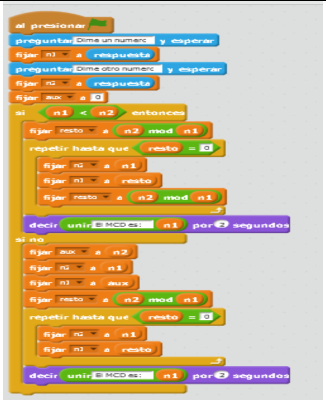
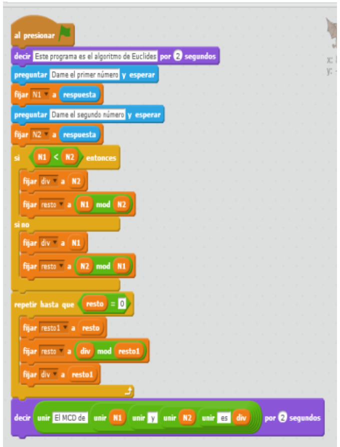
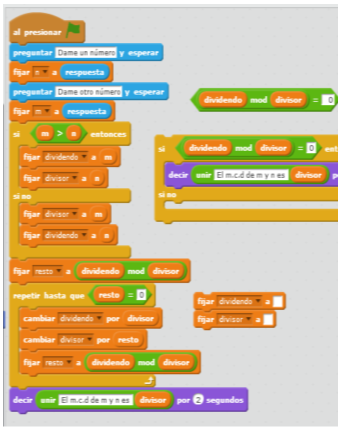
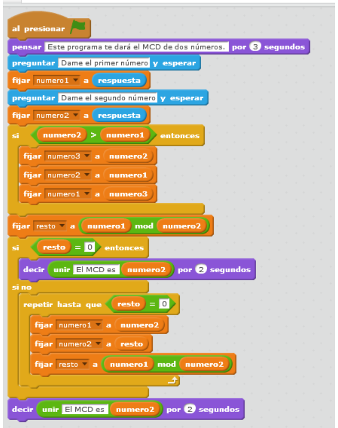
El grupo 8, pese a tener un diagrama de flujo correcto, no fue capaz de hacer funcionar el algoritmo. En este caso, aunque el algoritmo es aparentemente correcto, no han comprendido la diferencia en Scratch entre "cambiar" y "fijar". Al utilizar "cambiar" en lugar de "fijar" dentro del bucle, el algoritmo ingresa en un proceso infinito y no devuelve, por tanto, un valor para el máximo común divisor.

El grupo 9 consiguió que su algoritmo funcionase. Su diagrama de flujo fue correcto.

Podemos por tanto concluir que, de los cuatro grupos que diseñaron bien el diagrama de flujo, solamente tres han conseguido implementarlo correctamente. Además, dos grupos, de entre los cinco que no diseñaron correctamente el diagrama de flujo, han conseguido implementarlo correctamente. Llama la atención que cuatro grupos, de entre los nueve, no hayan conseguido que su algoritmo funcione correctamente en Scratch, lo que supone un 44,44% de los alumnos. Es necesario mencionar que el tiempo no fue un obstáculo. Todos los alumnos terminaron los programas en el tiempo establecido. Como peculiaridad mostramos que el grupo 8, aunque tenía el algoritmo terminado, al probarlo detectaban que lo tenían mal; aunque lo intentaron solucionar, fueron incapaces en el tiempo que les facilitamos. Quizá si hubiesen tenido 10 minutos más, hubiesen solucionado el problema que tenían en la asignación de las variables dividendo y divisor dentro del bucle.

Los algoritmos de cada uno de los grupos se muestran en la tabla 5:

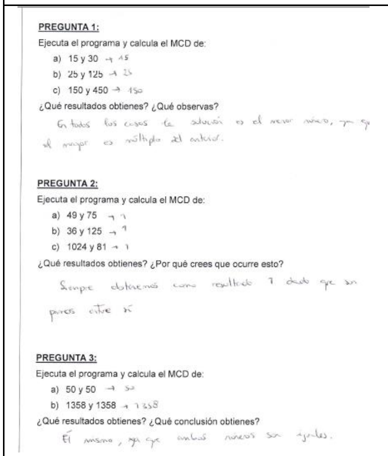
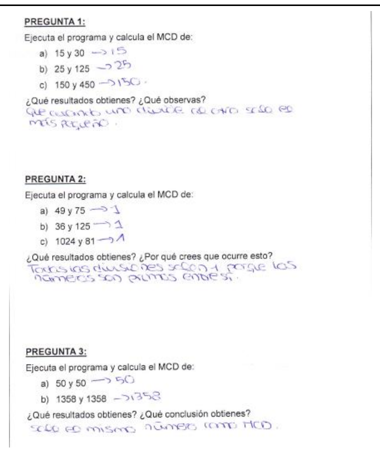
Tabla 5. Resultados del ejercicio 4 de la ficha 3

Grupo 1	Grupo 2	Grupo 3
		
Grupo 4	Grupo 5	Grupo 6
		
Grupo 7	Grupo 8	Grupo 9
		

En la siguiente sesión de clase (4ª sesión) se realizaron las fichas 4, 5 y 6. Antes de comenzar la sesión, se dedicaron 5 minutos para indicar en alto qué grupos habían sido capaces de programar correctamente el algoritmo y cuáles no. Tras este tiempo, se comenzó con la ficha número 4. Se dedicó a ella únicamente 10 minutos. El objetivo de esta ficha fue el de facilitar a los alumnos el diagrama de flujo correcto para el algoritmo de Euclides e intentar que con éste, todos los alumnos fueran capaces de realizar bien el programa en Scratch. El grupo 1 se dio rápidamente cuenta de su error en el algoritmo; comentaron que no llegaron a probar el algoritmo para pares de números en el que el primero fuese más grande que el segundo. Comentaron que en el caso en que lo hubiesen hecho, habrían entregado el algoritmo correcto. El grupo 2 no conseguía detectar qué estaba mal en su algoritmo. Lo probaban para casos concretos y sí les funcionaba y para otros casos no. No fueron capaces, ni tan siquiera con el diagrama de flujo correcto, de implementar bien el algoritmo. El grupo 5 localizó su error y supo subsanarlo. El grupo 8 ya sabía dónde estaba su error antes de ver el diagrama de flujo. Pensaban que "fijar" y "cambiar" variable significaba lo mismo. Buscaron por la red y encontraron su fallo antes de asistir a clase.

Tras los 10 minutos dedicados a la ficha 4, se repartió la ficha 5. En ésta se les facilita el código del algoritmo de Euclides para que todos puedan programarlo correctamente (solamente el grupo 2 no lo tenía bien programado previamente) y responder a una serie de preguntas que se proponen a continuación. De nuevo se dedica a esta ficha 10 minutos de clase. Todos los grupos responden correctamente a las preguntas que se hacen; en este caso son preguntas muy sencillas para alumnos de Máster. No obstante, resultan llamativas las respuestas que dan los alumnos que integran el grupo 2, en comparación a las dadas por el resto de los grupos; sobre todo a las preguntas 1 y 3 (véase tabla 6).

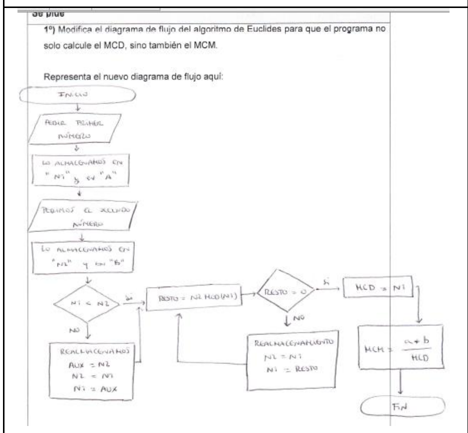
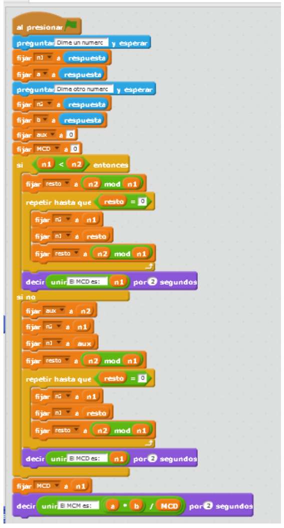
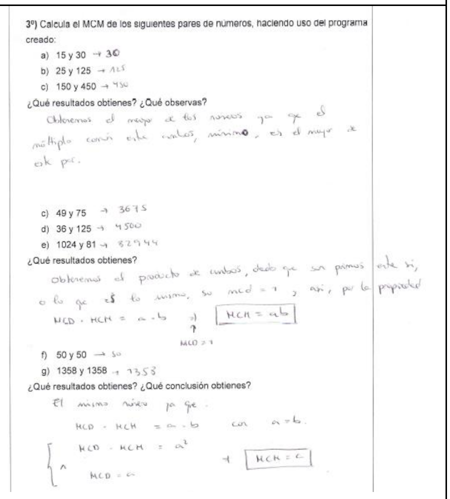
Tabla 6. Resultados de los grupos 1 y 2 a la ficha 5

Grupo 1	Grupo 2
 **PREGUNTA 1:** Ejecuta el programa y calcula el MCD de: a) 15 y 30 → 15 b) 25 y 125 → 25 c) 150 y 450 → 150 ¿Qué resultados obtienes? ¿Qué observas? En todos los casos la solución es el menor número, ya que el mayor es múltiplo del anterior. **PREGUNTA 2:** Ejecuta el programa y calcula el MCD de: a) 49 y 75 → 1 b) 36 y 125 → 1 c) 1024 y 81 → 1 ¿Qué resultados obtienes? ¿Por qué crees que ocurre esto? Siempre obtenemos como resultado 1 dado que son primos entre sí **PREGUNTA 3:** Ejecuta el programa y calcula el MCD de: a) 50 y 50 → 50 b) 1358 y 1358 → 1358 ¿Qué resultados obtienes? ¿Qué conclusión obtienes? El mismo, ya que ambos números son iguales.	 **PREGUNTA 1:** Ejecuta el programa y calcula el MCD de: a) 15 y 30 → 15 b) 25 y 125 → 25 c) 150 y 450 → 150. ¿Qué resultados obtienes? ¿Qué observas? Que cuando uno divide al otro sale el más pequeño. **PREGUNTA 2:** Ejecuta el programa y calcula el MCD de: a) 49 y 75 → 1 b) 36 y 125 → 1 c) 1024 y 81 → 1 ¿Qué resultados obtienes? ¿Por qué crees que ocurre esto? Todas las divisiones salen 1 porque los números son primos entre sí. **PREGUNTA 3:** Ejecuta el programa y calcula el MCD de: a) 50 y 50 → 50 b) 1358 y 1358 → 1358 ¿Qué resultados obtienes? ¿Qué conclusión obtienes? Sale el mismo número como MCD.

Transcurridos los 25 primeros minutos de la 4ª sesión, se comienza con la ficha 6. Se dedican 50 minutos a la misma. En esta ficha, se pretende que los alumnos hagan una modificación al algoritmo para que éste muestre también el M.C.M. En la ficha se pide el diagrama de flujo del algoritmo, el programa en Scratch y la respuesta a una serie de preguntas. Los resultados a dicha ficha se comentan a continuación:

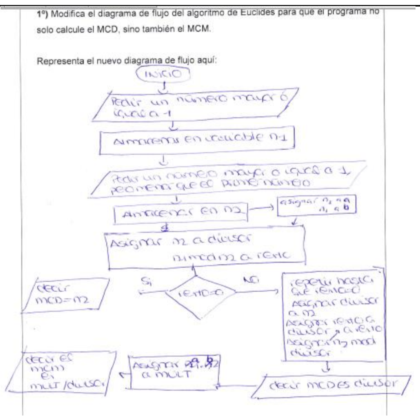
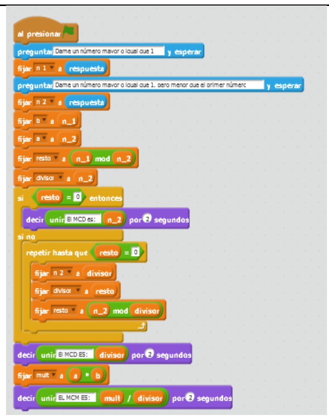
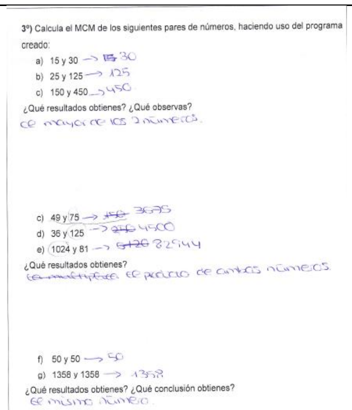
El grupo 1 diseña correctamente el diagrama de flujo, traduce correctamente dicho algoritmo a Scratch y responde correctamente a las preguntas que se efectúan.

Tabla 7. Resultado del grupo 1 a la ficha 6

Grupo 1		
Diagrama de flujo	Algoritmo	Conclusiones
		

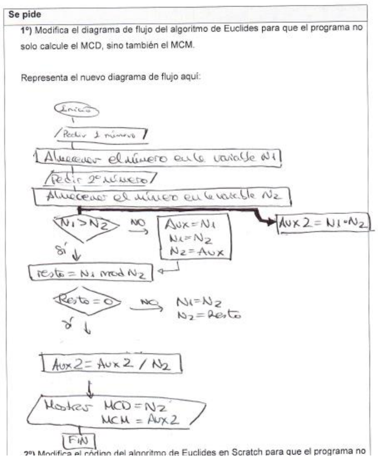
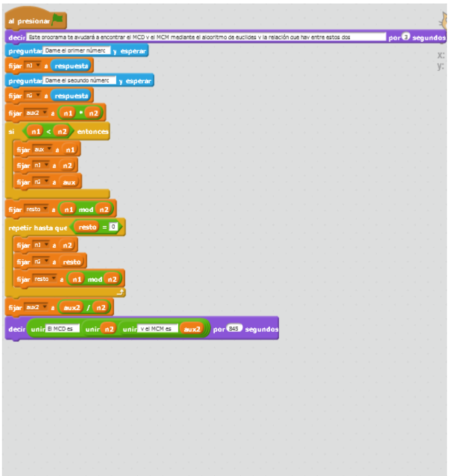
El grupo 2 es uno de los grupos que más dificultades ha tenido a lo largo del desarrollo de las tareas. Dicho grupo no logra diseñar correctamente el diagrama de flujo. Sin embargo, logra traducir su pensamiento correctamente a Scratch y consigue que finalmente funcione el algoritmo. Las respuestas a las preguntas que aparecen en la ficha, son también correctas, pero no argumentan el porqué. Se limitan a dar respuestas escuetas y sin justificación (tabla 8).

Tabla 8. Resultado del grupo 2 a la ficha 6

Grupo 2		
Diagrama de flujo	Algoritmo	Conclusiones
		

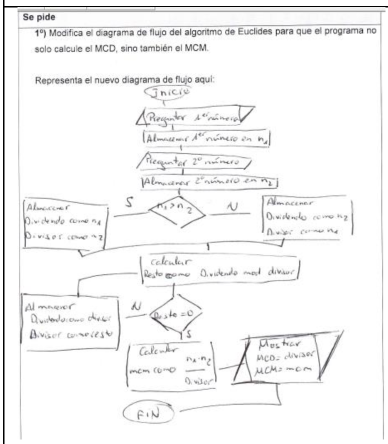
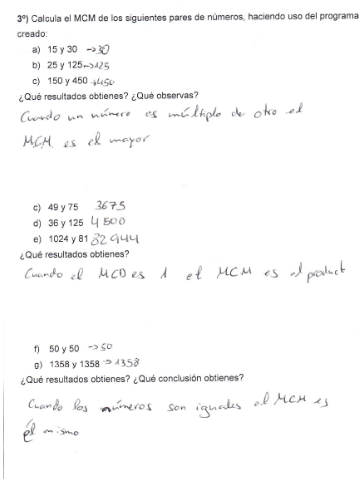
El grupo 3 no diseña correctamente el diagrama de flujo ya que olvidan marcar en un rectángulo la condición que está dentro del bucle, además de faltarles la línea de flujo que regresa esa condición al inicio del mismo. No obstante, traducen correctamente el algoritmo a Scratch y responden correctamente a las preguntas que se efectúan (tabla 9).

Tabla 9. Resultado del grupo 3 a la ficha 6

Grupo 3		
Diagrama de flujo	Algoritmo	Conclusiones
		

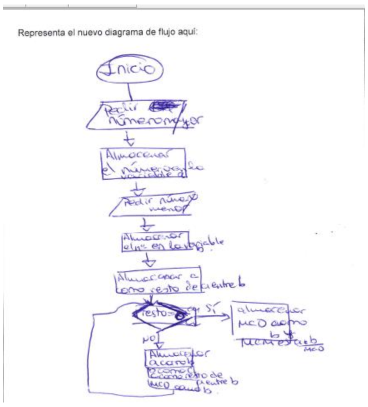
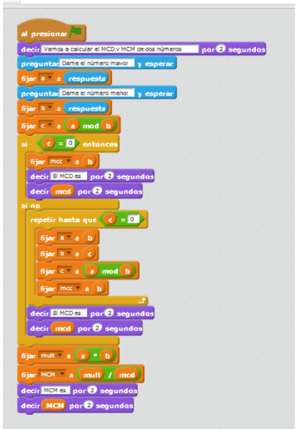
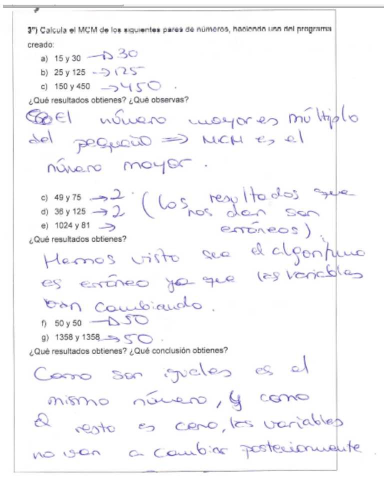
El grupo 4 diseña correctamente el diagrama de flujo, traduce correctamente dicho algoritmo a Scratch y responde correctamente a las preguntas que se efectúan (tabla 10).

Tabla 10. Resultado del grupo 4 a la ficha 6

Grupo 4		
Diagrama de flujo	Algoritmo	Conclusiones
		

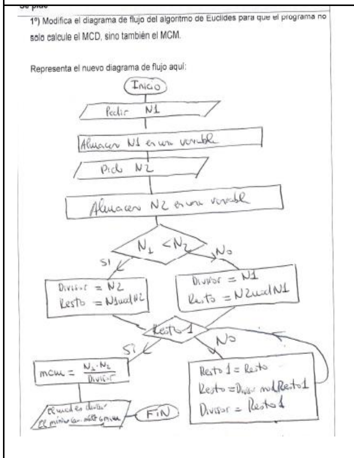
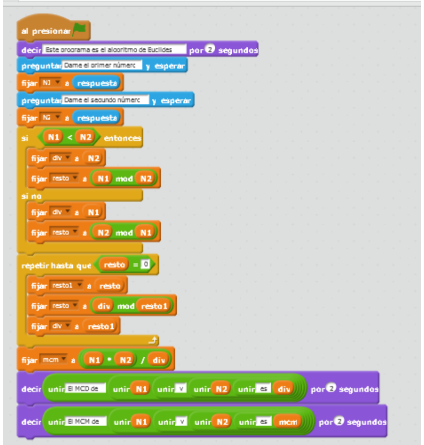
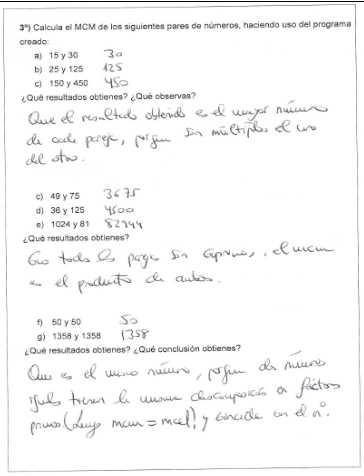
El grupo 5, sin embargo, devuelve el peor resultado de entre todos los grupos. Diseñan mal el diagrama de flujo, ya que, al utilizar continuamente las variables a y b y no guardar los valores iniciales pedidos en dos variables distintas, utilizan dichas variables pasadas por el bucle, para calcular el M.C.M. Por tanto, el algoritmo no devuelve el M.C.M de forma correcta. Además de esto, la escritura en el diagrama de flujo no es nada clara. Como consecuencia, arrastran este error al implementar el algoritmo en Scratch. A la hora de dar las respuestas, denotan que saben que su algoritmo es incorrecto; no obstante, no saben argumentar el porqué (tabla 11).

Tabla 11. Resultado del grupo 5 a la ficha 6

<table>
<tr><th colspan="3">Grupo 5</th></tr>
<tr><th>Diagrama de flujo</th><th>Algoritmo</th><th>Conclusiones</th></tr>
<tr><td></td><td></td><td></td></tr>
</table>

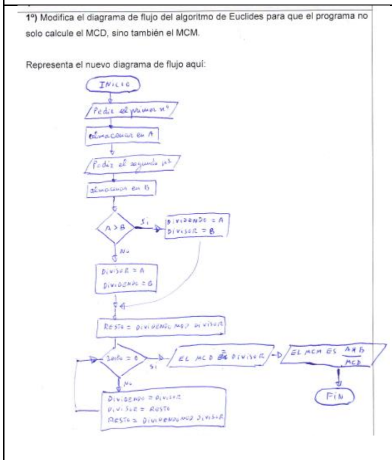
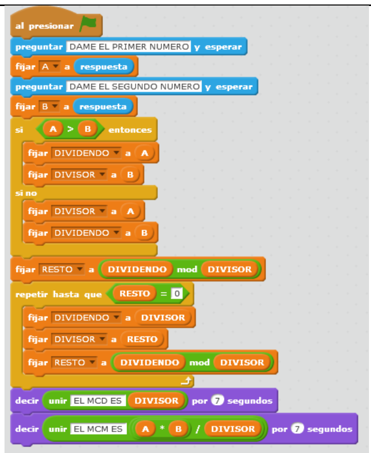
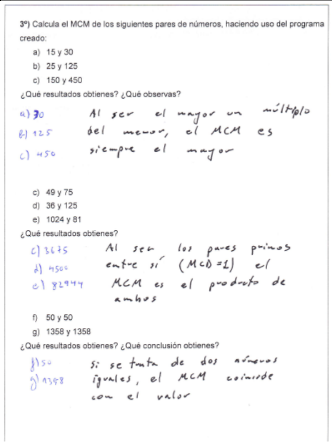
El grupo 6 diseña de forma incorrecta el diagrama de flujo ya que la condición para ingresar en el bucle es incorrecta. No obstante, implementan de forma correcta el algoritmo en Scratch y responden a las preguntas de forma correcta y argumentada (tabla 12).

Tabla 12. Resultado del grupo 6 a la ficha 6

Grupo 6		
Diagrama de flujo	Algoritmo	Conclusiones
		

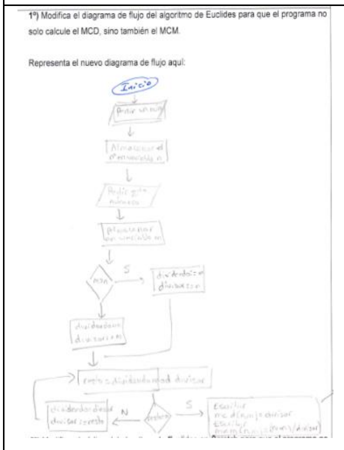
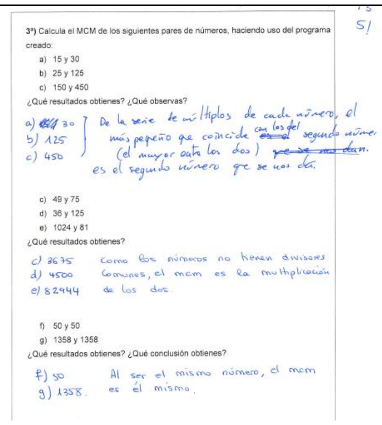
El grupo 7 diseña correctamente el diagrama de flujo, traduce correctamente dicho algoritmo a Scratch y responde correctamente a las preguntas que se efectúan (tabla 13).

Tabla 13. Resultado del grupo 7 a la ficha 6

Grupo 7		
Diagrama de flujo	Algoritmo	Conclusiones
		

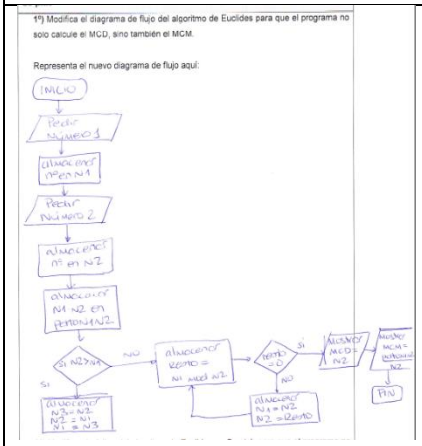
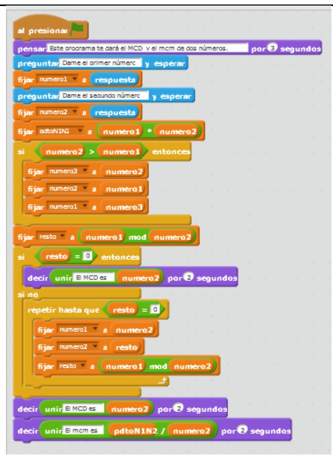
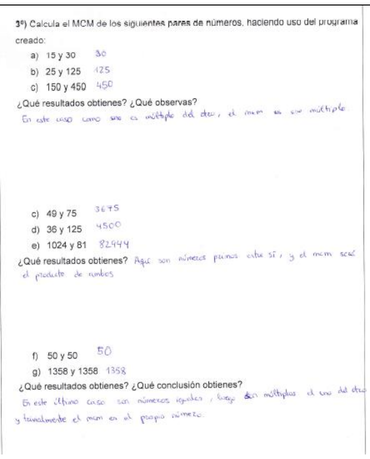
El grupo 8 diseña correctamente el diagrama de flujo, traduce correctamente dicho algoritmo a Scratch y responde correctamente a las preguntas que se efectúan, aunque no queda muy claro el argumento que dan en la pregunta 1 (tabla 14).

Tabla 14. Resultado del grupo 8 a la ficha 6

Grupo 8		
Diagrama de flujo	Algoritmo	Conclusiones
		

El grupo 9 diseña correctamente el diagrama de flujo, a excepción de una línea de flujo que falta uniendo la condición SI del condicional, con el resto del programa. Traduce correctamente dicho algoritmo a Scratch y responde correctamente a las preguntas que se efectúan (tabla 15).

Tabla 15. Resultado del grupo 9 a la ficha 6

Grupo 9		
Diagrama de flujo	Algoritmo	Conclusiones
		

6. Conclusiones y perspectivas futuras

En este documento se ha trabajado el M.C.D, un concepto que aparece por primera vez en 5º de primaria y que por su naturaleza y el tratamiento que se realiza habitualmente del mismo en el aula, se puede situar dentro de lo que denominamos pensamiento matemático elemental. A pesar de esto, podemos señalar que este concepto, tratado como se ha hecho en este documento, forma parte del Pensamiento Matemático Avanzado. Las razones que justifican este hecho son:

- Se ha trabajado el razonamiento lógico-matemático y la formalización a la hora de diseñar el diagrama de flujo para el algoritmo de Euclides. Esto puede observarse a través de la ficha 2 y de los resultados que se obtuvieron de la realización de la misma (véase tabla 2 del documento). Como consecuencia de esto podemos señalar que se ha cumplido el objetivo 1.
- Se ha potenciado la abstracción a la hora de traducir el diagrama de flujo para el algoritmo de Euclides al lenguaje de programación Scratch. Este hecho se constata a través de la realización y los resultados de la ficha 5. Por este motivo podemos decir que se ha cumplido el objetivo 2.
- Se han formulado hipótesis que han podido ser contrastadas mediante la ejecución del programa para diversos casos particulares. Este hecho se puede observar a través de la comprobación que hacen los alumnos de la aplicación creada en la ficha 5. En cualquier caso, la formulación de hipótesis ha sido un elemento presente durante todo el proceso de implementación del algoritmo de Euclides en Scratch. Como consecuencia de lo anterior podemos señalar que se ha cumplido el objetivo 3.
- Se ha trabajado el concepto de generalización. Los alumnos han podido inferir resultados generales, a partir de la ejecución del algoritmo para diversos casos particulares. Esto se observa tanto en la ficha 5 como en la ficha 6. Por ello podemos decir que se ha cumplido el objetivo 4.
- Ha supuesto un duro trabajo para alumnos con una sólida formación matemática. Tal y como se ha detallado en el apartado resultados, no todos los grupos han sido capaces de llevar a cabo de forma satisfactoria todas las fichas.
- Por último señalamos que los alumnos han valorado el algoritmo implementado en Scracth como un modelo matemático que, una vez probado y verificado, permite calcular el M.C.D y el M.C.M de cualquier par de números naturales. Esto, que se observa una vez que los alumnos han completado las fichas 5 y 6, implica la consecución del objetivo 5.

Como perspectivas futuras, se encuentra la puesta en práctica de esta secuencia de actividades con alumnos de 3º ESO, en la asignatura de Matemáticas para las Enseñanzas Académicas. Tal y como señala Carralero (2011), a partir de este curso los alumnos empiezan a poseer un desarrollo lógico-abstracto que permitiría desarrollar en ellos algunos elementos del Pensamiento Matemático Avanzado. Cabe señalar que los tiempos dedicados a cada una de las fichas que se han trabajado en este documento deberán ser modificados y adaptados a los alumnos de secundaria, pero el contenido de las mismas prevemos permanecerá intacto.

Referencias

[1] BOYER, Carl Benjamin. *Historia de la Matemática*, pp. 241, 146, 485, Alianza Editorial, Madrid, 2010.

[2] BARRERA, L. (2013). *Algoritmos y programación para la enseñanza y aprendizaje de la matemática escolar.* Actas del VII CIBEM ISSN, 2301(0797), 6680.

[3] BROUSSEAU, G. (1997). *La théorie des situations didactiques.* Le cours de Montréal 1997. Curso dictado por G. Brousseau con motivo de la atribución del título de Doctor Honoris

Causa de la Universidad de Montréal, Québec.

[4] BROUSSEAU, G. (1998). *Théorie des situations didactiques.* Grenoble: La Pensée Sauvage-Éditions, coll. Recherches en didactique des mathématiques.

[5] BROUSSEAU, G. (2011). La *théorie des situations didactiques en mathématiques* (Vol. 5, No. 1, pp. 101-104). Presses universitaires de Rennes.

[6] CARRALERO, N. (2011). *Scratch. Programación fácil para primaria y secundaria.* Revista digital sociedad de la Información, 29, 1-10.

[7] CARVAJAL, S., FONT, V., y GIMÉNEZ, J. (2014). *Uso de las TIC en las prácticas de la formación de profesores de secundaria de matemáticas.* Revista del Congrés Internacional de Docència Universitària i Innovació (CIDUI), (2).

[8] DREYFUS, T. (1990). *Advanced mathematical thinking.* En Nesher, P. Y Kilpatrick, J. (Eds.), Mathematics and cognition. Cambridge: Cambridge University Press, 113-133.

[9] DREYFUS, T. (1991). *Advanced mathematical thinking processes.* En Tall, D. (Ed), Advanced mathematical thinking. Dordrecht: Kluwer Academic Publisher, 25-41.

[10] EDWARDS, B.S., DUBINSKY, E., y MCDONALD, M.A. (2005). *Advanced mathematical thinking.* Mathematical Thinking and Learning, 7(1), 15-25.

[11] FERNÁNDEZ, J., y MUÑOZ, J. (2007). *Las TIC como herramienta educativa en matemáticas.* Unión, 119-147.

[12] GAIRÍN, J.M., y SANCHO, J. (2002). *Números y algoritmos.* Madrid: Síntesis.

[13] KIDRON, I. y DREYFUSS, T. (2014). *Proof image.* Educational Studies in Mathematics, vol. 87, 297-321.

[14] IFRAH, G. (2008). *Historia universal de las cifras.* Madrid: Espasa Calpe.

[15] PASCUAL, M.A. (1998). *La nueva frontera educativa con nuevas tecnologías.* Nuevas Tecnologías, medios de comunicación y educación. Formación inicial y permanente del profesorado. Madrid: CCS.

[16] PÉREZ, R.P. (1998). Nuevas *tecnologías y nuevos modelos de enseñanza.* Nuevas tecnologías, medios de comunicación y educación: formación inicial y permanente del profesorado (pp. 105-150). Madrid: CCS.

[17] RICOY, M.C., y COUTO, M.J. (2012). *El acercamiento al contexto profesional como móvil para indagar sobre las TIC: un estudio cualitativo.* Revista Complutense de Educación, 23 (2), 443-461.

[18] ROBERT, A. y SCHWARZENBERGER, T. (1991). *Research in the teaching and Learning Mathematics at an Advanced Level.* En Tall, D (Ed.). Advanced Mathematical thinking. Dordrecht: Kluwer Academic Publisher, 25-41.

[19] SKEMP, R. (1993). *Psicología del aprendizaje de las matemáticas.* Madrid: Morata.

[20] TALL, D. (1991). *The psychology of advanced mathematical thinking.* Advanced mathematical thinking (pp. 3-21). Springer Netherlands.

[21] USISKIN, Z. (1998). *Paper-and-Pencil Algorithms in a Calculator-and-Computer Age.* En L. J. Morrow y M. J. Kenney (Eds.) The Teaching and Learning of Algorithms in School

Mathematics (pp. 7-20). Reston, VA: National Council of Teachers of Mathematics.

[22] VIDAL, C.L., CABEZAS, C., PARRA, J.H., y LÓPEZ, L.P. (2015). *Experiencias Prácticas con el Uso del Lenguaje de Programación Scratch para Desarrollar el Pensamiento Algorítmico de Estudiantes en Chile.* Formación universitaria, 8 (4), 23-32.

Sobre los autores:

Nombre: Miguel Ángel Baeza Alba
Correo Electrónico: mbaeza@ucm.es
Institución: Universidad Complutense de Madrid, España.

Nombre: Francisco Javier Claros Mellado
Correo Electrónico: fclaros@ucm.es
Institución: Universidad Complutense de Madrid, España.

Nombre: María Teresa Sánchez Compaña
Correo Electrónico: teresasanchez@uma.es
Institución: Universidad de Málaga, España.

Nombre: Mónica Arnal Palacián
Correo Electrónico: monica.arnal@urjc.es
Institución: Universidad Rey Juan Carlos, España.