

**Diseño de actividades con el software de geometría dinámica CaRMetal, para la enseñanza de la relación de la gráfica y la ecuación de la recta.**

**Autores:**

**Yuri Paola Cárdenas Sánchez  
Katherin Esmeralda Pinzón Amaya**

**Dirigido por:**

**Martín Eduardo Acosta Gempeler**

**Universidad Distrital Francisco José de Caldas  
Facultad de Ciencias y Educación.  
Proyecto Curricular: Licenciatura en Educación básica con énfasis en matemáticas  
Bogotá D.C.  
Mayo de 2016**

## Tabla de contenido

<b>Introducción</b>	<b>5</b>
<b>Problema</b>	<b>7</b>
<b>Antecedentes y Justificación</b>	<b>8</b>
<b>Utilización de software de geometría dinámica en la educación matemática.</b>	<b>8</b>
<b>Enseñanza de la recta y la conversión de los registros semióticos gráfico y algebraico.</b>	<b>8</b>
<b>La utilización de tecnología para el trabajo de conversión de registros semióticos.</b>	<b>9</b>
<b>Objetivos</b>	<b>11</b>
<b>General</b>	<b>11</b>
<b>Específicos</b>	<b>11</b>
<b>Metodología</b>	<b>12</b>
<b>Marco Teórico Conceptual y Análisis Preliminar</b>	<b>13</b>
<b>Análisis a priori</b>	<b>16</b>
<b>Desarrollo de la propuesta</b>	<b>16</b>
<b>Descripción general del medio (figura de CaRMetal)</b>	<b>16</b>
Sobre los botones en la figura:	16
Sobre la cuadrícula y el sistema de coordenadas:	17
Sobre los registros gráficos y algebraicos:	17
<b>Primer Tomo: Conversión del registro Algebraico al Gráfico</b>	<b>17</b>
Descripción específica del medio- tomo 1:	17
Diseño de las tareas- Tomo 1	19
<b>Tarea 1- Experimentación.</b>	<b>19</b>
<b>Tarea 1- Anticipación:</b>	<b>20</b>
<b>Tarea 2- Experimentación con el numerador de la pendiente.</b>	<b>21</b>
<b>Tarea 2- Experimentación con el denominador de la pendiente.</b>	<b>22</b>
<b>Tarea 2- Anticipación.</b>	<b>23</b>
<b>Tarea 3- Experimentación.</b>	<b>24</b>
<b>Tarea 3- Anticipación.</b>	<b>25</b>
<b>Segundo Tomo: Conversión del registro Gráfico al Algebraico</b>	<b>26</b>
Descripción específica del medio- tomo 2:	26
Diseño de las actividades- Tomo 2.	28
<b>Tarea 1- Experimentación.</b>	<b>28</b>
<b>Tarea 1- Anticipación.</b>	<b>28</b>
<b>Tarea 2- Experimentación.</b>	<b>30</b>
<b>Tarea 2- Anticipación.</b>	<b>31</b>
<b>Tarea 3- Experimentación.</b>	<b>32</b>

<b>Tarea 3- Anticipación.</b>	33
<b>Pilotaje</b>	<b>35</b>
<b>Tomo 1- Conversión del registro algebraico al gráfico.</b>	<b>35</b>
<b>Tarea 1- Experimentación</b>	35
<b>Tarea 1- Anticipación</b>	36
<b>Tarea 2- Experimentación numerador pendiente</b>	38
<b>Tarea 2- Experimentación denominador pendiente</b>	42
<b>Tarea 2- Anticipación</b>	44
<b>Tarea 3- Experimentación</b>	45
<b>Tarea 3- Anticipación</b>	46
<b>Conclusiones generales primer tomo</b>	46
<b>Tomo 2- Conversión del registro gráfico al algebraico.</b>	<b>47</b>
<b>Tarea 1- Experimentación</b>	47
<b>Tarea 1- Anticipación</b>	48
<b>Tarea 2- Experimentación</b>	48
<b>Tarea 2- Anticipación</b>	49
<b>Tarea 3- Experimentación</b>	51
<b>Tarea 3- Anticipación</b>	53
<b>Conclusiones generales segundo tomo</b>	53
<b>Conclusiones y Recomendaciones</b>	<b>54</b>
<b>Referentes Bibliográficos</b>	<b>56</b>
<b>Anexos</b>	<b>59</b>
<b>Programación de la secuencia de actividades</b>	<b>59</b>
<b>Los scripts.</b>	59
<b>I. Tareas.</b>	59
<b>II. Verificar.</b>	64
<b>Sobre las funciones VerificarTarea_ ()</b>	66
<b>function VerificarTarea1() {</b>	67
<b>function VerificarTarea2() {</b>	72
<b>function VerificarTarea3() {</b>	79
<b>function VerificarTarea3Parte1() {</b>	85
<b>function VerificarTarea3Parte2() {</b>	90
<b>function VerificarTarea4() {</b>	96
<b>function VerificarTarea5() {</b>	105
<b>function VerificarTarea6() {</b>	111
<b>Sobre las funciones EntradaInteractiva_ ()</b>	120
<b>function entradainteractivaD () {</b>	120
<b>function entradainteractivaH () {</b>	121

<b>function entradainteractivaK ()</b>	121
<b>function entradainteractivaL ()</b>	122
III. <b>Continuar</b>	124
<b>Sobre las funciones ContinuarTarea_()</b>	128
<b>function ContinuarTarea2()</b>	128
<b>function ContinuarTarea3()</b>	135
<b>function ContinuarTarea3Parte1()</b>	139
<b>function ContinuarTarea3Parte2()</b>	143
<b>function ContinuarTarea4()</b>	148
<b>function ContinuarTarea5()</b>	155
<b>function ContinuarTarea6()</b>	162
IV. <b>Volver a la tarea</b>	171
<b>Sobre las funciones VolverTarea_()</b>	175
<b>function VolverTarea1()</b>	175
<b>function VolverTarea2() {</b>	181
<b>function VolverTarea3()</b>	188
<b>function VolverTarea3Parte1()</b>	193
<b>function VolverTarea3Parte2()</b>	198
<b>function VolverTarea4()</b>	204
<b>function VolverTarea5()</b>	214
<b>function VolverTarea6()</b>	223
V. <b>Reiniciar</b>	234
VI. <b>Contadores</b>	243
VII. <b>Ayuda</b>	247
<b>Sobre las funciones AyudaTarea_()</b>	250
<b>Configuración de la pantalla</b>	<b>252</b>

## Introducción

Este trabajo de grado nace de nuestro interés personal, cuando en nuestro proceso de formación como licenciadas en Educación Básica con Énfasis en Matemáticas de la Universidad Distrital Francisco José de Caldas, pudimos entender que generalmente no se aprovecha el potencial de la tecnología para producir aprendizaje. Según Sacco (2007) “existen gabinetes de computación arrumados y una de las causas (..) suele ser que los docentes y profesionales que podrían aprovecharlos no saben cómo hacerlo” (p. 2). Efectivamente, aunque una solución a este problema radique en la capacitación que necesita el docente, consideramos que un aporte para lograr esto es mostrar cómo hacer posible un diseño pedagógico de actividades. Nos proponemos entonces aprovechar al máximo el potencial del software de geometría dinámica CaRMetal procurando enriquecer con éste la enseñanza y el aprendizaje de las matemáticas.

Nosotros nos hemos centrado en el aprendizaje del concepto de la recta pues según Roldán (2013) este concepto “hace grandes aportes al desarrollo del pensamiento variacional que a su vez resulta fundamental en procesos de generalización y desarrollo del pensamiento abstracto” (p 9). Así, hemos diseñado actividades para la construcción de las relaciones entre la ecuación de una recta y su gráfica en el plano cartesiano con el software CaRMetal.

Para el desarrollo de esas actividades hemos empleado como metodología una ingeniería didáctica de la que desarrollamos sólo las dos primeras fases. La primera fase de análisis preliminar se enfoca en realizar un abordaje teórico para precisar cómo podemos tener un diseño fundamentado. De esta manera hemos tomado a Brousseau como nuestro fundamento didáctico y a Duval como nuestro fundamento cognitivo, con el propósito de involucrar el aprendizaje por adaptación y la conversión de registros de representación semiótica en el diseño.

La segunda fase de diseño y análisis a priori, presenta cómo fueron en general diseñadas las actividades, cuáles son los objetivos de aprendizaje de éstas y cómo garantizamos que el estudiante irá construyendo las relaciones entre la gráfica y la ecuación de la recta. Además, realizamos un pilotaje con una estudiante de octavo grado que nos permitió confirmar nuestros análisis e identificar aspectos que deberían modificarse en el diseño.

Una de las características de CaRMetal que aprovechamos en el diseño es la posibilidad de programar algunas acciones, lo que nos permitió automatizar la evaluación de las respuestas de los estudiantes para regular la secuencia de aprendizaje. En los anexos presentamos una descripción detallada de esa programación.

El presente documento está organizado en nueve apartados:

1. **PROBLEMA**, donde se define cuál y cómo surge la pregunta orientadora que determina el desarrollo y realización del tema de este trabajo.
2. **ANTECEDENTES Y JUSTIFICACIÓN**, donde se describen algunos aportes importantes para el desarrollo de este trabajo que nos llevan a reflexionar por qué se necesita el diseño de las actividades.
3. **OBJETIVOS**, donde establecemos las metas generales y específicas a alcanzar en el diseño.
4. **METODOLOGÍA**, donde se muestran los pasos que se siguen para el desarrollo de esta propuesta, especificando las fases que se ejecutaron de la ingeniería didáctica y como se ven desarrolladas en el trabajo.

5. MARCO TEÓRICO Y ANÁLISI PRELIMINAR, donde se exponen los referentes conceptual y didáctico tomados en el diseño y se precisa cómo se pueden ver incluidos en el mismo.
6. ANÁLISIS A PRIORI, donde se define el desarrollo de la propuesta, es decir, cómo fueron diseñadas las actividades en general y cómo pueden garantizar el aprendizaje de los estudiantes en relación al objeto matemático (relación entre la ecuación y la gráfica de la recta) escogido.
7. PILOTAJE, donde se presenta un análisis de lo que era esperado con las actividades y lo que realmente sucede en la experimentación de las mismas con una estudiante de octavo grado.
8. CONCLUSIONES Y RECOMENDACIONES, donde presentamos los resultados generales de nuestro trabajo, algunas proyecciones y algunas sugerencias sobre las que nos gustaría que la comunidad educativa matemática reflexionara.
9. REFERENTES BIBLIOGRÁFICOS, donde se señalan las fuentes de indagación que exclusivamente se han tomado en la escritura de este documento.

Finalmente, en los anexos se incluye: una descripción detallada sobre la programación que se ha hecho con el software para el diseño de las actividades y un CD con las dos figuras elaboradas en CaRMetal de los dos tomos de actividades diseñados para la conversión del registro algebraico al gráfico y para la conversión del registro gráfico al algebraico en este trabajo.

## Problema

Reconocemos que las nuevas tecnologías tienen un potencial transformador en la enseñanza y aprendizaje; sin embargo, para que estas tecnologías desarrollen todo su potencial de transformación “deben integrarse en el aula y convertirse en un instrumento cognitivo capaz de mejorar la inteligencia y potenciar la aventura de aprender” (Beltrán, Martín y Pérez, 2003, p. 7). Cómo usar entonces esas nuevas tecnologías en pro de aprovechar todo su potencial transformador en la enseñanza, es una de las primeras preguntas que surgen. Según Guzmán (1993) el profesor cuando trata de aceptar el desafío de diseñar e incorporar actividades con ayuda de las nuevas tecnologías debe empezar a pensar que con éstas se deben contestar preguntas como:

¿Cuáles serán las cuestiones o problemas que propondrán en las clases para que den sentido al conocimiento que están construyendo los alumnos, y cuáles serán las tareas rutinarias a delegar en estas nuevas tecnologías? ¿Cómo usarlas para que permitan establecer un trabajo en la clase más centrado en la búsqueda de soluciones a problemas, en tratar de probar conjeturas, etc., y no en un mero trabajo mecánico de cálculo algorítmico? (párr. 10).

Estas dos afirmaciones permiten ver que, aunque el diseño es necesario para la incorporación de las tecnologías, conseguirlo no es una tarea fácil. Es necesario ir más allá de ejercitar procesos y de repetir lo que se hace con el papel y el lápiz. Para el diseño, debe reconocerse cómo se construye el conocimiento de los estudiantes, qué tareas se les puede proponer usando tecnología para lograr una construcción de conocimientos, cuáles son las interacciones posibles con ésta, qué es lo que permiten esas interacciones y cuáles de esas interacciones ayudan a que se dé el aprendizaje. De esta manera se puede garantizar el aprendizaje por medio del diseño. Esas cuestiones nos conducen a mirar el fundamento cognitivo y didáctico que debe tener cualquier propuesta de diseño.

El contenido matemático de este trabajo son las relaciones entre la gráfica y la ecuación de la recta, pues tal como lo propone el Ministerio de Educación Nacional (2008) en uno de los Estándares Básicos de Competencias en Matemáticas, se debe lograr que los estudiantes identifiquen las “relaciones entre propiedades de las gráficas y propiedades de las ecuaciones algebraicas” (p. 87). Así que la pregunta general de este trabajo es:

¿Qué elementos cognitivos, didácticos y tecnológicos contribuyen a diseñar actividades para la enseñanza de la relación entre la gráfica y la ecuación de la recta utilizando el software CaRMetal?

## **Antecedentes y Justificación**

Para realizar nuestro trabajo revisamos algunos documentos sobre tres temas principales: 1) *la utilización de software de geometría dinámica en la educación matemática*, 2) *la enseñanza de la recta y la conversión de los registros semióticos gráfico y algebraico*, y 3) *la utilización de tecnología para el trabajo de conversión de registros semióticos*.

### **Utilización de software de geometría dinámica en la educación matemática.**

Sobre este tema se encuentran varios estudios referenciados para la enseñanza de un saber en específico. Gouvea (2005) y Bohórquez (2004), por ejemplo, han desarrollado investigaciones de aula, en las que se problematiza y reflexiona sobre las ventajas que tiene la incorporación de un software en el aula de clases.

Estos autores plantean que el software de geometría puede ser un material que permita un aprendizaje diferente a lo tradicional. Según Gouvea (2005), los software podrían contribuir “para el establecimiento de un ambiente de aprendizaje agradable y participativo” (p. 2), pues la enseñanza matemática se empieza a conectar con algo que hace parte del actual real de los estudiantes: las tecnologías informáticas. Sin embargo, el uso de un software no garantiza en sí un aprendizaje matemático (Bohórquez, 2004). Es decir, no basta con enseñar a manejar un software, debe buscarse y aprovecharse al máximo su potencial, creando prácticas que lleven a la comprensión de una noción matemática. El profesor debe reconocer que “la efectividad en el uso de un software para el aprendizaje matemático, y en particular para la comprensión de conceptos geométricos, puede relacionarse con el diseño pedagógico de actividades” (Bohórquez, 2004, p. 109).

Un software de geometría dinámica podría por ejemplo establecerse como una herramienta para: el proceso de toma de decisiones, reflexión, razonamiento y resolución de problemas (Bohórquez, 2004). El uso de un software puede servir como “una ventana hacia las concepciones de los alumnos, pues ayuda a la exteriorización del pensamiento por medio de la cantidad de acciones y experimentos que permite” (Bohórquez, 2004, p.108). Entonces, es fundamental analizar la pertinencia del software como herramienta de enseñanza.

De estos estudios podemos concluir que hay que prestarle una atención especial a las tareas que se proponen a los alumnos y en general a la actividad que realiza el alumno con el software. El uso de un software puede generar un ambiente diferente en el aula, pero las prácticas de enseñanza matemática no son buenas o malas por las herramientas que se usen, sino por el sentido pedagógico que tienen. Las herramientas y en este caso el software, no serviría de nada cuando no se tiene verdadera consciencia de lo que se va hacer con él.

### **Enseñanza de la recta y la conversión de los registros semióticos gráfico y algebraico.**

Según Duval (1981) la integración de los registros gráfico y algebraico requiere una interpretación global de las representaciones gráficas. Es decir, no considerar la recta como un conjunto de puntos sino como una unidad global. La interpretación global sirve para entender que existe una articulación entre los registros, lo que deja ver a la recta como “un objeto o entidad conceptual” (Benítez, 2010, p. 7) y no simplemente como el estudio de puntos aislados.

Las propuestas que se presentan para el trabajo de la conversión de registros en la construcción del concepto de recta, analizan las dificultades que los estudiantes tienen para relacionar la ecuación y la gráfica de la recta. Un ejemplo son los estudios de Peralta (s.f), Díaz, Haye, Montenegro y Córdoba (2013), quienes examinaron las dificultades de aprendizaje de la función lineal en estudiantes de reciente ingreso a carreras de ingeniería y universitarios de segundo semestre del área económico administrativa. Ellos concluyeron que esas dificultades están ligadas a los errores en la coordinación de los registros gráfico y algebraico. Los errores que los autores encuentran “se observan tanto en la noción de pendiente como en la de ordenada” (Díaz et al., 2013, p. 11).

La dificultad de la noción de pendiente revela su presencia con mayor fuerza “cuando el registro de partida es el gráfico” (Peralta, s.f, p. 171); el estudiante no comprende la alteración de la pendiente en la gráfica, limitándose solo a una caracterización algebraica de ésta. La exclusiva memorización del algoritmo para determinar la pendiente de una recta conlleva a que los estudiantes ni siquiera asocien “el signo de la pendiente con la inclinación de la recta” (ídem, s.f, p. 172).

Para superar estas dificultades, los autores reconocen que el trabajo con la conversión es ineludible, sólo ella logra la articulación entre los registros. Sin embargo, después de aplicar estrategias para el trabajo con esa conversión, “una considerable proporción de los estudiantes no logró establecer una articulación exenta de errores de sus representaciones” (Díaz et al., 2013). Los autores manifiestan que, aunque los estudiantes reconocen la conversión de registros, “no han mostrado una articulación espontánea y libre de contradicciones de sus diversas representaciones” (Peralta, s.f, p.173). Creemos que esto puede deberse a que las representaciones usadas para la conversión de los registros gráfico y algebraico en estos estudios, son estáticas. Al igual que los autores, consideramos la necesidad de elaborar propuestas que permitan el trabajo con la conversión de estos registros para la construcción del concepto de la recta. El software de geometría permite representaciones y manipulaciones dinámicas en los dos registros, elemento que podría tener un mayor impacto en el aprendizaje.

Este trabajo de grado propone entonces, un material de aula para la construcción de la noción de la recta, con actividades que requieren la conversión de los registros algebraico y gráfico de forma dinámica y en ambos sentidos.

### **La utilización de tecnología para el trabajo de conversión de registros semióticos.**

Para este tema es relevante el estudio de Lupiáñez y Moreno (2001), quienes hacen una reflexión en torno a la implementación de las tecnologías y su relación con los sistemas de representación y las representaciones semióticas. En este estudio se realiza un análisis en torno a la aplicación de la calculadora como un componente tecnológico. La calculadora graficadora puede considerarse como uno de los primeros instrumentos que permiten trabajar la conversión de registros semióticos. Los autores señalan la necesidad de: “conjugar las posibilidades gráficas de estas tecnologías con las analíticas, pues de estas relaciones es de donde surge un conocimiento útil y consistente” (p. 296). El uso de “la calculadora permite ver los objetos matemáticos como *manipulables*, y permite actuar sobre ellos. Por eso, la fuerza de la tecnología está basada, en gran medida, en esa reificación de objetos y relaciones matemáticas” (ídem, 2001, p. 297).

Nosotros reconocemos el potencial que Lupiañez y Moreno (2001) atribuyen a la calculadora graficadora. En nuestro caso, el uso de un software quiere aprovechar esas ventajas que al igual que en la calculadora permite la conversión entre los registros gráfico y algebraico.

Por otra parte, según Benítez (2010):

Los trabajos realizados por Moschkovich, Schoenfeld y Arcavi (1993) con apoyo de la computadora, mencionan la importancia que tiene para el estudiante moverse con mayor facilidad en el simbolismo algebraico, mientras éste mantenga un significado visual. Además, estiman que el razonamiento visual para observar las transformaciones como movimientos “sobre” o “del” plano tienen un poder potencialmente cognitivo en el alumno, el cual requiere tiempo, oportunidades y recursos para hacer construcciones, conjeturas y modificaciones (p. 7).

La ventaja de la utilización del software en nuestro trabajo, va más allá de que sea un software gratuito o de que permita un ambiente diferente de aula; radica en el tipo de tareas que permite realizar y en los procesos cognitivos que refuerzan estas tareas. A diferencia de la calculadora, en el software es posible trabajar la conversión del registro gráfico al registro algebraico y del registro algebraico al registro gráfico.

## Objetivos

### General

Proponer actividades de clase para estudiantes de octavo o noveno, haciendo uso de la conversión de registros semióticos aprovechando el potencial del software “CaRMetal” para promover el aprendizaje por adaptación.

### Específicos

- Diseñar actividades para trabajar la conversión del registro algebraico al registro gráfico y viceversa.
- Automatizar<sup>1</sup> la evaluación de las respuestas de los alumnos y la gestión de la secuencia de tareas.

---

<sup>1</sup> La automatización de la evaluación y de la gestión de la secuencia de tareas, es la programación con la que el software CaRMetal determina y evalúa de forma automática, el paso del usuario por los ejercicios y las tareas propuestas. Para encontrar una mayor explicación sobre qué significa esa automatización en el diseño, véase el apartado: Marco Teórico Conceptual y Análisis Preliminar.

## **Metodología**

Se tomará como base metodológica una ingeniería didáctica, puesto que ésta “se diseña para provocar, de manera controlada, la evolución de las concepciones” (Artigue, Douady y Moreno, 1995, p. 42).

La ingeniería didáctica comprende cuatro fases: análisis preliminar, diseño y análisis a priori, experimentación y análisis a posteriori. Sin embargo, en este proyecto se desarrollan sólo las dos primeras fases, incluyendo un pilotaje para controlar el diseño.

A continuación presentamos los análisis preliminares que comprenden un análisis cognitivo, fundamentado en la conversión de registros de representación semiótica de Duval, y un análisis didáctico, fundamentado en el aprendizaje por adaptación de la Teoría de Situaciones Didácticas de Brousseau. Posteriormente presentamos el diseño y análisis a priori donde se muestra el control de diversas variables didácticas y su efecto en el aprendizaje de los estudiantes.

## Marco Teórico Conceptual y Análisis Preliminar

Para el desarrollo de este trabajo de grado, se toman como base dos referentes teóricos: el Análisis sobre el uso de Registros Semióticos de Duval, del cual tomamos la necesidad de articular diferentes registros semióticos para la construcción de conceptos, y el concepto de aprendizaje por adaptación de la Teoría de Situaciones Didácticas planteada por Brousseau. En este sentido se acogerá a Duval como nuestro fundamento cognitivo y a Brousseau como nuestro fundamento didáctico.

Duval (2012) establece que “el acceso a los objetos matemáticos no es ni sensorial ni instrumental, como en física o en química, sino que éste pasa por la producción de representaciones semióticas que no deben jamás ser confundidas con los objetos que ellas representan” (p. 7).

“Desde el punto de vista cognitivo, comprender en matemática es, antes que todo, reconocer los objetos matemáticos representados” (Duval, 2012, p. 7); sin embargo, para hacer este reconocimiento es indispensable contar por lo menos con dos registros semióticos que puedan ser articulados entre sí, pues “un aprendizaje específicamente centrado en el cambio y en la coordinación de los diferentes registros de representación, produce efectos espectaculares sobre las macro-tareas de producción y de comprensión” (idem, 1999, p. 46).

Según Duval (2012, p. 8) “ser capaz de reconocer el mismo objeto en dos representaciones semióticas implica que, si solo una es dada, es posible convertirla espontáneamente en otra o igualmente en una tercera”. En palabras de San Martín (2007), Duval resalta “la importancia de la coordinación de diversos registros de representación semiótica. Establece que muchas de las dificultades encontradas por los estudiantes pueden ser descritas y explicadas como una falta de coordinación de registros de representación” (p. 4).

Como consecuencia, es necesario lograr que los estudiantes hagan conversiones de una representación a otra, en ambos sentidos. En nuestro caso debemos proponer la conversión entre los registros algebraico (ecuación) y geométrico (gráfica), como registros congruentes.

EL software de geometría dinámica CaRMetal permite representar simultáneamente una recta en los registros gráfico y algebraico y observar en cada uno de ellos los efectos del cambio en el otro. Nuestras actividades consisten en presentar las dos representaciones simultáneamente y solicitarle al estudiante un cambio en una de las dos, sobre la cual no puede actuar directamente, sino que debe actuar sobre la otra.

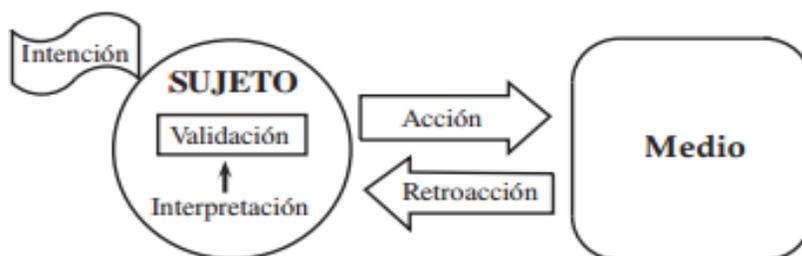
Para el fundamento didáctico de nuestras actividades, tomamos la Teoría de Situaciones Didácticas de Guy Brousseau, específicamente el concepto de aprendizaje por adaptación como el que se produce en la interacción entre un sujeto (estudiante) y un medio.

Según Brousseau (1986):

El alumno aprende adaptándose a un medio que es factor de contradicciones, de dificultades, de desequilibrios, un poco como lo ha hecho la sociedad humana. Este saber fruto de la adaptación del alumno, se manifiesta por respuestas nuevas que son la prueba del aprendizaje (p. 14).

Gracias a esta interacción, el alumno construye el conocimiento como “una respuesta natural en unas condiciones relativamente particulares, condiciones indispensables para que tengan sentido para él” (Brousseau, 1986, p. 7).

Según Acosta, Monroy. y Rueda (2010), el aprendizaje por adaptación comprende 5 elementos sintetizados en el siguiente diagrama:



Tomado de Acosta, Monroy. y Rueda (2010).

El sujeto tiene una *intención*, propósito u objetivo. Para lograrla, realiza una *acción* sobre el medio. El medio reacciona a la acción del sujeto (*retroacción*). El sujeto *interpreta* la retroacción del medio, es decir, toma conciencia de ella y le da un sentido. Finalmente, el sujeto *valida* su acción, decidiendo si su intención inicial fue cumplida o no. Si el sujeto decide que la acción que hizo no le sirvió para alcanzar lo que se proponía (invalidación- validación negativa), entonces, cambia de acción. Si el sujeto decide que la acción sí le sirvió para alcanzar lo que se proponía (validación positiva), refuerza la acción. El cambio de acción y el refuerzo de acción, son los signos observables del aprendizaje.

Según Rueda y Niño (2013):

La interacción del sujeto con el medio es cíclica, así que no hay que verla como un proceso que termina con la validación, sino que la validación conduce a un nuevo ciclo.

De los cinco elementos de la interacción el más importante desde el punto de vista didáctico, ya que es condición fundamental para el aprendizaje por adaptación, es la validación. Si el proceso de interacción no conduce a la validación de las acciones por parte del sujeto, no se produce un aprendizaje por adaptación (p. 25).

El medio es determinante en el aprendizaje por adaptación, ya que es el que determina las posibles acciones que puede realizar el alumno y las retroacciones que recibe éste para validar o invalidar sus acciones. Debe haber, por tanto, un control del medio que garantice que no haya por ejemplo retroacciones distractoras. Si toda acción fuera posible, no todas las retroacciones del medio generarían un aprendizaje que guíe al conocimiento personal. Las retroacciones del medio deben permitir invalidar las estrategias no matemáticas y validar las estrategias matemáticas.

En el diseño de actividades consideramos el software como un medio material con el cual interactúa el estudiante. Nos limitamos a examinar esa interacción y sus implicaciones en el aprendizaje. El proceso de enseñanza es mucho más amplio y comprende interacciones sociales entre los estudiantes y entre los estudiantes y el profesor, necesarias para la reconstrucción

colectiva del saber cultural. Las actividades que proponemos constituyen entonces solo una parte del proceso de enseñanza correspondiente a lo que la teoría llama *situación adidáctica*, centrada en la interacción entre el estudiante y el medio. Reconocemos que para la utilización en clase será necesario reflexionar sobre la fase didáctica en la que la interacción entre el profesor y los estudiantes es fundamental.

### **CaRMetal como medio:**

El software CaRMetal “es un programa desarrollado desde el 2006 por el profesor de matemáticas de Millau/Francia Eric Hakenholz. Fue desarrollado “a partir del motor matemático del software C.a.R. (Regla y compás)” (Debrabant, s.f, párr. 1). pero se le han añadido muchas más funciones a partir del software de origen. Una de las novedades de CaRMetal es la integración de “un editor de script que abarca el JavaScript más los comandos de geometría dinámica” (Debrabant, s.f, párr. 2).

CaRMetal es un software libre y multiplataforma que posee tres características que nosotros privilegiamos:

- Las representaciones algebraicas, que pueden ser calculadas, escritas y modificadas por el usuario.
- Las representaciones gráficas que permiten la visualización de objetos geométricos y su manipulación directa por medio del arrastre, cambio del aspecto, etc.
- Los controles y la programación como elementos que no son objetos reales del programa, pero que permiten poner en relación los dos sistemas de representación mencionados mediante el uso de scripts. Esto último no quiere decir que en sí no existan relaciones de los dos registros, al contrario, el software permite mostrar las relaciones matemáticas que tienen estos sin la necesidad de la programación.

Por medio de dichas características podemos condicionar el medio: software CaRMetal, para determinar las acciones que el estudiante puede realizar y las retroacciones que este recibirá. Las retroacciones que permiten validar o invalidar las llamamos retroacciones matemáticas, pues tienen que ver con el comportamiento matemático de los objetos representados. Pero el software comprende otro tipo de retroacciones que no están directamente ligadas al comportamiento de los objetos matemáticos representados y las llamamos retroacciones didácticas, pues tienen que ver con la organización de una secuencia de tareas y la evaluación del trabajo del alumno.

Es posible hacer que el software reaccione de manera automática para evaluar las respuestas del estudiante a las tareas propuestas, y con base en esta evaluación le presente o no nuevos ejercicios o tareas. Estas son decisiones didácticas que normalmente están a cargo del profesor, pero que pueden automatizarse por medio de la programación de scripts.

## Análisis a priori

### Desarrollo de la propuesta

Las actividades fueron diseñadas para realizarse con estudiantes de grados octavo y noveno, ya que uno de los Estándares Básicos de competencias en Matemáticas de este ciclo especifica que los estudiantes deberían: “identificar las relaciones entre propiedades de las gráficas y propiedades de las ecuaciones algebraicas” (MEN, 2008, p. 87). Se requiere entonces que el estudiante comprenda algunas de las nociones básicas del pensamiento variacional y del pensamiento geométrico para el trabajo con nuestras actividades como: comprensión de la letra como variable, comprensión de la variación y la covariación, trabajo con ecuaciones e igualdades, coordenadas cartesianas y ubicación en el plano, clasificación de triángulos, congruencia y puntos de intersección.

La idea básica de las actividades es presentar dos registros semióticos simultáneamente: el algebraico y el gráfico, y solicitar la modificación de uno de los dos registros, pero dando solo posibilidad de acción en el otro registro, con el fin de provocar la conversión. Tenemos entonces dos tomos de actividades: uno que trabaja con la conversión del registro algebraico al gráfico, y el otro que trabaja con la conversión del registro gráfico al algebraico.

### Descripción general del medio (figura de CaRMetal)

Antes de describir cada tomo de actividades, presentamos una descripción general del medio (figura de CaRMetal), mencionando los aspectos comunes a todos los ejercicios. La figura muestra la pantalla dividida en tres partes: Una franja verde, que muestra las representaciones algebraicas, una franja gris, con botones para mostrar el enunciado, verificar, continuar y volver a una tarea; y una franja blanca que comprende un sistema de coordenadas con una cuadrícula y las representaciones gráficas de la(s) recta(s).

#### Sobre los botones en la figura:

Una claridad que debe hacerse es que en general las actividades manejan 4 botones diferentes, así:

1. Tarea: Muestra el enunciado de la tarea.
2. Verificar: Muestra un mensaje de evaluación de la tarea (“muy bien o inténtalo de nuevo”).

3. Continuar: Este botón permite pasar a la siguiente tarea, por lo que sólo aparece cuando el estudiante ha resuelto por lo menos 5 ejercicios correctamente y el número de incorrectos es menor que el de correctos.
4. Volver a..: Permite al estudiante regresar a una de las tareas anteriores, por lo que solo aparece tras haber desarrollado la primera tarea en cada tomo.

#### Sobre la cuadrícula y el sistema de coordenadas:

La cuadrícula y el sistema de coordenadas son independientes del sistema de coordenadas del software, con el fin de evitar la posibilidad de agrandar o achicar la figura, ya que esta posibilidad, conduce, en algunos casos, a tener elementos de la figura por fuera de la pantalla.

#### Sobre los registros gráficos y algebraicos:

En la pantalla aparecen simultáneamente la ecuación de la recta y su representación gráfica. Toda modificación de la ecuación produce una modificación de la representación gráfica y viceversa. Las rectas pueden tener en general cualquier pendiente, pero se evita el caso de la recta vertical, es decir, con pendiente indeterminada.

### **Primer Tomo: Conversión del registro Algebraico al Gráfico**

En este tomo se solicita obtener una representación gráfica de la recta, a partir de la modificación de su ecuación.

#### Descripción específica del medio- tomo 1:

En el registro gráfico se pueden observar dos rectas, una azul y una roja. Sobre la recta azul aparecen dos puntos: Un punto B en la intersección con el *eje y*, y un punto A sobre la cuadrícula. En el registro algebraico aparecen tres ecuaciones: una ecuación verde con la forma general:  $y=ax+b$ , y dos ecuaciones rojas (con la pendiente escrita como fracción y con la pendiente escrita como aproximación decimal).

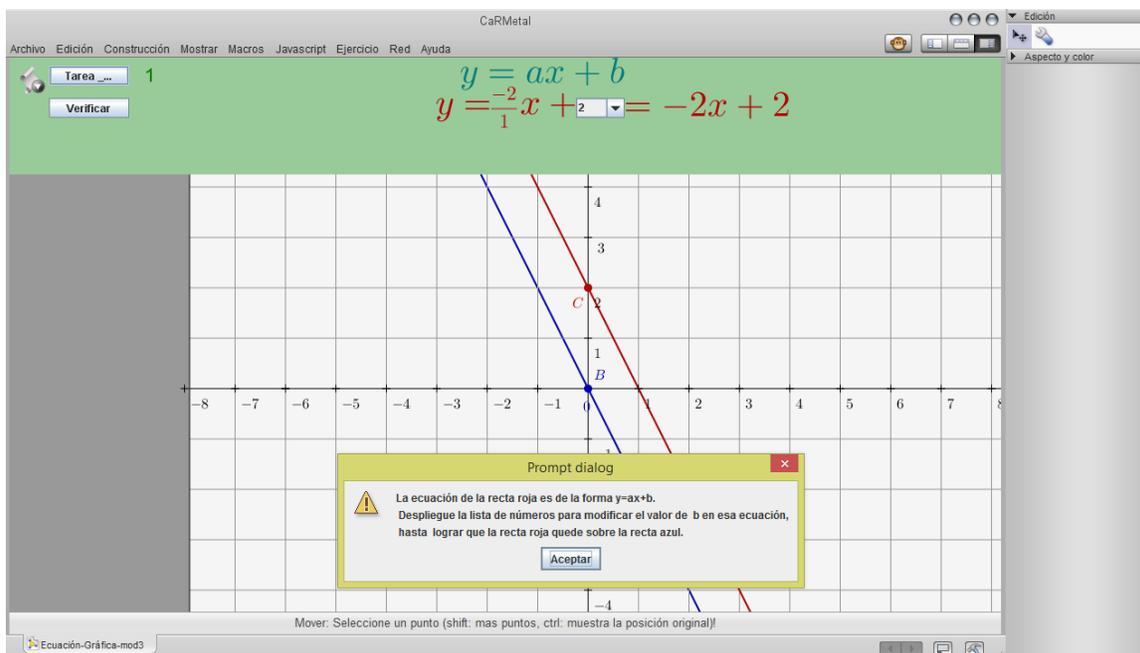


Imagen 1. Actividad del Tomo 1 Tarea 1.

Se le pide al estudiante que modifique la ecuación roja para lograr que la recta roja se superponga con la recta azul.

En este tomo hay 3 tareas que requieren:

1. Modificar el término independiente de la ecuación.
2. Modificar el coeficiente de la variable de la ecuación.
3. Modificar el término independiente y el coeficiente de la variable de la ecuación.

Cada tarea tiene dos partes: una, en la que el alumno puede experimentar modificando la ecuación roja y observando el efecto de esa modificación en la gráfica (al finalizar esta primera parte se solicita al estudiante que describa ese efecto, para que tome consciencia de las relaciones entre los dos registros); y otra, en la que el estudiante debe anticipar el valor que debe darle a la ecuación para lograr la superposición de la recta roja y la recta azul.

Cada parte comprende 5 o más ejercicios. En cada ejercicio aparece una posición diferente de la recta azul. Cuando el estudiante resuelve correctamente un ejercicio se presenta automáticamente uno nuevo (la posición de la recta azul se escoge al azar garantizando que pase por un punto B de coordenadas enteras sobre el eje y, y un punto A de coordenadas enteras visible en la pantalla). Una vez que el estudiante ha completado por lo menos 5 ejercicios y el número de intentos correctos es mayor que el número de intentos incorrectos, se le da la posibilidad de pasar a la siguiente tarea.

### **Tarea 1- Experimentación.**

Consiste en modificar el término independiente de la ecuación de la recta roja, mediante una lista desplegable, hasta hacer que las rectas roja y azul queden superpuestas.

Las rectas son paralelas, pero no pasan por el mismo punto de intersección con el *eje y*. El punto de intersección de la recta roja con el *eje y* aparece con nombre C. El punto de intersección de la recta azul con el *eje y* aparece con nombre B. Al cambiar el número de la lista desplegable correspondiente al término independiente de la ecuación roja, la posición de la recta roja cambia; se conserva la pendiente de ésta, y se modifica la posición de C, de manera que su ordenada corresponde al término independiente de la ecuación roja.

Existen dos posibilidades de acción del estudiante:

1. Elegir un valor que no corresponde al punto B, corte de la recta azul con el *eje y*. La retroacción del medio va a mostrar la recta roja paralela a la recta azul, pero cortando el *eje y* en un punto diferente al B de la recta azul, de manera que el estudiante puede invalidar esta acción.
2. Elegir el valor que corresponde al punto B, corte de la recta azul con el *eje y*. La retroacción del medio va a mostrar la recta roja sobre la recta azul por lo que el estudiante puede validar esta acción.

Cuando el estudiante considere que ha resuelto la tarea, debe hacer clic en el botón “verificar”. El software compara el valor del término independiente con el valor del punto de corte de la recta azul con el *eje y*.<sup>2</sup> Si los dos valores son diferentes, muestra como mensaje: “La recta roja aún no está sobre la recta azul, vuelve a intentarlo”. Si los dos valores coinciden muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva recta” y genera un nuevo ejercicio.

Se espera que el estudiante haga varios intentos de modificación del valor del término independiente y tome conciencia del efecto de este cambio en la posición de la recta roja. Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, se le solicita escribir en su cuaderno la relación entre el término b de la ecuación y la posición de la recta roja. Se espera que el estudiante mencione que el número correspondiente a b de la ecuación es el número en el que la recta corta al *eje y*.

---

<sup>2</sup> Esta programación del software, se describe con mayor detalle en el anexo de este documento.

### **Tarea 1- Anticipación:**

Consiste en modificar el término independiente de la ecuación de la recta roja, ingresando en una casilla el valor numérico que debe tener éste para hacer que las rectas roja y azul queden superpuestas. El estudiante deberá entonces anticipar el valor específico que tendrá el término independiente de la ecuación de la recta roja para que quede sobre la recta azul.

Las rectas son paralelas, pero no pasan por el mismo punto de intersección con el *eje y*. El punto de intersección de la recta roja con el *eje y* aparece con nombre D. Al ingresar un número en la casilla correspondiente al término independiente de la ecuación roja, la posición de la recta roja cambia; se conserva la pendiente de ésta, y se modifica la posición de D, de manera que su ordenada corresponde al término independiente de la ecuación roja.

Existen dos posibilidades de acción del estudiante:

1. Introducir un valor que no corresponde al punto B, corte de la recta azul con el *eje y*. La retroacción del medio va a mostrar la recta roja paralela a la recta azul, pero cortando el *eje y* en un punto diferente a B, de manera que el estudiante puede invalidar esta acción.
2. Introducir el valor que corresponde al punto B, corte de la recta azul con el *eje y*. La retroacción del medio va a mostrar la recta roja sobre la recta azul por lo que el estudiante puede validar esta acción.

Es posible que al comenzar esta tarea el estudiante aún no haya tomado conciencia de la relación entre el término independiente y el punto de corte de la recta con el *eje y*, de manera que no pueda anticipar correctamente el valor que debe escribir en la casilla. Por esta razón, a partir de esta tarea aparece el botón “volver a...” que en este caso permite regresar a la experimentación para determinar la relación entre el término independiente de la ecuación y el punto de corte de la recta con el *eje y*.

Al igual que en la tarea anterior, cada vez que el estudiante ha resuelto la tarea, debe hacer clic en el botón “verificar”. El software compara el valor del término independiente introducido por el estudiante con el valor del punto de corte de la recta azul con el *eje y*. Si los dos valores son diferentes, muestra como mensaje: “La recta roja aún no está sobre la recta azul, vuelve a intentarlo”. Si los dos valores coinciden muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva recta” y genera un nuevo ejercicio.

Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, podrá pasar a la siguiente tarea con el botón continuar, el cual aparecerá en ese momento en su pantalla.

## **Tarea 2- Experimentación con el numerador de la pendiente.**

Consiste en utilizar una lista desplegable para modificar el numerador del coeficiente de la variable, hasta hacer que las rectas roja y azul queden superpuestas.

Las rectas tienen el mismo punto B de intersección con el eje de las ordenadas. La recta azul pasa por un punto A de la cuadrícula y la recta roja pasa por un punto M de la cuadrícula. El punto A y el punto M tienen la misma abscisa, pero no la misma ordenada.

En el desarrollo de esta tarea, existen dos posibilidades de acción del estudiante:

1. Elegir un valor que no corresponde a la diferencia de las ordenadas en los puntos de la recta azul. La retroacción del medio va a mostrar que la recta roja no está sobre la recta azul, por lo que el estudiante puede invalidar esta acción.
2. Elegir el valor que corresponde a la diferencia de las ordenadas en los puntos de la recta azul. La retroacción del medio va a mostrar la recta roja sobre la recta azul por lo que el estudiante podrá validar esta acción.

Cuando el estudiante ha resuelto la tarea, oprime el botón “verificar”. El software compara el valor del coeficiente de la variable de la ecuación de la recta, con el valor de la pendiente de la recta azul. Si los dos valores son diferentes, muestra como mensaje: “La recta roja aún no está sobre la recta azul, vuelve a intentarlo”. Si los dos valores coinciden muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva recta” y genera un nuevo ejercicio.

Es posible que el estudiante no encuentre una relación entre el coeficiente de la variable y la gráfica de la recta, a pesar de observar los cambios en la gráfica cuando modifica el numerador del coeficiente de la variable. En efecto, para entender esta relación es necesario tener en cuenta dos puntos de la recta y las relaciones entre sus ordenadas. Por esta razón en ésta y las otras dos partes de esta tarea se incluye una ayuda que está disponible al hacer clic sobre el botón con el mismo nombre (“ayuda”). Cuando el estudiante hace clic sobre el botón “ayuda”, aparece el mensaje: “Modifica el valor del numerador del término  $a$  de la ecuación, hasta hacer que los triángulos queden superpuestos e iguales”. También aparecen dos triángulos rectángulos: uno azul y uno rojo; el triángulo azul tiene como vértices los puntos A y B de la recta azul, y el triángulo rojo tiene como vértices los puntos B y M de la recta roja. Al lado de los catetos del triángulo rojo aparecen números que corresponden respectivamente a la diferencia de las ordenadas y a la diferencia de las abscisas de los puntos M y B. Se espera que el estudiante constataste que el número correspondiente al numerador del coeficiente de la variable es el cateto “vertical” del triángulo rojo y que el número correspondiente al denominador del coeficiente de la variable es el cateto “horizontal” del triángulo rojo.

Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, se le solicitará escribir en su cuaderno la relación entre el numerador del coeficiente de la variable de la ecuación y la posición de la recta roja. Se espera que el estudiante mencione que el número asignado al numerador del coeficiente de la variable, es la distancia recorrida verticalmente a partir de los dos puntos que se muestran de la recta roja, y que sus signos corresponden a la dirección con la que se ha hecho ese movimiento: positivo si es hacia arriba, negativo si es hacia abajo.

### **Tarea 2- Experimentación con el denominador de la pendiente.**

Consiste en utilizar una lista desplegable para modificar el denominador del coeficiente de la variable, hasta hacer que las rectas roja y azul queden superpuestas.

Las rectas tienen el mismo punto B de intersección con el *eje y*. La recta azul pasa por un punto A de la cuadrícula y la recta roja pasa por un punto G de la cuadrícula. El punto A y el punto G tienen la misma ordenada, pero no la misma abscisa. El punto A y el punto B no tienen la misma ordenada, evitando que la recta azul tenga como pendiente cero, pues la tarea quedaría resuelta.

En el desarrollo de esta tarea, existen dos posibilidades de acción del estudiante:

1. Elegir un valor que no corresponde a la diferencia de las abscisas en los puntos de la recta azul. La retroacción del medio va a mostrar que la recta roja no está sobre la recta azul, por lo que el estudiante puede invalidar esta acción.
2. Elegir el valor que corresponde a la diferencia de las abscisas en los puntos de la recta azul. La retroacción del medio va a mostrar la recta roja sobre la recta azul, por lo que el estudiante puede validar esta acción.

Cuando el estudiante oprime el botón “verificar” el software compara el valor del coeficiente de la variable de la ecuación de la recta, con el valor de la pendiente de la recta azul. Si los dos valores son diferentes, muestra como mensaje: “La recta roja aún no está sobre la recta azul, vuelve a intentarlo”. Si los dos valores coinciden muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva recta” y genera un nuevo ejercicio.

En esta tarea también está disponible la ayuda descrita anteriormente, pero con un mensaje diferente: “Modifica el valor del denominador del término  $a$  de la ecuación, hasta hacer que los triángulos queden superpuestos e iguales”, y con unos vértices para el triángulo rojo distintos, el punto G y el punto B.

Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces y el número de intentos fallidos sea menor al número de éxitos, se le solicitará escribir en su cuaderno la relación que

existe entre el denominador del coeficiente de la variable de la ecuación de la recta y la posición de la recta roja. Se espera que el estudiante mencione que el número asignado al denominador del coeficiente de la variable es la distancia recorrida horizontalmente a partir de los dos puntos que se muestran de la recta roja, y que sus signos corresponden a la dirección en el que se ha hecho ese movimiento: positivo si es hacia la derecha, negativo si es hacia la izquierda. El estudiante también puede decir que el valor del denominador corresponde a la coordenada en  $x$  del punto G.

### **Tarea 2- Anticipación.**

Consiste en modificar el numerador y denominador del coeficiente de la variable en la ecuación de la recta roja, ingresando en las casillas los valores correspondientes a cada uno de éstos, para hacer que las rectas roja y azul queden superpuestas. El estudiante deberá entonces anticipar el valor específico que tendrá el coeficiente de la variable de la ecuación de la recta roja, para que la recta roja quede sobre la recta azul.

Las rectas tienen el mismo punto B de intersección con el *eje y*, La recta azul pasa por un punto A de la cuadrícula y la recta roja pasa por un punto H de la cuadrícula. El punto A y el punto H tienen coordenadas distintas.

En el desarrollo de esta tarea, existen dos posibilidades de acción del estudiante:

1. Introducir valores que no corresponden a la pendiente de la recta azul. La retroacción del medio va a mostrar que la recta roja no está sobre la recta azul, por lo que el estudiante podrá invalidar esta acción.
2. Introducir valores que corresponden a la pendiente de la recta azul. La retroacción del medio va a mostrar la recta roja sobre la recta azul, por lo que el estudiante podrá validar esta acción.

Cuando el estudiante haga clic en el botón “verificar”, el software compara el valor del coeficiente de la variable introducido por el estudiante con el valor de la pendiente de la recta azul. Si los dos valores son diferentes, muestra como mensaje: “La recta roja aún no está sobre la recta azul, vuelve a intentarlo”. Si los dos valores coinciden muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva recta” y genera un nuevo ejercicio.

Es posible que al comenzar esta tarea el estudiante aún no haya tomado conciencia de la relación entre el coeficiente de la variable y la pendiente de la recta, por lo que no puede anticipar correctamente los valores que debe escribir en las casillas. En dicho sentido, en esta tarea aparece el botón “volver a...” que le permite regresar a la experimentación para determinar la relación entre el coeficiente de la variable y la pendiente de la recta. Del mismo modo, en esta

tarea también está disponible la ayuda descrita anteriormente, pero con un mensaje diferente: “Introduce en las casillas los valores del término  $a$  de la ecuación, para que los triángulos queden superpuestos e iguales”, y con unos vértices para el triángulo rojo distintos, el punto H y el punto B.

Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, podrá pasar a la siguiente tarea con el botón continuar, el cual aparecerá en ese momento en su pantalla.

### **Tarea 3- Experimentación.**

Consiste en utilizar tres listas desplegables para modificar el término independiente y el coeficiente de la variable en la ecuación de la recta roja, hasta hacer que las rectas roja y azul queden superpuestas.

Las rectas azul y roja tienen un punto que indica la intersección con el eje de las ordenadas (B e I, respectivamente) y otro que se ubica en la cuadrícula (A y J respectivamente). Estos puntos tienen distintas coordenadas.

El estudiante debe hacer uso de las estrategias construidas mediante las tareas anteriores; sin embargo, como las modificaciones de los parámetros de la ecuación de la recta roja deben hacerse por medio de listas desplegables, el estudiante tiene la posibilidad de experimentar para su solución. Esto último querría por ejemplo que el estudiante tome conciencia y modifique inicialmente el término independiente de la ecuación roja, hasta que coincida el punto I con el punto B, para luego modificar el coeficiente de la ecuación roja hasta que coincidan las pendientes de las dos rectas.

Existen cuatro posibilidades de acción del estudiante:

1. Los valores escogidos hacen que las rectas sean paralelas, es decir, que tienen la misma pendiente, pero con el punto de intersección con el *eje* y distintos.
2. Los valores escogidos hacen que las rectas tengan el mismo punto de intersección con el *eje* y, pero que la pendiente sea distinta.
3. Los valores escogidos hacen las rectas con pendientes y puntos de corte con el *eje* y, distintos.
4. Los valores escogidos hacen que las rectas tengan la misma pendiente y punto de intersección con el *eje* y.

Para los 3 primeros casos, las retroacciones del software mostrarán que las rectas no se superponen. El estudiante entonces, está en capacidad de invalidar su acción. En el último caso,

las retroacciones del software mostrarán que las rectas se superponen. El estudiante entonces, está en capacidad de validar su acción.

Cuando el estudiante hace clic sobre el botón “verificar”, el software comparará respectivamente, el valor del término independiente y del coeficiente de la variable, con el valor del punto de corte de la recta azul con el *eje y*, y la pendiente de la recta azul. Si los valores son diferentes, muestra como mensaje: “La recta roja aún no está sobre la recta azul, vuelve a intentarlo”. Si los valores coinciden muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva recta” y genera un nuevo ejercicio.

Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, podrá pasar a la siguiente tarea con el botón continuar, el cual aparecerá en ese momento en su pantalla.

### **Tarea 3- Anticipación.**

Consiste en utilizar las tres casillas para modificar el término independiente y el coeficiente de la variable en la ecuación de la recta roja, hasta hacer que las rectas roja y azul queden superpuestas. El estudiante deberá entonces anticipar el valor específico que tendrán el término independiente y el coeficiente de la variable de la ecuación de la recta roja, para que la recta roja quede sobre la recta azul.

Las rectas azul y roja tienen un punto que indica la intersección con el eje de las ordenadas (B y L, respectivamente) y otro que se ubica en la cuadrícula (A y K respectivamente). Estos puntos tienen distintas coordenadas.

Existen cuatro posibilidades de acción del estudiante:

1. Los valores escogidos hacen que las rectas sean paralelas, es decir, que tienen la misma pendiente, pero con el punto de intersección con el *eje y* distintos.
2. Los valores ingresados hacen que las rectas tengan el mismo punto de intersección con el *eje y*, pero que la pendiente sea distinta.
3. Los valores ingresados hacen las rectas con pendientes y puntos de corte con el *eje y*, distintos.
4. Los valores ingresados hacen que las rectas tengan la misma pendiente y punto de intersección con el *eje y*.

Para los 3 primeros casos, las retroacciones del software mostrarán que las rectas no se superponen por lo que el estudiante está en capacidad de invalidar su acción. En el último caso, las retroacciones del software mostrarán que las rectas se superponen por lo que el estudiante está en capacidad de validar su acción.

Cuando el estudiante hace clic en el botón “verificar”, el software compara respectivamente el valor del término independiente y del coeficiente de la variable de la ecuación roja, con el valor del punto de corte de la recta azul con el *eje y*, y la pendiente de la misma. Si los dos valores son diferentes, muestra como mensaje: “La recta roja aún no está sobre la recta azul, vuelve a intentarlo”. Si los dos valores coinciden muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva recta” y genera un nuevo ejercicio.

El estudiante podrá regresar a alguna tarea en la que modifique por separado o en conjunto los parámetros de la ecuación de la recta, cuantas veces quiera. Para esto está el botón “volver a...”.

Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, recibirá como mensaje: “¡Felicitaciones! Has terminado todas las tareas”

### **Segundo Tomo: Conversión del registro Gráfico al Algebraico**

En este tomo se solicita modificar la representación algebraica de la recta, a partir de la manipulación de su representación gráfica. El estudiante podrá utilizar las relaciones encontradas en el tomo anterior para resolver las tareas.

#### Descripción específica del medio- tomo 2:

En el registro algebraico aparecen cinco ecuaciones: una ecuación verde con la forma general  $y=ax+b$ , dos ecuaciones azules (con la pendiente en forma de fracción y con la pendiente en forma de aproximación decimal) y dos ecuaciones rojas (con la pendiente escrita como fracción y con la pendiente escrita como aproximación decimal). En el registro gráfico se puede observar una recta roja.

Se le pide al estudiante que mueva la recta roja para que la ecuación roja sea igual a la ecuación azul.

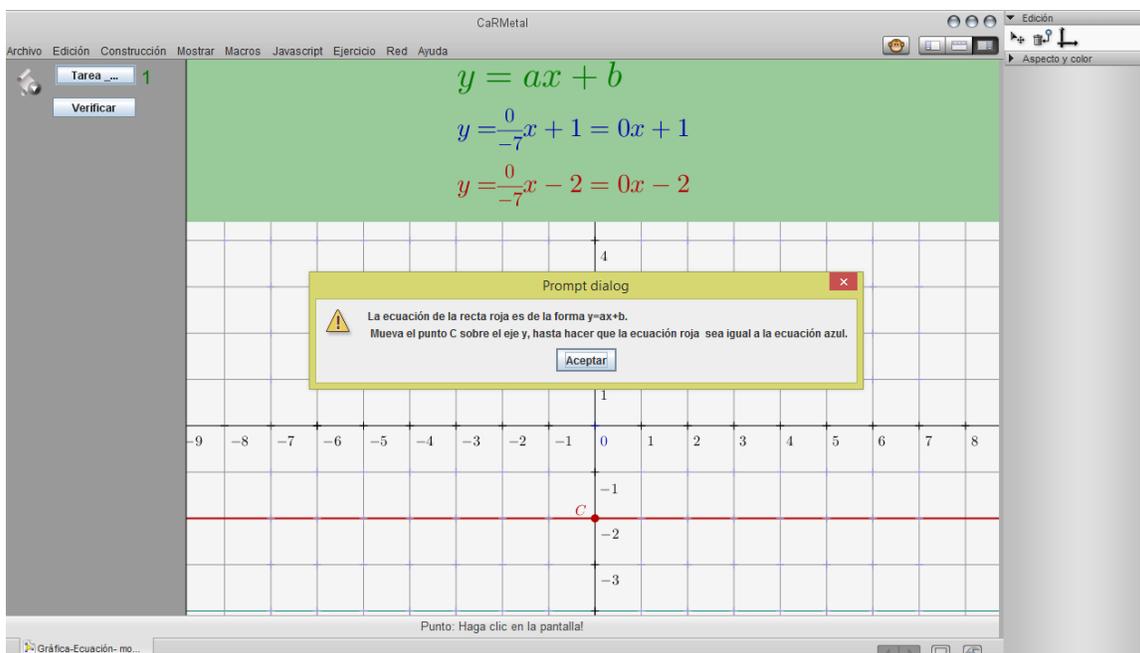


Imagen 2. Actividad del Tomo 2 Tarea 1.

En este tomo hay 3 tareas:

1. Modificar el punto de intersección de la recta con el *eje y*.
2. Modificar la inclinación de la recta.
3. Modificar el punto de intersección de la recta con el *eje y*, y su inclinación.

Cada tarea tiene dos partes: una, en la que el alumno puede experimentar modificando mediante el arrastre la gráfica roja y observando el efecto de esa modificación en la ecuación (al finalizar esta primera parte se solicita al estudiante que describa ese efecto, para que tome consciencia de las relaciones entre los dos registros); y otra, en la que el estudiante debe anticipar la posición que debe darle a la gráfica (seleccionando ésta en el sistema de coordenadas) para lograr la igualdad entre la ecuación roja y la ecuación azul.

Cada parte de la tarea comprende 5 o más ejercicios. En cada ejercicio aparecen dos ecuaciones azules diferentes. Cuando el estudiante resuelve correctamente el ejercicio, automáticamente se presenta uno nuevo (los parámetros de las ecuaciones azules se calculan con base en dos puntos A y B, ocultos, que se generan al azar, garantizando que B esté sobre el *eje y*, y A sobre la cuadrícula). Una vez que el estudiante ha completado por lo menos 5 ejercicios y el número de intentos correctos es mayor que el número de intentos incorrectos, se le da la posibilidad de pasar a la siguiente tarea.

### **Tarea 1- Experimentación.**

Consiste en arrastrar el punto C intersección entre la recta roja y el *eje* y sobre el eje de las ordenadas, hasta hacer que las ecuaciones roja y azul sean iguales.

Las ecuaciones tienen el mismo coeficiente de la variable, pero no tienen el mismo término independiente. Al arrastrar el punto C, que se mueve solo sobre la graduación del *eje* y, la ecuación roja cambia; se conserva el valor del coeficiente de la variable, y se modifica el valor del término independiente con el valor correspondiente a la ordenada de C.

Existen dos posibilidades de acción para el estudiante:

1. Mueve el punto C hasta un punto en el que su ordenada no coincide con el término independiente de la ecuación azul. La retroacción del medio va a mostrar que las ecuaciones roja y azul son diferentes y por lo tanto el estudiante puede invalidar esta acción.
2. Mueve el punto C hasta un punto en el que su ordenada coincide con el término independiente de la ecuación azul. La retroacción del medio va a mostrar la ecuación roja igual a la ecuación azul y por lo tanto el estudiante puede validar esta acción.

Cuando el estudiante haga clic en el botón “verificar” el software comparará la ordenada del punto C con el valor del término independiente de la ecuación azul. Si los dos valores son diferentes, muestra como mensaje: “La ecuación roja aún no es igual a la ecuación azul, vuelve a intentarlo”. Si los dos valores coinciden muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva ecuación” y genera un nuevo ejercicio.

Se espera que el estudiante haga varios intentos de modificación del punto C y tome conciencia del efecto de este cambio con el valor del término independiente de la ecuación roja. Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, se le solicita escribir en su cuaderno la relación entre el punto C que se mueve sobre el *eje* y, y la ecuación roja. Se espera que el estudiante mencione que la ordenada del punto C es el número correspondiente al parámetro b de la ecuación.

### **Tarea 1- Anticipación.**

Consiste en mover el punto D, seleccionando la posición en la que debe quedar sobre el *eje* y, para hacer que las ecuaciones roja y azul sean iguales. El estudiante deberá entonces anticipar el

valor específico de la ordenada de D sobre el *eje y*, para que la ecuación roja sea igual a la ecuación azul.

Las ecuaciones tienen el mismo coeficiente de la variable, pero no tienen el mismo término independiente. Al seleccionar el lugar donde quedará el punto D, la recta se mueve conservando su pendiente y la ecuación roja cambia; se conserva el valor del coeficiente de la variable, y se modifica el valor del término independiente con el valor correspondiente al punto de intersección de la recta roja con el *eje y*.

Existen dos posibilidades de acción para el estudiante:

1. Señala un punto cuya ordenada no coincide con el término independiente de la ecuación azul. La retroacción del medio va a mostrar que las ecuaciones roja y azul son diferentes y por lo tanto el estudiante puede invalidar esta acción.
2. Señala un punto cuya ordenada coincide con el término independiente de la ecuación azul. La retroacción del medio va a mostrar la ecuación roja igual a la ecuación azul y por lo tanto el estudiante puede validar esta acción.

El estudiante solo podrá seleccionar una vez la ubicación del punto D, pues éste se fijará automáticamente en ese lugar hasta que se oprima el botón “verificar”. Cuando el estudiante haga clic en el botón “verificar” el software examinará si el punto D está sobre el *eje y*, y posteriormente comparará la ordenada del punto D con el valor del término independiente de la ecuación azul. Si el punto D no está sobre el *eje y*, muestra como mensaje: “D no está sobre el *eje y*, vuelve a intentarlo”. Si D está sobre el *eje y*, pero la ordenada del punto D no coincide con el valor del término independiente de la ecuación azul, muestra como mensaje: “La ecuación roja aún no es igual a la ecuación azul, vuelve a intentarlo”. Si D está sobre el *eje y*, y la ordenada del punto D coincide con el valor del término independiente de la ecuación azul, muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva ecuación” y genera un nuevo ejercicio.

Es posible que al comenzar esta tarea el estudiante aún no haya tomado conciencia de la relación entre el punto de corte de la recta con el *eje y*, y el término independiente de la ecuación, de manera que no pueda anticipar correctamente el lugar o punto que debe seleccionar para que D se mueva hasta allí. Por esta razón, a partir de esta tarea aparece el botón “volver a...” que en este caso permite regresar a la experimentación para determinar la relación entre el punto de corte de la recta con el *eje y*, y el término independiente de la ecuación roja.

Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, podrá pasar a la siguiente tarea con el botón continuar, el cual aparecerá en ese momento en su pantalla.

## Tarea 2- Experimentación.

Consiste en arrastrar el punto G que pertenece a la recta roja sobre la cuadrícula, hasta hacer que las ecuaciones roja y azul sean iguales.

Las ecuaciones tienen el mismo valor del término independiente, pero no tienen el mismo coeficiente de la variable. Al arrastrar el punto G sobre la cuadrícula, la inclinación de la recta cambia y su ecuación correspondiente también. Al cambio de la ecuación de la recta roja, se conserva el valor del término independiente, y se modifica el valor del coeficiente de la variable.

En esta actividad no se realiza una variación por separado de la diferencia de las ordenadas y las abscisas en los puntos de la recta roja, sino que el cambio se hace en conjunto, esperando que el estudiante no encuentre mayor dificultad a partir de las acciones validadas con el primer tomo.

Existen dos posibilidades de acción para el estudiante:

1. Mueve el punto G hasta un punto en el que el valor de la pendiente de la recta no coincide con el coeficiente de la variable de la ecuación azul. La retroacción del medio va a mostrar que las ecuaciones roja y azul son diferentes, por lo que el estudiante puede invalidar esta acción.
2. Mueve el punto G hasta el punto en el que el valor de la pendiente de la recta coincide con el coeficiente de la variable de la ecuación azul. La retroacción del medio va a mostrar la ecuación roja igual a la ecuación azul, por lo que el estudiante puede validar esta acción.

Cuando el estudiante hace clic en el botón “verificar” el software compara la pendiente de la recta roja con el valor del coeficiente de la variable de la ecuación azul. Si los dos valores son diferentes, muestra como mensaje: “La ecuación roja no es igual a la ecuación azul, vuelve a intentarlo”. Si los dos valores coinciden muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva ecuación” y genera un nuevo ejercicio.

Se espera que el estudiante haga varios intentos de modificación del punto G y tome conciencia del efecto de este cambio con el valor del coeficiente de la variable de la ecuación roja. Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, se le solicita escribir en su cuaderno la relación entre el punto G que se mueve sobre la cuadrícula y la ecuación roja. Se espera que el estudiante mencione que:

1. La distancia recorrida verticalmente del punto G hasta el punto de corte de la recta con el *eje* y es el valor del numerador del coeficiente de la variable, donde su signo es

positivo si G se ha movido a arriba del punto de intersección de la recta con el *eje y*, o negativo si G se ha movido a abajo del punto de intersección de la recta con el *eje y*.

2. La distancia recorrida horizontalmente del punto G hasta el punto de corte de la recta con el *eje y*, es el valor del denominador del coeficiente de la variable, donde su signo es positivo si G se ha movido a la derecha del punto de intersección de la recta con el *eje y*, o negativo si G se ha movido a la izquierda del punto de intersección de la recta con el *eje y*.

## **Tarea 2- Anticipación.**

Consiste en mover el punto H, seleccionando la posición en la que debe quedar sobre la cuadrícula, para que la ecuación roja sea igual a la ecuación azul. El estudiante deberá entonces anticipar el valor específico de las coordenadas de H, en el que el valor de la pendiente de la recta coincide con el coeficiente de la variable de la ecuación azul para que la ecuación roja sea igual a ésta.

Las ecuaciones tienen el mismo valor del término independiente, pero no tienen el mismo coeficiente de la variable. Al seleccionar la ubicación del punto H sobre la cuadrícula la inclinación de la recta cambia y la ecuación roja también. Al cambio de la ecuación se conserva el valor del término independiente, y se modifica el valor del coeficiente de la variable.

Existen dos posibilidades de acción para el estudiante:

1. Señala un punto en el que el valor de la pendiente de la recta no coincide con el coeficiente de la variable de la ecuación azul. La retroacción del medio va a mostrar que las ecuaciones roja y azul son diferentes, por lo que el estudiante puede invalidar esta acción.
2. Señala un punto en el que el valor de la pendiente de la recta coincide con el coeficiente de la variable de la ecuación azul. La retroacción del medio va a mostrar la ecuación roja igual a la ecuación azul, por lo que el estudiante puede validar esta acción.

El estudiante solo podrá seleccionar una vez la ubicación del punto G, pues éste se fijará automáticamente en ese lugar hasta que se oprima el botón “verificar”. Cuando el estudiante hace clic en el botón “verificar” el software compara la pendiente de la recta roja con el valor del coeficiente de la variable de la ecuación azul. Si los dos valores son diferentes, muestra como mensaje: “La ecuación roja no es igual a la ecuación azul, vuelve a intentarlo”. Si los dos valores coinciden muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva ecuación” y genera un nuevo ejercicio. De ser necesario el

estudiante podrá regresar a la tarea en la que modifica la pendiente de forma experimental, cuantas veces quiera. Para esto está el botón “volver a...”.

Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, podrá pasar a la siguiente tarea con el botón continuar, el cual aparecerá en ese momento en su pantalla.

### **Tarea 3- Experimentación.**

Consiste en arrastrar el punto I sobre el eje de las ordenadas y el punto J sobre la cuadrícula, hasta hacer que las ecuaciones roja y azul sean iguales.

Los puntos I y J pertenecen a la recta roja. El punto I indica la intersección con el eje de las ordenadas. Las ecuaciones no tienen ni el mismo valor del término independiente, ni el mismo coeficiente de la variable. Al arrastrar los puntos J e I cambia la posición de la recta y su ecuación. La ecuación de la recta roja se modifica tanto en el valor del término independiente, como el valor del coeficiente de la variable.

Existen cuatro posibilidades de acción para el estudiante:

1. Arrastra los puntos I y J haciendo que las ecuaciones tengan el mismo coeficiente de la variable, pero distinto valor en el término independiente.
2. Arrastra los puntos I y J haciendo que las ecuaciones tengan el mismo término independiente, pero distinto valor en el coeficiente de la variable.
3. Arrastra los puntos I y J haciendo que las ecuaciones no tengan ni el mismo valor en el término independiente, ni el mismo valor en el coeficiente de la variable.
4. Arrastra los puntos I y J haciendo que las ecuaciones tengan el mismo término independiente y el mismo coeficiente de la variable.

Para los 3 primeros casos, las retroacciones del software mostrarán que las ecuaciones no son iguales, por lo que el estudiante está en capacidad de invalidar su acción. En el último caso, las retroacciones del software mostrarán que las ecuaciones serán iguales, por lo que el estudiante está en capacidad de validar su acción.

Cuando el estudiante hace clic en el botón “verificar” el software compara respectivamente la pendiente de la recta roja y la ordenada del punto I, con el valor del coeficiente de la variable y el término independiente de la ecuación azul. Si los valores son diferentes, muestra como mensaje: “La ecuación roja no es igual a la ecuación azul, vuelve a intentarlo”. Si los valores coinciden muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva ecuación” y genera un nuevo ejercicio.

El estudiante podrá regresar a alguna tarea en la que modifique por separado conjunto la pendiente y la intersección de la recta con el *eje* y cuantas veces quiera. Para esto está el botón “volver a...”.

Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, podrá pasar a la siguiente tarea con el botón continuar, el cual aparecerá en ese momento en su pantalla.

### **Tarea 3- Anticipación.**

Consiste en seleccionar la posición de los puntos K y L para hacer que las ecuaciones roja y azul sean iguales. El punto K debe seleccionarse como la intersección entre la recta y el eje de las ordenadas. El punto L debe quedar sobre la cuadrícula de forma que la pendiente de la recta sea el coeficiente de la variable de la ecuación azul.

Los puntos K y L pertenecen a la recta. Las ecuaciones no tienen ni el mismo valor del término independiente, ni el mismo coeficiente de la variable. Al mover los puntos K y L cambia la posición de la recta y su ecuación. La ecuación de la recta roja se modifica tanto en el valor del término independiente, como el valor del coeficiente de la variable.

Existen cuatro posibilidades de acción para el estudiante:

1. Mueve los puntos K y L haciendo que las ecuaciones tengan el mismo coeficiente de la variable, pero distinto valor en el término independiente.
2. Mueve los puntos K y L haciendo que las ecuaciones tengan el mismo término independiente, pero distinto valor en el coeficiente de la variable.
3. Mueve los puntos K y L haciendo que las ecuaciones no tengan ni el mismo valor en el término independiente, ni el mismo valor en el coeficiente de la variable.
4. Mueve los puntos K y L haciendo que las ecuaciones tengan el mismo término independiente y el mismo coeficiente de la variable.

Para los 3 primeros casos, las retroacciones del software mostrarán que las ecuaciones no son iguales, por lo que el estudiante está en capacidad de invalidar su acción. En el último caso, las retroacciones del software mostrarán que las ecuaciones serán iguales, por lo que el estudiante está en capacidad de validar su acción.

El estudiante solo podrá seleccionar una vez la ubicación de los puntos K y L pues éstos se fijarán automáticamente en el lugar escogido hasta que se oprima el botón “verificar”. Cuando el estudiante hace clic en el botón “verificar” el software compara 3 cosas: La abscisa del punto B con la abscisa del punto I, la ordenada del punto I con el valor del término independiente de la

ecuación azul y la pendiente de la recta con el coeficiente de la variable. Recordemos, el punto B está oculto sobre el eje de las ordenadas.

Si los puntos B e I tienen diferentes abscisas se muestra como mensaje: “I no está sobre el *eje* y, vuelve a intentarlo”. Si los puntos B e I tienen la misma abscisa, pero la ordenada del punto I no coincide con el valor del término independiente de la ecuación azul, muestra como mensaje: “La ecuación roja aún no es igual a la ecuación azul, vuelve a intentarlo”. Si los puntos B e I tienen la misma abscisa y la ordenada del punto I coincide con el valor del término independiente de la ecuación azul, pero la pendiente de la recta no coincide con el coeficiente de la variable de la ecuación azul, muestra como mensaje: “La ecuación roja aún no es igual a la ecuación azul, vuelve a intentarlo”. Si los puntos B e I tienen la misma abscisa, la ordenada del punto I coincide con el valor del término independiente de la ecuación azul y la pendiente de la recta coincide con el coeficiente de la variable de la ecuación azul, muestra como mensaje: “Muy bien, Ahora hazlo con esta nueva ecuación” y genera un nuevo ejercicio.

El estudiante podrá regresar a alguna tarea en la que modifique por separado o en conjunto la pendiente y la intersección de la recta con el *eje* y cuantas veces quiera. Para esto está el botón “volver a...”.

Cuando el estudiante haya resuelto esta tarea por lo menos 5 veces, y el número de intentos fallidos sea menor al número de éxitos, recibirá como mensaje: “¡Felicitaciones! Has terminado todas las tareas”.

## Pilotaje

Este pilotaje se desarrolló con una estudiante de 8°. A continuación, comparamos el análisis a priori, presentado en el desarrollo de la propuesta, con las conductas reales que presenta la alumna durante la aplicación de las actividades.

### Tomo 1- Conversión del registro algebraico al gráfico.

#### Tarea 1- Experimentación

En esta tarea se evidencia que todas las acciones y retroacciones previstas se alcanzan. Inicialmente la estudiante determina valores aleatorios en la lista desplegable para modificar el punto de intersección de la recta roja con el *eje y*. En principio la estudiante cambia muchas veces el valor en la lista hasta superponer las dos rectas. Cuando logra superponer las dos rectas, dice: “*listo, y ¿ahora qué?*”. Con lo cual se deduce que validó sus acciones mediante las retroacciones matemáticas del medio.

La niña no usa de manera espontánea el botón “verificar”, probablemente porque piensa que no necesita otra verificación, fue necesario entonces solicitarle que lo hiciera. Con esto se considera pertinente cambiar el nombre de este botón para que los estudiantes lo utilicen espontáneamente. Algunas ideas son: “tarea realizada” o “nuevo ejercicio”.

En los siguientes ejercicios se observa una disminución de los valores que la estudiante selecciona sucesivamente para solucionar la tarea, de lo cual se deduce que desarrolla una estrategia que le permite seleccionar este valor.

Tras realizar 5 ejercicios y emplear el botón “Verificar”, la estudiante nota que el mensaje que le da el software es diferente. Ella pregunta: “*¿pero ¿qué es lo que tengo que escribir ahí?, no entiendo ahí?*”, haciendo referencia al mensaje que indica: “Escribe en tu cuaderno la relación que hay entre el valor  $b$  de la ecuación y la gráfica de la recta (...)”. Cuando se le pregunta qué es lo que ha entendido, responde que debe escribir la relación entre  $b$  y la gráfica pero que no sabe cuál es, que “ *$b$  es la azul*” (señalando el punto B de la recta azul). Al pedirle que leyera más despacio, dijo: “*ah! es el  $b$  de la ecuación*”

E: El  $b$  de la ecuación es dos.

P1: Ahora cómo ves ese 2 en la recta.

E: Acá mira es el número debajo del puntico (la estudiante lo señala en la pantalla tal como se muestra en la imagen 3), el que está sobre la línea.

P1: ¿Cómo se llama esa línea?

E: Es el *eje y*.

P2: Cuando tu cambias el valor de  $b$  en la ecuación, entonces ¿qué le pasa a la recta?

E: Que se mueve.

P1: ¿A dónde?

P2: ¿Cómo se mueve, se mueve a cualquier lado?

E: No, se mueve de acuerdo al número que yo ponga ahí (señalando la lista desplegable)

P1: ¿Y ese número cómo se ve en la recta? ¿Cómo sabes dónde va a quedar si selecciono por ejemplo el 1 en la casilla?

E. Como se ve en la recta, está en el punto, ósea es en el punto ahí está el numerito (señalando el punto de la recta roja y el valor de su ordenada en el *eje y*. Luego coloca el 1 en la casilla y muestra el 1 en el eje de las ordenadas). Está acá, el punto en el numerito, en el *y* (haciendo referencia al eje *y*).

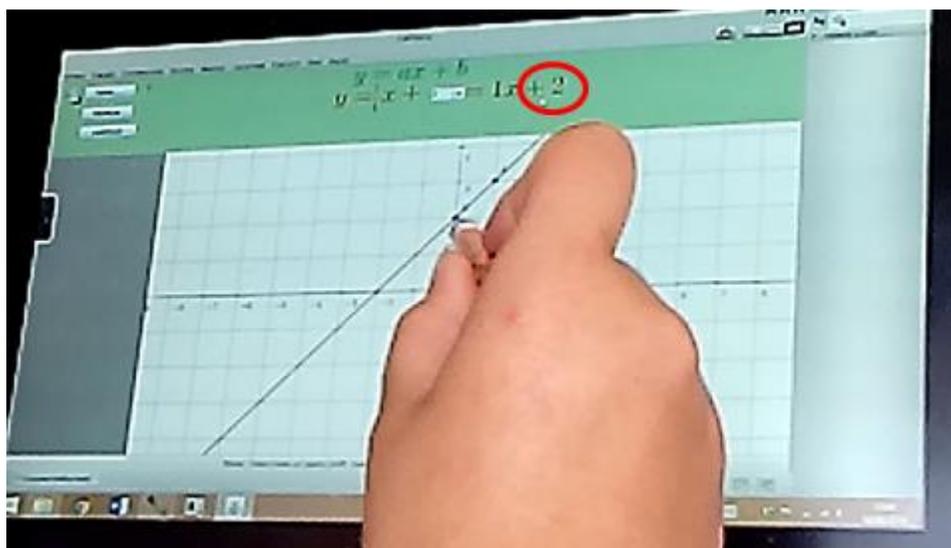


Imagen 3. Tarea 1, determina el parámetro  $b$ .

Cuando se le solicita a la estudiante que escriba lo mencionado, ella plantea que:

Si cambio el valor de  $b$  se cambia  
el punto de intersección en  $1$  de  $y$   
y la recta roja

Imagen 4. Respuesta dada por la estudiante.

### Tarea 1- Anticipación

Después de oprimir el botón “continuar” y de que aparece el mensaje: “la ecuación de la recta roja es de la forma  $y=ax+b$ . Escriba en la casilla el valor correspondiente a  $b$  en la ecuación, para que la recta roja quede sobre la recta azul”, la estudiante pregunta: “¿qué hay que hacer?”, sin embargo, en ese momento identifica la casilla y escribe un número cualquiera. Constata el movimiento correspondiente de la recta roja y reconoce que corresponde a lo que había trabajado en los ejercicios anteriores. A partir de este momento escribe en la casilla el valor

correspondiente a la ordenada del punto B. Se deduce así que logra anticipar el valor del término independiente de la ecuación como se había previsto en el análisis a priori.

Hubo momentos donde la estudiante ingresa valores erróneos, aunque no porque predijera mal el valor que debía introducir en la casilla, sino porque visualizaba mal la escala del eje y. Esto nos lleva a pensar dos cosas: las marcaciones en las escalas deberían ubicarse más cerca del lugar que señalan (la niña las veía por ejemplo como el numerito abajo del punto) y debería restringirse la posibilidad de ingresarse varias veces un valor en la casilla, pues fuese error o no la mala elección de la estudiante, el que pueda corregirse la elección implica un proceso que el estudiante puede tomar como experimentación hasta encontrar el valor que hace que las rectas se superpongan.

Después de que la niña completa los cinco ejercicios se le pregunta sobre el ejercicio que aparece en la pantalla antes de que continúe a la otra tarea:

P1: ¿Qué número deberías colocar en la casilla para que la recta roja quede sobre la recta azul?

E: El cero (señala la intersección de la recta azul con el *eje y*, para posteriormente introducir ese valor en la casilla (ver imagen 5)).

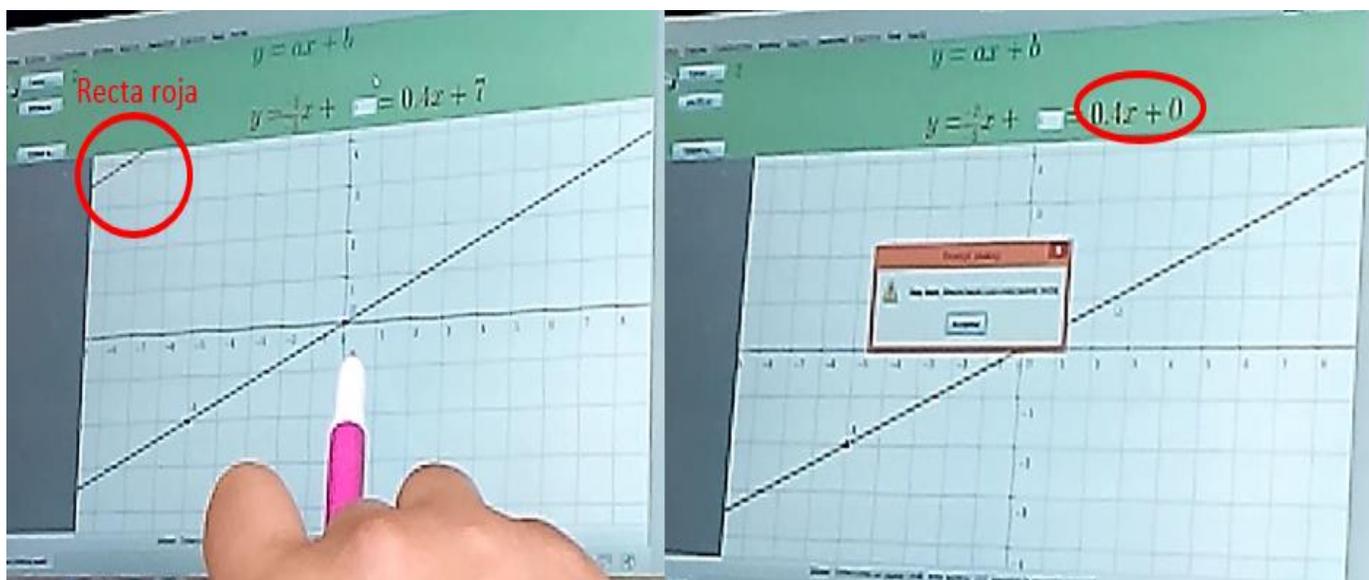


Imagen 5. Anticipación para determinar el valor del parámetro  $b$ .

El desarrollo de los 5 ejercicios de esta tarea fue rápido y efectivo, evidenciando la relación que la estudiante establece entre el valor del  $b$  de la ecuación y su gráfica.

## Tarea 2- Experimentación numerador pendiente

Después de que aparece y oprime el botón “continuar”, aparece el mensaje: “La ecuación de la recta roja es de la forma  $y=ax+b$ . La recta roja corta el eje y en -1 (*este valor corresponde a la ordenada de B, punto de intersección de la recta roja con el eje y*) utilice la lista desplegable para modificar el numerador del término  $a$  en la ecuación, hasta lograr que la recta roja quede sobre la recta azul”. La estudiante lee detenidamente el mensaje e identifica señalando, que las rectas y el eje y se unen en el punto B (ver imagen 6) y que ahí “está el -1, en el y”. Además, dice: “*toca cambiar este número de acá* (señalando la lista desplegable ubicada en el numerador) *para que esta* (señala la recta roja) *quede sobre la azul*”. Planteamiento que nos lleva deducir que la estudiante ha entendido la tarea.

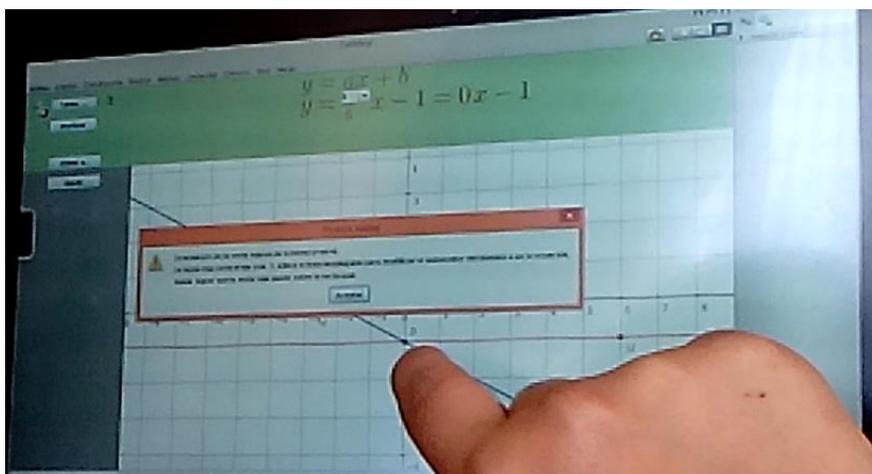


Imagen 6. Identifica la intersección entre las rectas y el eje y.

Inicialmente la estudiante determina valores aleatorios en la lista desplegable, sin ninguna regularidad. Ante esto se interviene preguntando:

P1: ¿Qué está pasando?

E: Pues que el valor cambia (señalando la lista) y la recta se mueve.

P2: Pero, ¿cómo se mueve, se mueve de cualquier manera?

E: No, de acuerdo al número que yo ponga arriba

P1: Digamos, si pones el -3, ¿Qué va a pasar?

E: Sube (En ese momento la estudiante asigna ese valor en la lista desplegable). ¡Ah no!, baja.

P1: y el 3 cómo cambia la recta (ante esto no se obtiene respuesta)

La estudiante entiende que debe encontrar una relación entre el movimiento de la recta roja respecto al número que seleccione de la lista desplegable, pero, aunque sigue cambiando el valor del numerador ella no nota ninguna relación. Tal como lo habíamos previsto, se necesita una ayuda, sin embargo, la estudiante ignora la presencia del botón ayuda.

P1: ¿Ves algo diferente en la pantalla con respecto a la actividad anterior, algo que te pueda servir?

E: no, nada.

P1: En los botones.

E: ¡Ah sí!, el que dice ayuda.

Podemos reconocer que se hace necesario hacer tomar conciencia de la presencia del botón “ayuda” cuando el estudiante ha realizado varios intentos y no ha encontrado la solución de la tarea. Pensamos que podría entonces programarse un mensaje automático que indique que puede usar este botón, luego de que el estudiante ha seleccionado varias veces datos erróneos o que ha pasado un tiempo prudencial y aún no ha culminado los primeros 5 ejercicios correctos.

Cuando la estudiante hace clic sobre el botón “ayuda”, aparece el mensaje: “Modifica el valor del numerador del término  $a$  de la ecuación, hasta hacer que los triángulos queden superpuestos e iguales”. La estudiante empieza a variar los valores de la lista desplegable, hasta hacer que el triángulo rojo sea igual al azul. La estudiante dice:

E: Listo

P1: ¿Qué ves en el triángulo?

E: veo un -4 (y señala en la pantalla el cateto horizontal del triángulo, imagen 7), ¡ah!, mira aquí está, es el número de abajo (señalando el denominador de la fracción, imagen7).

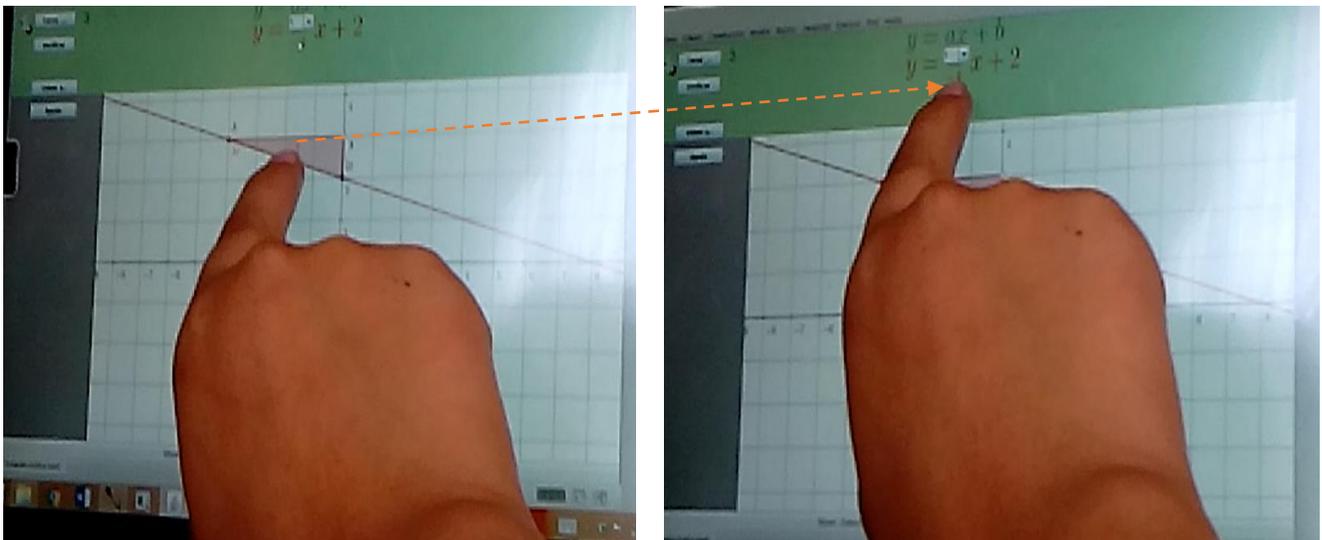


Imagen 7. Relación entre el valor del cateto horizontal y el denominador de  $a$ .

P1: ¿y qué más?

E: Aquí hay un 3 (refiriéndose a un punto del eje  $y$ ), ¡ay no!, es un 1 (señala el otro cateto del triángulo, imagen 8).

P2: ¿y ese número dónde lo ves?

E: Aquí (señalando el numerador de la fracción, imagen 8), es el que cambia.

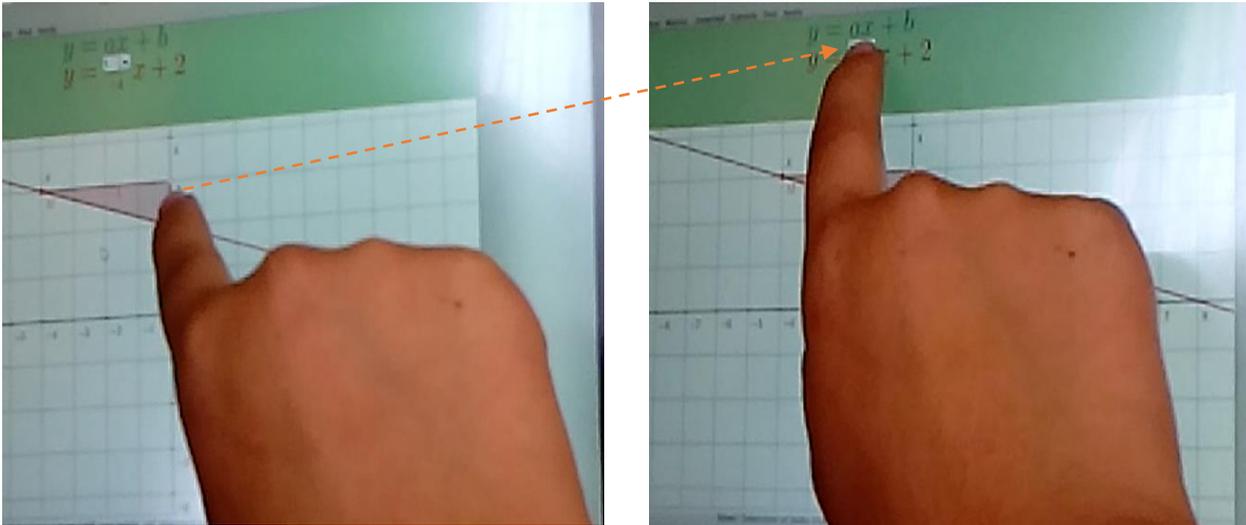


Imagen 8. Relación entre el valor del cateto vertical y el numerador de  $b$ .

En este momento la estudiante continúa realizando los respectivos ejercicios, pero haciendo uso de la ayuda. Ella constata que el número correspondiente al numerador del coeficiente de la variable es el número de cuadros de la cuadrícula del cateto “vertical” del triángulo rojo y que el número correspondiente al denominador del coeficiente de la variable es el número de cuadros de la cuadrícula del cateto “horizontal” del triángulo rojo. Sin embargo, aún no relaciona el triángulo con el signo de los números; ella, escribe un número correspondiente al conteo y en el caso en que la recta no queda superpuesta le cambia de signo. En uno de los ejercicios se le pregunta:

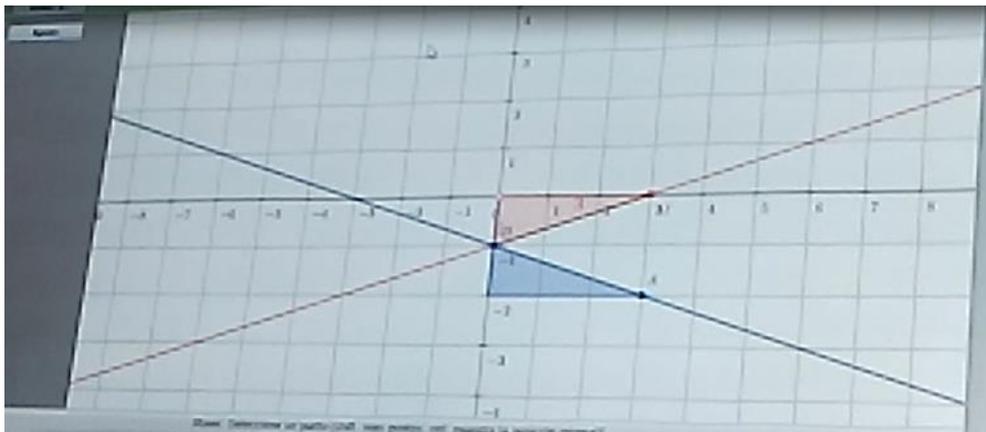


Imagen 9. Tarea 2: Triángulos iguales.

P1: ¿Qué te pedía la ayuda?

E: Que los triángulos eran iguales y ya lo son (señala la pantalla (imagen 9)).

P1: Ya quedaron iguales, y ¿qué te dijeron más en la ayuda?

E: Que... cómo se dice, que encajara los dos triángulos en uno. En un lugar.

P2: Ok, y ¿qué pasa cuando pusiste 1?

E: Se abre.

P1: ¿Hacia dónde?

E: Hacia arriba.

P2: y entonces, ¿qué debes hacer?

E: Realizar lo opuesto. Que vaya hacia abajo, por eso... -1

P2: ¿Entonces a qué se debe el cambio de posición?

E: Pues que siempre los positivos van hacia arriba y los negativos hacia abajo

La estudiante empieza entonces a tener en cuenta la diferencia entre las ordenadas de los puntos que se muestran de la recta azul. Según la distancia vertical del cateto del triángulo asigna un valor, pero antes se asegura del signo que debe tener mirando: “B está arriba de A, entonces es menos”. Esto no había sido previsto en el análisis a priori. La estudiante empieza a realizar adecuadamente los ejercicios propuestos, sin embargo, sigue usando la ayuda. En aras de evidenciar qué pasaba si ya la ayuda no estuviera, se le pide que no haga uso de ésta y se le pregunta:

P1: ¿Qué debes hacer?

E: Construir el triángulo.

P2: Indica el triángulo en la pantalla.

E: Sus vértices están sobre la recta azul (señalando el punto A y B), entonces es esté (la estudiante indica con el cursor un triángulo tal como lo muestra la imagen 10).

P2: ¿Qué características tenía el triángulo que te mostraba la ayuda?

E: era un triángulo.

P1: si, ¿pero qué tipo de triángulo es, o qué características tiene?

E: yo no me acuerdo de eso.

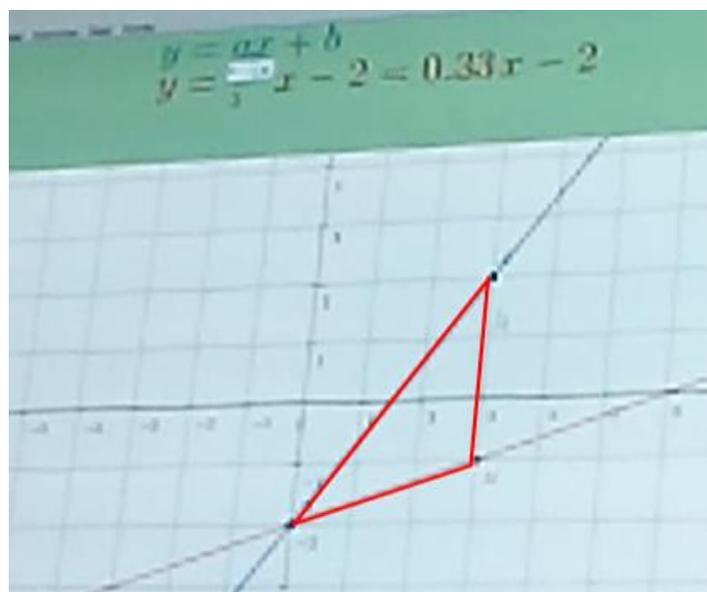


Imagen 10. Triángulo formado por el alumno.

Lo que se infiere es que hay una necesidad de reforzar la clasificación de triángulos, pues la estudiante no se ha fijado que el triángulo debe ser rectángulo. Partiendo de ello, las docentes intervienen un momento y le solicitan que active la ayuda y compare los triángulos que aparecen con el formado anteriormente por ella. La estudiante dice: *“el triángulo se mueve hacia arriba y hacia el lado”* (desplaza el dedo sobre la línea horizontal y sobre la línea vertical de la cuadrícula). La estudiante emplea el botón “verificar” para obtener un nuevo ejercicio. Sin utilizar la ayuda, cuenta los cuadrados moviendo el dedo sobre la cuadrícula y dice: *“tiene que ser un triángulo recto con A y B”*.

En el momento en el que la estudiante usa el botón “Verificar” y aparece el mensaje: “Escribe en tu cuaderno la relación entre el valor del numerador de a en la ecuación y la gráfica de la recta, (...)”, la estudiante plantea: *“se mueven los cuadrados del número, ¿y qué?, cuando la A está aquí arriba es positivo, porque la B está abajo y se cuentan los cuadritos hacia arriba cuando la A está hacia arriba, desde donde está B y, al contrario, abajo si es negativo”*. Es decir, la estudiante plantea que el número asignado al numerador del coeficiente de la variable es el número de cuadritos verticales entre A y B y decide el signo dependiendo de la posición relativa de A y B: si A está arriba es positivo, si A está abajo es negativo.

## Tarea 2- Experimentación denominador pendiente

Al oprimir el botón “continuar”, aparece el mensaje: “La ecuación de la recta roja es de la forma  $y=ax+b$ . La recta roja corta el eje y en 0 (este valor corresponde a la ordenada de B, punto de intersección de la recta roja con el eje y), utilice la lista desplegable para modificar el denominador del término a en la ecuación, hasta lograr que la recta roja quede sobre la recta azul”.

La estudiante lee detenidamente el mensaje, identifica que “*toca cambiar ahora el número de abajo (el denominador), para encontrar el número y que ésta (recta roja) quede sobre ésta (recta azul)*”. El punto A tiene coordenadas (5, -3) y el punto B tiene coordenadas (0,0). La estudiante cuenta y elige el valor -3 de la lista despegable, empleando la estrategia de la tarea anterior (imagen 11) y dice “no me sale”.

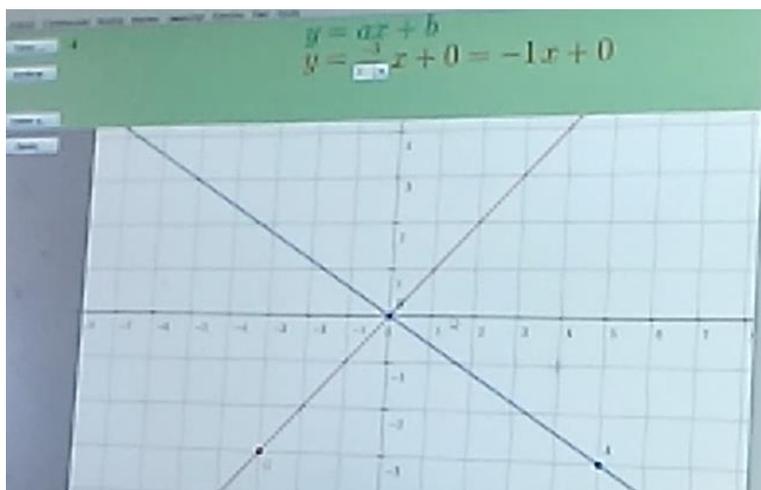


Imagen 11.

Ante esto, se le pregunta a la estudiante:

P1: ¿Qué hacías en la anterior actividad?

E: Contar los cuadritos de B a A.

P2: ¿Que modificabas?

E: El de arriba (haciendo referencia al numerador del parámetro de  $a$ ).

P2: ¿Estas modificando lo mismo?

E: No, estoy modificando el de abajo (haciendo referencia al denominador del parámetro de  $a$ ).

P1: ¿Cuánto es el numerador en este caso? (refiere a la imagen 11).

E: -3.

P1: ¿Por qué?

E: Porque se cuenta hacia abajo, tres cuadritos hacia la A, entonces negativos

P1: Entonces que puede pasar con el denominador.

E: Lo del otro lado, (la estudiante cuenta los cuadros correspondientes al cateto horizontal del triángulo, empezando desde el *eje y*, hasta el punto A) cinco, (pone en la lista despegable -5) ¡ay no!, es positivo (al cambiar el valor, las rectas se superponen) ya.

P1: ¿Entonces qué puedes deducir de ahí?

E: Pues que cuando la A está a la derecha es positivo y cuando A está a la izquierda es negativo.

P1: y ¿qué debes contar?

E: los cuadritos hacia la derecha.

P1: ¿A partir de qué?

E: de la B.

Lo anterior denota que la estudiante relaciona el número asignado al denominador del coeficiente de la variable con la distancia horizontal entre los puntos A y B, y que sus signos corresponden al sentido en el que se ha hecho ese movimiento: positivo si es hacia la derecha, negativo si es hacia la izquierda. La estudiante aplica esta estrategia para resolver los 5 ejercicios. Cuando ella emplea el botón “Verificar” y nota que el mensaje que le da el software es diferente, este es: “Escribe en tu cuaderno la relación entre el valor del denominador de  $a$  en la ecuación y la gráfica de la recta, (...)”, la estudiante manifiesta no entender el enunciado, por lo cual se le pregunta:

P1: ¿Cuál es la relación entre el denominador de  $a$  y la gráfica?

E: Como así

P1: ¿Cuál es el denominador de  $a$ ?

E: El de abajo.

P2: Y qué relación tiene con la gráfica.

E: Ya lo dije, contando los cuadritos hacia la derecha o la izquierda.

P1: ¿entre qué y qué?

E: Entre A y B y si es a la derecha es positivo y a la izquierda negativo.

Es decir, la estudiante establece una relación de cómo se moverá la recta dependiendo de dicho valor.

## Tarea 2- Anticipación

Después de oprimir el botón “continuar” aparece el mensaje: “la ecuación de la recta roja es de la forma  $y=ax+b$ . La recta roja corta el eje  $y$  en  $-1$  (*este valor corresponde a la ordenada de B, punto de intersección de la recta roja con el eje  $y$* ). Escriba en las casillas los valores del numerador y denominador de  $a$  para que la recta roja quede sobre la recta azul.”). Se le pregunta a la estudiante ¿que debes realizar?, ella plantea “*debo escribir en las dos casillas para que las rectas queden una sobre otra*”.

La estudiante escribe en la casilla del numerador el valor correspondiente a la distancia recorrida verticalmente y en el denominador la distancia recorrida horizontalmente (imagen 12), a partir de los dos puntos que se muestran de la recta roja. Se deduce que ella logra anticipar el valor del coeficiente de la variable de la ecuación de la recta como se había previsto en el análisis a priori; pero, en algunos casos volvemos a ver la posibilidad de corrección que tiene ella a su elección. La estudiante presenta algunas veces errores en sus conteos por lo que ve la necesidad de corregir.

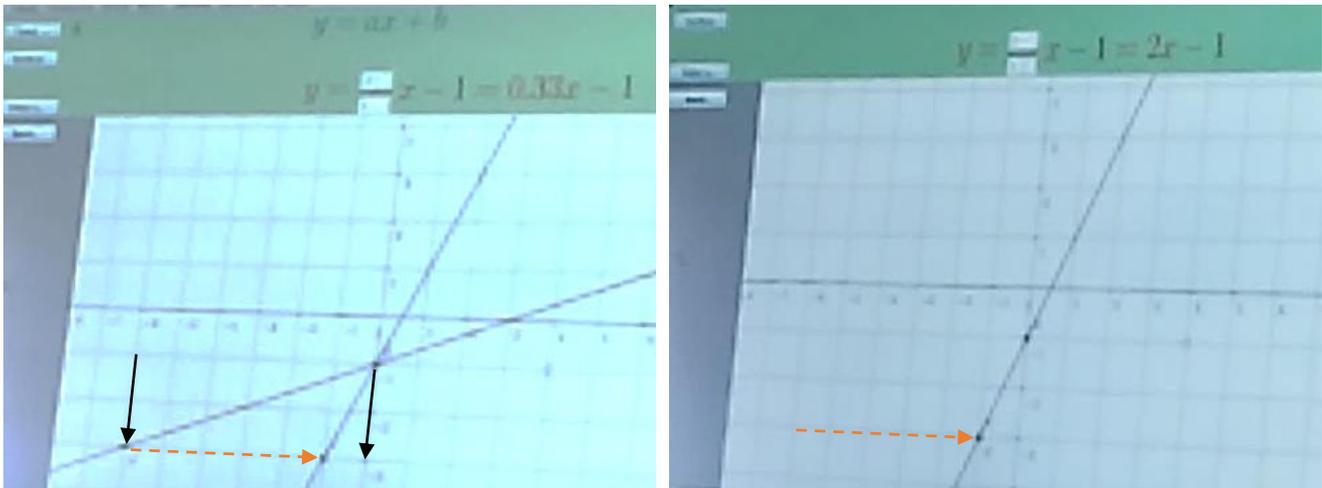


Imagen 12.

La estudiante resolvió los 5 ejercicios rápidamente, con lo cual deducimos que: comprende la relación entre el cociente  $a$  de la ecuación y la gráfica de la recta.

### Tarea 3- Experimentación

Después de oprimir el botón “continuar”, aparece el mensaje: “La ecuación de la recta roja es de la forma  $y=ax+b$ . Utilice las listas desplegables para modificar el valor de  $a$  y de  $b$  en la ecuación hasta lograr que la recta roja quede sobre la recta azul.”.

En el primer ejercicio, la estudiante empieza a modificar las listas: inicialmente cambia el punto de intersección de la recta roja con el *eje* y (imagen 13) y posteriormente determina el coeficiente de la variable de la ecuación, haciendo que las rectas se superpongan.

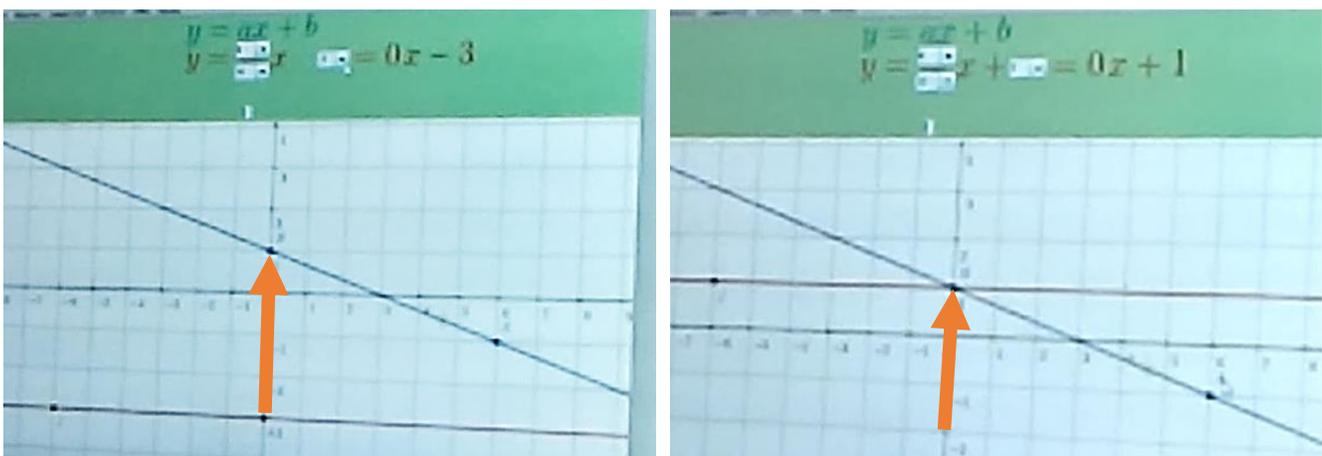


Imagen 13.

En el segundo ejercicio, la estudiante modifica primero el valor del parámetro  $a$  de manera que el punto J coincide con el punto A. se le pregunta:

P1: ¿Por qué las rectas no se superponen?

E: Porque no tienen igual este punto (señalando con el cursor el punto B).

P1: Entonces que deberíamos hacer.

E: Pues que le pongo un -3 (y se dirige a la lista desplegable del parámetro b), ¡ah no! -4.

La estudiante constata que los puntos I y B coinciden, pero los puntos A y J no. La estudiante entonces se da cuenta que ha calculado mal la pendiente y procede a su corrección. Es importante resaltar que pueden existir dos estrategias:

- El estudiante determina inicialmente el valor de  $a$ , por lo cual las rectas quedarán paralelas y al cambiar el valor  $b$  correctamente quedarán superpuestas.
- El estudiante determina inicialmente el punto de corte de la recta roja con el eje  $y$ , por lo cual las rectas se encontrarán en dicho punto y al cambiar el valor  $a$  correctamente quedarán superpuestas.

En los siguientes ejercicios la estudiante emplea la segunda estrategia.

### **Tarea 3- Anticipación**

Al oprimir el botón “continuar”, aparece el mensaje: “la ecuación de la recta roja es de la forma  $y = ax + b$ . Escriba en las casillas los valores correspondientes de  $a$  y de  $b$  en la ecuación, para que la recta roja quede sobre la recta azul”. En el momento en que la estudiante empieza la tarea, comenta “está fácil, es lo mismo de antes”. Escribe en las casillas los valores correspondientes a los parámetros  $a$  y  $b$  de la ecuación. Se deduce así, que logra anticipar los valores de los términos de la ecuación como se había previsto en el análisis a priori.

### **Conclusiones generales primer tomo**

En general, la estudiante utilizó las acciones previstas en el análisis a priori y las retroacciones del medio le permitieron realizar las validaciones correspondientes. Sin embargo, encontramos la necesidad de realizar algunas modificaciones al diseño.

- La programación requiere que el estudiante haya resuelto 5 ejercicios y que el número de intentos fallidos sea menor al número de éxitos para pasar a la siguiente tarea. Sin embargo, no existen intentos fallidos, puesto que la validación es visual. El estudiante siempre validara su ejercicio cuando vea que las rectas están superpuestas y no hace clic en el botón verificar si las rectas no están superpuestas. Esto nos lleva a pensar que no es necesario esta verificación del software para mostrar el botón “continuar”, al menos

no hasta que no se proponga una solución al problema de la anticipación descrito en las tareas 2 y 3 de este tipo.

- Debe cambiársele el nombre al botón “verificar” para que su uso sea espontáneo.
- Debe indicarse la ayuda disponible para cuando haya repetidos intentos fallidos en la resolución de las partes de la tarea 2.
- Las marcaciones en las escalas deben ubicarse más cerca del lugar que señalan.

## **Tomo 2- Conversión del registro gráfico al algebraico.**

### **Tarea 1- Experimentación**

En esta tarea se evidencia que todas las acciones y retroacciones previstas se alcanzan.

Inicialmente, cuando se abre la figura, se le muestra a la estudiante el siguiente mensaje: “La ecuación de la recta roja es de la forma  $y=ax+b$ . Mueva el punto C sobre el *eje y*, hasta hacer que la ecuación roja sea igual a la ecuación azul”.

La estudiante manifiesta que debe mover el punto C que se encuentra sobre el *eje y*, y al preguntarle ¿Cómo se puede mover?, ella indica en la pantalla que el punto realiza un movimiento vertical sobre el *eje y* moviendo su dedo para arriba y para abajo.

Se le pregunta sobre el propósito de la tarea, ella responde:

E: Que las ecuaciones deben ser iguales.

P1: ¿Son iguales ahorita?

E: No porque éste es -2 (Señalando el parámetro b de la ecuación roja) y éste es -4 (señalando el parámetro b de la ecuación azul).

La estudiante selecciona el punto C y dice: “*debe ir hasta el -4, pero ¿dónde está?*”, luego mueve C hasta la posición (0,-4), finalmente oprime el botón verificar y se le presenta un nuevo ejercicio.

La solución de esta tarea fue rápida. Al realizar los 5 ejercicios y emplear el botón “Verificar”, el mensaje que el medio muestra es: “Escribe en tu cuaderno la relación que hay entre el punto C que se mueve sobre el *eje y*, y la ecuación roja, (...)”. La estudiante plantea que: “*el Punto C, es intersección, ese cambia y en la ecuación cambia el b*”, lo que indica que ella relaciona la posición del punto C con el parámetro b de la ecuación.

## Tarea 1- Anticipación

Después de oprimir el botón “continuar” y de que aparece el mensaje: “La ecuación de la recta roja es de la forma  $y=ax+b$ . El punto D es el punto de la recta roja que está sobre el *eje y*. Señale la posición donde debe quedar D para que la ecuación roja sea igual a la ecuación azul.”. La estudiante reconoce que debe señalar la posición en que debe quedar el punto D, para hacer que las ecuaciones roja y azul sean iguales. La estudiante lee la ecuación azul, dice “2” y señala con el dedo el punto (0,2). Se deduce que la estudiante logra anticipar la posición de la intersección, relacionándola con el valor del término independiente de la ecuación azul, como se había previsto en el análisis a priori, aunque posteriormente señala equivocadamente otro punto con el mouse.

Esta tarea y las otras tareas de anticipación del segundo tomo requieren que el estudiante haga clic sobre un punto de la cuadrícula. A diferencia de las tareas de anticipación del tomo 1, una vez que han hecho clic sobre un punto ya no pueden modificar su elección. Si eligen un punto equivocado no pueden corregir antes de oprimir el botón “verificar”. Esta situación provocó confusión en la estudiante, quien manifiesta que se equivocó de punto, constata que las ecuaciones no son iguales, pero no puede corregir el error. Se necesitó intervenir para explicarle que debía hacer clic sobre el botón verificar y se le daba una nueva oportunidad de elegir la posición del punto. En los siguientes ejercicios la estudiante verifica con atención cuál es el punto que selecciona en el *eje y*, y no volvió a cometer errores.

Cuando la estudiante resolvió los 5 ejercicios, con un número de intentos fallidos menor al número de éxitos, pudo pasar a la siguiente tarea con el botón continuar. La resolución de esta tarea fue rápida.

## Tarea 2- Experimentación

Después de oprimir el botón “continuar”, aparece el mensaje: “La ecuación de la recta roja es de la forma  $y=ax+b$ . La recta roja pasa por el punto 3 (*este valor corresponde a la intersección de la recta roja con el eje y*), mueva el punto G por la cuadrícula, hasta hacer que la ecuación roja sea igual a la ecuación azul.” La estudiante lee detenidamente el mensaje e identifica que las ecuaciones tienen el mismo valor del término independiente, y este corresponde al determinado por el enunciado, es decir 3, pero no tienen el mismo coeficiente de la variable: señala las ecuaciones y dice “*aquí, en el a, son diferentes*”.

En esta tarea la estudiante tuvo muchas dificultades. Inicialmente relaciono el numerador y el denominador de la fracción azul con las coordenadas del punto G tratando de ubicar el punto G en esas coordenadas. Se necesitó la intervención de las docentes para que tomara consciencia de que el numerador y el denominador de la fracción azul no corresponden a las coordenadas del punto G, sino a las distancias horizontal y vertical a partir del punto de intersección de la recta

con el eje  $y$ . Consideramos necesario proponer una ayuda para este tomo, de manera que no sea indispensable la intervención del profesor.

Después de resolver 5 ejercicios, aparece el letrero: “Escribe en tu cuaderno la relación entre el punto  $G$  que se mueve sobre la cuadrícula, y la ecuación roja. (...)”. La estudiante argumenta que la relación es “*que se mueve el punto hacia arriba si es positivo o hacia abajo si es negativo, de donde se chocan según el número de arriba (numerador) de la azul y también se mueve el punto horizontalmente, positivo hacia la derecha o negativo hacia la izquierda, según el número de abajo (denominador) de esa* (señalando el coeficiente de la variable de la ecuación azul)”. Con esto nos damos cuenta que la estudiante ha establecido una relación correcta entre el cociente  $a$  de la ecuación con la gráfica de la recta.

## Tarea 2- Anticipación

Después de oprimir el botón “continuar” aparece el mensaje: “la ecuación de la recta roja es de la forma  $y=ax+b$ . La recta roja pasa por el punto  $-1$  del eje  $y$  (*este valor corresponde a la intersección de la recta roja con el eje  $y$* ), señale la posición donde debe quedar el punto  $H$  para que la ecuación roja sea igual a la ecuación azul.”). La estudiante tiene presente que tiene solo una oportunidad de seleccionar la posición en la que debe quedar el punto  $H$  de la recta para que la ecuación roja sea igual a la ecuación azul. Ella no tiene mayor dificultad para señalar correctamente ese punto.

A continuación, se presenta un ejercicio (ver imagen 14) sobre el que se le pregunta a la estudiante qué es lo que haría:

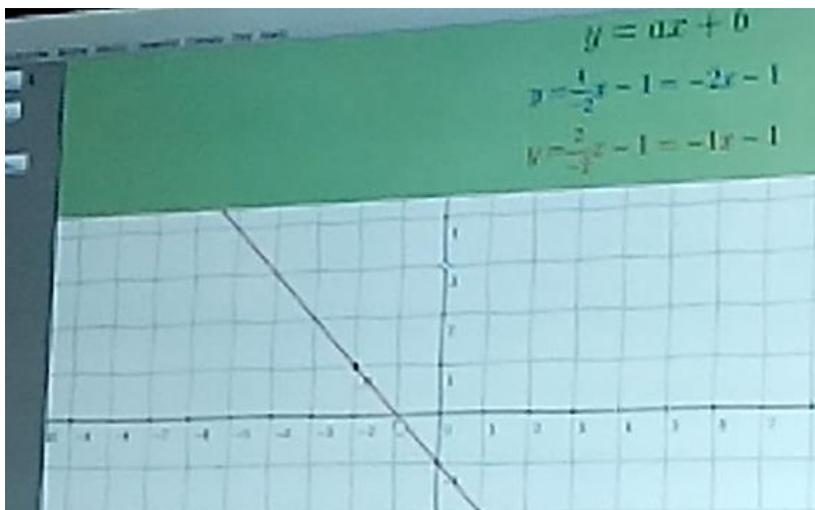


Imagen 14. Ejercicio 1, Tarea 2 anticipación.

P1: ¿Cómo hacemos para que las ecuaciones queden iguales?

E: Poner el punto como el 4 (haciendo referencia al numerador de la ecuación azul de la imagen 14).

P1: ¿Qué determina ese número?

E: Que debo moverme hacia arriba desde acá (señala el punto de intersección de la recta con el *eje y*) y cuento los cuatro cuadros: uno, dos, tres, cuatro (cuenta los cuadros verticalmente hacia arriba como se muestra en la imagen 15).

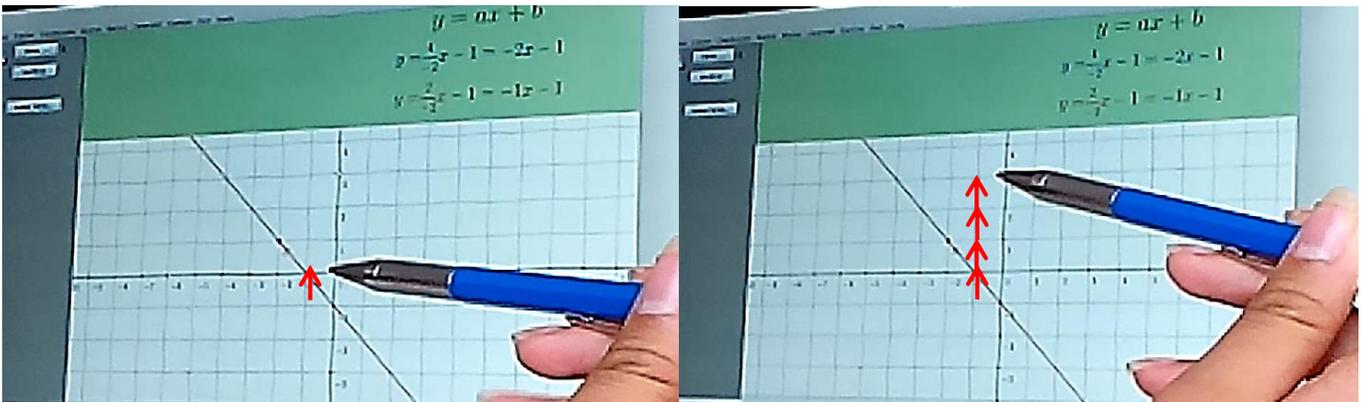


Imagen 15. Desplazamiento vertical, numerador del parámetro  $a$ .

E: Estoy acá.

P1: ¿Y Ahora?

E: El denominador, o sea -2.

P1: ¿Cómo te mueves?

E: Hacia la izquierda porque es negativo. Entonces, uno y dos (cuenta los cuadros de forma horizontal hacia la izquierda como se muestra en la imagen 16).

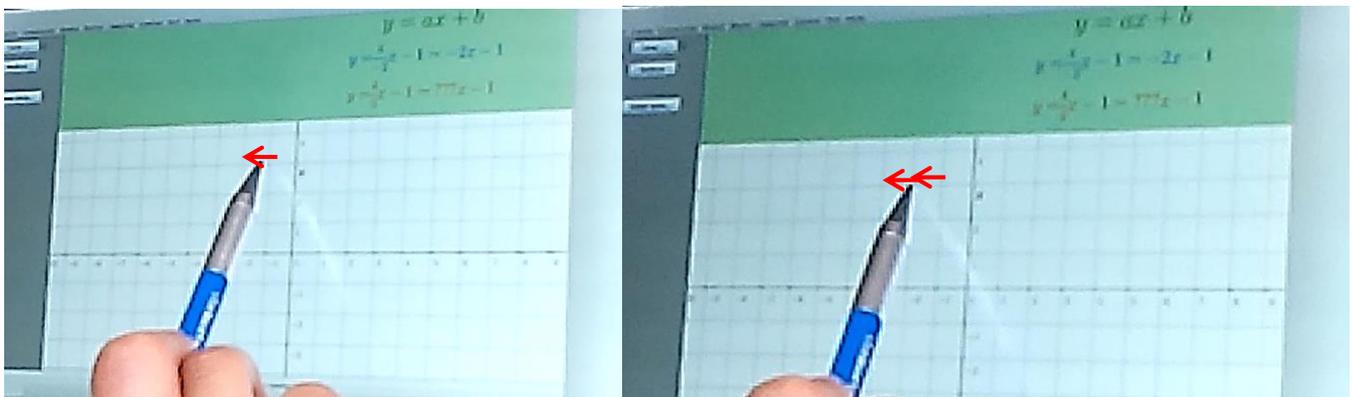


Imagen 16. Desplazamiento horizontal, denominador del parámetro  $a$ .

La estudiante determina que ha resuelto el ejercicio cuando al dar clic en ese punto las ecuaciones quedan iguales (ver imagen 17). Vemos entonces que de nuevo fueron suficientes las

retroacciones matemáticas para que ella supiera que había hecho bien el proceso y pudiera oprimir el botón verificar.

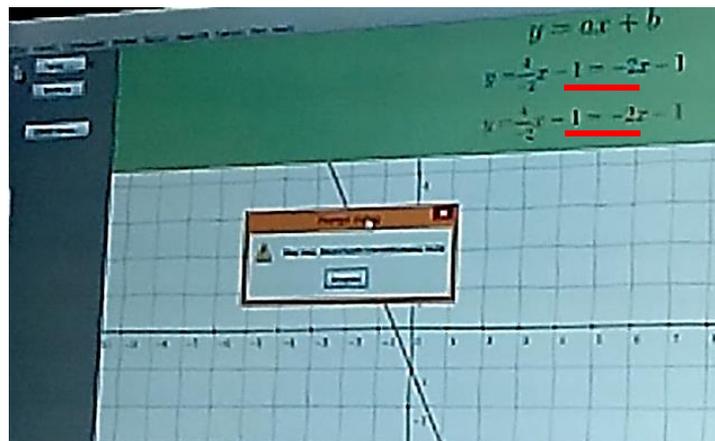


Imagen 17. Ejercicio 1. Tarea 2 anticipación resuelto.

Los siguientes ejercicios son solucionados rápidamente por la estudiante lo que muestra que ella no tiene mayor inconveniente en la tarea propuesta.

### Tarea 3- Experimentación

Después de oprimir el botón “continuar”, aparece el mensaje: “La ecuación de la recta roja es de la forma  $y=ax+b$ . Mueva el punto I sobre el *eje* y y el punto J sobre la cuadrícula, hasta hacer que la ecuación roja sea igual a la ecuación azul.” La estudiante empieza a arrastrar los respectivos puntos. Inicialmente el punto de corte de la recta roja con el eje y (parámetro b), y posteriormente cambia la pendiente (parámetro a), hasta hacer que las ecuaciones sean iguales:

P1: ¿Qué ubicaras primero?

E: El  $b$  (refiriendo al valor de la ecuación azul).

P1: ¿Por qué?

E: Porque desde ahí cuento (hace referencia que a partir de este se forma el triángulo).

P1: ¿El  $b$  qué representa?

E: El choque entre la recta y el eje y.

P1: ¿Qué debemos mover?

E: I (la estudiante mueve el punto I hasta la coordenada (0,1), ver imagen 18).

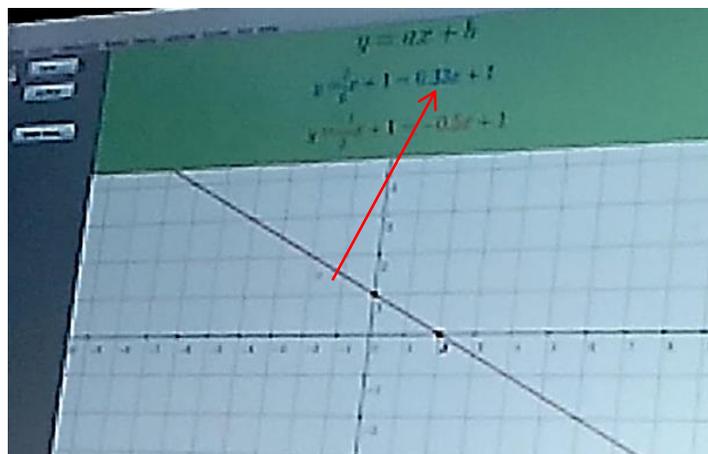


Imagen 18. Tarea 3-Experimentación.

P1: ¿Siguen siendo diferentes las ecuaciones?

E: Sí.

P2: ¿En dónde?

E: En la fracción.

P1: ¿Qué debes mover?

E: El J.

P2: ¿Qué haces para que las fracciones sean iguales?

E: Haciendo el triángulo (la estudiante mueve el punto J según la fracción  $a$ , de la ecuación azul, imagen 19), listo. (La estudiante entonces nos muestra que las ecuaciones son iguales).

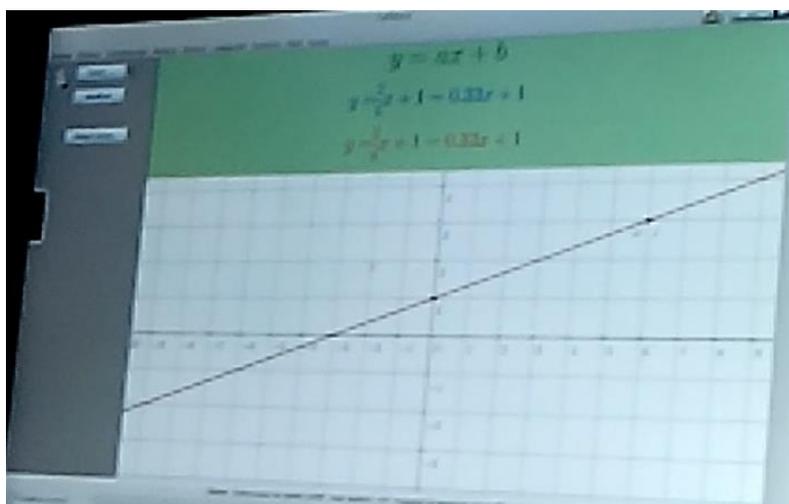


Imagen 19. Tarea 3-Experimentación resuelta.

Al resolver los 5 ejercicios correctamente, y tener la opción de continuar se le pregunta:

P1: ¿Cuál es la relación entre la gráfica y el valor de  $b$  de la ecuación?

E: Pues es que el punto de intersección entre el *eje* y la línea, es el valor de  $b$ .

P1: ¿Y cuál es la relación entre la gráfica y el valor de  $a$  de la ecuación?

E: Es una fracción, donde el de arriba, corre el punto a partir del signo, arriba positivo y abajo negativo, y se cuenta a partir del punto de intersección.

P2: ¿Y el denominador?

E: El que se cuenta de izquierda a derecha, izquierda negativa, derecha positiva, a partir del punto de intersección.

Con esto constatamos nuevamente las explicaciones en las que la estudiante muestra clara comprensión de las relaciones de los registros gráfico y algebraico de la recta.

### **Tarea 3- Anticipación**

En esta tarea el software no funcionó como se había previsto porque los puntos no se movían a la posición señalada por la estudiante. Por lo tanto, finaliza aquí el pilotaje y se hacen las correcciones necesarias.

### **Conclusiones generales segundo tomo**

En general, la estudiante utilizó las acciones previstas en el análisis a priori y las retroacciones del medio le permitieron realizar las validaciones correspondientes. Sin embargo, encontramos la necesidad de realizar algunas modificaciones al diseño.

- Al igual que en el tomo 1 debe cambiarse el nombre al botón “verificar” para que su uso sea espontáneo.
- Debe incluirse una ayuda en la parte experimental de la tarea 3.

## Conclusiones y Recomendaciones

Hemos logrado un diseño de actividades en el software de geometría dinámica CaRMetal, para la enseñanza de la relación de la gráfica y la ecuación de la recta, utilizando el aprendizaje por adaptación y la conversión de registros semióticos.

El análisis a priori muestra que es posible lograr un aprendizaje por adaptación de las conversiones entre los registros gráfico y algebraico de representación de la línea recta. Las retroacciones matemáticas y didácticas del software que fueron programadas en el diseño garantizan la validación perceptiva de las tareas propuestas. Además, la experimentación con los cambios en uno de los registros mientras se observan sus efectos en el otro registro permite al estudiante construir las relaciones de coordinación entre ellos.

El pilotaje sirvió para verificar cómo con las actividades el estudiante es capaz de decidir qué hacer y qué relaciones establecer entre los registros gráfico y algebraico de la recta, aun cuando no tenga los conocimientos teóricos de esto. Por lo tanto, creemos que estas actividades diseñadas constituyen un material utilizable para la enseñanza en matemáticas.

Nuestro diseño se fundamenta en un análisis de los registros gráfico y algebraico de la recta, cuando las escalas de los ejes coordenados son enteras. Es necesario entonces elaborar otras actividades utilizando escalas decimales. Otra posible ampliación de este trabajo es realizar actividades para trabajar la conversión entre registros con funciones de mayor grado (cuadráticas, cúbicas, etc.).

Consideramos que nuestro trabajo aprovecha el gran potencial del software CaRMetal para organizar secuencias de actividades interactivas. En efecto, la posibilidad de organizar retroacciones matemáticas y programar retroacciones didácticas da lugar al desarrollo de secuencias automatizadas de tareas que conducen a un aprendizaje por adaptación. El potencial del software para el aprendizaje no radica en su novedad o en la motivación que puede producir en los estudiantes, sino en la naturaleza de las tareas que se le plantean y en el control de las respuestas de los estudiantes.

Nosotros logramos ver que la automatización de la secuencia y de la evaluación de los alumnos es posible mediante elementos tecnológicos. Damos muestra de esto con las actividades y con una explicación detallada de lo que es esta programación en los anexos de este trabajo. Esperamos que los profesores de matemáticos interesados en el diseño de actividades y en el control automatizado de secuencias encuentren en este trabajo un material que les permita alcanzar sus objetivos. Pero más que nada, a ver que todo este trabajo puede servir para una reflexión sobre las posibilidades de programación del software CaRMetal en un diseño sobre muchos otros temas matemáticos.

Reconocemos que este trabajo no agota las posibilidades de uso de CaRMetal, al contrario, siempre hubo y habrá algo nuevo posible de hacer con él. De esta manera, nuestras recomendaciones se encaminan a que se vea que el software ya en si es una herramienta llena de posibilidades y que su explotación como profesores es necesaria.

## Referentes Bibliográficos

- Acosta, M., Monroy, L. y Rueda, K. (2010). Situaciones a-didácticas para la enseñanza de la simetría axial utilizando Cabri como medio. *Revista Integración*, Vol. 28, No. 2, pp. 173–189. Recuperado de: <http://matematicas.uis.edu.co/~integracion/Ediciones/vol28N2/V28N2-6Acosta.pdf> el día 1 de noviembre 2015.
- Artigue, M., Douady, R. y Moreno, L. (1995). *Ingeniería didáctica en educación matemática. Un esquema de la investigación y la innovación en la enseñanza y el aprendizaje de la matemática*. Bogotá: Grupo editorial Iberoamérica.
- Barrón, J.; et al. (2009). *La Ecuación de la Línea Recta en la Modelación de Fenómenos Físicos*. Recuperado de: <https://dialnet.unirioja.es/servlet/articulo?codigo=3238578> el día 3 de enero del 2016
- Benítez, A. (2010). Estudio numérico de la gráfica para construir su expresión algebraica. El caso de los polinomios de grado 2 y 3. *Educación Matemática*, vol. 22, núm. 1, pp. 5-29. Recuperado de: <http://www.scielo.org.mx/pdf/ed/v22n1/v22n1a2.pdf>. el día 16 de abril de 2016.
- Beltrán, J., Martín, J. y Pérez, L. (2003). *Cómo aprender con Internet*. Madrid: Fundación Encuentro.
- Bohórquez, L. (2004). Sobre las formas efectivas de incorporar el software cabri-geometrie en la enseñanza de conceptos geométricos en el bachillerato. *Revista de Estudios Sociales*, núm. 19, pp. 106-109. Recuperado de <http://www.redalyc.org/pdf/815/81501909.pdf> el día 7 de abril de 2016.
- Brousseau, G. (1986). *Fundamentos y métodos de la didáctica de las matemáticas*. Traducción Centeno, J.; Melendo, B. y Murillo, J.; pp. 1- 57. Recuperado de: [http://cimate.uagro.mx/ivanlopez/seminario/archivos/Brousseau\\_Fondements.pdf](http://cimate.uagro.mx/ivanlopez/seminario/archivos/Brousseau_Fondements.pdf) el día 12 de diciembre del 2015.
- Debrabant, P. (s.f). *Dgpad CaRMetal free softwares in dynamic geometry*. Recuperado de: <http://carmetal.org/index.php/es/> el día 27 de abril de 2016.
- Díaz, M., Haye, E., MONTENEGRO, F. y CÓRDOBA, L. (2013). Dificultades de los alumnos para articular representaciones gráficas y algebraicas de funciones lineales y cuadráticas. Recuperado de: <http://www.centroedumatematica.com/memorias-icemacyc/373-401-2-DR-C.pdf> el día 24 enero 2016.
- Duval, R. (1981). Gráficos e equações: a articulação de dois registros [Gráficos y ecuaciones: la articulación de dos registros]. Recuperado de: <https://periodicos.ufsc.br/index.php/revemat/article/view/1981-1322.2011v6n2p96/21794> el día 4 de marzo de 2016.
- Duval, R. (1981). A leitura das representações gráficas requer dos alunos a discriminação das diferentes variáveis visuais pertinentes constituintes deste tipo de representação. *Requer*

também que os alunos tenham consciência das correspondências entre as variações. Traducción nuestra.

- Duval, R. (1981). Quando se trata de partir da representação gráfica para encontrar, por exemplo, a equação correspondente ou para utilizar o conceito de inclinação ou de direção, é esta abordagem de interpretação global que se torna necessária. A razão disto se deve ao fato de que o recurso à abordagem ponto a ponto é totalmente inoperante uma vez que tira a atenção das variáveis visuais. A prática sistemática da abordagem ponto a ponto não favorece a abordagem de interpretação global que é em geral deixada de lado no ensino. Traducción nuestra.
- Duval, R. (1999). Semiosis y pensamiento humano. Registros semióticos y aprendizajes intelectuales. [Traducido al español de Sémiosis et Pensée Humaine. Registres sémiotiques et apprentissages intellectuels]. Cali: Universidad del Valle.
- Duval, R. (2012). Quais teorias e métodos para a pesquisa sobre o ensino da matemática? [¿Cuáles teorías y métodos para la investigación sobre la enseñanza de la matemática?]. Recuperado de: <http://www.redalyc.org/pdf/894/89424874002.pdf> el día 3 de marzo de 2016.
- Duval, R. (2012). O ponto de vista cognitivo, compreender em matemática é, antes de tudo, reconhecer os objetos matemáticos representados. Traducción nuestra.
- Duval, R. (2012). Ser capaz de reconhecer o mesmo objeto em duas representações semi-óticas diferentes implica que, se uma só é dada, é possível espontaneamente convertê-la em outra e mesmo em uma terceira. Traducción nuestra.
- Duval, R. (2012). O acesso aos objetos matemáticos não é nem sensorial nem instrumental, como em física ou em química, mas ele passa pela produção de representações semióticas que não devem jamais ser confundidas com os objetos que elas representam. Traducción nuestra.
- Gouvea, F. (2005). Um Estudo de Fractais Geométricos através de Caleidoscópios e Softwares de Geometria Dinâmica [Un estudio de fractales geométricos a través de Caleidoscopios y softwares de geometría dinámica]. Recuperado de: <http://www.redalyc.org/bdigital.udistrital.edu.co:8080/pdf/2912/291221859011.pdf> el día 7 de abril de 2016.
- Gouvea, F. (2005). Para o estabelecimento de um ambiente de aprendizagem agradável e participativo. Traducción nuestra.
- Guzmán, M. (1993). La inserción de las tecnologías ¿puede cambiar las prácticas matemáticas actuales? Recuperado de: [http://repositoriorecursos-download.educ.ar/dinamico/UnidadHtml\\_get\\_11f016e2-7a07-11e1-8388-ed15e3c494af/index.html](http://repositoriorecursos-download.educ.ar/dinamico/UnidadHtml_get_11f016e2-7a07-11e1-8388-ed15e3c494af/index.html) el día 26 de enero 2016.
- Lupiáñez, J. y Moreno, L. (2001). Tecnología y representaciones semióticas en el aprendizaje de las matemáticas. En Gómez, P.; Rico, L. (Eds.), Iniciación a la investigación en didáctica de la matemática. Homenaje al profesor Mauricio Castro (pp. 291-300). Granada: Editorial Universidad de Granada. Recuperado de: <http://www.uv.es/aprenggeom/archivos2/homenaje/00Indice.PDF> el 2 de enero del 2016.

- MEN (2008). Estándares básicos de competencias en matemáticas. Recuperado de: <http://www.colombiaaprende.edu.co/html/mediateca/1607/article-167733.html> el día 13 de enero de 2016.
- Peralta, J. (s.f). Dificultades para articular los registros gráfico, algebraico y tabular: el caso de la función lineal. Recuperado de: <http://semana.mat.uson.mx/MemoriasXVII/XII/Peralta%20Garcia.pdf> el día 29 de marzo de 2016.
- Roldán, E. (2013). El aprendizaje de la función lineal, propuesta didáctica para estudiantes de 8° y 9° grados de educación básica. Recuperado de: <http://www.bdigital.unal.edu.co/12943/1/1186875.2013.pdf> el día 08 de marzo del 2016.
- Rueda, M. y Niño, Á. (2013). Automatización de actos de devolución en el software Cabri LM. Recuperado de: <https://www.dropbox.com/s/u67nz75eoax888n/Automatizaci%C3%B3n%20de%20actos%20de%20devoluci%C3%B3n%20en%20el%20software%20CABRI%20LM.pdf?dl=0> el día 6 de abril de 2016.
- Sacco, A. (2007). Algunos factores que obstaculizan el uso de la informática en la educación especial. Recuperado de: [http://www.antoniosacco.com.ar/docu/factores\\_que\\_obstaculizan\\_original.pdf](http://www.antoniosacco.com.ar/docu/factores_que_obstaculizan_original.pdf) el día 29 de abril de 2016.
- San Martín, O. (2007). Un registro de representación semiótica de naturaleza geométrica para la trigonometría. Memorias XII congreso nacional de investigación educativa- Educación y conocimientos disciplinares. Recuperado de: <http://www.comie.org.mx/congreso/memoriaelectronica/v09/ponencias/at05/PRE1178828913.pdf> el día 13 de diciembre de 2015.

## Anexos

### Programación de la secuencia de actividades

En el diseño de las actividades se hace uso de una programación con scripts, los cuales permitirán entre algunas opciones, trabajar todas las actividades en una sola figura o ventana de CaRMetal y controlar el momento en el que un estudiante pueda pasar de una tarea a otra (navegar). A continuación, se detalla la programación de los scripts utilizados, junto con una breve descripción de lo que esta programación hace.

#### Los scripts.

##### I. Tareas.

En este script se controla la aparición de los enunciados de las tareas de la actividad, mediante un mensaje o alerta (Alert) que se ejecuta al hacer clic sobre el botón Tarea. El estudiante puede ejecutar este script en cualquier momento.

Se utiliza una variable escrita en una expresión para controlar el número de la tarea que se le está planteando al estudiante. El script recupera esa variable y muestra el enunciado de la tarea correspondiente.

Lo que internamente hará este script cuando se oprima el botón es:

<b><u>Tomo 1: Conversión algebraica a la gráfica.</u></b>	<b><u>Tomo 2: Conversión gráfica a la algebraica.</u></b>	<b>Descripción</b>
<code>tarea=GetExpressionValue("Tarea")</code>	<code>tarea=GetExpressionValue("Tarea");</code>	Este comando recupera y asigna a la variable tarea el valor que tiene la expresión "Tarea" en la figura. Según sea el caso de la actividad de conversión, los valores recuperados serán un número entero entre el 1 y el 7 o entre el 1 y el 6. En cada actividad los valores de la expresión corresponden con la secuencia de actividades que fue descrita en el diseño de cada tomo.
<code>switch (tarea)</code>	<code>switch (tarea)</code>	

<pre> { case (1): EnunciadoTarea1 (); break;  case (2): EnunciadoTarea2 (); break;  case (3): EnunciadoTarea3Parte1 (); break;  case (4): EnunciadoTarea3 Parte2 (); break;  case (5): EnunciadoTarea4 (); break;  case (6): EnunciadoTarea5 (); break;  case (7): EnunciadoTarea6 (); break; } </pre>	<pre> { case (1): EnunciadoTarea1 (); break;  case (2): EnunciadoTarea2 (); break;  case (3): EnunciadoTarea3 (); break;  case (4): EnunciadoTarea4 (); break;  case (5): EnunciadoTarea5 (); break;  case (6): EnunciadoTarea6 (); break; } </pre>	<p>Aquí, se ejecuta un condicional de casos posibles (switch case), con el que se evalúa el valor de la variable tarea. Cada caso (case()) tiene asociado una función que hemos denominado como EnunciadoTarea_ (). En el momento en que un caso se ejecuta, el script busca el nombre de la función asociada y la ejecuta.</p>
--	---	---

```
function EnunciadoTarea1 ()
```

```
{  
Alert ("La ecuación de la  
recta roja es de la forma  
 $y=ax+b$ . \n Despliegue la  
lista de números para  
modificar el valor de b en  
esa ecuación, \n hasta lograr  
que la recta roja quede sobre  
la recta azul.");  
}
```

```
function EnunciadoTarea2 ()
```

```
{  
Alert ("La ecuación de la  
recta roja es de la forma  
 $y=ax+b$ . \n Escriba en la  
casilla el valor  
correspondiente a b en la  
ecuación, \n para que la  
recta roja quede sobre la  
recta azul.");  
}
```

```
function EnunciadoTarea3Parte1 ()
```

```
{  
OrdenadaB=GetExpression
```

```
function EnunciadoTarea1()
```

```
{  
Alert ("La ecuación de la  
recta roja es de la forma  
 $y=ax+b$ . \n Mueva el punto  
C sobre el eje y, hasta  
hacer que la ecuación roja  
sea igual a la ecuación  
azul.");  
}
```

```
function EnunciadoTarea2()
```

```
{  
Alert ("La ecuación de la  
recta roja es de la forma  
 $y=ax+b$ . \n El punto D es  
el punto de la recta roja  
que está sobre el eje Y. \n  
Señale la posición donde  
debe quedar D para que la  
ecuación roja sea igual a la  
ecuación azul.");  
}
```

```
function EnunciadoTarea3()
```

```
{
```

Las funciones asociadas en el script, describen de manera ordenada lo que en cada caso de las tareas se ejecutará. En general, las funciones aquí mencionadas ejecutan un Alert; mensaje que se mostrará como un cuadro de dialogo automático en la pantalla (ver imagen 20), describiendo el enunciado de la tarea.

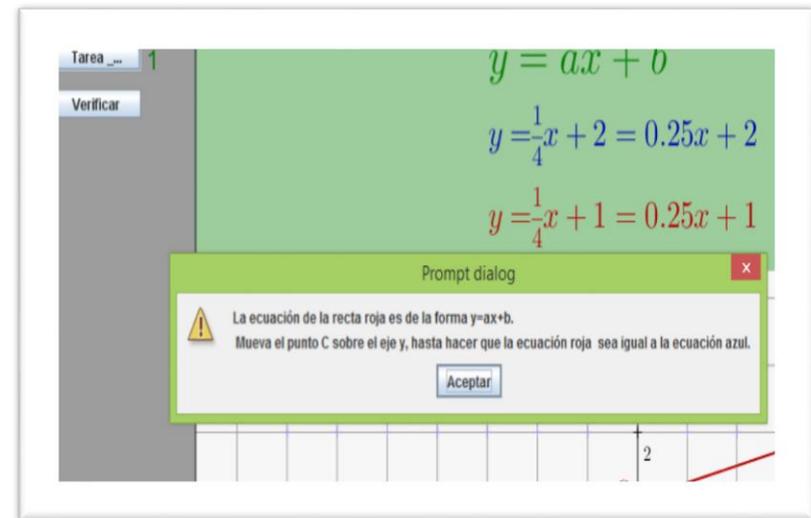


Imagen 20.

<pre>Value("OrdenadaB"); Alert ("La ecuación de la recta roja es de la forma <math>y=ax+b</math>.\n La recta roja corta el eje y en "+OrdenadaB+", utilice la lista desplegable para modificar el numerador del término a en la ecuación, \n hasta lograr que la recta roja quede sobre la recta azul."); }</pre>	<pre>OrdenadaB=GetExpressio nValue("ORDENADAB") ; Alert ("La ecuación de la recta roja es de la forma <math>y=ax+b</math>. \n La recta roja pasa por el punto "+OrdenadaB+" del eje y, mueva el punto G por la cuadrícula, hasta hacer que la ecuación roja sea igual a la ecuación azul."); }</pre>	
<pre>function EnunciadoTarea3Parte2 () { OrdenadaB=GetExpression Value("OrdenadaB"); Alert ("La ecuación de la recta roja es de la forma <math>y=ax+b</math>.\n La recta roja corta el eje y en "+OrdenadaB+", utilice la lista desplegable para modificar el denominador del término a en la ecuación, \n hasta lograr que la recta roja quede sobre la recta azul."); }</pre>	<pre>function EnunciadoTarea4() { OrdenadaB=GetExpressio nValue("ORDENADAB") ; Alert ("La ecuación de la recta roja es de la forma <math>y=ax+b</math>. \n La recta roja pasa por el punto "+OrdenadaB+" del eje y,  señale la posición donde debe quedar el punto H para que la ecuación roja sea igual a la ecuación azul."); }</pre>	

<pre>function EnunciadoTarea4 () { OrdenadaB=GetExpression Value("OrdenadaB"); Alert ("La ecuación de la recta roja es de la forma y=ax+b.\n La recta corta el eje y en "+OrdenadaB+", escriba en las casillas los valores del numerador y denominador de a \n para que la recta roja quede sobre la recta azul."); }  function EnunciadoTarea5 () { Alert ("La ecuación de la recta roja es de la forma y=ax+b.\n Utilice las listas desplegables para modificar el valor de a y de b en la ecuación \n hasta lograr que la recta roja quede sobre la recta azul."); }  function EnunciadoTarea6 ()</pre>	<pre>}  function EnunciadoTarea5() { Alert ("La ecuación de la recta roja es de la forma y=ax+b. \n Mueva el punto I sobre el eje y y el punto J sobre la cuadrícula, hasta hacer que la ecuación roja sea igual a la ecuación azul."); }  function EnunciadoTarea6() { Alert ("La ecuación de la recta roja es de la forma y=ax+b. \n Señale sucesivamente la posición donde deben quedar los puntos K y L, para que la ecuación roja sea igual a la ecuación azul. \n Tenga en cuenta que el punto K es el punto de la recta roja que</pre>	
--	---	--

<pre>{ Alert ("La ecuación de la recta roja es de la forma y=ax+b.\n  Escriba en las casillas los valores correspondientes a a y a b en la ecuación, \n para que la recta roja quede sobre la recta azul."); }</pre>	<pre>está sobre el eje Y. "); }</pre>	
--	---------------------------------------	--

## II. Verificar.

Este script evalúa la respuesta a una tarea del estudiante, mostrándole un mensaje de evaluación y presentándole un nuevo ejercicio en caso de que la respuesta sea correcta. El script se ejecuta al hacer clic sobre el botón “verificar”.

El script incrementa las variables “Correctos\_” y “ContadorMT\_”, almacenando sus valores en expresiones ocultas en la figura. Estas expresiones permiten determinar por tarea la cantidad de ejercicios solucionados bien y mal. Cuando el estudiante ha resuelto correctamente 5 ejercicios y el número de intentos incorrectos es menor que el número de intentos correctos, este script muestra el botón continuar.

<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
t = getZC().JCM.CPs;	t = getZC().JCM.CPs;	Este comando recupera en la variable t, todos los controles de la figura (botones, casillas, listas desplegables, etc.). Los controles no son reconocidos como objetos de la figura en el script, razón por la que debe utilizarse este comando para posteriormente llamarlos y utilizarlos en él.

<pre>for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getName().equals("continuar")) { continuar=t.get(i); break; } }</pre>	<pre>for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getName().equals("continuar")) { continuar=t.get(i); break; } }</pre>	<p>Este bucle recorre la lista de los controles almacenados en la variable t, recuperando el botón continuar, para poder mostrarlo cuando el estudiante ha completado 5 o más ejercicios.</p>
<pre>tarea=GetExpressionValue("Tarea");</pre>	<pre>tarea=GetExpressionValue("Tarea");</pre>	<p>Este comando recupera y asigna a la variable tarea el valor que tiene la expresión “Tarea” en la figura. Según sea el caso de la actividad de conversión, los valores recuperados serán un número entero entre el 1 y el 7 o entre el 1 y el 6. En cada actividad los valores de la expresión corresponden con la secuencia de actividades que fue descrita en el diseño de cada tomo.</p>
<pre>switch (tarea) { case 1: VerificarTarea1 (); break;  case 2: VerificarTarea2 (); break;  case 3: VerificarTarea3Parte1 (); break;</pre>	<pre>switch (tarea) { case (1): VerificarTarea1 (); break;  case (2): VerificarTarea2 (); break;  case (3): VerificarTarea3 (); break;</pre>	<p>Aquí se ejecuta un condicional de casos posibles (switch case), con el que se evalúa el valor de la variable tarea. Cada caso (case()) tiene asociado una función que hemos denominado como VerificarTarea_(). En el momento en que un caso se ejecuta, el script busca el nombre de la función asociada y la ejecuta.</p>

<pre> case 4: VerificarTarea3Parte2 (); break;  case 5: VerificarTarea4 (); break;  case 6: VerificarTarea5 (); break;  case 7: VerificarTarea6 (); break; } </pre>	<pre> case (4): VerificarTarea4 (); break;  case (5): VerificarTarea5 (); break;  case (6): VerificarTarea6 (); break; } </pre>	
---	---	--

**Sobre las funciones *VerificarTarea ()***

Como ya lo señalamos, estas funciones se ejecutan a partir del comando *switch case*. Estas funciones evalúan para cada tarea, si el estudiante la resolvió o no. Si la resolvió, le muestra un nuevo ejercicio, si no, le pide que vuelva a trabajar en el mismo ejercicio. Cada tarea verifica cosas distintas, por lo que se propone mostrar una descripción detallada de cada función, así:

Para la lectura de las tablas se debe tener en cuenta, la variación de colores (gris y verde) y las celdas sin llenar. Los colores estarán indicando dónde empieza un condicional y dónde termina el mismo, esto sólo para los más extensos. Las celdas vacías indicarán que la descripción del script solo se hace para uno de los tomos, donde está escrito.

<i>function VerificarTarea1() {</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ct=GetExpressionValue("ContadorT1");	ct=GetExpressionValue("ContadorT1");	Recupera y asigna a la variable ct, el valor que tiene la expresión “ContadorT1” en la figura. Esta expresión almacena la cantidad de veces que el estudiante ha oprimido el botón “verificar”.
SetExpressionValue("ContadorT1", ct+1);	SetExpressionValue("ContadorT1", ct+1);	Aumenta en 1 la variable ct y lo almacena en la expresión “ContadorT1” de la figura.
if((Math.abs(Y("C")-Y("B")))<0.0001)	if((Math.abs(Y("C")-Y("B")))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre la ordenada de C y la ordenada de B, es menor que 0.0001. Esto, para saber si la tarea ha sido resuelta o no por el alumno, ya que la tarea se considera resuelta cuando los puntos B y C, sobre el eje y, coinciden. El software reconoce diferencias sumamente pequeñas entre las coordenadas, por lo que no se utiliza una igualdad sino una aproximación.
{	{	<i>Si lo anterior sucede entonces:</i>
Alert ("Muy bien, Ahora hazlo con esta nueva recta");	Alert ("Muy bien, Ahora hazlo con esta nueva ecuación");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
g=GetExpressionValue("c	g=GetExpressionValue("c	Recupera y asigna a la variable g el valor que tiene la expresión

orrectos");	orrectos");	“correctos” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” la tarea está resuelta bien.
SetExpressionValue("correctos",g+1);	SetExpressionValue("correctos",g+1);	Aumenta en 1 la variable g y lo almacena en la expresión “correctos” de la figura.
ordeC=GetExpressionValue("ValorListaDesplegableCorteOrdenadaT3");	ordeC=GetExpressionValue("ORDENADAC");	Recupera y asigna a la variable ordeC el valor de la expresión que calcula la ordenada del punto C, en nuestro sistema de coordenadas. Para el tomo 1 esa expresión es: "ValorListaDesplegableCorteOrdenadaT3") y para el tomo 2 es: "ORDENADAC".
ordeB=Math.round(Math.random()*8-4);	ordeB=Math.round(Math.random()*8-4);	Calculo de la nueva Ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.
if ((Math.abs(ordeC-ordeB))<0.0001)	if ((Math.abs(ordeC-ordeB))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeC y ordeB del script, es menor que 0.0001, o lo que es casi lo mismo, que la ordeC sea igual a la ordeB. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos B y C sean iguales y por tanto que el nuevo ejercicio aparezca resuelto al estudiante. La razón por la que la diferencia entre ordeC y ordeB no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.
{	{	Si lo anterior sucede, este comando asigna a la variable OrdeB, el

<pre>ordeB=(ordeB+5) %8-4 }</pre>	<pre>ordeB=(ordeB+5) %8-4 }</pre>	<p>valor del módulo de OrdeB+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>absiA=Math.round(Math.random()*14-7);</pre>	<pre>absiA=Math.round(Math.random()*14-7);</pre>	<p>Calculo de la nueva abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.</p>
<pre>if (absiA==0) { absiA=1 }</pre>	<pre>if (absiA==0) { absiA=1 }</pre>	<p>Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.</p>
<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<p>Calculo de la nueva ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.</p>
<pre>SetExpressionValue("ORDENADAB",ordeB);</pre>	<pre>SetExpressionValue("ORDENADAB",ordeB);</pre>	<p>Asigna a la expresión "ORDENADAB" de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada "interna" de</p>

		B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así: $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
SetExpressionValue("ABSCISAA",absiA);	SetExpressionValue("ABSCISAA",absiA);	Asigna a la expresión "ABSCISAA" de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa "interna" de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula: $\text{ABSCISAA} * \text{FactorH} + x(P2)$
SetExpressionValue("OrdenadaA",ordeA); }	SetExpressionValue("ORDENADAA",ordeA); }	Asigna a la expresión "ORDENADAA" de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada "interna" de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula: $\text{ORDENADAA} * \text{FactorH} + y(P2)$
else {	else {	<u>Si esto no sucede entonces:</u>
Alert ("La recta roja aún no está sobre la recta azul, vuelve a intentarlo.");	Alert ("La ecuación roja aún no es igual a la ecuación azul, vuelve a intentarlo.")	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
ctm=GetExpressionValue("ContadorMT1");	ctm=GetExpressionValue("ContadorMT1");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT1" en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón "verificar" la tarea está resuelta mal.
SetExpressionValue("ContadorMT1", ctm+1) }	SetExpressionValue("ContadorMT1", ctm+1) }	Aumenta en 1 la variable ctm y lo almacena en la expresión "ContadorMT1" de la figura.

<code>g=GetExpressionValue("correctos");</code>	<code>g=GetExpressionValue("correctos");</code>	Recupera y asigna a la variable g, el valor que tiene la expresión “correctos” en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional anterior. Al finalizarse el condicional la variable ya no es reconocida por el script.
<code>ctm=GetExpressionVale("ContadorMT1");</code>	<code>ctm=GetExpressionValue("ContadorMT1");</code>	Recupera y asigna a la variable ctm el valor que tiene la expresión “ContadorMT1” en la figura.
<code>if (g&gt;=5&amp;&amp;ctm&lt;g)</code>	<code>if (g&gt;=5&amp;&amp;ctm&lt;g)</code>	Este condicional es el que permite que aparezca el botón continuar para ir a otra tarea. Es un condicional que verifica si: el valor de la variable g es mayor o igual a 5 y si el valor de la variable ctm es menor que el valor de la variable g; recordando con esto que el botón solo aparece cuando el estudiante resuelve 5 ejercicios correctamente y el número de éxitos en el desarrollo de la tarea es mayor al de fracasos.
<code>{</code>	<code>{</code>	<u>Si lo anterior sucede entonces:</u>
<code>Alert ("Escribe en tu cuaderno la relación que hay entre el valor b de la ecuación y la gráfica de la recta, \n luego haz clic en continuar si quieres pasar a la siguiente tarea");</code>	<code>Alert ("Escribe en tu cuaderno la relación que hay entre el punto C que se mueve sobre el eje y, y la ecuación roja, \n luego haz clic en continuar si quieres pasar a la siguiente tarea");</code>	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
<code>continuar.setHidden (false); }</code>	<code>continuar.setHidden (false); }</code>	Coloca el botón “continuar” visible.

else {	else {	<i>Si esto no sucede entonces:</i>
continuar.setHidden (true); } }	continuar.setHidden (true); } }	El botón “continuar” debe estar invisible.

<i>function VerificarTarea2() {</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ct=GetExpressionValue("ContadorT2");	ct=GetExpressionValue("ContadorT2");	Recupera y asigna a la variable ct el valor que tiene la expresión “ContadorT2” en la figura. Esta expresión almacena la cantidad de veces que el estudiante ha oprimido el botón “verificar”.
SetExpressionValue("ContadorT2", ct+1);	SetExpressionValue("ContadorT2", ct+1);	Aumenta en 1 la variable ct y lo almacena en la expresión “ContadorT2” de la figura.
	if((Math.abs(X("D")-X("B")))<0.0001)	Condiciona que solo se ejecuta en el tomo 2 verificando si el valor absoluto de la diferencia entre la abscisa de D y la abscisa de B es menor que 0.0001. Esto para saber si el punto D esta sobre el eje y. La tarea da la posibilidad de colocar el punto D en cualquier parte del plano, pero solo está resuelta si las coordenadas de D y B coinciden. B es intersección de la recta azul con el eje y.
	{	<i>Si lo anterior sucede entonces:</i>
if((Math.abs(Y("D")-Y("B")))<0.0001)	if((Math.abs(Y("D")-Y("B")))<0.0001)	Condiciona que verifica si el valor absoluto de la diferencia entre la ordenada de D y la ordenada de B, es menor que 0.0001. Esto, para

		saber si la tarea ha sido resuelta o no por el alumno, ya que la tarea se considera resuelta cuando los puntos B y D, sobre el eje y, coinciden. El software reconoce diferencias sumamente pequeñas entre las coordenadas, por lo que no se utiliza una igualdad sino una aproximación.
{	{	<i>Si lo anterior sucede entonces:</i>
Alert ("Muy bien, Ahora hazlo con esta nueva recta");	Alert ("Muy bien, Ahora hazlo con esta nueva ecuación");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
g=GetExpressionValue("correctos2");	g=GetExpressionValue("correctos2");	Recupera y asigna a la variable g el valor que tiene la expresión "correctos2" en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón "verificar" la tarea está resuelta bien.
SetExpressionValue("correctos2", g+1);	SetExpressionValue("correctos2", g+1);	Aumenta en 1 la variable g y lo almacena en la expresión "correctos2" de la figura.
ordeD=GetExpressionValue("OrdenadaD");	ordeD=GetExpressionValue("ORDENADAD");	Recupera y asigna a la variable ordeD el valor de la expresión que calcula la ordenada del punto D, en nuestro sistema de coordenadas. Para el tomo 1 esa expresión es: "OrdenadaD" y para el tomo 2 es: "ORDENADAD".
ordeB=Math.round(Math.random()*8-4);	ordeB=Math.round(Math.random()*8-4);	Calculo de la nueva ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego

		redondearlo (Math.round) y asignarlo a la variable ordeB.
if ((Math.abs(ordeD - ordeB))<0.0001)	if ((Math.abs(ordeD - ordeB))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeD y ordeB del script, es menor que 0.0001, o lo que es casi lo mismo, que la ordeD sea igual a la ordeB. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos B y D sean iguales y por tanto que el nuevo ejercicio aparezca resuelto al estudiante. La razón por la que la diferencia entre ordeD y ordeB no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.
{ ordeB=(ordeB+5) %8-4 }	{ ordeB=(ordeB+5)%8-4 }	Si lo anterior sucede, este comando asigna a la variable OrdeB, el valor del módulo de OrdeB+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
absiA=Math.round(Math.random()*14-7);	absiA=Math.round(Math.random()*14-7);	Calculo de la nueva abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
if (absiA==0) { absiA=1 }	if (absiA==0) { absiA=1 }	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.

<code>ordeA=Math.round(Math.random()*8-4);</code>	<code>ordeA=Math.round(Math.random()*8-4);</code>	Calculo de la nueva ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.
<code>SetExpressionValue("ORDENADAB",ordeB);</code>	<code>SetExpressionValue("ORDENADAB",ordeB);</code>	Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así: $\text{PosicionBy}=\text{OrdenadaB}*\text{FactorH}+\text{y}(\text{P2})$
<code>SetExpressionValue("ABSCISAA", absiA);</code>	<code>SetExpressionValue("ABSCISAA", absiA);</code>	Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula: $\text{ABSCISAA}*\text{FactorH}+\text{x}(\text{P2})$
<code>SetExpressionValue("ORDENADAA", ordeA);</code> <code>}</code>	<code>SetExpressionValue("ORDENADAA", ordeA);</code>	Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula: $\text{ORDENADAA}*\text{FactorH}+\text{y}(\text{P2})$
	<code>ctm=GetExpression</code>	Recupera y asigna a la variable ctm el valor que tiene la expresión

	Value("ContadorMT2");	“ContadorMT2” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” la tarea está resuelta mal.
	if (g>=5&&ctm<g)	Este condicional es el que permite que aparezca el botón continuar para ir a otra tarea. Es un condicional que verifica si: el valor de la variable g es mayor o igual a 5 y si el valor de la variable ctm es menor que el valor de la variable g; recordando con esto que el botón solo aparece cuando el estudiante resuelve 5 ejercicios correctamente y el número de éxitos en el desarrollo de la tarea es mayor al de fracasos.
	{	<i>Si lo anterior sucede entonces:</i>
	Alert ("Haz clic en continuar si quieres pasar a la siguiente tarea");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
	continuar.se tHidden (false); }	Coloca el botón “continuar” visible.
	else {	<i>Si esto no sucede entonces:</i>
	continuar.se	El botón “continuar” debe estar invisible.

	<pre>tHidden (true); }</pre>	
	<pre>entradainteractivaD (); }</pre>	El script busca la función con este nombre y la ejecuta.
else	else	<u>Si esto no sucede entonces:</u>
{	{	
Alert ("La recta roja aún no está sobre la recta azul, vuelve a intentarlo.");	Alert ("La ecuación roja aún no es igual a la ecuación azul, vuelve a intentarlo.");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
ctm=GetExpressionValue("ContadorMT2");	ctm=GetExpressionValue("ContadorMT2");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT2" en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón "verificar" la tarea está resuelta mal.
SetExpressionValue("ContadorMT2", ctm+1)	SetExpressionValue("ContadorMT2", ctm+1)	Aumenta en 1 la variable ctm y lo almacena en la expresión "ContadorMT2" de la figura.
}	}	
	<pre>entradainteractivaD (); }</pre>	El script busca la función con este nombre y la ejecuta.
	else	<u>Si esto no sucede entonces:</u>
	{	

	Alert ("D no está sobre el eje Y, vuelve a intentarlo.");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
	ctm=GetExpressionValue("ContadorMT2");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT2" en la figura. Se debe recuperar nuevamente el valor de esta expresión, ya que se ha cerrado el condicional en el tomo 2.
	SetExpressionValue("ContadorMT2", ctm+1)	Aumenta en 1 la variable ctm y lo almacena en la expresión "ContadorMT2" de la figura.
	<pre> entradainteractivaD (); } } </pre>	El script busca la función con este nombre y la ejecuta.
g=GetExpressionValue("correctos2");		Recupera y asigna a la variable g el valor que tiene la expresión "correctos2" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente en el tomo 1. Al finalizarse el condicional la variable ya no es reconocida por el script.
ctm=GetExpressionVale("ContadorMT2");		Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT2" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente en el tomo 1. Al finalizarse el condicional la variable ya no es reconocida por el script.
if (g>=5&&ctm<g)		Este condicional es el que permite que aparezca el botón continuar para ir a otra tarea. Es un condicional que verifica si: el valor de la variable g es mayor o igual a 5 y si el valor de la variable ctm es

		menor que el valor de la variable g; recordando con esto que el botón solo aparece cuando el estudiante resuelve 5 ejercicios correctamente y el número de éxitos en el desarrollo de la tarea es mayor al de fracasos.
{		<i>Si lo anterior sucede entonces:</i>
Alert ("Haz clic en continuar si quieres pasar a la siguiente tarea");		Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
continuar.setHidden (false); }		Coloca el botón “continuar” visible.
else {		<i>Si esto no sucede entonces:</i>
continuar.setHidden (true); } }		El botón “continuar” debe estar invisible.

<i>function VerificarTarea3() {</i>	
<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ct=GetExpressionValue("ContadorT3");	Recupera y asigna a la variable ct el valor que tiene la expresión “ContadorT3” en la figura. Está expresión almacena la cantidad de

	veces que el estudiante ha oprimido el botón “verificar”.
SetExpressionValue("ContadorT3", ct+1);	Aumenta en 1 la variable ct y lo almacena en la expresión “ContadorT3” de la figura.
if((Math.abs(((Y("A")-Y("B"))/(X("A")-X("B")))-((Y("B")-Y("G"))/(X("B")-X("G")))))<0.0001)	Condiciona que verifica si el valor absoluto de la diferencia entre la pendiente de la recta AB, y la pendiente de la recta GB, es menor que 0.0001. Esto para saber si la tarea ha sido resuelta o no por el alumno, ya que la tarea se considera resuelta cuando las pendientes de esas rectas son iguales. El software reconoce diferencias sumamente pequeñas entre las coordenadas, por lo que no se utiliza una igualdad sino una aproximación.
{	<i>Si lo anterior sucede entonces:</i>
Alert ("Muy bien, Ahora hazlo con esta nueva ecuación");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de diálogo automático en la pantalla.
g=GetExpressionValue("correctos3");	Recupera y asigna a la variable g el valor que tiene la expresión “correctos3” en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón “verificar” la tarea está resuelta bien.
SetExpressionValue("correctos3", g+1);	Aumenta en 1 la variable g y lo almacena en la expresión “correctos3” de la figura.
ordeG=GetExpressionValue("ORDENADAG");	Recupera y asigna a la variable ordeG el valor de la expresión que calcula la ordenada del punto G, en nuestro sistema de coordenadas.

<pre>absiG=GetExpressionValue("ABSCISAG");</pre>	<p>Recupera y asigna a la variable absiG el valor de la expresión que calcula la abscisa del punto G, en nuestro sistema de coordenadas.</p>
<pre>ordeB=Math.round(Math.random()*8-4);</pre>	<p>Calculo de la nueva ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.</p>
<pre>absiA=Math.round(Math.random()*14-7);</pre>	<p>Calculo de la nueva abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.</p>
<pre>if ((Math.abs(absiA-absiG))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiG del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las abscisas de los puntos A y G sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y GB tienen la misma pendiente estos puntos quedan superpuestos.</p>
<pre>{ absiA=(absiA+8)%14-7 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).</p>

<pre>if (absiA==0) { absiA=1 }</pre>	<p>Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.</p>
<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<p>Calculo de la nueva ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.</p>
<pre>if ((Math.abs(ordeA-ordeG))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeG del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos A y G sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y GB tienen la misma pendiente estos puntos quedan superpuestos.</p>
<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>SetExpressionValue("ORDENADAB",ordeB);</pre>	<p>Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de</p>

	<p>coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy}=\text{OrdenadaB}*\text{FactorH}+\text{y}(\text{P2})$
<pre>SetExpressionValue("ABSCISAA",absiA);</pre>	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA}*\text{FactorH}+\text{x}(\text{P2})$
<pre>SetExpressionValue("ORDENADAA",ordeA); }</pre>	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA}*\text{FactorH}+\text{y}(\text{P2})$
<pre>else {</pre>	<p><u>Si esto no sucede entonces:</u></p>
<pre>Alert ("La ecuación roja aún no es igual a la ecuación azul, vuelve a intentarlo.")</pre>	<p>Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.</p>
<pre>ctm=GetExpressionValue("ContadorMT3");</pre>	<p>Recupera y asigna a la variable ctm el valor que tiene la expresión “ContadorMT3” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” la tarea está resuelta mal.</p>
<pre>SetExpressionValue("ContadorMT3", ctm+1)</pre>	<p>Aumenta en 1 la variable ctm y lo almacena en la expresión</p>

}	"ContadorMT3" de la figura.
g=GetExpressionValue("correctos3");	Recupera y asigna a la variable g el valor que tiene la expresión "correctos3" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente Al finalizarse el condicional la variable ya no es reconocida por el script.
ctm=GetExpressionValue("ContadorMT3");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT3" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente Al finalizarse el condicional la variable ya no es reconocida por el script.
if (g>=5&&ctm<g)	Este condicional es el que permite que aparezca el botón continuar para ir a otra tarea. Es un condicional que verifica si: el valor de la variable g es mayor o igual a 5 y si el valor de la variable ctm es menor que el valor de la variable g; recordando con esto que el botón solo aparece cuando el estudiante resuelve 5 ejercicios correctamente y el número de éxitos en el desarrollo de la tarea es mayor al de fracasos.
{	<i>Si lo anterior sucede entonces:</i>
Alert ("Escribe en tu cuaderno la relación entre el punto G que se mueve sobre la cuadrícula, y la ecuación roja, \n luego haz clic en continuar si quieres pasar a la siguiente tarea ");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de diálogo automático en la pantalla.
continuar.setHidden (false); }	Coloca el botón "continuar" visible.

else	<i>Si esto no sucede entonces:</i>
{	
continuar.setHidden (true);	El botón “continuar” debe estar invisible.
}	

<i>function VerificarTarea3Parte1() {</i>	
<b><u>Tomo 1: Conversión algebraica a la gráfica.</u></b>	<b>Descripción</b>
ct=GetExpressionValue("ContadorT3Parte1");	Recupera y asigna a la variable ct el valor que tiene la expresión “ContadorT3Parte1” en la figura. Esta expresión almacena la cantidad de veces que el estudiante ha oprimido en esta tarea, el botón “verificar”.
SetExpressionValue("ContadorT3Parte1", ct+1);	Aumenta en 1 la variable ct y lo almacena en la expresión “ContadorT3Parte1” de la figura.
if((Math.abs(((Y("A")-Y("B"))/(X("A")-X("B")))-((Y("B")-Y("M"))/(X("B")-X("M")))))<0.0001)	Condiciona que verifica si el valor absoluto de la diferencia entre la pendiente de la recta AB, y la pendiente de la recta MB, es menor que 0.0001. Esto, para saber si la tarea ha sido resuelta o no por el alumno, ya que la tarea se considera resuelta cuando las pendientes de estas rectas coinciden. El software reconoce diferencias sumamente pequeñas entre las coordenadas, por lo que no se utiliza una igualdad sino una aproximación.
{	<i>Si lo anterior sucede entonces:</i>

Alert ("Muy bien, Ahora hazlo con esta nueva recta");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
if(SetHide("OrdenadaM,TrianguloRectaAzul,TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1",true))	Condicional que verifica si los objetos de la figura mencionados entre paréntesis y separados por comas, están visibles en la pantalla. Este condicional permite que la ayuda que usa el estudiante solo aparezca en el ejercicio que él así decida, y que en el nuevo ejercicio los objetos de la ayuda desaparezcan.
{	<i>Si lo anterior sucede entonces:</i>
SetHide("OrdenadaM,TrianguloRectaAzul,TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1",false) }	Coloca a los objetos de la figura mencionados entre paréntesis y separados por comas, invisibles en la pantalla.
g=GetExpressionValue("correctos3Parte1");	Recupera y asigna a la variable g el valor que tiene la expresión “correctos3Parte1” en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón “verificar” la tarea está resuelta bien.
SetExpressionValue("correctos3Parte1", g+1);	Aumenta en 1 la variable g y lo almacena en la expresión “correctos3Parte1” de la figura.
ordeM=GetExpressionValue("ValorListaDesplegableNumerada");	Recupera y asigna a la variable ordeM el valor de la expresión que

<code>dorPendiente");</code>	calcula la ordenada del punto M, en nuestro sistema de coordenadas.
<code>ordeB=Math.round(Math.random()*8-4);</code>	Calculo de la nueva ordenada de B: Calcula un valor aleatorio ( <code>Math.random()</code> ), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo ( <code>Math.round</code> ) y asignarlo a la variable <code>ordeB</code> .
<code>absiA=Math.round(Math.random()*14-7);</code>	Calculo de la nueva abscisa de A: Calcula un valor aleatorio ( <code>Math.random()</code> ), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo ( <code>Math.round</code> ) y asignarlo a la variable <code>absiA</code> .
<code>if (absiA==0)   {     absiA=1   }</code>	Condicional que verifica si <code>absiA</code> es igual a 0. Si esto sucede, entonces, asigna a la variable <code>absiA</code> el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
<code>ordeA=Math.round(Math.random()*8-4);</code>	Calculo de la nueva ordenada de A: Calcula un valor aleatorio ( <code>Math.random()</code> ), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo ( <code>Math.round</code> ) y asignarlo a la variable <code>ordeA</code> .
<code>if ((Math.abs(ordeA-ordeM))&lt;0.0001)</code>	Condicional que verifica si el valor absoluto de la diferencia entre las variables <code>ordeA</code> y <code>ordeM</code> del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos A y M sean iguales y por tanto que el ejercicio aparezca resuelto al estudiante, pues cuando las rectas

	AB y MB tienen la misma pendiente estos puntos quedan superpuestos.
<pre>{ ordeA=(ordeA+8)%14-7 }</pre>	Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).
<pre>SetExpressionValue("ORDENADAB", ordeB);</pre>	Asigna a la expresión "ORDENADAB" de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada "interna" de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así: $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
<pre>SetExpressionValue("ABSCISAA", absiA);</pre>	Asigna a la expresión "ABSCISAA" de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa "interna" de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula: $\text{ABSCISAA} * \text{FactorH} + x(P2)$
<pre>SetExpressionValue("ORDENADAA", ordeA); }</pre>	Asigna a la expresión "ORDENADAA" de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada "interna" de A con la expresión PosicionAy (oculta en la figura). La expresión

	PosicionAy calcula: $\text{ORDENADAA} * \text{FactorH} + y(P2)$
else	<i>Si esto no sucede entonces:</i>
{	
Alert ("La recta roja aún no está sobre la recta azul, vuelve a intentarlo.")	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
ctm=GetExpressionValue("ContadorMT3Parte1");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT3Parte1" en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón "verificar" la tarea está resuelta mal.
SetExpressionValue("ContadorMT3Parte1", ctm+1)	Aumenta en 1 la variable ctm y lo almacena en la expresión "ContadorMT3Parte1" de la figura.
}	
g=GetExpressionValue("correctos3Parte1");	Recupera y asigna a la variable g el valor que tiene la expresión "correctos3Parte1" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente. Al finalizarse el condicional la variable ya no es reconocida por el script.
ctm=GetExpressionValue("ContadorMT3");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT3" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente. Al finalizarse el condicional la variable ya no es reconocida por el script.
if (g>=5&&ctm<g)	Este condicional es el que permite que aparezca el botón continuar

	para ir a otra tarea. Es un condicional que verifica si: el valor de la variable g es mayor o igual a 5 y si el valor de la variable ctm es menor que el valor de la variable g; recordando con esto que el botón solo aparece cuando el estudiante resuelve 5 ejercicios correctamente y el número de éxitos en el desarrollo de la tarea es mayor al de fracasos.
{	<i>Si lo anterior sucede entonces:</i>
Alert "Escribe en tu cuaderno la relación entre el valor del numerador de a en la ecuación y la gráfica de la recta, \n luego haz clic en continuar si quieres pasar a la siguiente tarea");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
continuar.setHidden (false); }	Coloca el botón “continuar” visible.
else {	<i>Si esto no sucede entonces:</i>
continuar.setHidden (true); } }	El botón “continuar” debe estar invisible.

<i>function VerificarTarea3Parte2() {</i>	
<b><u>Tomo 1: Conversión algebraica a la gráfica.</u></b>	<b>Descripción</b>
ct=GetExpressionValue("ContadorT3Parte2");	Recupera y asigna a la variable ct el valor que tiene la expresión

	“ContadorT3Parte2” en la figura. Está expresión almacena la cantidad de veces que el estudiante ha oprimido el botón “verificar”.
SetExpressionValue("ContadorT3Parte2", ct+1);	Aumenta en 1 la variable ct y lo almacena en la expresión “ContadorT3Parte2” de la figura.
if((Math.abs(((Y("A")-Y("B"))/(X("A")-X("B")))-((Y("B")-Y("G"))/(X("B")-X("G")))))<0.0001)	Condiciona que verifica si el valor absoluto de la diferencia entre la pendiente de la recta AB, y la pendiente de la recta GB, es menor que 0.0001. Esto para saber si la tarea ha sido resuelta o no por el alumno, ya que la tarea se considera resuelta cuando las pendientes de esas rectas son iguales. El software reconoce diferencias sumamente pequeñas entre las coordenadas, por lo que no se utiliza una igualdad sino una aproximación.
{	<i>Si lo anterior sucede entonces:</i>
Alert ("Muy bien, Ahora hazlo con esta nueva recta");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de diálogo automático en la pantalla.
if(SetHide("OrdenadaG,TrianguloRectaAzul,TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,Segmento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Segmento3TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2",true))	Condiciona que verifica si los objetos de la figura mencionados entre paréntesis y separados por comas, están visibles en la pantalla. Este condicional permite que la ayuda que usa el estudiante solo aparezca en el ejercicio que él así decida, y que en el nuevo ejercicio los objetos de la ayuda desaparezcan.
{	<i>Si lo anterior sucede entonces:</i>
SetHide("OrdenadaG,TrianguloRectaAzul,TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaAzul,Seg	Coloca a los objetos de la figura mencionados entre paréntesis y separados por comas, invisibles en la pantalla.

<pre> mento2TrianguloRectaAzul,Segmento1TrianguloRecta RojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2 ,Segmento3TrianguloRectaRojaT3Parte2,MedidaSegm ento1TrianguloRectaRojaT3Parte2,MedidaSegmento2 TrianguloRectaRojaT3Parte2",false) }; </pre>	
<pre> g=GetExpressionValue("correctos3Parte2"); </pre>	<p>Recupera y asigna a la variable g el valor que tiene la expresión “correctos3Parte2” en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón “verificar” la tarea está resuelta bien.</p>
<pre> SetExpressionValue("correctos3Parte2", g+1); </pre>	<p>Aumenta en 1 la variable g y lo almacena en la expresión “correctos3Parte2” de la figura.</p>
<pre> absiG=GetExpressionValue("AbscisaG"); </pre>	<p>Recupera y asigna a la variable absiG el valor de la expresión que calcula la abscisa del punto G, en nuestro sistema de coordenadas.</p>
<pre> ordeB=Math.round(Math.random()*8-4); </pre>	<p>Calculo de la nueva ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.</p>
<pre> absiA=Math.round(Math.random()*14-7); </pre>	<p>Calculo de la nueva abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.</p>
<pre> if ((Math.abs(absiA-absiG))&lt;0.0001) </pre>	<p>Condiciona que verifica si el valor absoluto de la diferencia entre las</p>

	<p>variables absiA y absiG del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las abscisas de los puntos A y G sean iguales y por tanto que el ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y GB tienen la misma pendiente estos puntos quedan superpuestos.</p>
<pre>{ absiA=(absiA+8)%14-7 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).</p>
<pre>if (absiA==0) { absiA=1 }</pre>	<p>Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.</p>
<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<p>Calculo de la nueva ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.</p>
<pre>if ((Math.abs(ordeA-ordeB))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeB del script, es menor que 0.0001. Esto, para evitar que la recta AB tenga como pendiente cero, pues de lo contrario la tarea quedaría resuelta ya que el estudiante solo modifica la diferencia de las abscisas de la recta GB; mientras que la diferencia de</p>

	las ordenadas de las rectas AB y GB son iguales.
<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	Si lo anterior sucede este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
<pre>SetExpressionValue("ORDENADAB", ordeB);</pre>	Asigna a la expresión "ORDENADAB" de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada "interna" de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así: $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
<pre>SetExpressionValue("ABSCISAA", absiA);</pre>	Asigna a la expresión "ABSCISAA" de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa "interna" de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula: $\text{ABSCISAA} * \text{FactorH} + x(P2)$
<pre>SetExpressionValue("ORDENADAA", ordeA); }</pre>	Asigna a la expresión "ORDENADAA" de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada "interna" de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula: $\text{ORDENADAA} * \text{FactorH} + y(P2)$

else	<i>Si esto no sucede entonces:</i>
{	
Alert ("La recta roja aún no está sobre la recta azul, vuelve a intentarlo.")	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
ctm=GetExpressionValue("ContadorMT3Parte2");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT3Parte2" en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón "verificar" la tarea está resuelta mal.
SetExpressionValue("ContadorMT3Parte2", ctm+1)	Aumenta en 1 la variable ctm y lo almacena en la expresión "ContadorMT3Parte2" de la figura.
}	
g=GetExpressionValue("correctos3Parte2");	Recupera y asigna a la variable g el valor que tiene la expresión "correctos3Parte2" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente Al finalizarse el condicional la variable ya no es reconocida por el script.
ctm=GetExpressionValue("ContadorMT3Parte2");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT3Parte2" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente Al finalizarse el condicional la variable ya no es reconocida por el script.
if (g>=5&&ctm<g)	Este condicional es el que permite que aparezca el botón continuar para ir a otra tarea. Es un condicional que verifica si: el valor de la variable g es mayor o igual a 5 y si el valor de la variable ctm es menor que el valor de la variable g; recordando con esto que el botón solo aparece cuando el estudiante resuelve 5 ejercicios correctamente y

	el número de éxitos en el desarrollo de la tarea es mayor al de fracasos.
{	<i>Si lo anterior sucede entonces:</i>
Alert ("Escribe en tu cuaderno la relación entre el valor del denominador de a en la ecuación y la gráfica de la recta, \n luego haz clic en continuar si quieres pasar a la siguiente tarea");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
continuar.setHidden (false); }	Coloca el botón “continuar” visible.
else {	<i>Si esto no sucede entonces:</i>
continuar.setHidden (true); } }	El botón “continuar” debe estar invisible.

<i>function VerificarTarea4() {</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ct=GetExpressionValue("Contador T4");	ct=GetExpressionValue("ContadorT4");	Recupera y asigna a la variable ct, el valor que tiene la expresión “ContadorT4” en la figura. Esta expresión almacena la cantidad de veces que el estudiante ha oprimido el botón “verificar”.
SetExpressionValue("ContadorT4”,	SetExpressionValue("ContadorT	Aumenta en 1 la variable ct y lo almacena en la expresión

ct+1);	4", ct+1);	“ContadorT4” de la figura.
if((Math.abs(((Y("A")-Y("B"))/(X("A")-X("B")))-((Y("B")-Y("H"))/(X("B")-X("H")))))<0.0001)	if((Math.abs(((Y("A")-Y("B"))/(X("A")-X("B")))-((Y("B")-Y("H"))/(X("B")-X("H")))))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre la pendiente de la recta AB, y la pendiente de la recta HB, es menor que 0. 0001. Esto para saber si la tarea ha sido resuelta o no por el alumno, ya que la tarea se considera resuelta cuando las pendientes de esas rectas son iguales. El software reconoce diferencias sumamente pequeñas entre las coordenadas, por lo que no se utiliza una igualdad sino una aproximación.
{	{	<i>Si lo anterior sucede entonces:</i>
Alert ("Muy bien, Ahora hazlo con esta nueva recta");	Alert ("Muy bien, Ahora hazlo con esta nueva ecuación");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
if(SetHide("OrdenadaH,TrianguloRectaAzul,TrianguloRectaRojaT4,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,Segmento1TrianguloRectaRojaT4, Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT4, MedidaSegmento2TrianguloRectaRojaT4",true))		Condicional que verifica si los objetos de la figura mencionados entre paréntesis y separados por comas, están visibles en la pantalla. Este condicional permite que la ayuda que usa el estudiante solo aparezca en el ejercicio que él así decida, y que en el nuevo ejercicio los objetos de la ayuda desaparezcan.
{		<i>Si lo anterior sucede entonces:</i>

<pre>SetHide("Ordenada H,TrianguloRectaAzul,TrianguloRectaRojaT4,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,Segmento1TrianguloRectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4",false) };</pre>		<p>Coloca a los objetos de la figura mencionados entre paréntesis y separados por comas, invisibles en la pantalla.</p>
<pre>g=GetExpressionValue("correctos4");</pre>	<pre>g=GetExpressionValue("correctos4");</pre>	<p>Recupera y asigna a la variable g el valor que tiene la expresión “correctos4” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” la tarea está resuelta bien.</p>
<pre>SetExpressionValue("correctos4", g+1);</pre>	<pre>SetExpressionValue("correctos4", g+1);</pre>	<p>Aumenta en 1 la variable g y lo almacena en la expresión “correctos4” de la figura.</p>
<pre>ordeH=GetExpressionValue("ORDENADAH");</pre>	<pre>ordeH=GetExpressionValue("ORDENADAH");</pre>	<p>Recupera y asigna a la variable ordeH el valor de la expresión que calcula la ordenada del punto H, en nuestro sistema de coordenadas.</p>
<pre>absiH=GetExpressionValue("ABSCISAH");</pre>	<pre>absiH=GetExpressionValue("ABSCISAH");</pre>	<p>Recupera y asigna a la variable absiH el valor de la expresión que calcula la abscisa del punto H, en nuestro sistema de coordenadas.</p>

<code>ordeB=Math.round(Math.random()*8-4);</code>	<code>ordeB=Math.round(Math.random()*8-4);</code>	Calculo de la nueva ordenada de B: Calcula un valor aleatorio ( <code>Math.random()</code> ), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo ( <code>Math.round</code> ) y asignarlo a la variable <code>ordeB</code> .
<code>absiA=Math.round(Math.random()*14-7);</code>	<code>absiA=Math.round(Math.random()*14-7);</code>	Calculo de la nueva abscisa de A: Calcula un valor aleatorio ( <code>Math.random()</code> ), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo ( <code>Math.round</code> ) y asignarlo a la variable <code>absiA</code> .
<code>if ((Math.abs(absiA-absiH))&lt;0.0001)</code>	<code>if ((Math.abs(absiA-absiH))&lt;0.0001)</code>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables <code>absiA</code> y <code>absiH</code> del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las abscisas de los puntos A y H sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y HB tienen la misma pendiente estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre <code>absiA</code> y <code>absiH</code> no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<code>{ absiA=(absiA+8)%14-7 }</code>	<code>{ absiA=(absiA+8)%14-7 }</code>	Si lo anterior sucede, este comando asigna a la variable <code>absiA</code> , el valor del módulo de <code>absiA+8</code> (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad

		total de números positivos o negativos en la escala del eje horizontal).
<pre>if (absiA==0) { absiA=1 }</pre>	<pre>if (absiA==0) { absiA=1 }</pre>	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<pre>ordeA=Math.round(Math.random()*8-4);</pre>	Calculo de la nueva ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.
<pre>if ((Math.abs(ordeA-ordeH))&lt;0.0001)</pre>	<pre>if ((Math.abs(ordeA-ordeH))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeH del script, es menor que 0.0001 o lo que es casi lo mismo que los valores de ordeA y ordeH sean iguales. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos A y H sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y HB tienen la misma pendiente estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre ordeA y ordeH no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ ordeA=(ordeA+5)%</pre>	<pre>{ ordeA=(ordeA+5)</pre>	Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al

<pre>8-4 }</pre>	<pre>%8-4 }</pre>	<p>adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>SetExpressionValue("ORD ENADAB",ordeB);</pre>	<pre>SetExpressionValue("OR DENADAB",ordeB);</pre>	<p>Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(\text{P2})$
<pre>SetExpressionValue("ABSC ISAA",absiA);</pre>	<pre>SetExpressionValue("AB SCISAA",absiA);</pre>	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA} * \text{FactorH} + x(\text{P2})$
<pre>SetExpressionValue("ORD ENADAA",ordeA); }</pre>	<pre>SetExpressionValue("OR DENADAA",ordeA);</pre>	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA} * \text{FactorH} + y(\text{P2})$
	<pre>ctm=GetExpressionValu e("ContadorMT4");</pre>	<p>Recupera y asigna a la variable ctm el valor que tiene la expresión “ContadorMT4” en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional</p>

		terminado anteriormente Al finalizarse el condicional la variable ya no es reconocida por el script.
	if (g>=5&&ctm<g)	Este condicional es el que permite que aparezca el botón continuar para ir a otra tarea. Es un condicional que verifica si: el valor de la variable g es mayor o igual a 5 y si el valor de la variable ctm es menor que el valor de la variable g; recordando con esto que el botón solo aparece cuando el estudiante resuelve 5 ejercicios correctamente y el número de éxitos en el desarrollo de la tarea es mayor al de fracasos.
	{	<i>Si lo anterior sucede entonces:</i>
	Alert ("Haz clic en continuar si quieres pasar a la siguiente tarea ");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
	continuar.setHidde n (false); }	Coloca el botón “continuar” visible.
	else {	<i>Si esto no sucede entonces:</i>
	continuar.setHidde n (true); }	El botón “continuar” debe estar invisible.
	entradainteractivaH ();	

	}	
else {	else {	<i>Si esto no sucede entonces:</i>
Alert ("La recta roja aún no está sobre la recta azul, vuelve a intentarlo... \n Consejo: revisa la relación que escribiste que existía, en la anterior tarea");	Alert ("La ecuación roja aún no es igual a la ecuación azul, vuelve a intentarlo.")	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
ctm=GetExpressionValue("ContadorMT4");	ctm=GetExpressionValue("ContadorMT4");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT4" en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón "verificar" la tarea está resuelta mal.
SetExpressionValue("ContadorMT4", ctm+1) }	SetExpressionValue("ContadorMT4", ctm+1)	Aumenta en 1 la variable ctm y lo almacena en la expresión "ContadorMT4" de la figura.
	entradainteractivaH (); } }	
g=GetExpressionValue("correctos4");		Recupera y asigna a la variable g el valor que tiene la expresión "correctos4" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente Al finalizarse el condicional la variable ya no es reconocida por el script.
ctm=GetExpressionValue("Contad		Recupera y asigna a la variable ctm el valor que tiene la expresión

orMT4");		“ContadorMT4” en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente Al finalizarse el condicional la variable ya no es reconocida por el script.
if (g>=5&&ctm<g)		Este condicional es el que permite que aparezca el botón continuar para ir a otra tarea. Es un condicional que verifica si: el valor de la variable g es mayor o igual a 5 y si el valor de la variable ctm es menor que el valor de la variable g; recordando con esto que el botón solo aparece cuando el estudiante resuelve 5 ejercicios correctamente y el número de éxitos en el desarrollo de la tarea es mayor al de fracasos.
{		<u>Si lo anterior sucede entonces:</u>
Alert ("Haz clic en continuar si quieres pasar a la siguiente tarea ");		Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
continuar.setHidden (false); }		Coloca el botón “continuar” visible.
else {		<u>Si esto no sucede entonces:</u>
continuar.setHidden (true); } }		El botón “continuar” debe estar invisible.

<i>function VerificarTarea5() {</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ct=GetExpressionValue("ContadorT5");	ct=GetExpressionValue("ContadorT5");	Recupera y asigna a la variable ct, el valor que tiene la expresión “ContadorT5” en la figura. Esta expresión almacena la cantidad de veces que el estudiante ha oprimido el botón “verificar”.
SetExpressionValue("ContadorT5", ct+1);	SetExpressionValue("ContadorT5", ct+1);	Aumenta en 1 la variable ct y lo almacena en la expresión “ContadorT5” de la figura.
if(((Math.abs(((Y("A")-Y("B"))/(X("A")-X("B")))-((Y("I")-Y("J"))/(X("I")-X("J")))))<0.0001)&&(Math.abs(Y("B")-Y("I"))<0.0001))	if(((Math.abs(((Y("A")-Y("B"))/(X("A")-X("B")))-((Y("I")-Y("J"))/(X("I")-X("J")))))<0.0001)&&(Math.abs(Y("B")-Y("I"))<0.0001)){	Condicional que verifica si el valor absoluto de la diferencia de las ordenadas de los puntos B e I es menor que 0.0001 y si el valor absoluto de la diferencia entre la pendiente de la recta AB, y la pendiente de la recta JI, es menor que 0.0001. Esto para saber si la tarea ha sido resuelta o no por el alumno, ya que la tarea se considera resuelta cuando las rectas AB y JI tienen la misma pendiente y el mismo corte con el eje de las ordenadas.
{	{	<u>Si lo anterior sucede entonces:</u>
Alert ("Muy bien, Ahora hazlo con esta nueva recta");	Alert ("Muy bien, Ahora hazlo con esta nueva ecuación");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
g=GetExpressionValue("correctos5");	g=GetExpressionValue("correctos5");	Recupera y asigna a la variable g el valor que tiene la expresión “correctos5” en la figura. Esta expresión almacena la cantidad de

		veces que al oprimir el botón “verificar” la tarea está resuelta bien.
SetExpressionValue("correctos5", g+1);	SetExpressionValue("correctos5", g+1);	Aumenta en 1 la variable g y lo almacena en la expresión “correctos5” de la figura.
ordeJ=GetExpressionValue("ORDENADAJ");	ordeJ=GetExpressionValue("ORDENADAJ");	Recupera y asigna a la variable ordeJ el valor de la expresión que calcula la ordenada del punto J, en nuestro sistema de coordenadas.
absiJ=GetExpressionValue("ABSCISAJ");	absiJ=GetExpressionValue("ABSCISAJ");	Recupera y asigna a la variable absiJ el valor de la expresión que calcula la abscisa del punto J, en nuestro sistema de coordenadas.
ordeI=GetExpressionValue("ORDENADAI");	ordeI=GetExpressionValue("ORDENADAI");	Recupera y asigna a la variable ordeI el valor de la expresión que calcula la ordenada del punto I, en nuestro sistema de coordenadas.
ordeB=Math.round(Math.random()*8-4);	ordeB=Math.round(Math.random()*8-4);	Calculo de la nueva ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.
if ((Math.abs(ordeI-ordeB))<0.0001)	if ((Math.abs(ordeI-ordeB))<0.0001)	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeI y ordeB del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos I y B sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante.</p> <p>La razón por la que la diferencia entre ordeI y ordeB no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>

<pre>{ ordeB=(ordeB+5) %8-4 }</pre>	<pre>{ ordeB=(ordeB+5) %8-4 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable ordeB, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>absiA=Math.round(Math.random()*14-7);</pre>	<pre>absiA=Math.round(Math.random()*14-7);</pre>	<p>Calculo de la nueva abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.</p>
<pre>if ((Math.abs(absiA-absiJ))&lt;0.0001)</pre>	<pre>if ((Math.abs(absiA-absiJ))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiJ del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las abscisas de los puntos A y J sean iguales y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y JI tienen la misma pendiente estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre absiA y absiJ no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ absiA=(absiA+8)% 14-7</pre>	<pre>{ absiA=(absiA+8)% 14-7</pre>	<p>Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la</p>

}	}	escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).
<pre>if (absiA==0) { absiA=1 }</pre>	<pre>if (absiA==0) { absiA=1 }</pre>	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<pre>ordeA=Math.round(Math.random()*8-4);</pre>	Calculo de la nueva ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.
<pre>if ((Math.abs(ordeA-ordeJ))&lt;0.0001)</pre>	<pre>if ((Math.abs(ordeA-ordeJ))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeJ del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos A y J sean iguales y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y JI tienen la misma pendiente, estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre ordeA y ordeJ no se busca exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
{	{	Si lo anterior sucede, este comando asigna a la variable ordeA, el

<pre>ordeA=(ordeA+5) %8-4 }</pre>	<pre>ordeA=(ordeA+5) %8-4 }</pre>	<p>valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>SetExpressionValue("ORDENADAB",ordeB);</pre>	<pre>SetExpressionValue("ORDENADAB",ordeB);</pre>	<p>Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(\text{P2})$
<pre>SetExpressionValue("ABSCISAA", absiA);</pre>	<pre>SetExpressionValue("ABSCISAA", absiA);</pre>	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA} * \text{FactorH} + x(\text{P2})$
<pre>SetExpressionValue("ORDENADAA", ordeA); }</pre>	<pre>SetExpressionValue("ORDENADAA", ordeA); }</pre>	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA} * \text{FactorH} + y(\text{P2})$
<pre>else {</pre>	<pre>else {</pre>	<p><u>Si esto no sucede entonces:</u></p>

Alert ("La recta roja aún no está sobre la recta azul, vuelve a intentarlo");	Alert ("La ecuación roja aún no es igual a la ecuación azul, vuelve a intentarlo.")	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
ctm=GetExpressionValue("ContadorMT5");	ctm=GetExpressionValue("ContadorMT5");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT5" en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón "verificar" la tarea está resuelta mal.
SetExpressionValue("ContadorMT5", ctm+1) }	SetExpressionValue("ContadorMT4", ctm+1)	Aumenta en 1 la variable ctm y lo almacena en la expresión "ContadorMT5" de la figura.
g=GetExpressionValue("correctos5");	g=GetExpressionValue("correctos5");	Recupera y asigna a la variable g el valor que tiene la expresión "correctos5" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente Al finalizarse el condicional la variable ya no es reconocida por el script.
ctm=GetExpressionValue("ContadorMT5");	ctm=GetExpressionValue("ContadorMT5");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT5" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente Al finalizarse el condicional la variable ya no es reconocida por el script.
if (g>=5&&ctm<g)	if (g>=5&&ctm<g)	Este condicional es el que permite que aparezca el botón continuar para ir a otra tarea. Es un condicional que verifica si: el valor de la variable g es mayor o igual a 5 y si el valor de la variable ctm es menor que el valor de la variable g; recordando con esto que el botón solo aparece cuando el estudiante resuelve 5 ejercicios correctamente y el número de éxitos en el desarrollo de la tarea es mayor al de fracasos.

{	{	<i>Si lo anterior sucede entonces:</i>
Alert ("Haz clic en continuar si quieres pasar a la siguiente tarea ");	Alert ("Haz clic en continuar si quieres pasar a la siguiente tarea ");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
continuar.setHidden (false); }	continuar.setHidden (false); }	Coloca el botón “continuar” visible.
else	else	<i>Si esto no sucede entonces:</i>
{	{	
continuar.setHidden (true); }	continuar.setHidden (true); }	El botón “continuar” debe estar invisible.
}	}	

<i>function VerificarTarea6() {</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ct=GetExpressionValue("ContadorT6");	ct=GetExpressionValue("ContadorT6");	Recupera y asigna a la variable ct, el valor que tiene la expresión “ContadorT6” en la figura. Esta expresión almacena la cantidad de veces que el estudiante ha oprimido el botón “verificar”.
SetExpressionValue("ContadorT6", ct+1);	SetExpressionValue("ContadorT6", ct+1);	Aumenta en 1 la variable ct y lo almacena en la expresión “ContadorT6” de la figura.

	$\text{if}((\text{Math.abs}(X("K"))-X("B"))<0.0001)$	Condicional que solo se ejecuta en el tomo 2 verificando si el valor absoluto de la diferencia de la abscisa de K y la abscisa de B es menor que 0.0001. El punto K debe estar sobre el eje y. La tarea da la posibilidad de colocar el punto K en cualquier parte del plano, pero solo está resuelta si las coordenadas de K y B coinciden. B es intersección de la recta azul con el eje y.
	{	<i>Si lo anterior sucede entonces:</i>
$\text{if}(((\text{Math.abs}(((Y("A")-Y("B"))/(X("A")-X("B")))-((Y("K")-Y("L"))/(X("K")-X("L")))))<0.0001)\&\&(\text{Math.abs}(Y("B")-Y("K"))<0.0001))$	$\text{if}(((\text{Math.abs}(((Y("A")-Y("B"))/(X("A")-X("B")))-((Y("K")-Y("L"))/(X("K")-X("L")))))<0.0001)\&\&(\text{Math.abs}(Y("B")-Y("K"))<0.0001))$	Condicional que verifica si el valor absoluto de la diferencia de las ordenadas de los puntos B e I es menor que 0.0001 y si el valor absoluto de la diferencia entre la pendiente de la recta AB, y la pendiente de la recta LK, es menor que 0.0001. Esto para saber si la tarea ha sido resuelta o no por el alumno, ya que la tarea se considera resuelta cuando las rectas AB y LK tienen la misma pendiente y el mismo corte con el eje de las ordenadas.
{	{	<i>Si lo anterior sucede entonces:</i>
Alert ("Muy bien, Ahora hazlo con esta nueva recta");	Alert ("Muy bien, Ahora hazlo con esta nueva ecuación");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
$g=\text{GetExpressionValue}("correctos6");$	$g=\text{GetExpressionValue}("correctos6");$	Recupera y asigna a la variable g el valor que tiene la expresión "correctos6" en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón "verificar" la tarea está resuelta bien.
$\text{SetExpressionValue}("correctos6", g+1);$	$\text{SetExpressionValue}("correctos6", g+1);$	Aumenta en 1 la variable g y lo almacena en la expresión "correctos6" de la figura.
$\text{ordeK}=\text{GetExpressionValu}$	$\text{ordeK}=\text{GetExpression}$	Recupera y asigna a la variable ordeK el valor de la expresión que

<code>e("ORDENADAK");</code>	<code>Value("ORDENADA K");</code>	calcula la ordenada del punto K, en nuestro sistema de coordenadas.
<code>absiL=GetExpressionValue("ABSCISAL");</code>	<code>absiL=GetExpressionValue("ABSCISAL");</code>	Recupera y asigna a la variable absiL el valor de la expresión que calcula la abscisa del punto L, en nuestro sistema de coordenadas.
<code>ordeL=GetExpressionValue("ORDENADAL");</code>	<code>ordeL=GetExpressionValue("ORDENADAL");</code>	Recupera y asigna a la variable ordeL el valor de la expresión que calcula la ordenada del punto L, en nuestro sistema de coordenadas.
<code>ordeB=Math.round(Math.random()*8-4);</code>	<code>ordeB=Math.round(Math.random()*8-4);</code>	Calculo de la nueva ordenada de B: Calcula un valor aleatorio ( <code>Math.random()</code> ), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo ( <code>Math.round</code> ) y asignarlo a la variable <code>ordeB</code> .
<code>if ((Math.abs(ordeK-ordeB))&lt;0.0001)</code>	<code>if ((Math.abs(ordeK-ordeB))&lt;0.0001)</code>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables <code>ordeK</code> y <code>ordeB</code> del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos K y B sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante.</p> <p>La razón por la que la diferencia entre <code>ordeK</code> y <code>ordeB</code> no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<code>{ ordeB=(ordeB+5)</code>	<code>{ ordeB=(ordeB</code>	Si lo anterior sucede, este comando asigna a la variable <code>ordeB</code> , el valor del módulo de <code>ordeA+5</code> (5 como el número obtenido al

<pre>%8-4 }</pre>	<pre>+5)%8-4 }</pre>	<p>adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>absiA=Math.round(Math.random()*14-7);</pre>	<pre>absiA=Math.round(Math.random()*14-7);</pre>	<p>Calculo de la nueva abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.</p>
<pre>if ((Math.abs(absiA-absiL))&lt;0.0001)</pre>	<pre>if ((Math.abs(absiA-absiL))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiL del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las abscisas de los puntos A y L sean iguales y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y LK tienen la misma pendiente estos puntos quedan superpuestos.</p>
<pre>{ absiA=(absiA+8)%14-7 }</pre>	<pre>{ absiA=(absiA+8)%14-7 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).</p>
<pre>if (absiA==0)</pre>	<pre>if (absiA==0)</pre>	<p>Condicional que verifica si absiA es igual a 0. Si esto sucede,</p>

<pre>{ absiA=1 }</pre>	<pre>{ absiA=1 }</pre>	<p>entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.</p>
<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<p>Calculo de la nueva ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.</p>
<pre>if ((Math.abs(ordeA-ordeL))&lt;0.0001)</pre>	<pre>if ((Math.abs(ordeA-ordeL))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeL del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos A y L sean iguales y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y LK tienen la misma pendiente, estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre ordeA y ordeL no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje</p>

		vertical).
SetExpressionValue("ORDENADAB", ordeB);	SetExpressionValue("ORDENADAB", ordeB);	Asigna a la expresión "ORDENADAB" de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada "interna" de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así: $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
SetExpressionValue("ABSCISAA", absiA);	SetExpressionValue("ABSCISAA", absiA);	Asigna a la expresión "ABSCISAA" de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa "interna" de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula: $\text{ABSCISAA} * \text{FactorH} + x(P2)$
SetExpressionValue("ORDENADAA", ordeA); }	SetExpressionValue("ORDENADAA", ordeA);	Asigna a la expresión "ORDENADAA" de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada "interna" de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula: $\text{ORDENADAA} * \text{FactorH} + y(P2)$
	ctm=GetExpressionValue("ContadorMT6");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT6" en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón "verificar" la tarea está resuelta mal.
	if (g>=5&&ctm<g)	Este verifica si: el valor de la variable g es mayor o igual a 5 y si

		el valor de la variable ctm es menor que el valor de la variable g; recordando con esto que el botón solo aparece cuando el estudiante resuelve 5 ejercicios correctamente y el número de éxitos en el desarrollo de la tarea es mayor al de fracasos.
	{	<i>Si lo anterior sucede entonces:</i>
	Alert ("¡Felicitaciones! Haz terminado todas las tareas"); }	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
	entradainteractivaK ();	El script busca la función con este nombre y la ejecuta.
	entradainteractivaL (); }	El script busca la función con este nombre y la ejecuta.
else	else	<i>Si esto no sucede entonces:</i>
{	{	
Alert ("La recta roja aún no está sobre la recta azul, vuelve a intentarlo.");	Alert ("La ecuación roja aún no es igual a la ecuación azul, vuelve a intentarlo.");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
ctm=GetExpressionValue("ContadorMT6");	ctm=GetExpressionValue("ContadorMT6");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT6" en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón "verificar" la tarea está resuelta mal.

SetExpressionValue("ContadorMT6", ctm+1) }	SetExpressionValue("ContadorMT6", ctm+1)	Aumenta en 1 la variable ctm y lo almacena en la expresión "ContadorMT6" de la figura.
	entradainteractivaK ();	El script busca la función con este nombre y la ejecuta.
	entradainteractivaL (); }	El script busca la función con este nombre y la ejecuta.
	else {	<i>Si esto no sucede entonces:</i>
	Alert ("K no está sobre el eje Y, vuelve a intentarlo.");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
	ctm=GetExpressionValue("ContadorMT6");	Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT6" en la figura. Se debe recuperar nuevamente el valor de esta expresión, ya que se ha cerrado el condicional en el tomo 2.
	SetExpressionValue("ContadorMT6", ctm+1)	Aumenta en 1 la variable ctm y lo almacena en la expresión "ContadorMT6" de la figura.
	entradainteractivaK ();	El script busca la función con este nombre y la ejecuta.
	entradainteractivaL (); }	El script busca la función con este nombre y la ejecuta.
g=GetExpressionValue("correctos		Recupera y asigna a la variable g el valor que tiene la expresión

6");		"correctos6" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente en el tomo 1. Al finalizarse el condicional la variable ya no es reconocida por el script.
ctm=GetExpressionValue("ContadorMT6");		Recupera y asigna a la variable ctm el valor que tiene la expresión "ContadorMT6" en la figura. Este valor se recupera nuevamente, pues la primera vez que lo hicimos fue dentro del condicional terminado anteriormente en el tomo 1. Al finalizarse el condicional la variable ya no es reconocida por el script.
if (g>=5&&g>ctm)		Este condicional es el que verifica si: el valor de la variable g es mayor o igual a 5 y si el valor de la variable ctm es menor que el valor de la variable g; recordando con esto que el botón solo aparece cuando el estudiante resuelve 5 ejercicios correctamente y el número de éxitos en el desarrollo de la tarea es mayor al de fracasos.
{		<u>Si lo anterior sucede entonces:</u>
Alert ("¡Felicitaciones! Has terminado todas las tareas"); }  }		Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.

**Sobre las funciones EntradaInteractiva ()**

Estas funciones son llamadas en algunas de las funciones VerificarTarea\_() del tomo 2. Solo hasta dicho llamamiento las funciones se ejecutarán. El uso de estas funciones simplifica la cantidad de líneas a escribir en el script, ya que almacena acciones que de otro modo tendrían que repetirse muchas veces en el script de verificar. Cuando se ejecute la función *entradainteractiva\_()* lo que esto hará en cada caso es:

<i>function entradainteractivaD () {</i>	
Orde=InteractiveInput ("Haga clic en el punto donde debe quedar D", "Point");	Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto. El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar D", "Point"); se mostrará en la barra de estado (parte inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
factorh=GetExpressionValue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión "FactorH" de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y (1,0) en nuestro sistema de coordenadas.
ordeD=(Y(Orde)-Y("P2"))/factorh;	Asigna a la variable ordeD el valor de la ordenada que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
absciD=(X(Orde)-X("P2"))/factorh;	Asigna a la variable absciD el valor de la abscisa que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
Move ("D",absciD+"*FactorH+x(P2)",ordeD+"*FactorH+y(P2)");	Mueve al punto D en la posición (absciD,ordeD) de nuestro sistema de coordenadas.
SetFixed("D",true); }	Fija al punto D (Impide el desplazamiento)

<i>function entradainteractivaH (){</i>	
Orde=InteractiveInput ("Haga clic en el punto donde debe quedar H", "Point");	Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto. El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar H", "Point"); se mostrará en la barra de estado (parte inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
factorh=GetExpressionValue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión "FactorH" de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y (1,0) en nuestro sistema de coordenadas.
ordeH=(Y(Orde)-Y("P2"))/factorh;	Asigna a la variable ordeH el valor de la ordenada que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
absciH=(X(Orde)-X("P2"))/factorh;	Asigna a la variable absciH el valor de la abscisa que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
Move ("H",absciH+"*FactorH+x(P2)",ordeH+"*FactorH+y(P2));	Mueve al punto H en la posición (absciH,ordeH) de nuestro sistema de coordenadas.
SetFixed("H",true); }	Fija al punto H (Impide el desplazamiento)

<i>function entradainteractivaK (){</i>	
Orde=InteractiveInput ("Haga clic en el punto donde debe quedar K", "Point");	Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto. El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar K", "Point"); se mostrará en la barra de estado (parte inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el

	usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
factorh=GetExpressionValue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión "FactorH" de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y (1,0) en nuestro sistema de coordenadas.
ordeK=(Y(Orde)-Y("P2"))/factorh;	Asigna a la variable ordeK el valor de la ordenada que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
absciK=(X(Orde)-X("P2"))/factorh;	Asigna a la variable absciK el valor de la abscisa que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
Move ("K",absciK+"*FactorH+x(P2)",ordeK+"*FactorH+y(P2));	Mueve al punto K en la posición (absciK,ordeK) de nuestro sistema de coordenadas.
SetFixed("K",true); }	Fija al punto K (Impide el desplazamiento)

<i>function entradainteractivaL (){</i>	
Orde=InteractiveInput ("Haga clic en el punto donde debe quedar L", "Point");	Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto. El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar L", "Point"); se mostrará en la barra de estado (parte inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
factorh=GetExpressionValue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión "FactorH" de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y (1,0) en nuestro sistema de coordenadas.
ordeL=(Y(Orde)-Y("P2"))/factorh;	Asigna a la variable ordeL el valor de la ordenada que tendrá el punto seleccionado por el

	usuario con el InteractiveInput en nuestro sistema de coordenadas.
Move ("L",absciL+"*FactorH+x(P2)",ordeL+"*FactorH+y(P2)");	Mueve al punto L en la posición (absciL,ordeL) de nuestro sistema de coordenadas.
SetFixed("L",true); }	Fija al punto L (Impide el desplazamiento)

La siguiente imagen (21) ejemplifica la descripción de las funciones tomando como caso la entrada interactiva D.

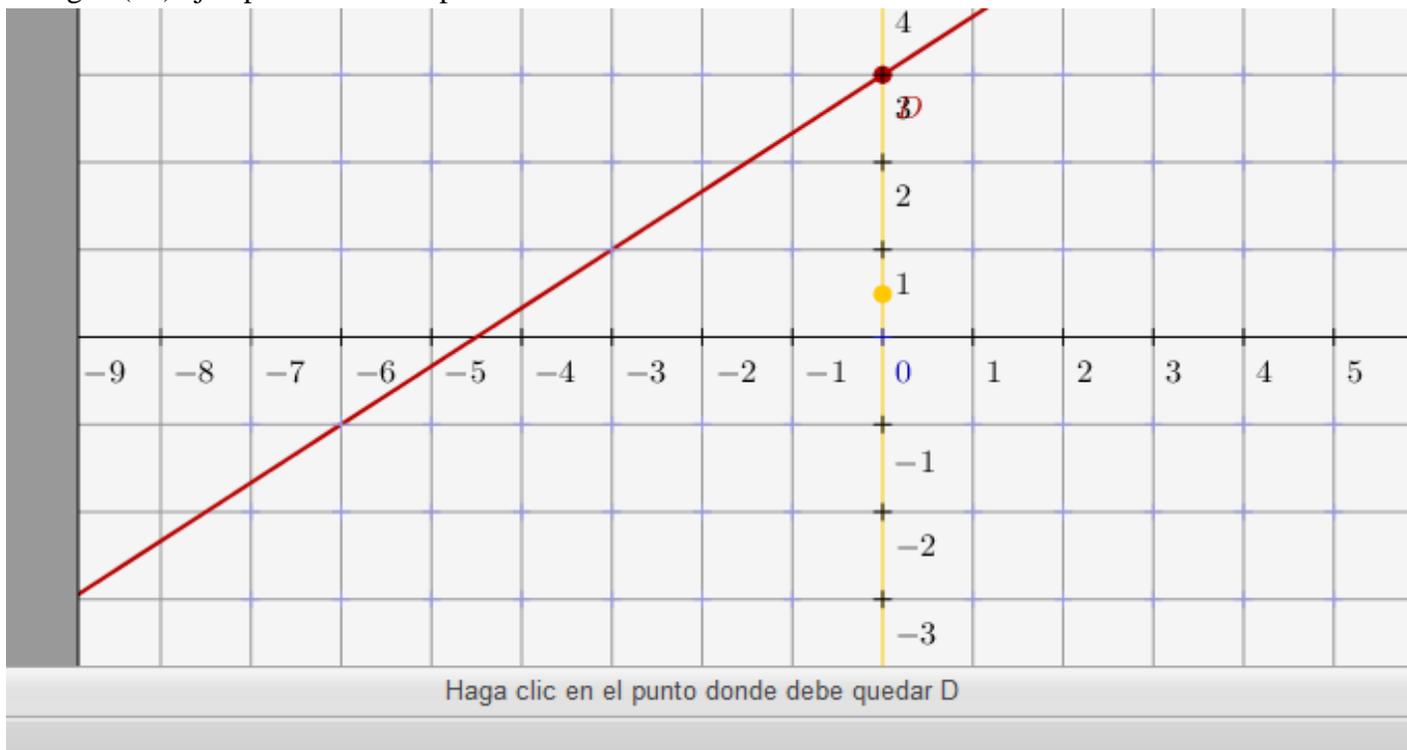


Imagen 21.

Nótese en la imagen (21), el mensaje que se verá en la barra de estado (parte inferior de la figura) cuando se ejecuta la función `entradainteractivaD`. El punto de color amarillo es donde está nuestro cursor del mouse, esperando que demos clic en algún lugar del sistema de coordenadas para mover hasta este sitio nuestro punto D.

### III. Continuar

Este script muestra los elementos necesarios para desarrollar cada tarea, y oculta los elementos innecesarios, que en su mayoría pertenecen a la tarea anterior. El script se ejecuta al hacer clic sobre el botón “continuar”. Como ya lo hemos venido diciendo el botón continuar permanece oculto en la figura hasta que el estudiante complete como mínimo 5 ejercicios de cada parte de la tarea y el número de éxitos en las mismas sea mayor al de fracasos cuando las resuelve.

<b><u>Tomo 1: Conversión algebraica a la gráfica</u></b>	<b><u>Tomo 2: Conversión gráfica a la algebraica</u></b>	<b>Descripción</b>
<code>t = getZC().JCM.CPs;</code>	<code>t = getZC().JCM.CPs;</code>	Este comando recupera en la variable t, todos los controles de la figura (botones, casillas, listas desplegadas, etc.). Los controles no son reconocidos como objetos de la figura en el script, razón por la que debe utilizarse este comando para posteriormente llamarlos y utilizarlos en él.
<pre> for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getName().equals("continuar")) { continuar=t.get(i); break; } } </pre>	<pre> for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getName().equals("continuar")) { continuar=t.get(i); break; } } </pre>	<p>En los dos tomos vemos que este bucle recorre la lista de los controles almacenados en la variable t, recuperando los botones “continuar” y “volver a”, para respectivamente ocultarlos y mostrarlos.</p> <p>En el tomo 1 el bucle también recupera el botón “ayuda” y las casillas “b2”, “CasillaDenominadorA” y “CasillaNumeradorA”. Esto para poder ocultarlas y mostrarlas cuando así se requiera en las tareas.</p>

```
for(i=0; i<t.size(); i++)
{
    if(t.get(i).O.getName().equals("volver"))
        {
            volvertarea=t.get(i);
            break;
        }
}
```

```
for(i=0; i<t.size(); i++)
{
    if(t.get(i).O.getName().equals("b2"))
        {
            CasillaCorteOrdenada=t.get(i);
            break;
        }
}
```

```
for(i=0; i<t.size(); i++)
{
    if(t.get(i).O.getName().equals("CasillaDenominadorA"))
        {
            CasillaDenominadorPendiente=t.get(i);
            break;
        }
}
```

<pre>     } } for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getName().equals("CasillaNumeradorA")) { CasillaNumeradorPendiente=t.get(i); break; } }  for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getName().equals("ayuda")) { ayuda=t.get(i); break; } } </pre>		
<pre>if(!continuar.isHidden())</pre>	<pre>if(!continuar.isHidden())</pre>	<p>Condicional que no permite la ejecución del script hasta que el botón “continuar” este visible en la pantalla.</p>
<pre>{</pre>	<pre>{</pre>	<p>Si lo anterior sucede, es decir si el botón continuar está visible, entonces:</p>
<pre>tarea=GetExpressionValue("Tarea");</pre>	<pre>tarea=GetExpressionValue("Tarea");</pre>	<p>Este comando recupera y asigna a la variable tarea el valor que tiene la expresión “Tarea” en la figura. Según sea el caso de la</p>

		<p>actividad de conversión, los valores recuperados serán un número entero entre el 1 y el 7 o entre el 1 y el 6. En cada actividad los valores de la expresión corresponden con la secuencia de actividades que fue descrita en el diseño de cada tomo.</p>
<pre>switch (tarea) { case (1) : ContinuarTarea2 (); break;  case (2) : ContinuarTarea3Parte1 (); break;  case (3) : ContinuarTarea3Parte2 (); break;  case (4) : ContinuarTarea4 (); break;  case (5) : ContinuarTarea5 (); break;</pre>	<pre>switch (tarea) { case (1) : ContinuarTarea2 (); break;  case (2) : ContinuarTarea3(); break;  case (3) : ContinuarTarea4 (); break;  case (4) : ContinuarTarea5 (); break;  case (5) :</pre>	<p>Aquí se ejecuta un condicional de casos posibles (switch case), con el que se evalúa el valor de la variable tarea. Cada caso (case()) tiene asociado una función que hemos denominado como ContinuarTarea_ (). En el momento en que un caso se ejecuta, el script busca el nombre de la función asociada y la ejecuta.</p>

<pre> case (6) : ContinuarTarea6 (); break; } </pre>	<pre> ContinuarTarea6 (); break; } </pre>	
--	---	--

**Sobre las funciones ContinuarTarea ()**

Como ya lo señalamos, estas funciones se ejecutan a partir del comando *switch case*. Estas funciones permiten avanzar en la secuencia de actividades mostrando los elementos diseñados en cada una y ocultando los que no son requeridos. A continuación, se propone mostrar una descripción detallada de cada función, así:

Para la lectura de las tablas se debe tener en cuenta las celdas sin llenar, que indicarán que la descripción del script solo se hace para uno de los tomos, donde está escrito.

<i>function ContinuarTarea2(){</i>		
<b><u>Tomo 1: Conversión algebraica a la gráfica.</u></b>	<b><u>Tomo 2: Conversión gráfica a la algebraica.</u></b>	<b>Descripción</b>
ayuda.setHidden(true)		Coloca oculto al botón “ayuda” de la figura.
continuar.setHidden (true);	continuar.setHidden (true);	Coloca oculto al botón “continuar” de la figura.
volvartarea.setHidden (false);	volvartarea.setHidden (false);	Coloca visible al botón “volver a...” de la figura.
CasillaCorteOrdenada.setHidden (false);		Coloca visible la casilla “b2” de la figura.

CasillaNumeradorPendiente.setHidden (true);		Coloca oculta la “CasillaNumeradorA” de la figura.
CasillaDenominadorPendiente.setHidden (true);		Coloca oculta la “CasillaDenominadorA” de la figura.
Show("D,RectaRojaT2,EcuacionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2");	Show("D,RectaRojaT2,EcuacionRojaBPositivoT2,EcuacionRojaBNegativoT2");	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
Hide("C,M,G,H,I,J,K,L,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT1,RectaRojaT3Parte1,RectaRojaT3Parte2,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT1,EcuacionRectaRojaBNegativoAFraccionT1,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPostivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNumeradorPendienteT3,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPostivoDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3,EcuacionRectaRojaYIgual,Ecuacion	Hide("C,F,G,H,I,J,K,L,RectaRojaT1,RectaRojaT3,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3,EcuacionRojaBPositivoT4,EcuacionRojaBNegativoT4,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5,EcuacionRojaBPositivoT6,EcuacionRojaBNegativoT6");	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.

<p>nRectaRojaAFraccionLínea,EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4,EcuacionRectaRojaXT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5,EcuacionRectaRojaXBPositivoT6,EcuacionRectaRojaXBNegativoT6,EcuacionRectaRojaBPostivoADecimalT6,EcuacionRectaRojaBNegativoADecimalT6,TrianguloRectaAzul,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,TrianguloRectaRojaT3Parte2,TrianguloRectaRojaT3Parte1,TrianguloRectaRojaT4,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Segmento3TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRe</p>		
---	--	--

<p>ctaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4");</p>		
<p>SetExpressionValue("Tarea",2);</p>	<p>SetExpressionValue("Tarea",2);</p>	<p>Cambia por dos el valor de la expresión “Tarea” que esta oculta en la figura. Esta expresión es la que permite saber, en cada tomo, en cuál tarea va el estudiante.</p>
<p>ordeD=GetExpressionValue("b2");</p>	<p>ordeD=GetExpressionValue("ORDENADAD");</p>	<p>Recupera y asigna a la variable ordeD el valor de la expresión que calcula la ordenada del punto D, en nuestro sistema de coordenadas. Para el tomo 1 esa expresión es: "b2" y para el tomo 2 es: "ORDENADAD".</p>
<p>ordeB=Math.round(Math.random()*8-4);</p>	<p>ordeB=Math.round(Math.random()*8-4);</p>	<p>Calculo de la ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.</p>
<p>if ((Math.abs(ordeD - ordeB))&lt;0.0001)</p>	<p>if ((Math.abs(ordeD - ordeB))&lt;0.0001)</p>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeD y ordeB del script, es menor que 0.0001, o lo que es casi lo mismo, que la ordeD sea igual a la ordeB. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las ordenadas de los puntos B y D sean iguales y por tanto que el nuevo ejercicio aparezca resuelto al estudiante. La razón por la que la diferencia entre ordeD y ordeB no se busque exacta, es porque éstos valores son un resultado de un</p>

		cálculo entre las coordenadas de los puntos que aparecen en la pantalla.
<pre>{ ordeB=(ordeB+5)%8-4 }</pre>	<pre>{ ordeB=(ordeB+5)%8-4 }</pre>	Si lo anterior sucede, este comando asigna a la variable OrdeB, el valor del módulo de OrdeB+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
<pre>absiA=Math.round(Math.random() *14-7);</pre>	<pre>absiA=Math.round(Math.random( )*14-7);</pre>	Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
<pre>if (absiA==0) { absiA=1 }</pre>	<pre>if (absiA==0) { absiA=1 }</pre>	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
<pre>ordeA=Math.round(Math.random() *8-4);</pre>	<pre>ordeA=Math.round(Math.random( )*8-4);</pre>	Calculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.

SetExpressionValue("ORDENADA B",ordeB);	SetExpressionValue("ORDENAD AB",ordeB);	<p>Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy}=\text{OrdenadaB}*\text{FactorH}+y(\text{P2})$
SetExpressionValue("ABSCISAA",absiA);	SetExpressionValue("ABSCISAA",absiA);	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA}*\text{FactorH}+x(\text{P2})$
SetExpressionValue("ORDENADA A",ordeA);	SetExpressionValue("ORDENAD AA",ordeA);	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA}*\text{FactorH}+y(\text{P2})$
Alert("La ecuación de la recta roja es de la forma $y=ax+b$ . \n Escriba en la casilla el valor correspondiente a b en la ecuación, \n para que la recta roja quede sobre la recta azul.");	Alert("La ecuación de la recta roja es de la forma $y=ax+b$ . \n El punto D es el punto de la recta roja que está sobre el eje Y. \n Señale la posición donde debe	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.

}	quedar D para que la ecuación roja sea igual a la ecuación azul");	
	Orde=InteractiveInput ("Haga clic en el punto donde debe quedar D", "Point");	Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto. El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar D", "Point"); se mostrará en la barra de estado (parte inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
	factorh=GetExpressionValue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión "FactorH" de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y (1,0) en nuestro sistema de coordenadas.
	ordeD=(Y(Orde)-Y("P2))/factorh;	Asigna a la variable ordeD el valor de la ordenada que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
	Move ("D",absciD+"*FactorH+x(P2)",ordeD+"*FactorH+y(P2));	Mueve al punto D en la posición (absiD,ordeD) de nuestro sistema de coordenadas.
	SetFixed("D",true); }	Fija al punto D (Impide el desplazamiento)

<i>function ContinuarTarea3(){</i>	
<b><u>Tomó 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
<code>continuar.setHidden (true);</code>	Coloca oculto al botón “continuar” de la figura.
<code>Show("G,RectaRojaT3,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3");</code>	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
<code>Hide("C,D,H,I,J,K,L,RectaRojaT1,RectaRojaT2,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1,EcuacionRojaBPositivoT2,EcuacionRojaBNegativoT2,EcuacionRojaBPositivoT4,EcuacionRojaBNegativoT4,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5,EcuacionRojaBPositivoT6,EcuacionRojaBNegativoT6");</code>	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
<code>SetExpressionValue("Tarea",3);</code>	Cambia por tres el valor de la expresión “Tarea” que esta oculta en la figura. Esta expresión es la que permite saber en cuál tarea va el estudiante.
<code>ordeG=GetExpressionValue("ORDENADAG");</code>	Recupera y asigna a la variable ordeG el valor de la expresión que calcula la ordenada del punto G, en nuestro sistema de coordenadas.
<code>absiG=GetExpressionValue("ABSCISAG");</code>	Recupera y asigna a la variable absiG el valor de la expresión que calcula la abscisa del punto G, en nuestro sistema de coordenadas.
<code>ordeB=Math.round(Math.random()*8-4);</code>	Calculo de la ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego

	redondearlo (Math.round) y asignarlo a la variable ordeB.
<code>absiA=Math.round(Math.random()*14-7);</code>	Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
<code>if ((Math.abs(absiA-absiG))&lt;0.0001)</code>	Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiG del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las abscisas de los puntos A y G sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y GB tienen la misma pendiente estos puntos quedan superpuestos.
<code>{ absiA=(absiA+8)%14-7 }</code>	Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).
<code>if (absiA==0) { absiA=1 }</code>	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
<code>ordeA=Math.round(Math.random()*8-4);</code>	Calculo de la ordenada de A: Calcula un valor aleatorio

	(Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.
if ((Math.abs(ordeA-ordeG))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeG del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las ordenadas de los puntos A y G sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y GB tienen la misma pendiente estos puntos quedan superpuestos.
{ ordeA=(ordeA+5)%8-4 }	Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
SetExpressionValue("ORDENADAB",ordeB);	Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así: $\text{PosicionBy}=\text{OrdenadaB}*\text{FactorH}+y(P2)$

<pre>SetExpressionValue("ABSCISAA",absiA);</pre>	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA} * \text{FactorH} + x(\text{P2})$
<pre>SetExpressionValue("ORDENADAA",ordeA);</pre>	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA} * \text{FactorH} + y(\text{P2})$
<pre>OrdenadaB=GetExpressionValue("ORDENADAB");</pre>	<p>Recupera y asigna a la variable OrdenadaB el valor de la expresión que almacena el valor de la ordenada del punto B en el sistema de coordenadas de la pantalla. Esto para poder tomar este valor e incluirlo en el mensaje que le aparecerá al estudiante al inicio de la tarea.</p>
<pre>Alert("La ecuación de la recta roja es de la forma y=ax+b. \n La recta roja pasa por el punto "+OrdenadaB+" del eje y, mueva el punto G por la cuadrícula, hasta hacer que la ecuación roja sea igual a la ecuación azul."); }</pre>	<p>Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.</p>

*function ContinuarTarea3Parte1(){*

<b><u>Tomo 2</u>: Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ayuda.setHidden(false)	Coloca visible al botón “ayuda” de la figura.
continuar.setHidden (true);	Coloca oculto al botón “continuar” de la figura.
volvartarea.setHidden (false);	Coloca visible al botón “volver a...” de la figura.
CasillaCorteOrdenada.setHidden (true);	Coloca oculta la casilla “b2” de la figura.
CasillaNumeradorPendiente.setHidden (true);	Coloca oculta la “CasillaNumeradorA” de la figura.
CasillaDenominadorPendiente.setHidden (true);	Coloca oculta la “CasillaDenominadorA” de la figura.
Show("M,RectaRojaT3Parte1,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPositivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNumeradorPendienteT3");	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
Hide("C,D,G,H,I,J,K,L,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT1,RectaRojaT2,RectaRojaT3Parte2,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPositivoDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3,	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.

<p>EcuacionRectaRojaYIgual,EcuacionRectaRojaAFraccionLínea,EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4,EcuacionRectaRojaXT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5,EcuacionRectaRojaXBPositivoT6,EcuacionRectaRojaXBNegativoT6,EcuacionRectaRojaBPostivoADecimalT6,EcuacionRectaRojaBNegativoADecimalT6,TrianguloRectaAzul,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,TrianguloRectaRojaT3Parte2,TrianguloRectaRojaT3Parte1,TrianguloRectaRojaT4,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Segmento3TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4");</p>	
<p>SetExpressionValue("Tarea",3);</p>	<p>Cambia por tres el valor de la expresión “Tarea” que esta oculta en la figura. Esta expresión es la que permite saber en cuál tarea va el estudiante.</p>
<p>ordeM=GetExpressionValue("ValorListaDesplegableNumeradorPendiente");</p>	<p>Recupera y asigna a la variable ordeM el valor de la expresión que calcula la ordenada del punto M, en nuestro sistema de coordenadas.</p>
<p>ordeB=Math.round(Math.random()*8-4);</p>	<p>Calculo de la ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego</p>

	redondearlo (Math.round) y asignarlo a la variable ordeB.
absiA=Math.round(Math.random()*14-7);	Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
if (absiA==0) { absiA=1 }	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
ordeA=Math.round(Math.random()*8-4);	Calculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.
if ((Math.abs(ordeA-ordeM))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeM del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las ordenadas de los puntos A y M sean iguales y por tanto que el ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y MB tienen la misma pendiente estos puntos quedan superpuestos.
{ ordeA=(ordeA+8)%14-7 }	Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la

	<p>escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).</p>
<p>SetExpressionValue("ORDENADAB",ordeB);</p>	<p>Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy}=\text{OrdenadaB}*\text{FactorH}+\text{y}(\text{P2})$
<p>SetExpressionValue("ABSCISAA",absiA);</p>	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA}*\text{FactorH}+\text{x}(\text{P2})$
<p>SetExpressionValue("ORDENADAA",ordeA);</p>	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA}*\text{FactorH}+\text{y}(\text{P2})$

<pre>OrdenadaB=GetExpressionValue("ORDENADAB");</pre>	<p>Recupera y asigna a la variable OrdenadaB el valor de la expresión que almacena el valor de la ordenada del punto B en el sistema de coordenadas de la pantalla. Esto para poder tomar este valor e incluirlo en el mensaje que le aparecerá al estudiante al inicio de la tarea.</p>
<pre>Alert("La ecuación de la recta roja es de la forma y=ax+b.\n La recta roja corta el eje y en "+OrdenadaB+", utilice la lista desplegable para modificar el numerador del término a en la ecuación, \n hasta lograr que la recta roja quede sobre la recta azul.");\n };</pre>	<p>Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.</p>

<i>function ContinuarTarea3Parte2(){</i>	
<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ayuda.setHidden(false)	Coloca visible al botón “ayuda” de la figura.
continuar.setHidden (true);	Coloca oculto al botón “continuar” de la figura.
voltarea.setHidden (false);	Coloca visible al botón “volver a...” de la figura.
CasillaCorteOrdenada.setHidden (true);	Coloca oculta la casilla “b2” de la figura.
CasillaNumeradorPendiente.setHidden (true);	Coloca oculta la “CasillaNumeradorA” de la figura.
CasillaDenominadorPendiente.setHidden (true);	Coloca oculta la “CasillaDenominadorA” de la figura.

<pre>Show("G,RectaRojaT3Parte2,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPositivoDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3");</pre>	<p>Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.</p>
<pre>Hide("C,D,M,H,I,J,K,L,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT1,RectaRojaT2,RectaRojaT3Parte1,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPositivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNumeradorPendienteT3,EcuacionRectaRojaYIgual,EcuacionRectaRojaAFraccionLínea,EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4,EcuacionRectaRojaXT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5,EcuacionRectaRojaXBPostivoT6,EcuacionRectaRojaXBNegativoT6,EcuacionRectaRojaBPostivoADecimalT6,EcuacionRectaRojaBNegativoADecimalT6,TrianguloRectaAzul,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,TrianguloRectaRojaT3Parte2,TrianguloRectaRojaT3Parte1,TrianguloRectaRojaT4,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Segmento3TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte2,MedidaSegmento2TrianguloRecta</pre>	<p>Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.</p>

RojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4");	
SetExpressionValue("Tarea",4);	Cambia por cuatro el valor de la expresión “Tarea” que esta oculta en la figura. Esta expresión es la que permite saber en cuál tarea va el estudiante.
absiG=GetExpressionValue("AbscisaG");	Recupera y asigna a la variable absiG el valor de la expresión que calcula la abscisa del punto G, en nuestro sistema de coordenadas.
ordeB=Math.round(Math.random()*8-4);	Calculo de la ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.
absiA=Math.round(Math.random()*14-7);	Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
if ((Math.abs(absiA-absiG))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiG del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las abscisas de los puntos A y G sean iguales y por tanto que el ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y GB tienen la misma pendiente estos puntos quedan superpuestos.
{	Si lo anterior sucede, este comando asigna a la variable absiA, el

<pre>absiA=(absiA+8)%14-7 }</pre>	<p>valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).</p>
<pre>if (absiA==0) { absiA=1 }</pre>	<p>Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.</p>
<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<p>Calculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.</p>
<pre>if ((Math.abs(ordeA-ordeB))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeB del script, es menor que 0.0001. Esto, para evitar que la recta AB tenga como pendiente cero, pues de lo contrario la tarea quedaría resuelta ya que el estudiante solo modifica la diferencia de las abscisas de la recta GB; mientras que la diferencia de las ordenadas de las rectas AB y GB son iguales.</p>
<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	<p>Si lo anterior sucede este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de</p>

	números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
SetExpressionValue("ORDENADAB",ordeB);	<p>Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy}=\text{OrdenadaB}*\text{FactorH}+\text{y}(\text{P2})$
SetExpressionValue("ABSCISAA",absiA);	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA}*\text{FactorH}+\text{x}(\text{P2})$
SetExpressionValue("ORDENADAA",ordeA);	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA}*\text{FactorH}+\text{y}(\text{P2})$
OrdenadaB=GetExpressionValue("ORDENADAB");	Recupera y asigna a la variable OrdenadaB el valor de la expresión que almacena el valor de la ordenada del punto B en el sistema de coordenadas de la pantalla. Esto para poder tomar este valor e incluirlo en el mensaje que le aparecerá al estudiante al inicio de la

	tarea.
Alert("La ecuación de la recta roja es de la forma $y=ax+b$ . La recta roja corta el eje y en "+OrdenadaB+", utilice la lista desplegable para modificar el denominador del término a en la ecuación, hasta lograr que la recta roja quede sobre la recta azul."); }	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.

<i>function ContinuarTarea4(){</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ayuda.setHidden(false)		Coloca visible al botón “ayuda” de la figura en el tomo 1.
continuar.setHidden (true);	continuar.setHidden (true);	Coloca oculto al botón “continuar” de la figura.
volvertarea.setHidden (false);		Coloca visible al botón “volver a...” de la figura.
CasillaCorteOrdenada.setHidden (true);		Coloca oculta la casilla “b2” de la figura.
CasillaNumeradorPendiente.setHidden (false);		Coloca visible la “CasillaNumeradorA” de la figura.
CasillaDenominadorPendiente.set		Coloca visible la “CasillaDenominadorA” de la figura.

Hidden (false);		
Show("H,RectaRojaT4,EcuacionRectaRojaYIgual,EcuacionRectaRojaAFraccionLínea,EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4");	Show("H,RectaRojaT4,EcuacionRojaBPositivoT4,EcuacionRojaBNegativoT4");	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
Hide("C,D,M,G,I,J,K,L,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT1,RectaRojaT2,RectaRojaT3Parte1,RectaRojaT3Parte2,RectaRojaT5,RectaRojaT6,EcuacionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPositivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNumeradorPendienteT3,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPositivoDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBP	Hide("C,D,G,I,J,K,L,RectaRojaT1,RectaRojaT2,RectaRojaT3,RectaRojaT5,RectaRojaT6,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1,EcuacionRojaBPositivoT2,EcuacionRojaBNegativoT2,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5,EcuacionRojaBPositivoT6,EcuacionRojaBNegativoT6");	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.

<p>ostivoT5,EcuacionRectaRojaBNe gativoT5,EcuacionRectaRojaXBP ositivoT6,EcuacionRectaRojaXB NegativoT6,EcuacionRectaRojaB PostivoADecimalT6,EcuacionRec taRojaBNegativoADecimalT6,Tri anguloRectaAzul,Segmento1Trian guloRectaAzul,Segmento2Triang uloRectaAzul,TrianguloRectaRoja T3Parte2,TrianguloRectaRojaT3P arte1,TrianguloRectaRojaT4,Seg mento1TrianguloRectaRojaT3Part e1,Segmento2TrianguloRectaRoja T3Parte1,Segmento3TrianguloRe ctaRojaT3Parte1,Segmento1Trian guloRectaRojaT3Parte2,Segmento 2TrianguloRectaRojaT3Parte2,Se gmento3TrianguloRectaRojaT3Pa rte2,Segmento1TrianguloRectaRo jaT4,Segmento2TrianguloRectaR ojaT4,Segmento3TrianguloRecta RojaT4,MedidaSegmento1Triang uloRectaRojaT3Parte1,MedidaSe gmento2TrianguloRectaRojaT3Pa rte1,MedidaSegmento1Triangulo RectaRojaT3Parte2,MedidaSegme nto2TrianguloRectaRojaT3Parte2, MedidaSegmento1TrianguloRecta RojaT4,MedidaSegmento2Triang</p>		
---	--	--

uloRectaRojaT4");		
SetExpressionValue("Tarea",5);	SetExpressionValue("Tarea",4);	Cambia por cuatro el valor de la expresión “Tarea” que esta oculta en la figura del tomo 2, y a cinco el valor de la expresión “Tarea” que esta oculta en la figura del tomo 1. Esta expresión es la que permite, en ambos tomos, saber en cuál tarea va el estudiante.
ordeH=GetExpressionValue("ORDENADAH");	ordeH=GetExpressionValue("ORDENADAH");	Recupera y asigna a la variable ordeH el valor de la expresión que calcula la ordenada del punto H, en nuestro sistema de coordenadas.
absiH=GetExpressionValue("ABSCISAH");	absiH=GetExpressionValue("ABSCISAH");	Recupera y asigna a la variable absiH el valor de la expresión que calcula la abscisa del punto H, en nuestro sistema de coordenadas.
ordeB=Math.round(Math.random()*8-4);	ordeB=Math.round(Math.random()*8-4);	Calculo de la ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.
absiA=Math.round(Math.random()*14-7);	absiA=Math.round(Math.random()*14-7);	Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
if ((Math.abs(absiA-absiH))<0.0001)	if ((Math.abs(absiA-absiH))<0.0001)	Condiciona que verifica si el valor absoluto de la diferencia entre las variables absiA y absiH del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las abscisas de los puntos A y H sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y HB tienen la misma

		<p>pendiente estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre absiA y absiH no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ absiA=(absiA+8)%14-7 }</pre>	<pre>{ absiA=(absiA+8)%14-7 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).</p>
<pre>if (absiA==0) { absiA=1 }</pre>	<pre>if (absiA==0) { absiA=1 }</pre>	<p>Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.</p>
<pre>ordeA=Math.round(Math.random( )*8-4);</pre>	<pre>ordeA=Math.round(Math.random( )*8-4);</pre>	<p>Calculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.</p>
<pre>if ((Math.abs(ordeA- ordeH))&lt;0.0001)</pre>	<pre>if ((Math.abs(ordeA- ordeH))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeH del script, es menor que 0.0001 o lo que es casi lo mismo que los valores de ordeA y ordeH sean iguales. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las ordenadas de los puntos A y H</p>

		<p>sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y HB tienen la misma pendiente estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre ordeA y ordeH no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>SetExpressionValue("ORDENADAB",ordeB);</pre>	<pre>SetExpressionValue("ORDENADAB",ordeB);</pre>	<p>Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, cálculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
<pre>SetExpressionValue("ABSCISAA",absiA);</pre>	<pre>SetExpressionValue("ABSCISAA",absiA);</pre>	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión</p>

		<p>PosicionAx calcula:</p> $ABSCISAA * FactorH + x(P2)$
<pre>SetExpressionValue("ORDENADAA",ordeA);</pre>	<pre>SetExpressionValue("ORDENADAA",ordeA);</pre>	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $ORDENADAA * FactorH + y(P2)$
<pre>OrdenadaB=GetExpressionValue("ORDENADAB");</pre>	<pre>OrdenadaB=GetExpressionValue("ORDENADAB");</pre>	<p>Recupera y asigna a la variable OrdenadaB el valor de la expresión que almacena el valor de la ordenada del punto B en el sistema de coordenadas de la pantalla. Esto para poder tomar este valor e incluirlo en el mensaje que le aparecerá al estudiante al inicio de la tarea.</p>
<pre>Alert("La ecuación de la recta roja es de la forma y=ax+b.\n La recta corta el eje y en "+OrdenadaB+", escriba en las casillas los valores del numerador y denominador de a \n para que la recta roja quede sobre la recta azul."); }</pre>	<pre>Alert("La ecuación de la recta roja es de la forma y=ax+b. \n La recta roja pasa por el punto "+OrdenadaB+" del eje y, señale la posición donde debe quedar el punto H para que la ecuación roja sea igual a la ecuación azul.");</pre>	<p>Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.</p>
	<pre>Orde=InteractiveInput ("Haga clic en el punto donde debe quedar H", "Point");</pre>	<p>Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto. El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar H", "Point"); se mostrará en la barra de estado (parte</p>

		inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
	factorh=GetExpressionValue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión “FactorH” de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y (1,0) en nuestro sistema de coordenadas.
	ordeH=(Y(Orde)-Y("P2"))/factorh;	Asigna a la variable ordeH el valor de la ordenada que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
	Move ("H",absiH+"*FactorH+x(P2)",ordeH+"*FactorH+y(P2));	Mueve al punto H en la posición (absiH,ordeH) de nuestro sistema de coordenadas.
	SetFixed("H",true); }	Fija al punto H (Impide el desplazamiento)

<i>function ContinuarTarea5(){</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ayuda.setHidden(true)		Coloca oculto al botón “ayuda” de la figura en el tomo 1.
continuar.setHidden (true);	continuar.setHidden (true);	Coloca oculto al botón “continuar” de la figura.
volvartarea.setHidden (false);		Coloca visible al botón “volver a...” de la figura.

CasillaCorteOrdenada.setHidden (true);		Coloca oculta la casilla “b2” de la figura.
CasillaNumeradorPendiente.setHidden (true);		Coloca oculta la “CasillaNumeradorA” de la figura.
CasillaDenominadorPendiente.setHidden (true);		Coloca oculta la “CasillaDenominadorA” de la figura.
Show("I,J,RectaRojaT5,EcuacionRectaRojaXT5,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5");	Show("I,J,RectaRojaT5,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5");	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
Hide("C,D,M,G,H,K,L,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT1,RectaRojaT2,RectaRojaT3Parte1,RectaRojaT3Parte2,RectaRojaT4,RectaRojaT6,EcuacionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPositivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNumeradorPendienteT3,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPositivoDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3");	Hide("C,D,G,H,K,L,RectaRojaT1,RectaRojaT2,RectaRojaT3,RectaRojaT4,RectaRojaT6,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1,EcuacionRojaBPositivoT2,EcuacionRojaBNegativoT2,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3,EcuacionRojaBPositivoT4,EcuacionRojaBNegativoT4,EcuacionRojaBPositivoT6,EcuacionRojaBNegativoT6");	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.

<p>onRectaRojaBNegativoNumeradorPendienteT3,EcuacionRectaRojaBPositivoDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3,EcuacionRectaRojaYIgual,EcuacionRectaRojaAFraccionLínea,EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4,EcuacionRectaRojaXBPositivoT6,EcuacionRectaRojaXBNegativoT6,EcuacionRectaRojaBPostivoADecimalT6,EcuacionRectaRojaBNegativoADecimalT6,TrianguloRectaAzul,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,TrianguloRectaRojaT3Parte2,TrianguloRectaRojaT3Parte1,TrianguloRectaRojaT4,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Segmento3TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegm</p>		
---	--	--

ento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4");		
SetExpressionValue("Tarea",6);	SetExpressionValue("Tarea",5);	Cambia por cinco el valor de la expresión “Tarea” que esta oculta en la figura del tomo 2, y a seis el valor de la expresión “Tarea” que esta oculta en la figura del tomo 1. Esta expresión es la que permite, en ambos tomos, saber en cuál tarea va el estudiante.
ordeJ=GetExpressionValue("ORDENADAJ");	ordeJ=GetExpressionValue("ORDENADAJ");	Recupera y asigna a la variable ordeJ el valor de la expresión que calcula la ordenada del punto J, en nuestro sistema de coordenadas.
absiJ=GetExpressionValue("ABSCISAJ");	absiJ=GetExpressionValue("ABSCISAJ");	Recupera y asigna a la variable absiJ el valor de la expresión que calcula la abscisa del punto J, en nuestro sistema de coordenadas.
ordeI=GetExpressionValue("ORDENADAI");	ordeI=GetExpressionValue("ORDENADAI");	Recupera y asigna a la variable ordeI el valor de la expresión que calcula la ordenada del punto I, en nuestro sistema de coordenadas.
ordeB=Math.round(Math.random()*8-4);	ordeB=Math.round(Math.random()*8-4);	Calculo de la ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.

<pre>if ((Math.abs(ordeI- ordeB))&lt;0.0001)</pre>	<pre>if ((Math.abs(ordeI- ordeB))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeI y ordeB del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las ordenadas de los puntos I y B sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante.</p> <p>La razón por la que la diferencia entre ordeI y ordeB no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ ordeB=(ordeB+5)%8-4 }</pre>	<pre>{ ordeB=(ordeB+5)%8-4 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable ordeB, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>absiA=Math.round(Math.random( )*14-7);</pre>	<pre>absiA=Math.round(Math.random( )*14-7);</pre>	<p>Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.</p>
<pre>if ((Math.abs(absiA- absiJ))&lt;0.0001)</pre>	<pre>if ((Math.abs(absiA- absiJ))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiJ del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las abscisas de los puntos A y J sean iguales</p>

		<p>y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y JI tienen la misma pendiente estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre absiA y absiJ no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ absiA=(absiA+8)%14-7 }</pre>	<pre>{ absiA=(absiA+8)%14-7 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).</p>
<pre>if (absiA==0) { absiA=1 }</pre>	<pre>if (absiA==0){absiA=1 }</pre>	<p>Condiciona que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.</p>
<pre>ordeA=Math.round(Math.random( )*8-4);</pre>	<pre>ordeA=Math.round(Math.random( )*8-4);</pre>	<p>Cálculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.</p>
<pre>if ((Math.abs(ordeA- ordeJ))&lt;0.0001)</pre>	<pre>if ((Math.abs(ordeA- ordeJ))&lt;0.0001)</pre>	<p>Condiciona que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeJ del script, es menor que 0.0001. Este</p>

		<p>condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las ordenadas de los puntos A y J sean iguales y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y JI tienen la misma pendiente, estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre ordeA y ordeJ no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>SetExpressionValue("ORDENAD AB",ordeB);</pre>	<pre>SetExpressionValue("ORDENAD AB",ordeB);</pre>	<p>Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
<pre>SetExpressionValue("ABSCISAA ",absiA);</pre>	<pre>SetExpressionValue("ABSCISAA ",absiA);</pre>	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de</p>

		A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:  $\text{ABSCISAA} * \text{FactorH} + x(P2)$
SetExpressionValue("ORDENADAA",ordeA);	SetExpressionValue("ORDENADAA",ordeA);	Asigna a la expresión "ORDENADAA" de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada "interna" de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:  $\text{ORDENADAA} * \text{FactorH} + y(P2)$
Alert("La ecuación de la recta roja es de la forma $y=ax+b$ . Utilice las listas desplegadas para modificar el valor de a y de b en la ecuación hasta lograr que la recta roja quede sobre la recta azul."); }	Alert ("La ecuación de la recta roja es de la forma $y=ax+b$ . Mueva el punto I sobre el eje y y el punto J sobre la cuadrícula, hasta hacer que la ecuación roja sea igual a la ecuación azul."); }	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.

<i>function ContinuarTarea6(){</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ayuda.setHidden(true)		Coloca oculto al botón "ayuda" de la figura en el tomo 1.

<code>continuar.setHidden (true);</code>	<code>continuar.setHidden (true);</code>	Coloca oculto al botón “continuar” de la figura.
<code>volvartarea.setHidden (false);</code>		Coloca visible al botón “volver a...” de la figura.
<code>CasillaCorteOrdenada.setHidden (false);</code>		Coloca visible la casilla “b2” de la figura.
<code>CasillaNumeradorPendiente.setHidden (false);</code>		Coloca visible la “CasillaNumeradorA” de la figura.
<code>CasillaDenominadorPendiente.setHidden (false);</code>		Coloca visible la “CasillaDenominadorA” de la figura.
<code>Show("K,L,RectaRojaT6,EcuacionRectaRojaYIgual,EcuacionRectaRojaAFraccionLinea,EcuacionRectaRojaXBPositivoT6,EcuacionRectaRojaXBNegativoT6,EcuacionRectaRojaBPostivoADecimalT6,EcuacionRectaRojaBNegativoADecimalT6");</code>	<code>Show("K,L,RectaRojaT6,EcuacionRojaBPositivoT6,EcuacionRojaBNegativoT6");</code>	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
<code>Hide("C,D,M,G,H,I,J,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT1,RectaRojaT2,RectaRojaT3Parte1,RectaRojaT3Parte2,RectaRojaT4,RectaRojaT5,EcuacionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT2,Ecuacion</code>	<code>Hide("C,D,G,H,I,J,RectaRojaT1,RectaRojaT2,RectaRojaT3,RectaRojaT4,RectaRojaT5,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1,EcuacionRojaBPositivoT2,EcuacionRojaBNegativoT2,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3,EcuacionRojaB</code>	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.

<p>nRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPositivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNumeradorPendienteT3,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPositivoDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3,EcuacionRectaRojaAFraccionYIgualT4,EcuacionRectaRojaAFraccionLíneaFracciónT4,EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4,EcuacionRectaRojaXT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5,TrianguloRectaAzul,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,TrianguloRectaRojaT3Parte2,TrianguloRectaRojaT3Parte1,TrianguloRectaRojaT4,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Seg</p>	<p>PositivoT4,EcuacionRojaBNegativoT4,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5");</p>	
---	--	--

<p>mento3TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Segmento3TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4");</p>		
<p>SetExpressionValue("Tarea",7);</p>	<p>SetExpressionValue("Tarea",6);</p>	<p>Cambia por seis el valor de la expresión “Tarea” que esta oculta en la figura del tomo 2, y por siete, el valor de la expresión “Tarea” que esta oculta en la figura del tomo 1. Esta expresión es la que permite, en ambos tomos, saber en cuál tarea va el estudiante.</p>
<p>ordeK=GetExpressionValue("ORDENADAK");</p>	<p>ordeK=GetExpressionValue("ORDENADAK");</p>	<p>Recupera y asigna a la variable ordeK el valor de la expresión que calcula la ordenada del punto K, en nuestro sistema de coordenadas.</p>
<p>absiL=GetExpressionValue("ABSCISAL");</p>	<p>absiL=GetExpressionValue("ABSCISAL");</p>	<p>Recupera y asigna a la variable absiL el valor de la expresión que calcula la abscisa del punto L, en nuestro sistema de coordenadas.</p>
<p>ordeL=GetExpressionValue("OR</p>	<p>ordeL=GetExpressionValue("OR</p>	<p>Recupera y asigna a la variable ordeL el valor de la expresión que</p>

DENADAL");	DENADAL");	calcula la ordenada del punto L, en nuestro sistema de coordenadas.
ordeB=Math.round(Math.random() ) * 8 - 4);	ordeB=Math.round(Math.random() ) * 8 - 4);	Calculo de la ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.
if ((Math.abs(ordeK - ordeB)) < 0.0001)	if ((Math.abs(ordeK - ordeB)) < 0.0001)	<p>Condiciona que verifica si el valor absoluto de la diferencia entre las variables ordeK y ordeB del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las ordenadas de los puntos K y B sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante.</p> <p>La razón por la que la diferencia entre ordeK y ordeB no se busca exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
{ ordeB=(ordeB+5)%8-4 }	{ ordeB=(ordeB+5)%8-4 }	Si lo anterior sucede, este comando asigna a la variable ordeB, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
absiA=Math.round(Math.random() ) * 14 - 7);	absiA=Math.round(Math.random() ) * 14 - 7);	Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de

		números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
if ((Math.abs(absiA-absiL))<0.0001)	if ((Math.abs(absiA-absiL))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiL del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las abscisas de los puntos A y L sean iguales y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y LK tienen la misma pendiente estos puntos quedan superpuestos.
{ absiA=(absiA+8)%14-7 }	{ absiA=(absiA+8)%14-7 }	Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).
if (absiA==0) { absiA=1 }	if (absiA==0) { absiA=1 }	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
ordeA=Math.round(Math.random()*8-4);	ordeA=Math.round(Math.random()*8-4);	Calculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.

<pre>if ((Math.abs(ordeA- ordeL))&lt;0.0001)</pre>	<pre>if ((Math.abs(ordeA- ordeL))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeL del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las ordenadas de los puntos A y L sean iguales y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y LK tienen la misma pendiente, estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre ordeA y ordeL no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>SetExpressionValue("ORDENAD AB",ordeB);</pre>	<pre>SetExpressionValue("ORDENAD AB",ordeB);</pre>	<p>Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, cálculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p>

		$PosicionBy=OrdenadaB*FactorH+y(P2)$
SetExpressionValue("ABSCISAA",absiA);	SetExpressionValue("ABSCISAA",absiA);	Asigna a la expresión "ABSCISAA" de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa "interna" de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:  $ABSCISAA*FactorH+x(P2)$
SetExpressionValue("ORDENADAA",ordeA);	SetExpressionValue("ORDENADAA",ordeA);	Asigna a la expresión "ORDENADAA" de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada "interna" de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:  $ORDENADAA*FactorH+y(P2)$
Alert("La ecuación de la recta roja es de la forma $y=ax+b$ . Escriba en las casillas los valores correspondientes a a y a b en la ecuación, para que la recta roja quede sobre la recta azul."); }	Alert ("La ecuación de la recta roja es de la forma $y=ax+b$ . Señale sucesivamente la posición donde deben quedar los puntos K y L, para que la ecuación roja sea igual a la ecuación azul. Tenga en cuenta que el punto K es el punto de la recta roja que está sobre el eje Y.");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.
	Orde=InteractiveInput ("Haga clic en el punto donde debe quedar K", "Point");	Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto. El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar K", "Point"); se mostrará en la barra de estado (parte

		inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
	factorh=GetExpressionValue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión "FactorH" de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y (1,0) en nuestro sistema de coordenadas.
	ordeK=(Y(Orde)-Y("P2"))/factorh;	Asigna a la variable ordeK el valor de la ordenada que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
	Move("K",absiK+"*FactorH+x(P2)",ordeK+"*FactorH+y(P2));	Mueve al punto K en la posición (absiK,ordeK) de nuestro sistema de coordenadas.
	SetFixed("K",true);	Fija al punto K (Impide el desplazamiento)
	Orde=InteractiveInput("Haga clic en el punto donde debe quedar L", "Point");	Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto. El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar L", "Point"); se mostrará en la barra de estado (parte inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
	factorh=GetExpressionValue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión "FactorH" de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y (1,0) en nuestro sistema de coordenadas.
	ordeL=(Y(Orde)-Y("P2"))/factorh;	Asigna a la variable ordeL el valor de la ordenada que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro

		sistema de coordenadas.
	Move ("L",absciL+"*FactorH+x(P2)",ordeL+"*FactorH+y(P2)");	Mueve al punto L en la posición (absiL,ordeL) de nuestro sistema de coordenadas.
	SetFixed("L",true); }	Fija al punto L (Impide el desplazamiento)

#### IV. Volver a la tarea

Este script se ejecuta con el botón “volver a...”, permitiendo regresar a las condiciones de una tarea. El script hace una recuperación de todas las todas las expresiones que, por tarea, almacenan la cantidad de ejercicios resueltos bien; esto con el fin de hacer que el estudiante solo regrese a las tareas donde ha resuleto por lo menos 5 ejercicios correctamente o lo que es lo mismo, que no avance saltándose la secuencia de actividades.

<b><u>Tomo 1: Conversión algebraica a la gráfica</u></b>	<b><u>Tomo 2: Conversión gráfica a la algebraica</u></b>	<b>Descripción</b>
t = getZC().JCM.CPs;	t = getZC().JCM.CPs;	Este comando recupera en la variable t, todos los controles de la figura (botones, casillas, listas desplegables, etc.). Los controles no son reconocidos como objetos de la figura en el script, razón por la que debe utilizarse este comando para posteriormente llamarlos y utilizarlos en él.
for(i=0; i<t.size(); i++) { if(t.get(i).O.getNa me().equals("b2")) {  CasillaCorteOrdena	for(i=0; i<t.size(); i++) { if(t.get(i).O.getName().equ als("continuar")) {	En los dos tomos vemos que este bucle recorre la lista de los controles almacenados en la variable t, recuperando los botones “continuar” y “volver a”, para respectivamente ocultarlos y mostrarlos.

<pre> da=t.get(i);         break;     } } for(i=0; i&lt;t.size(); i++) {     if(t.get(i).O.getNa me().equals("CasillaDeno minadorA")) {          CasillaDenominado rPendiente=t.get(i);             break;         }     } for(i=0; i&lt;t.size(); i++) {     if(t.get(i).O.getNa me().equals("CasillaNume radorA")) {          CasillaNumeradorP endiente=t.get(i);             break;         }     } for(i=0; i&lt;t.size(); i++) {     if(t.get(i).O.getNa me().equals("continuar")) { </pre>	<pre> continuar=t.get(i);         break;     } } for(i=0; i&lt;t.size(); i++) {     if(t.get(i).O.getName().equ als("volver"))         {             volvertarea1=t.get(i );             break;         } } </pre>	<p>En el tomo 1 el bucle también recupera el botón “ayuda” y las casillas “b2”, “CasillaDenominadorA” y “CasillaNumeradorA”. Esto para poder ocultarlas y mostrarlas cuando así se requiera en las tareas.</p>
---	--	--

<pre> continuar=t.get(i); break; } } for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getNa me().equals("volver")) {  volvartarea=t.get(i) ; break; } } for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getNa me().equals("ayuda")) {  ayuda=t.get(i); break; } } </pre>		
if(!volvartarea.isHidden())	if(!volvartarea1.isHidden())	Condicional que no permite la ejecución del script hasta que el botón “continuar” este visible en la pantalla.
{	{	Si lo anterior sucede, es decir si el botón continuar está visible, entonces:
tarea=Input("A cual tarea quiere volver");	tarea=Input("A cual tarea quiere volver");	Asigna a la variable tarea, el valor o texto que el comando <i>input</i> pedirá al usuario que ingrese. Cuando el estudiante ingresa el texto, el <i>input</i> traerá un diálogo modal al frente que será interrumpido

		hasta que el usuario pulse el botón OK.
Alert("la tarea que usted eligió fue "+tarea);	Alert("la tarea que usted eligió fue "+tarea);	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado que aparece en el paréntesis con la modificación de que mostrará el valor o texto que el usuario haya ingresado con el <i>input</i> anterior.
<pre> switch (+tarea){     case (1) :         VolverTarea1 ();         break;      case (2) :         VolverTarea2 ();         break;      case (3) :         VolverTarea3Parte1 ();         break;      case (4) :         VolverTarea3Parte2 ();         break;      case (5) :         VolverTarea4 ();         break; </pre>	<pre> switch (+tarea) {     case (1) :         VolverTarea1 ();         break;      case (2) :         VolverTarea2 ();         break;      case (3) :         VolverTarea3 ();         break;      case (4) :         VolverTarea4 ();         break;      case (5) :         VolverTarea5 ();         break; </pre>	Aquí se ejecuta un condicional de casos posibles ( <i>switch case</i> ), con el que se evalúa el valor de la variable tarea. Cada caso (case()) tiene asociado una función que hemos denominado como VolverTarea_ (). En el momento en que un caso se ejecuta, el script busca el nombre de la función asociada y la ejecuta.

<pre> case (6) :   VolverTarea5 ();   break;  case (7) :   VolverTarea6 ();   break;  } </pre>	<pre> case (6) :   VolverTarea6 ();   break; } </pre>	
--	---	--

**Sobre las funciones *VolverTarea ()***

Como ya lo señalamos, estas funciones se ejecutan a partir del comando *switch case*. Estas funciones permiten devolverse en la secuencia de actividades mostrando los elementos diseñados en cada una y ocultando los que no son requeridos. A continuación, se propone mostrar una descripción detallada de cada función, así:

Para la lectura de las tablas se debe tener en cuenta las celdas sin llenar, que indicarán que la descripción del script solo se hace para uno de los tomos, donde está escrito.

<i>function VolverTarea1(){</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
ayuda.setHidden(true);		Coloca oculto al botón “ayuda” de la figura.
continuar.setHidden (true);	continuar.setHidden (true);	Coloca oculto al botón “continuar” de la figura.
volvartarea.setHidden (true);		Coloca oculto al botón “volver a...” de la figura.

CasillaCorteOrdenada.setHidden (true);		Coloca oculta la casilla “b2” de la figura.
CasillaNumeradorPendiente.setHidden (true);		Coloca oculta la “CasillaNumeradorA” de la figura.
CasillaDenominadorPendiente.setHidden (true);		Coloca oculta la “CasillaDenominadorA” de la figura.
Show("C,RectaRojaT1,EcuacionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT1,EcuacionRectaRojaBNegativoAFraccionT1");	Show ("C,RectaRojaT1,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1");	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
Hide("D,M,G,H,I,J,K,L,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT2,RectaRojaT3Parte1,RectaRojaT3Parte2,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPositivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNumeradorPendienteT3,EcuacionRectaRojaBPostivoDenominadorPendiente	Hide("D,F,G,H,I,J,K,L,RectaRojaT2,RectaRojaT3,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRojaBPositivoT2,EcuacionRojaBNegativoT2,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3,EcuacionRojaBPositivoT4,EcuacionRojaBNegativoT4,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5,EcuacionRojaBPositivoT6,EcuacionRojaBNegativoT6");	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.

<p>T3,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3,EcuacionRectaRojaYIgual,EcuacionRectaRojaAFraccionLínea,EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4,EcuacionRectaRojaXT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5,EcuacionRectaRojaXBPositivoT6,EcuacionRectaRojaXBNegativoT6,EcuacionRectaRojaBPostivoADecimalT6,EcuacionRectaRojaBNegativoADecimalT6,TrianguloRectaAzul,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,TrianguloRectaRojaT3Parte2,TrianguloRectaRojaT3Parte1,TrianguloRectaRojaT4,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Segmento3TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaRojaT4,Seg</p>		
---	--	--

<p>mento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4,CursorPendiente,ParteEnteraCursorPendiente,ValorCursorNumeradorPendiente,ListaDesplegableNumeradorPendiente");</p>		
<p>SetExpressionValue("Tarea",1);</p>	<p>SetExpressionValue("Tarea",1);</p>	<p>Cambia por uno el valor de la expresión “Tarea” que esta oculta en la figura. Esta expresión es la que permite saber, en cada tomo, en cuál tarea va el estudiante.</p>
<p>ordeC=GetExpressionValue("ValorListaDesplegableCorteOrdenadaT3");</p>	<p>ordeC=GetExpressionValue("ORDENADAC");</p>	<p>Recupera y asigna a la variable ordeC el valor de la expresión que calcula la ordenada del punto C, en nuestro sistema de coordenadas. Para el tomo 1 esa expresión es: "ValorListaDesplegableCorteOrdendaT3" y para el tomo 2 es: "ORDENADAC".</p>
<p>ordeB=Math.round(Math.random()*8-4);</p>	<p>ordeB=Math.round(Math.random()*8-4);</p>	<p>Calculo de la nueva ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB</p>
<p>if ((Math.abs(ordeC-ordeB))&lt;0.0001)</p>	<p>if ((Math.abs(ordeC-ordeB))&lt;0.0001)</p>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeC y ordeB del script, es menor que 0.0001, o lo</p>

		<p>que es casi lo mismo, que la ordeC sea igual a la ordeB. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos B y C sean iguales y por tanto que el nuevo ejercicio aparezca resuelto al estudiante.</p> <p>La razón por la que la diferencia entre ordeC y ordeB no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{   ordeB=(ordeB+5)%8-4 }</pre>	<pre>{   ordeB=(ordeB+5)%8-4 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable OrdeB, el valor del módulo de OrdeB+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>absiA=Math.round(Math.random( )*14-7);</pre>	<pre>absiA=Math.round(Math.random( )*14-7);</pre>	<p>Cálculo de la nueva abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.</p>
<pre>if (absiA==0) {   absiA=1 }</pre>	<pre>if (absiA==0) {   absiA=1 }</pre>	<p>Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.</p>
<pre>ordeA=Math.round(Math.random( )*8-4);</pre>	<pre>ordeA=Math.round(Math.random( )*8-4);</pre>	<p>Cálculo de la nueva ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA</p>
<pre>SetExpressionValue("ORDENAD</pre>	<pre>SetExpressionValue("ORDENAD</pre>	<p>Asigna a la expresión "ORDENADAB" de la figura, el valor de la</p>

AB",ordeB);	AB",ordeB);	<p>variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
SetExpressionValue("ABSCISAA",absiA);	SetExpressionValue("ABSCISAA",absiA);	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA} * \text{FactorH} + x(P2)$
SetExpressionValue("ORDENADAA",ordeA);	SetExpressionValue("ORDENADAA",ordeA);	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA} * \text{FactorH} + y(P2)$
Alert("La ecuación de la recta roja es de la forma $y=ax+b$ .\n Despliegue la lista de números para modificar el valor de b en esa ecuación,\n hasta lograr que la recta roja quede sobre la recta azul.")\n}	Alert("La ecuación de la recta roja es de la forma $y=ax+b$ . \n Mueva el punto C sobre el eje y, hasta hacer que la ecuación roja sea igual a la ecuación azul.");\n}\n}	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.

<i>function VolverTarea2() {</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
correctos1=GetExpressionValue("correctos");	correctos1=GetExpressionValue("correctos");	Recupera y asigna a la variable correctos1, el valor que tiene la expresión “correctos” en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 1, ésta está resuelta bien.
if (correctos1>=5)	if (correctos1>=5)	Condicional que verifica si la variable correctos1 es menor o igual a 5. Esto para saber si el estudiante ya ha resuelto la tarea 1 antes de poder volver a la tarea 2.
{	{	<i>Si lo anterior sucede, entonces:</i>
ayuda.setHidden(true)		Coloca oculto al botón “ayuda” de la figura.
continuar.setHidden (true);	continuar.setHidden (true);	Coloca oculto al botón “continuar” de la figura.
volvartarea.setHidden (false);	volvartarea.setHidden (false);	Coloca visible al botón “volver a...” de la figura.
CasillaCorteOrdenada.setHidden (false);		Coloca visible la casilla “b2” de la figura.
CasillaNumeradorPendiente.setHidden (true);		Coloca oculta la “CasillaNumeradorA” de la figura.
CasillaDenominadorPendiente.setHidden (true);		Coloca oculta la “CasillaDenominadorA” de la figura.
Show("D,RectaRojaT2,Ecu	Show("D,RectaRojaT2,Ec	Este comando muestra los objetos que en la figura se han

<p>acionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2");</p>	<p>uacionRojaBPositivoT2,EcuacionRojaBNegativoT2");</p>	<p>denominado con los nombres que se muestran entre los paréntesis, separados con comas.</p>
<p>Hide("C,M,G,H,I,J,K,L,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT1,RectaRojaT3Parte1,RectaRojaT3Parte2,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT1,EcuacionRectaRojaBNegativoAFraccionT1,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPositivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNumeradorPendienteT3,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPositivoDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenomi</p>	<p>Hide("C,F,G,H,I,J,K,L,RectaRojaT1,RectaRojaT3,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3,EcuacionRojaBPositivoT4,EcuacionRojaBNegativoT4,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5,EcuacionRojaBPositivoT6,EcuacionRojaBNegativoT6");</p>	<p>Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.</p>

<p>nadorPendienteT3,Ecuacion RectaRojaYIgual,Ecuacion RectaRojaAFraccionLínea, EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4,EcuacionRectaRojaXT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5,EcuacionRectaRojaXBPositivoT6,EcuacionRectaRojaXBNegativoT6,EcuacionRectaRojaBPostivoADecimalT6,EcuacionRectaRojaBNegativoADecimalT6,TrianguloRectaAzul,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,TrianguloRectaRojaT3Parte2,TrianguloRectaRojaT3Parte1,TrianguloRectaRojaT4,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Seg</p>		
---	--	--

<p>mento3TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4");</p>		
<p>SetExpressionValue("Tarea",2);</p>	<p>SetExpressionValue("Tarea",2);</p>	<p>Cambia por dos el valor de la expresión “Tarea” que esta oculta en la figura. Esta expresión es la que permite saber, en cada tomo, en cuál tarea va el estudiante.</p>
<p>ordeD=GetExpressionValue("b2");</p>	<p>ordeD=GetExpressionValue("ORDENADAD");</p>	<p>Recupera y asigna a la variable ordeD el valor de la expresión que calcula la ordenada del punto D, en nuestro sistema de coordenadas. Para el tomo 1 esa expresión es: "b2" y para el tomo 2 es: "ORDENADAD".</p>
<p>ordeB=Math.round(Math.random()*8-4);</p>	<p>ordeB=Math.round(Math.random()*8-4);</p>	<p>Calculo de la nueva ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números</p>

		positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.
if ((Math.abs(ordeD - ordeB))<0.0001)	if ((Math.abs(ordeD - ordeB))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeD y ordeB del script, es menor que 0.0001, o lo que es casi lo mismo, que la ordeD sea igual a la ordeB. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos B y D sean iguales y por tanto que el nuevo ejercicio aparezca resuelto al estudiante. La razón por la que la diferencia entre ordeD y ordeB no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.
{ ordeB=(ordeB+5)%8 -4 }	{ ordeB=(ordeB+5) %8-4 }	Si lo anterior sucede, este comando asigna a la variable OrdeB, el valor del módulo de OrdeB+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
absiA=Math.round(Math.random()*14-7);	absiA=Math.round(Math.random()*14-7);	Calculo de la nueva abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
if (absiA==0) {	if (absiA==0) {	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional

absiA=1 }	absiA=1 }	es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
ordeA=Math.round(Math.random()*8-4);	ordeA=Math.round(Math.random()*8-4);	Calculo de la nueva ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.
SetExpressionValue("ORDENADAB",ordeB);	SetExpressionValue("ORDENADAB",ordeB);	Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:  $\text{PosicionBy}=\text{OrdenadaB}*\text{FactorH}+\text{y}(\text{P2})$
SetExpressionValue("ABSCISAA",absiA);	SetExpressionValue("ABSCISAA",absiA);	Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:  $\text{ABSCISAA}*\text{FactorH}+\text{x}(\text{P2})$
SetExpressionValue("ORDENADAA",ordeA);	SetExpressionValue("ORDENADAA",ordeA);	Asigna a la expresión “ORDENADAA” de la figura, el valor de la

ENADAA",ordeA);	DENADAA",ordeA);	variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:  ORDENADAA*FactorH+y(P2)
Alert("La ecuación de la recta roja es de la forma $y=ax+b$ . \n Escriba en la casilla el valor correspondiente a b en la ecuación, \n para que la recta roja quede sobre la recta azul."); }	Alert("La ecuación de la recta roja es de la forma $y=ax+b$ . \n El punto D es el punto de la recta roja que está sobre el eje Y. \n Señale la posición donde debe quedar D para que la ecuación roja sea igual a la ecuación azul");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.
	Orde=InteractiveInput ("Haga clic en el punto donde debe quedar D", "Point");	Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto. El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar D", "Point"); se mostrará en la barra de estado (parte inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
	factorh=GetExpressionValue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión “FactorH” de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y (1,0) en nuestro sistema de coordenadas.
	ordeD=(Y(Orde)-	Asigna a la variable ordeD el valor de la ordenada que tendrá el

	Y("P2"))/factorh;	punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
	Move ("D",absciD+"*FactorH+x (P2)",ordeD+"*FactorH+y (P2)");	Mueve al punto D en la posición (absiD,ordeD) de nuestro sistema de coordenadas.
	SetFixed("D",true); }	Fija al punto D (Impide el desplazamiento)
else {	else {	<i>Si lo anterior no sucede, entonces:</i>
Alert("Error: No has completado la tarea 1, debes realizar como mínimo 5 ejercicios para continuar"); } }	Alert("Error: No has completado la tarea 1, debes realizar como mínimo 5 ejercicios para continuar"); } }	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla para que de está forma se le indique al estudiante su no posible paso por tareas cuando no ha realizado las anteriores.

<i>function VolverTarea3(){</i>	
<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
correctos1=GetExpressionValue("correctos");	Recupera y asigna a la variable correctos1, el valor que tiene la expresión “correctos” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 1, ésta está resuelta bien.
correctos2=GetExpressionValue("correctos2");	Recupera y asigna a la variable correctos2, el valor que tiene la

	expresión “correctos2” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 2, ésta está resuelta bien.
if (correctos1>=5&&correctos2>=5)	Condicional que verifica si las variables correctos1 y correctos 2 son menores o iguales a 5. Esto para saber si el estudiante ya ha resuelto la tarea 1 y 2 antes de poder volver a la tarea 3.
{	<i>Si lo anterior sucede, entonces:</i>
continuar.setHidden (true);	Coloca oculto al botón “continuar” de la figura.
Show("G,RectaRojaT3,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3");	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
Hide("C,D,H,I,J,K,L,RectaRojaT1,RectaRojaT2,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1,EcuacionRojaBPositivoT2,EcuacionRojaBNegativoT2,EcuacionRojaBPositivoT4,EcuacionRojaBNegativoT4,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5,EcuacionRojaBPositivoT6,EcuacionRojaBNegativoT6");	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
SetExpressionValue("Tarea",3);	Cambia por tres el valor de la expresión “Tarea” que esta oculta en la figura. Esta expresión es la que permite saber en cuál tarea va el estudiante.
ordeG=GetExpressionValue("ORDENADAG");	Recupera y asigna a la variable ordeG el valor de la expresión que calcula la ordenada del punto G, en nuestro sistema de coordenadas.
absiG=GetExpressionValue("ABSCISAG");	Recupera y asigna a la variable absiG el valor de la expresión que calcula la abscisa del punto G, en nuestro sistema de coordenadas.
ordeB=Math.round(Math.random()*8-4);	Calculo de la ordenada de B: Calcula un valor aleatorio

	(Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.
absiA=Math.round(Math.random()*14-7);	Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
if ((Math.abs(absiA-absiG))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiG del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las abscisas de los puntos A y G sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y GB tienen la misma pendiente estos puntos quedan superpuestos.
{ absiA=(absiA+8)%14-7 }	Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).
if (absiA==0) { absiA=1	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes

}	indeterminadas.
ordeA=Math.round(Math.random()*8-4);	Calculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.
if ((Math.abs(ordeA-ordeG))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeG del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos A y G sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y GB tienen la misma pendiente estos puntos quedan superpuestos.
<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
SetExpressionValue("ORDENADAB",ordeB);	Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de

	<p>B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
<pre>SetExpressionValue("ABSCISAA",absiA);</pre>	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA} * \text{FactorH} + x(P2)$
<pre>SetExpressionValue("ORDENADAA",ordeA);</pre>	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA} * \text{FactorH} + y(P2)$
<pre>OrdenadaB=GetExpressionValue("ORDENADAB");</pre>	<p>Recupera y asigna a la variable OrdenadaB el valor de la expresión que almacena el valor de la ordenada del punto B en el sistema de coordenadas de la pantalla. Esto para poder tomar este valor e incluirlo en el mensaje que le aparecerá al estudiante al inicio de la tarea.</p>
<pre>Alert("La ecuación de la recta roja es de la forma y=ax+b. \n La recta roja pasa por el punto "+OrdenadaB+" del eje y, mueva el punto G por la cuadrícula, hasta hacer que la ecuación roja sea igual a la ecuación azul."); }</pre>	<p>Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.</p>
<pre>else</pre>	<p><i>Si lo anterior no sucede, entonces:</i></p>

{	
Alert("Error: No has completado todos los ejercicios propuestos en la tarea 1 y/o 2, debes realizar como mínimo 5 ejercicios por cada una de ellas para continuar");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla para que de está forma se le indique al estudiante su no posible paso por tareas cuando no ha realizado las anteriores.
}	

<i>function VolverTarea3Parte1(){</i>	
<b><u>Tomo 2: Conversión gráfica a la algebraica.</u></b>	<b>Descripción</b>
correctos1=GetExpressionValue("correctos");	Recupera y asigna a la variable correctos1, el valor que tiene la expresión “correctos” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 1, ésta está resuelta bien.
correctos2=GetExpressionValue("correctos2");	Recupera y asigna a la variable correctos2, el valor que tiene la expresión “correctos2” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 2, ésta está resuelta bien.
if (correctos1>=5&&correctos2>=5)	Condicional que verifica si las variables correctos1 y correctos 2 son menores o iguales a 5. Esto para saber si el estudiante ya ha resuelto la tarea 1 y 2 antes de poder volver a la tarea 3.
{	<i>Si lo anterior sucede, entonces:</i>
ayuda.setHidden(false)	Coloca visible al botón “ayuda” de la figura.

<code>continuar.setHidden (true);</code>	Coloca oculto al botón “continuar” de la figura.
<code>volvertarea.setHidden (false);</code>	Coloca visible al botón “volver a...” de la figura.
<code>CasillaCorteOrdenada.setHidden (true);</code>	Coloca oculta la casilla “b2” de la figura.
<code>CasillaNumeradorPendiente.setHidden (true);</code>	Coloca oculta la “CasillaNumeradorA” de la figura.
<code>CasillaDenominadorPendiente.setHidden (true);</code>	Coloca oculta la “CasillaDenominadorA” de la figura.
<code>Show("M,RectaRojaT3Parte1,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPositivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNumeradorPendienteT3");</code>	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
<code>Hide("C,D,G,H,I,J,K,L,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT1,RectaRojaT2,RectaRojaT3Parte2,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPositivoDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3,EcuacionRectaRojaYIgual,EcuacionRectaRojaAFraccionLínea,EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4,EcuacionRectaRojaXT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5,EcuacionRectaRojaXBPositivoT6,EcuacionRectaRoja</code>	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.

<p>XBNegativoT6,EcuacionRectaRojaBPostivoADecimalT6,EcuacionRectaRojaBNegativoADecimalT6,TrianguloRectaAzul,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,TrianguloRectaRojaT3Parte2,TrianguloRectaRojaT3Parte1,TrianguloRectaRojaT4,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Segmento3TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4");</p>	
<p>SetExpressionValue("Tarea",3);</p>	<p>Cambia por tres el valor de la expresión “Tarea” que esta oculta en la figura. Esta expresión es la que permite saber en cuál tarea va el estudiante.</p>
<p>ordeM=GetExpressionValue("ValorListaDesplegableNumeradorPendiente");</p>	<p>Recupera y asigna a la variable ordeM el valor de la expresión que calcula la ordenada del punto M, en nuestro sistema de coordenadas.</p>
<p>ordeB=Math.round(Math.random()*8-4);</p>	<p>Calculo de la ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.</p>
<p>absiA=Math.round(Math.random()*14-7);</p>	<p>Calculo de la abscisa de A: Calcula un valor aleatorio</p>

	(Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
<pre>if (absiA==0) { absiA=1 }</pre>	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
<pre>ordeA=Math.round(Math.random()*8-4);</pre>	Calculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.
<pre>if ((Math.abs(ordeA-ordeM))&lt;0.0001)</pre>	Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeM del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos A y M sean iguales y por tanto que el ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y MB tienen la misma pendiente estos puntos quedan superpuestos.
<pre>{ ordeA=(ordeA+8)%14-7 }</pre>	Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje

	horizontal).
SetExpressionValue("ORDENADAB",ordeB);	<p>Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy}=\text{OrdenadaB}*\text{FactorH}+\text{y}(\text{P2})$
SetExpressionValue("ABSCISAA",absiA);	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA}*\text{FactorH}+\text{x}(\text{P2})$
SetExpressionValue("ORDENADAA",ordeA);	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA}*\text{FactorH}+\text{y}(\text{P2})$
OrdenadaB=GetExpressionValue("ORDENADAB");	<p>Recupera y asigna a la variable OrdenadaB el valor de la expresión que almacena el valor de la ordenada del punto B en el sistema de coordenadas de la pantalla. Esto para poder tomar este valor e</p>

	incluirlo en el mensaje que le aparecerá al estudiante al inicio de la tarea.
<pre>Alert("La ecuación de la recta roja es de la forma <math>y=ax+b</math>.\n La recta roja corta el eje y en "+OrdenadaB+", utilice la lista desplegable para modificar el numerador del término a en la ecuación, \n hasta lograr que la recta roja quede sobre la recta azul."); }</pre>	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.
<pre>else {</pre>	<i>Si lo anterior no sucede, entonces:</i>
<pre>Alert("Error: No has completado todos los ejercicios propuestos en la tarea 1 y/o 2, debes realizar como mínimo 5 ejercicios por cada una de ellas para continuar"); } }</pre>	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla para que de está forma se le indique al estudiante su no posible paso por tareas cuando no ha realizado las anteriores.

<i>function VolverTarea3Parte2(){</i>	
<b><u>Tomo 2: Conversión gráfica a la algebraica.</u></b>	<b>Descripción</b>
<pre>correctos1=GetExpressionValue("correctos");</pre>	Recupera y asigna a la variable correctos1, el valor que tiene la expresión “correctos” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 1, ésta está resuelta bien.

correctos2=GetExpressionValue("correctos2");	Recupera y asigna a la variable correctos2, el valor que tiene la expresión “correctos2” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 2, ésta está resuelta bien.
correctos3Parte1=GetExpressionValue("correctos3Parte1");	Recupera y asigna a la variable correctos3Parte1, el valor que tiene la expresión “correctos3Parte1” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 3, ésta está resuelta bien.
if (correctos1>=5&&correctos2>=5&&correctos3Parte1>=5)	Condicional que verifica si las variables correctos1, correctos 2 y correctos3Parte1 son menores o iguales a 5. Esto para saber si el estudiante ya ha resuelto la tarea 1, 2 y 3 antes de poder volver a la tarea 4.
{	<i>Si lo anterior sucede, entonces:</i>
ayuda.setHidden(false)	Coloca visible al botón “ayuda” de la figura.
continuar.setHidden (true);	Coloca oculto al botón “continuar” de la figura.
volvertarea.setHidden (false);	Coloca visible al botón “volver a...” de la figura.
CasillaCorteOrdenada.setHidden (true);	Coloca oculta la casilla “b2” de la figura.
CasillaNumeradorPendiente.setHidden (true);	Coloca oculta la “CasillaNumeradorA” de la figura.
CasillaDenominadorPendiente.setHidden (true);	Coloca oculta la “CasillaDenominadorA” de la figura.
Show("G,RectaRojaT3Parte2,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPositivoDenomina	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis,



<p>guloRectaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4");</p>	
<p>SetExpressionValue("Tarea",4);</p>	<p>Cambia por cuatro el valor de la expresión “Tarea” que esta oculta en la figura. Esta expresión es la que permite saber en cuál tarea va el estudiante.</p>
<p>absiG=GetExpressionValue("AbscisaG");</p>	<p>Recupera y asigna a la variable absiG el valor de la expresión que calcula la abscisa del punto G, en nuestro sistema de coordenadas.</p>
<p>ordeB=Math.round(Math.random()*8-4);</p>	<p>Calculo de la ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.</p>
<p>absiA=Math.round(Math.random()*14-7);</p>	<p>Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.</p>
<p>if ((Math.abs(absiA-absiG))&lt;0.0001)</p>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiG del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las abscisas de los puntos A y G sean iguales y por tanto que el ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y GB tienen la misma pendiente estos puntos quedan superpuestos.</p>

<pre>{ absiA=(absiA+8)%14-7 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).</p>
<pre>if (absiA==0) { absiA=1 }</pre>	<p>Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.</p>
<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<p>Calculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.</p>
<pre>if ((Math.abs(ordeA-ordeB))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeB del script, es menor que 0.0001. Esto, para evitar que la recta AB tenga como pendiente cero, pues de lo contrario la tarea quedaría resuelta ya que el estudiante solo modifica la diferencia de las abscisas de la recta GB; mientras que la diferencia de las ordenadas de las rectas AB y GB son iguales.</p>
<pre>{ ordeA=(ordeA+5)%8-4 }</pre>	<p>Si lo anterior sucede este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la</p>

	escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
SetExpressionValue("ORDENADAB",ordeB);	Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así: $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
SetExpressionValue("ABSCISAA",absiA);	Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula: $\text{ABSCISAA} * \text{FactorH} + x(P2)$
SetExpressionValue("ORDENADAA",ordeA);	Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula: $\text{ORDENADAA} * \text{FactorH} + y(P2)$
OrdenadaB=GetExpressionValue("ORDENADAB");	Recupera y asigna a la variable OrdenadaB el valor de la expresión que almacena el valor de la ordenada del punto B en el sistema de coordenadas de la pantalla. Esto para poder tomar este valor e

	incluirlo en el mensaje que le aparecerá al estudiante al inicio de la tarea.
<pre>Alert("La ecuación de la recta roja es de la forma y=ax+b.\n La recta roja corta el eje y en "+OrdenadaB+", utilice la lista desplegable para modificar el denominador del término a en la ecuación, \n hasta lograr que la recta roja quede sobre la recta azul."); }</pre>	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.
<pre>else {</pre>	<i>Si lo anterior no sucede, entonces:</i>
<pre>Alert("Error: Verifica que hayas completado todos los ejercicios propuestos en las tareas anteriores para poder continuar"); } }</pre>	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla para que de esta forma se le indique al estudiante su no posible paso por tareas cuando no ha realizado las anteriores.

<i>function VolverTarea4(){</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
correctos1=GetExpressionValue("correctos");	correctos1=GetExpressionValue("correctos");	Recupera y asigna a la variable correctos1, el valor que tiene la expresión “correctos” en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 1, ésta está resuelta bien.

correctos2=GetExpressionValue("correctos2");	correctos2=GetExpressionValue("correctos2");	Recupera y asigna a la variable correctos2, el valor que tiene la expresión “correctos2” en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 2, ésta está resuelta bien.
	correctos3=GetExpressionValue("correctos3");	Recupera y asigna a la variable correctos3, el valor que tiene la expresión “correctos3” en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 3, ésta está resuelta bien.
correctos3Parte1=GetExpressionValue("correctos3Parte1");		Recupera y asigna a la variable correctos3Parte1, el valor que tiene la expresión “correctos3Parte1” en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 3, ésta está resuelta bien.
correctos3Parte2=GetExpressionValue("correctos3Parte2");		Recupera y asigna a la variable correctos3Parte2, el valor que tiene la expresión “correctos3Parte2” en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 4, ésta está resuelta bien.
	if (correctos1>=5&&correctos2>=5&&correctos3>=5)	Condicional que verifica si las variables correctos1, correctos 2 y correctos3 son menores o iguales a 5. Esto para saber si el estudiante ya ha resuelto la tarea 1, 2 y 3 antes de poder volver a la tarea 4.
if (correctos1>=5&&correctos2>=5&&correctos3Parte1>=5&&correctos3Parte2>=5)		Condicional que verifica si las variables correctos1, correctos 2, correctos3Parte1 y correctos3Parte2 son menores o iguales a 5. Esto para saber si el estudiante ya ha resuelto la tarea 1, 2, 3 y 4 antes de poder volver a la tarea 5.
{	{	<i>Si lo anterior sucede, entonces:</i>

<code>ayuda.setHidden(false)</code>		Coloca visible al botón “ayuda” de la figura en el tomo 1.
<code>continuar.setHidden (true);</code>	<code>continuar.setHidden (true);</code>	Coloca oculto al botón “continuar” de la figura.
<code>volvartarea.setHidden (false);</code>		Coloca visible al botón “volver a...” de la figura.
<code>CasillaCorteOrdenada.set Hidden (true);</code>		Coloca oculta la casilla “b2” de la figura.
<code>CasillaNumeradorPendient e.setHidden (false);</code>		Coloca visible la “CasillaNumeradorA” de la figura.
<code>CasillaDenominadorPendi ente.setHidden (false);</code>		Coloca visible la “CasillaDenominadorA” de la figura.
<code>Show("H,RectaRojaT4,EcuacionRectaRojaYIgual,EcuacionRectaRojaAFraccionLínea,EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4");</code>	<code>Show("H,RectaRojaT4,EcuacionRojaBPositivoT4,EcuacionRojaBNegativoT4");</code>	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
<code>Hide("C,D,M,G,I,J,K,L,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT1,RectaRojaT2,RectaRojaT3Parte1,RectaRojaT3Parte2,RectaRojaT5,RectaRojaT6,Ecu</code>	<code>Hide("C,D,G,I,J,K,L,RectaRojaT1,RectaRojaT2,RectaRojaT3,RectaRojaT5,RectaRojaT6,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1,EcuacionRoja</code>	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.

<p>acionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPostivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNumeradorPendienteT3,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPostivoDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5,EcuacionRectaRojaXBPostivoT6,EcuacionRectaRojaXBNegativoT6,EcuacionRectaRojaBPostivoADeci</p>	<p>BPositivoT2,EcuacionRojaBNegativoT2,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5,EcuacionRojaBPositivoT6,EcuacionRojaBNegativoT6");</p>	
---	---	--

<p>malT6,EcuacionRectaRoja        BNegativoADecimalT6,Tr        ianguloRectaAzul,Segmen        to1TrianguloRectaAzul,Se        gmento2TrianguloRectaAz        ul,TrianguloRectaRojaT3P        arte2,TrianguloRectaRoja        T3Parte1,TrianguloRectaR        ojaT4,Segmento1Triangul        oRectaRojaT3Parte1,Segm        ento2TrianguloRectaRoja        T3Parte1,Segmento3Trian        guloRectaRojaT3Parte1,Se        gmento1TrianguloRectaRo        jaT3Parte2,Segmento2Tria        nguloRectaRojaT3Parte2,        Segmento3TrianguloRecta        RojaT3Parte2,Segmento1        TrianguloRectaRojaT4,Se        gmento2TrianguloRectaRo        jaT4,Segmento3Triangulo        RectaRojaT4,MedidaSegm        ento1TrianguloRectaRoja        T3Parte1,MedidaSegment        o2TrianguloRectaRojaT3P        arte1,MedidaSegmento1Tr        ianguloRectaRojaT3Parte2        ,MedidaSegmento2Triang        uloRectaRojaT3Parte2,Me</p>		
--	--	--

<code>didasegmento1TrianguloRectaRojaT4,Medidasegmento2TrianguloRectaRojaT4");</code>		
<code>SetExpressionValue("Tarea",5);</code>	<code>SetExpressionValue("Tarea",4);</code>	Cambia por cuatro el valor de la expresión “Tarea” que esta oculta en la figura del tomo 2, y a cinco el valor de la expresión “Tarea” que esta oculta en la figura del tomo 1. Esta expresión es la que permite, en ambos tomos, saber en cuál tarea va el estudiante.
<code>ordeH=GetExpressionValue("ORDENADAH");</code>	<code>ordeH=GetExpressionValue("ORDENADAH");</code>	Recupera y asigna a la variable ordeH el valor de la expresión que calcula la ordenada del punto H, en nuestro sistema de coordenadas.
<code>absiH=GetExpressionValue("ABSCISAH");</code>	<code>absiH=GetExpressionValue("ABSCISAH");</code>	Recupera y asigna a la variable absiH el valor de la expresión que calcula la abscisa del punto H, en nuestro sistema de coordenadas.
<code>ordeB=Math.round(Math.random()*8-4);</code>	<code>ordeB=Math.round(Math.random()*8-4);</code>	Calculo de la ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.
<code>absiA=Math.round(Math.random()*14-7);</code>	<code>absiA=Math.round(Math.random()*14-7);</code>	Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
<code>if ((Math.abs(absiA-absiH))&lt;0.0001)</code>	<code>if ((Math.abs(absiA-absiH))&lt;0.0001)</code>	Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiH del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de

		<p>esta tarea, las abscisas de los puntos A y H sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y HB tienen la misma pendiente estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre absiA y absiH no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ absiA=(absiA+8)% 14-7 }</pre>	<pre>{ absiA=(absiA+8)% 14-7 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).</p>
<pre>if (absiA==0) { absiA=1 }</pre>	<pre>if (absiA==0) { absiA=1 }</pre>	<p>Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.</p>
<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<p>Cálculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.</p>
<pre>if ((Math.abs(ordeA-</pre>	<pre>if ((Math.abs(ordeA-</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre</p>

<pre>ordeH))&lt;0.0001)</pre>	<pre>ordeH))&lt;0.0001)</pre>	<p>las variables ordeA y ordeH del script, es menor que 0.0001 o lo que es casi lo mismo que los valores de ordeA y ordeH sean iguales. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos A y H sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y HB tienen la misma pendiente estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre ordeA y ordeH no se busca exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ ordeA=(ordeA+5) %8-4 }</pre>	<pre>{ ordeA=(ordeA+5) %8-4 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>SetExpressionValue("ORDENADAB",ordeB);</pre>	<pre>SetExpressionValue("ORDENADAB",ordeB);</pre>	<p>Asigna a la expresión "ORDENADAB" de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada "interna" de B, cálculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$

SetExpressionValue("ABSCISAA",absiA);	SetExpressionValue("ABSCISAA",absiA);	Asigna a la expresión "ABSCISAA" de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa "interna" de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:  $\text{ABSCISAA} * \text{FactorH} + x(P2)$
SetExpressionValue("ORDENADAA",ordeA);	SetExpressionValue("ORDENADAA",ordeA);	Asigna a la expresión "ORDENADAA" de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada "interna" de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:  $\text{ORDENADAA} * \text{FactorH} + y(P2)$
OrdenadaB=GetExpressionValue("ORDENADAB");	OrdenadaB=GetExpressionValue("ORDENADAB");	Recupera y asigna a la variable OrdenadaB el valor de la expresión que almacena el valor de la ordenada del punto B en el sistema de coordenadas de la pantalla. Esto para poder tomar este valor e incluirlo en el mensaje que le aparecerá al estudiante al inicio de la tarea.
Alert("La ecuación de la recta roja es de la forma $y=ax+b$ . \n La recta corta el eje y en "+OrdenadaB+", escriba en las casillas los valores del numerador y denominador de a \n para que la recta roja quede sobre la recta azul.");	Alert("La ecuación de la recta roja es de la forma $y=ax+b$ . \n La recta roja pasa por el punto "+OrdenadaB+" del eje y, señale la posición donde debe quedar el punto H para que la ecuación roja sea igual a la ecuación	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.

}	azul.");	
	Orde=InteractiveInput ("Haga clic en el punto donde debe quedar H", "Point");	Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto. El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar H", "Point"); se mostrará en la barra de estado (parte inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
	factorh=GetExpressionVal ue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión "FactorH" de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y (1,0) en nuestro sistema de coordenadas.
	ordeH=(Y(Orde)- Y("P2"))/factorh;	Asigna a la variable ordeH el valor de la ordenada que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
	Move ("H",absclH+"*FactorH+x (P2)",ordeH+"*FactorH+y (P2));	Mueve al punto H en la posición (absclH,ordeH) de nuestro sistema de coordenadas.
	SetFixed("H",true); }	Fija al punto H (Impide el desplazamiento)
else  {	else  {	<i>Si lo anterior no sucede, entonces:</i>
Alert("Error: Verifica que hayas completado todos los ejercicios propuestos	Alert("Error: Verifica que hayas completado todos los ejercicios propuestos	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla para que de esta forma se le indique al estudiante su no posible paso por tareas

<pre> en las tareas anteriores para poder continuar"); } } </pre>	<pre> en las tareas anteriores para poder continuar"); } } </pre>	cuando no ha realizado las anteriores.
---	---	--

<i>function VolverTarea5(){</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
correctos1=GetExpressionValue("correctos");	correctos1=GetExpressionValue("correctos");	Recupera y asigna a la variable correctos1, el valor que tiene la expresión “correctos” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 1, ésta está resuelta bien.
correctos2=GetExpressionValue("correctos2");	correctos2=GetExpressionValue("correctos2");	Recupera y asigna a la variable correctos2, el valor que tiene la expresión “correctos2” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 2, ésta está resuelta bien.
	correctos3=GetExpressionValue("correctos3");	Recupera y asigna a la variable correctos3, el valor que tiene la expresión “correctos3” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 3, ésta está resuelta bien.
correctos3Parte1=GetExpression Value("correctos3Parte1");		Recupera y asigna a la variable correctos3Parte1, el valor que tiene la expresión “correctos3Parte1” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en

		la tarea 3, ésta está resuelta bien.
correctos3Parte2=GetExpressionValue("correctos3Parte2");		Recupera y asigna a la variable correctos3Parte2, el valor que tiene la expresión “correctos3Parte2” en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 4, ésta está resuelta bien.
correctos4=GetExpressionValue("correctos5");	correctos4=GetExpressionValue("correctos4");	Recupera y asigna a la variable correctos4, el valor que tiene la expresión “correctos4” en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 4 del tomo 2 y en la tarea 5 del tomo 1, éstas están resueltas bien.
	if (correctos1>=5&&correctos2>=5 &&correctos3>=5&&correctos4>=5)	Condicional que verifica si las variables correctos1, correctos 2, correctos3 y correctos4 son menores o iguales a 5. Esto para saber si el estudiante ya ha resuelto la tarea 1, 2, 3 y 4 antes de poder volver a la tarea 5.
if (correctos1>=5&&correctos2>=5 &&correctos3Parte1>=5&&correctos3Parte2>=5&&correctos4>=5 )		Condicional que verifica si las variables correctos1, correctos 2, correctos3Parte1, correctos3Parte2 y correctos4 son menores o iguales a 5. Esto para saber si el estudiante ya ha resuelto la tarea 1, 2, 3, 4 y 5 antes de poder volver a la tarea 6.
{	{	<i>Si lo anterior sucede, entonces:</i>
ayuda.setHidden(true)		Coloca oculto al botón “ayuda” de la figura en el tomo 1.
continuar.setHidden (true);	continuar.setHidden (true);	Coloca oculto al botón “continuar” de la figura.
volvartarea.setHidden (false);		Coloca visible al botón “volver a...” de la figura.

CasillaCorteOrdenada.setHidden (true);		Coloca oculta la casilla “b2” de la figura.
CasillaNumeradorPendiente.setHidden (true);		Coloca oculta la “CasillaNumeradorA” de la figura.
CasillaDenominadorPendiente.setHidden (true);		Coloca oculta la “CasillaDenominadorA” de la figura.
Show("I,J,RectaRojaT5,EcuacionRectaRojaXT5,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5");	Show("I,J,RectaRojaT5,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5");	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
Hide("C,D,M,G,H,K,L,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT1,RectaRojaT2,RectaRojaT3Parte1,RectaRojaT3Parte2,RectaRojaT4,RectaRojaT6,EcuacionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADeci	Hide("C,D,G,H,K,L,RectaRojaT1,RectaRojaT2,RectaRojaT3,RectaRojaT4,RectaRojaT6,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1,EcuacionRojaBPositivoT2,EcuacionRojaBNegativoT2,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3,EcuacionRojaBPositivoT4,EcuacionRojaBNegativoT4,EcuacionRojaBPositivoT6,Ecu	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.

<p>malT2,EcuacionRectaRoja BNegativoADecimalT2,Ec uacionRectaRojaYIgualNu meradorPendienteT3,Ecu acionRectaRojaBPositivoN umeradorPendienteT3,Ecu acionRectaRojaBNegativo NumeradorPendienteT3,E cuacionRectaRojaBPositiv oDenominadorPendienteT 3,EcuacionRectaRojaBNe gativoDenominadorPendie nteT3,EcuacionRectaRoja YIgual,EcuacionRectaRoja AFraccionLínea,Ecuacion RectaRojaABpositivoT4,E cuacionRectaRojaABNega tivoT4,EcuacionRectaRoja XBPositivoT6,EcuacionRe ctaRojaXBNegativoT6,Ec uacionRectaRojaBPostivo ADecimalT6,EcuacionRec taRojaBNegativoADecima lT6,TrianguloRectaAzul,S egmento1TrianguloRectaA zul,Segmento2TrianguloR ectaAzul,TrianguloRectaR ojaT3Parte2,TrianguloRec taRojaT3Parte1,Triangulo</p>	<p>acionRojaBNegativoT6");</p>	
---	--------------------------------	--

<p>RectaRojaT4,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Segmento3TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4");</p>		
<p>SetExpressionValue("Tarea",6);</p>	<p>SetExpressionValue("Tarea",5);</p>	<p>Cambia por cinco el valor de la expresión “Tarea” que esta oculta en la figura del tomo 2, y a seis el valor de la expresión “Tarea” que esta oculta en la figura del tomo 1. Esta expresión es la que permite,</p>

		en ambos tomos, saber en cuál tarea va el estudiante.
<code>ordeJ=GetExpressionValue("ORDENADAJ");</code>	<code>ordeJ=GetExpressionValue("ORDENADAJ");</code>	Recupera y asigna a la variable ordeJ el valor de la expresión que calcula la ordenada del punto J, en nuestro sistema de coordenadas.
<code>absiJ=GetExpressionValue("ABSCISAJ");</code>	<code>absiJ=GetExpressionValue("ABSCISAJ");</code>	Recupera y asigna a la variable absiJ el valor de la expresión que calcula la abscisa del punto J, en nuestro sistema de coordenadas.
<code>ordeI=GetExpressionValue("ORDENADAI");</code>	<code>ordeI=GetExpressionValue("ORDENADAI");</code>	Recupera y asigna a la variable ordeI el valor de la expresión que calcula la ordenada del punto I, en nuestro sistema de coordenadas.
<code>ordeB=Math.round(Math.random()*8-4);</code>	<code>ordeB=Math.round(Math.random()*8-4);</code>	Calculo de la ordenada de B: Calcula un valor aleatorio ( <code>Math.random()</code> ), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo ( <code>Math.round</code> ) y asignarlo a la variable ordeB.
<code>if ((Math.abs(ordeI-ordeB))&lt;0.0001)</code>	<code>if ((Math.abs(ordeI-ordeB))&lt;0.0001)</code>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeI y ordeB del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos I y B sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante.</p> <p>La razón por la que la diferencia entre ordeI y ordeB no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<code>{</code>	<code>{</code>	Si lo anterior sucede, este comando asigna a la variable ordeB, el

<pre>ordeB=(ordeB+5) %8-4 }</pre>	<pre>ordeB=(ordeB+5) %8-4 }</pre>	<p>valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).</p>
<pre>absiA=Math.round(Math.random()*14-7);</pre>	<pre>absiA=Math.round(Math.random()*14-7);</pre>	<p>Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.</p>
<pre>if ((Math.abs(absiA-absiJ))&lt;0.0001)</pre>	<pre>if ((Math.abs(absiA-absiJ))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiJ del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las abscisas de los puntos A y J sean iguales y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y JI tienen la misma pendiente estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre absiA y absiJ no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ absiA=(absiA+8)% 14-7 }</pre>	<pre>{ absiA=(absiA+8)% 14-7 }</pre>	<p>Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad</p>

		total de números positivos o negativos en la escala del eje horizontal).
<pre>if (absiA==0) { absiA=1 }</pre>	<pre>if (absiA==0){absiA=1}</pre>	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
<pre>ordeA=Math.round(Math.random()*8-4);</pre>	<pre>ordeA=Math.round(Math.random()*8-4);</pre>	Calculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.
<pre>if ((Math.abs(ordeA-ordeJ))&lt;0.0001)</pre>	<pre>if ((Math.abs(ordeA-ordeJ))&lt;0.0001)</pre>	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeJ del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos A y J sean iguales y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y JI tienen la misma pendiente, estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre ordeA y ordeJ no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
<pre>{ ordeA=(ordeA+5) %8-4</pre>	<pre>{ ordeA=(ordeA+5) %8-4</pre>	Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la

}	}	escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
SetExpressionValue("ORDENADAB",ordeB);	SetExpressionValue("ORDENADAB",ordeB);	<p>Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:</p> $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
SetExpressionValue("ABSCISAA",absiA);	SetExpressionValue("ABSCISAA",absiA);	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA} * \text{FactorH} + x(P2)$
SetExpressionValue("ORDENADAA",ordeA);	SetExpressionValue("ORDENADAA",ordeA);	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA} * \text{FactorH} + y(P2)$
Alert("La ecuación de la recta roja es de la forma	Alert ("La ecuación de la recta roja es de la forma	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje

$y=ax+b$ . Utilice las listas desplegadas para modificar el valor de $a$ y de $b$ en la ecuación hasta lograr que la recta roja quede sobre la recta azul."); }	$y=ax+b$ . Mueva el punto I sobre el eje $y$ y el punto J sobre la cuadrícula, hasta hacer que la ecuación roja sea igual a la ecuación azul."); }	describe el enunciado de la tarea al comenzar la tarea.
else {	else {	<u>Si lo anterior no sucede entonces:</u>
Alert("Error: Verifica que hayas completado todos los ejercicios propuestos en las tareas anteriores para poder continuar"); } }	Alert("Error: Verifica que hayas completado todos los ejercicios propuestos en las tareas anteriores para poder continuar"); } }	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla para que de esta forma se le indique al estudiante su no posible paso por tareas cuando no ha realizado las anteriores.

<i>function VolverTarea6(){</i>		
<b><u>Tomo 1:</u> Conversión algebraica a la gráfica.</b>	<b><u>Tomo 2:</u> Conversión gráfica a la algebraica.</b>	<b>Descripción</b>
correctos1=GetExpressionValue("correctos");	correctos1=GetExpressionValue("correctos");	Recupera y asigna a la variable correctos1, el valor que tiene la expresión "correctos" en la figura. Esta expresión almacena la cantidad de veces que al oprimir el botón "verificar" en la tarea 1,

		ésta está resuelta bien.
correctos2=GetExpressionValue("correctos2");	correctos2=GetExpressionValue("correctos2");	Recupera y asigna a la variable correctos2, el valor que tiene la expresión “correctos2” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 2, ésta está resuelta bien.
	correctos3=GetExpressionValue("correctos3");	Recupera y asigna a la variable correctos3, el valor que tiene la expresión “correctos3” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 3, ésta está resuelta bien.
correctos3Parte1=GetExpressionValue("correctos3Parte1");		Recupera y asigna a la variable correctos3Parte1, el valor que tiene la expresión “correctos3Parte1” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 3, ésta está resuelta bien.
correctos3Parte2=GetExpressionValue("correctos3Parte2");		Recupera y asigna a la variable correctos3Parte2, el valor que tiene la expresión “correctos3Parte2” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 4, ésta está resuelta bien.
correctos4=GetExpressionValue("correctos4");	correctos4=GetExpressionValue("correctos4");	Recupera y asigna a la variable correctos4, el valor que tiene la expresión “correctos4” en la figura. Está expresión almacena la cantidad de veces que al oprimir el botón “verificar” en la tarea 4 del tomo 2 y en la tarea 5 del tomo 1, éstas están resueltas bien.
correctos5=GetExpressionValue("correctos5");	correctos5=GetExpressionValue("correctos5");	Recupera y asigna a la variable correctos5, el valor que tiene la expresión “correctos5” en la figura. Está expresión almacena la

		cantidad de veces que al oprimir el botón “verificar” en la tarea 5 del tomo 2 y en la tarea 6 del tomo 1, éstas están resueltas bien.
	<code>if (correctos1&gt;=5&amp;&amp;correctos2&gt;=5&amp;&amp;correctos3&gt;=5&amp;&amp;correctos4&gt;=5&amp;&amp;correctos5&gt;=5)</code>	Condicional que verifica si las variables correctos1, correctos 2, correctos3, correctos4 y correctos5 son menores o iguales a 5. Esto para saber si el estudiante ya ha resuelto la tarea 1, 2, 3, 4 y 5 antes de poder volver a la tarea 6.
	<code>if (correctos1&gt;=5&amp;&amp;correctos2&gt;=5&amp;&amp;correctos3Parte1&gt;=5&amp;&amp;correctos3Parte2&gt;=5&amp;&amp;correctos4&gt;=5&amp;&amp;correctos5&gt;=5)</code>	Condicional que verifica si las variables correctos1, correctos 2, correctos3Parte1, correctos3Parte2, correctos4 y correctos5 son menores o iguales a 5. Esto para saber si el estudiante ya ha resuelto la tarea 1, 2, 3, 4, 5 y 6 antes de poder volver a la tarea 7.
<code>{</code>	<code>{</code>	<i>Si lo anterior sucede, entonces:</i>
<code>ayuda.setHidden(true)</code>		Coloca oculto al botón “ayuda” de la figura en el tomo 1.
<code>continuar.setHidden (true);</code>	<code>continuar.setHidden (true);</code>	Coloca oculto al botón “continuar” de la figura.
<code>volvertarea.setHidden (false);</code>		Coloca visible al botón “volver a...” de la figura.
<code>CasillaCorteOrdenada.setHidden (false);</code>		Coloca visible la casilla “b2” de la figura.
<code>CasillaNumeradorPendiente.setHidden (false);</code>		Coloca visible la “CasillaNumeradorA” de la figura.
<code>CasillaDenominadorPendiente.setHidden (false);</code>		Coloca visible la “CasillaDenominadorA” de la figura.
<code>Show("K,L,RectaRojaT6, EcuacionRectaRojaYIgual,</code>	<code>Show("K,L,RectaRojaT6, EcuacionRojaBPositivoT6</code>	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis,

<p>EcuacionRectaRojaAFraccionLínea,EcuacionRectaRojaXBPositivoT6,EcuacionRectaRojaXBNegativoT6,EcuacionRectaRojaBPositivoADecimalT6,EcuacionRectaRojaBNegativoADecimalT6");</p>	<p>,EcuacionRojaBNegativoT6");</p>	<p>separados con comas.</p>
<p>Hide("C,D,M,G,H,I,J,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT1,RectaRojaT2,RectaRojaT3Parte1,RectaRojaT3Parte2,RectaRojaT4,RectaRojaT5,EcuacionRectaRojaBPostivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPositivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNu</p>	<p>Hide("C,D,G,H,I,J,RectaRojaT1,RectaRojaT2,RectaRojaT3,RectaRojaT4,RectaRojaT5,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1,EcuacionRojaBPositivoT2,EcuacionRojaBNegativoT2,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3,EcuacionRojaBPositivoT4,EcuacionRojaBNegativoT4,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5");</p>	<p>Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.</p>

<p> meradorPendienteT3,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBPositivoDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3,EcuacionRectaRojaAFraccionYIgualT4,EcuacionRectaRojaAFraccionLíneaFracciónT4,EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4,EcuacionRectaRojaXT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5, TrianguloRectaAzul,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul, TrianguloRectaRojaT3Parte2, TrianguloRectaRojaT3Parte1, TrianguloRectaRojaT4, Segmento1TrianguloRectaRojaT3Parte1, Segmento2TrianguloRectaRojaT3Parte1, Segmento3TrianguloRectaRojaT3Parte1, Segm </p>		
---	--	--

<p>ento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Segmento3TrianguloRectaRojaT3Parte2,Segmento1TrianguloRectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4");</p>		
<p>SetExpressionValue("Tarea",7);</p>	<p>SetExpressionValue("Tarea",6);</p>	<p>Cambia por seis el valor de la expresión “Tarea” que esta oculta en la figura del tomo 2, y por siete, el valor de la expresión “Tarea” que esta oculta en la figura del tomo 1. Esta expresión es la que permite, en ambos tomos, saber en cuál tarea va el estudiante.</p>
<p>ordeK=GetExpressionValue("ORDENADAK");</p>	<p>ordeK=GetExpressionValue("ORDENADAK");</p>	<p>Recupera y asigna a la variable ordeK el valor de la expresión que calcula la ordenada del punto K, en nuestro sistema de coordenadas.</p>
<p>absiL=GetExpressionValue("ORDENADAK");</p>	<p>absiL=GetExpressionValue("ORDENADAK");</p>	<p>Recupera y asigna a la variable absiL el valor de la expresión que</p>

<code>e("ABSCISAL");</code>	<code>e("ABSCISAL");</code>	calcula la abscisa del punto L, en nuestro sistema de coordenadas.
<code>ordeL=GetExpressionValu e("ORDENADAL");</code>	<code>ordeL=GetExpressionValu e("ORDENADAL");</code>	Recupera y asigna a la variable ordeL el valor de la expresión que calcula la ordenada del punto L, en nuestro sistema de coordenadas.
<code>ordeB=Math.round(Math.r andom()*8-4);</code>	<code>ordeB=Math.round(Math.r andom()*8-4);</code>	Calculo de la ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.
<code>if ((Math.abs(ordeK- ordeB))&lt;0.0001)</code>	<code>if ((Math.abs(ordeK- ordeB))&lt;0.0001)</code>	Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeK y ordeB del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos K y B sean iguales y por tanto que la mitad del ejercicio aparezca resuelto al estudiante.  La razón por la que la diferencia entre ordeK y ordeB no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.
<code>{ ordeB=(ordeB+5) %8-4 }</code>	<code>{ ordeB=(ordeB+5) %8-4 }</code>	Si lo anterior sucede, este comando asigna a la variable ordeB, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
<code>absiA=Math.round(Math.r andom()*14-7);</code>	<code>absiA=Math.round(Math.r andom()*14-7);</code>	Calculo de la abscisa de A: Calcula un valor aleatorio (Math.random()), que multiplica por 14 (cantidad total de números

		en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
<code>if ((Math.abs(absiA-absiL))&lt;0.0001)</code>	<code>if ((Math.abs(absiA-absiL))&lt;0.0001)</code>	Condicional que verifica si el valor absoluto de la diferencia entre las variables absiA y absiL del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las abscisas de los puntos A y L sean iguales y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y LK tienen la misma pendiente estos puntos quedan superpuestos.
<code>{ absiA=(absiA+8)% 14-7 }</code>	<code>{ absiA=(absiA+8)% 14-7 }</code>	Si lo anterior sucede, este comando asigna a la variable absiA, el valor del módulo de absiA+8 (8 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje horizontal, con 1) multiplicado por 14 (cantidad total de números en la escala del eje horizontal) y restado con 7 (cantidad total de números positivos o negativos en la escala del eje horizontal).
<code>if (absiA==0) { absiA=1 }</code>	<code>if (absiA==0) { absiA=1 }</code>	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
<code>ordeA=Math.round(Math.random()*8-4);</code>	<code>ordeA=Math.round(Math.random()*8-4);</code>	Calculo de la ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego

		redondearlo (Math.round) y asignarlo a la variable ordeA.
if ((Math.abs(ordeA-ordeL))<0.0001)	if ((Math.abs(ordeA-ordeL))<0.0001)	<p>Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeA y ordeL del script, es menor que 0.0001. Este condicional se establece evitando que, al comenzar otro ejercicio de esta tarea, las ordenadas de los puntos A y L sean iguales y por tanto que una parte del ejercicio aparezca resuelto al estudiante, pues cuando las rectas AB y LK tienen la misma pendiente, estos puntos quedan superpuestos.</p> <p>La razón por la que la diferencia entre ordeA y ordeL no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.</p>
{ ordeA=(ordeA+5) %8-4 }	{ ordeA=(ordeA+5) %8-4 }	Si lo anterior sucede, este comando asigna a la variable ordeA, el valor del módulo de ordeA+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
SetExpressionValue("ORDENADAB",ordeB);	SetExpressionValue("ORDENADAB",ordeB);	Asigna a la expresión "ORDENADAB" de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada "interna" de B, calculo que está almacenado en la expresión PosicionBy (oculta en la figura), así:

		$\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
<pre>SetExpressionValue("ABSCISAA",absiA);</pre>	<pre>SetExpressionValue("ABSCISAA",absiA);</pre>	<p>Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula:</p> $\text{ABSCISAA} * \text{FactorH} + x(P2)$
<pre>SetExpressionValue("ORDENADAA",ordeA);</pre>	<pre>SetExpressionValue("ORDENADAA",ordeA);</pre>	<p>Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula:</p> $\text{ORDENADAA} * \text{FactorH} + y(P2)$
<pre>Alert("La ecuación de la recta roja es de la forma y=ax+b.\n Escriba en las casillas los valores correspondientes a a y a b en la ecuación, \n para que la recta roja quede sobre la recta azul."); }</pre>	<pre>Alert ("La ecuación de la recta roja es de la forma y=ax+b. \n Señale sucesivamente la posición donde deben quedar los puntos K y L, para que la ecuación roja sea igual a la ecuación azul. \n Tenga en cuenta que el punto K es el punto de la recta roja que está sobre el eje Y.");</pre>	<p>Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla. Este mensaje describe el enunciado de la tarea al comenzar la tarea.</p>
	<pre>Orde=InteractiveInput ("Haga clic en el punto</pre>	<p>Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto.</p>

	donde debe quedar K", "Point");	El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar K", "Point"); se mostrará en la barra de estado (parte inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
	factorh=GetExpressionValue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión "FactorH" de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y (1,0) en nuestro sistema de coordenadas.
	ordeK=(Y(Orde)-Y("P2"))/factorh;	Asigna a la variable ordeK el valor de la ordenada que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
	Move ("K",absciK+"*FactorH+x (P2)",ordeK+"*FactorH+y (P2)");	Mueve al punto K en la posición (absiK,ordeK) de nuestro sistema de coordenadas.
	SetFixed("K",true);	Fija al punto K (Impide el desplazamiento)
	Orde=InteractiveInput ("Haga clic en el punto donde debe quedar L", "Point");	Asigna a la variable Orde, el nombre del objeto sobre el cual el usuario hace clic. En nuestro caso el objeto a señalar será un punto. El mensaje indicado entre paréntesis, ("Haga clic en el punto donde debe quedar L", "Point"); se mostrará en la barra de estado (parte inferior de la figura). El comando interactiveinput le ordena al sistema detener los procesos hasta que el usuario haga clic sobre un objeto determinado (punto, recta, segmento, etc.) en la pantalla.
	factorh=GetExpressionValue("FactorH");	Asigna a la variable factorh del script, el valor que tiene la expresión "FactorH" de la figura. La expresión FactorH de la figura es la que calcula el valor de la distancia de los puntos P2 y P3 designados respectivamente por nosotros como los puntos (0,0) y

		(1,0) en nuestro sistema de coordenadas.
	ordeL=(Y(Orde)-Y("P2"))/factorh;	Asigna a la variable ordeL el valor de la ordenada que tendrá el punto seleccionado por el usuario con el InteractiveInput en nuestro sistema de coordenadas.
	Move ("L",absciL+"*FactorH+x(P2)",ordeL+"*FactorH+y(P2));	Mueve al punto L en la posición (absiL,ordeL) de nuestro sistema de coordenadas.
	SetFixed("L",true); }	Fija al punto L (Impide el desplazamiento)
else {	else {	<u>Si lo anterior no sucede, entonces:</u>
Alert("Error: Verifica que hayas completado todos los ejercicios propuestos en las tareas anteriores para poder continuar"); } }	Alert("Error: Verifica que hayas completado todos los ejercicios propuestos en las tareas anteriores para poder continuar"); } }	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla para que de está forma se le indique al estudiante su no posible paso por tareas cuando no ha realizado las anteriores.

## V. Reiniciar

Vuelve a la figura a un estado inicial; colocando a la secuencia de actividades desde ceros, es decir, como si no se hubiese dado antes una interacción con el sujeto. Este script no requiere de un botón, se ejecuta inmediatamente se abra la actividad.

<b><u>Tomo 1: Conversión algebraica a la gráfica</u></b>	<b><u>Tomo 2: Conversión gráfica a la algebraica</u></b>	<b>Descripción</b>
t = getZC().JCM.CPs;	t = getZC().JCM.CPs;	Este comando recupera en la variable t, todos los

		<p>controles de la figura (botones, casillas, listas desplegables, etc.). Los controles no son reconocidos como objetos de la figura en el script, razón por la que debe utilizarse este comando para posteriormente llamarlos y utilizarlos en él.</p>
<pre> for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getName().equals("b2")) { CasillaCorteOrdenada=t.get(i); break; } }  for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getName().equals("CasillaDenominadorA")) { CasillaDenominadorPendiente=t.get(i); break; } }  for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getName().equals("CasillaN </pre>	<pre> for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getName().equals("continuar")) { continuar=t.get(i); break; } }  for(i=0; i&lt;t.size(); i++) { if(t.get(i).O.getName().equals("volver")) { volvertarea=t.get(i); break; } } </pre>	<p>En los dos tomos vemos que este bucle recorre la lista de los controles almacenados en la variable t, recuperando los botones “continuar” y “volver a”. En el tomo 1 el bucle recupera además el botón “ayuda” y las casillas “b2”, “CasillaDenominadorA” y “CasillaNumeradorA”. Esto para poder ocultarlas y mostrarlas cuando así se requiera en las tareas, llamándolas con los nombres que después del condicional se le asigna a estos controles recuperados.</p>

<pre>umeradorA"))     {         CasillaNumeradorPendiente=t.g         et(i);         break;     } }  for(i=0; i&lt;t.size(); i++)     {         if(t.get(i).O.getName().equals("continua r"))             {                 continuar=t.get(i);                 break;             }     }  for(i=0; i&lt;t.size(); i++)     {         if(t.get(i).O.getName().equals("volver") )             {                 volvertarea=t.get(i);                 break;             }     }  for(i=0; i&lt;t.size(); i++)     {</pre>		
--	--	--

<pre> if(t.get(i).O.getName().equals("ayuda")) { ayuda=t.get(i); break; } } </pre>		
<pre> ayuda.setHidden(true); </pre>		Coloca oculto al botón “ayuda” de la figura.
<pre> continuar.setHidden (true); </pre>	<pre> continuar.setHidden (true); </pre>	Coloca oculto al botón “continuar” de la figura.
<pre> volvartarea.setHidden (true); </pre>	<pre> volvartarea.setHidden (true); </pre>	Coloca oculto al botón “volver a...” de la figura.
<pre> CasillaCorteOrdenada.setHidden (true); </pre>		Coloca oculta la casilla “b2” de la figura.
<pre> CasillaNumeradorPendiente.setHidden (true); </pre>		Coloca oculta la “CasillaNumeradorA” de la figura.
<pre> CasillaDenominadorPendiente.setHidden (true); </pre>		Coloca oculta la “CasillaDenominadorA” de la figura.
<pre> Show("C,RectaRojaT1,EcuacionRectaRojaBPositivoT1,EcuacionRectaRojaBNegativoT1,EcuacionRectaRojaBPostivoAFraccionT1,EcuacionRectaRojaBNegativoAFraccionT1"); </pre>	<pre> Show ("C,RectaRojaT1,EcuacionRojaBPositivoT1,EcuacionRojaBNegativoT1"); ; </pre>	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
<pre> Hide("D,M,G,H,I,J,K,L,OrdenadaM,OrdenadaG,OrdenadaH,RectaRojaT2,RectaRojaT3Parte1,RectaRojaT3Parte2,RectaRojaT4,RectaRojaT </pre>	<pre> Hide("D,F,G,H,I,J,K,L,RectaRojaT2,RectaRojaT3,RectaRojaT4,RectaRojaT5,RectaRojaT6,EcuacionRojaBPo </pre>	Este comando oculta los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.

<p>5,RectaRojaT6,EcuacionRectaRojaBPostivoAFraccionT2,EcuacionRectaRojaBNegativoAFraccionT2,EcuacionRectaRojaBPostivoADecimalT2,EcuacionRectaRojaBNegativoADecimalT2,EcuacionRectaRojaYIgualNumeradorPendienteT3,EcuacionRectaRojaBPositivoNumeradorPendienteT3,EcuacionRectaRojaBNegativoNumeradorPendienteT3,EcuacionRectaRojaBPositivoDenominadorPendienteT3,EcuacionRectaRojaYIgualDenominadorPendienteT3,EcuacionRectaRojaBNegativoDenominadorPendienteT3,EcuacionRectaRojaYIgual,EcuacionRectaRojaAFraccionLínea,EcuacionRectaRojaABpositivoT4,EcuacionRectaRojaABNegativoT4,EcuacionRectaRojaXT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBPostivoT5,EcuacionRectaRojaBNegativoT5,EcuacionRectaRojaXBPositivoT6,EcuacionRectaRojaXBNegativoT6,EcuacionRectaRojaBPostivoADecimalT6,EcuacionRectaRojaBNegativoADecimalT6,TrianguloRectaAzul,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,TrianguloRectaRojaT3Parte2,TrianguloRectaRojaT3Parte1,TrianguloRectaRojaT4,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaRojaT3Parte2,Segmento2TrianguloRectaRojaT3Parte2,Segmento3TrianguloRectaRojaT3Parte2,Segmento1Triangulo</p>	<p>sitivoT2,EcuacionRojaBNegativoT2,EcuacionRojaBPositivoT3,EcuacionRojaBNegativoT3,EcuacionRojaBPositivoT4,EcuacionRojaBNegativoT4,EcuacionRojaBPositivoT5,EcuacionRojaBNegativoT5,EcuacionRojaBPositivoT6,EcuacionRojaBNegativoT6" );</p>	
--	---	--

RectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte2,MedidaSegmento2TrianguloRectaRojaT3Parte2,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4,CursorPendiente,ParteEnteraCursorPendiente,ValorCursorNumeradorPendiente,ListaDesplegableNumeradorPendiente");		
SetExpressionValue("correctos",0);	SetExpressionValue("correctos",0);	Cambia por cero el valor de la expresión “correctos” que esta oculta en la figura. La expresión “correctos” es la que almacena la cantidad de ejercicios resueltos bien en la tarea 1.
SetExpressionValue("correctos2",0);	SetExpressionValue("correctos2",0);	Cambia por cero el valor de la expresión “correctos2” que esta oculta en la figura. La expresión “correctos2” es la que almacena la cantidad de ejercicios resueltos bien en la tarea 2.
	SetExpressionValue("correctos3",0);	Cambia por cero el valor de la expresión “correctos3” que esta oculta en la figura. La expresión “correctos3” es la que almacena la cantidad de ejercicios resueltos bien en la tarea 3 del tomo 2.
SetExpressionValue("correctos3Parte1",0);		Cambia por cero el valor de la expresión “correctos3Parte1” que esta oculta en la figura. La expresión “correctos3Parte1” es la que almacena la cantidad de ejercicios resueltos bien en la tarea 3 del tomo 1.
SetExpressionValue("correctos3Parte2",0);		Cambia por cero el valor de la expresión

		“correctos3Parte2” que esta oculta en la figura. La expresión “correctos3Parte2” es la que almacena la cantidad de ejercicios resueltos bien en la tarea 4 del tomo 1.
SetExpressionValue("correctos4",0);	SetExpressionValue("correctos4",0);	Cambia por cero el valor de la expresión “correctos4” que esta oculta en la figura. La expresión “correctos4” es la que almacena la cantidad de ejercicios resueltos bien en la tarea 4 del tomo 2 y en la tarea 5 del tomo 1.
SetExpressionValue("correctos5",0);	SetExpressionValue("correctos5",0);	Cambia por cero el valor de la expresión “correctos5” que esta oculta en la figura. La expresión “correctos5” es la que almacena la cantidad de ejercicios resueltos bien en la tarea 5 del tomo 2 y en la tarea 6 del tomo 1.
SetExpressionValue("correctos6",0);	SetExpressionValue("correctos6",0);	Cambia por cero el valor de la expresión “correctos6” que esta oculta en la figura. La expresión “correctos6” es la que almacena la cantidad de ejercicios resueltos bien en la tarea 6 del tomo 2 y en la tarea 7 del tomo 1.
SetExpressionValue("Tarea",1);	SetExpressionValue("Tarea",1);	Cambia por uno el valor de la expresión “Tarea” que esta oculta en la figura. Esta expresión es la que permite saber, en cada tomo, en cuál tarea va el estudiante.
ordeC=GetExpressionValue("ValorListaDesplegableCorteOrdenadaT3");	ordeC=GetExpressionValue("ORDENADAC");	Recupera y asigna a la variable ordeC el valor de la expresión que calcula la ordenada del punto C, en nuestro sistema de coordenadas. Para el tomo 1 esa expresión es: "ValorListaDesplegableCorteOrdenadaT3") y para el tomo 2 es: "ORDENADAC".
ordeB=Math.round(Math.random()*8-4);	ordeB=Math.round(Math.random()*8-4);	Calculo de la nueva Ordenada de B: Calcula un valor aleatorio (Math.random()), que multiplica por 8

		(cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeB.
if ((Math.abs(ordeC-ordeB))<0.0001)	if ((Math.abs(ordeC-ordeB))<0.0001)	Condicional que verifica si el valor absoluto de la diferencia entre las variables ordeC y ordeB del script, es menor que 0.0001, o lo que es casi lo mismo, que la ordeC sea igual a la ordeB. Este condicional se establece evitando que, al comenzar el primer ejercicio de esta tarea, las ordenadas de los puntos B y C sean iguales y por tanto que el nuevo ejercicio aparezca resuelto al estudiante. La razón por la que la diferencia entre ordeC y ordeB no se busque exacta, es porque éstos valores son un resultado de un cálculo entre las coordenadas de los puntos que aparecen en la pantalla.
{ ordeB=(ordeB+5)%8-4 }	{ ordeB=(ordeB+5)%8-4 }	Si lo anterior sucede, este comando asigna a la variable OrdeB, el valor del módulo de OrdeB+5 (5 como el número obtenido al adicionar la cantidad total de números positivos o negativos en la escala del eje vertical, con 1) multiplicado por 8 (cantidad total de números en la escala del eje vertical) y restado con 4 (cantidad total de números positivos o negativos en la escala del eje vertical).
absiA=Math.round(Math.random()*14-7);	absiA=Math.round(Math.random()*1	Calculo de la nueva abscisa de A: Calcula un valor

	4-7);	aleatorio (Math.random()), que multiplica por 14 (cantidad total de números en la escala del eje horizontal) y le resta 7 (cantidad total de números positivos o negativos en la escala del eje horizontal), para luego redondearlo (Math.round) y asignarlo a la variable absiA.
if (absiA==0) { absiA=1 }	if (absiA==0) { absiA=1 }	Condicional que verifica si absiA es igual a 0. Si esto sucede, entonces, asigna a la variable absiA el valor de 1. Este condicional es el que evita que haya rectas verticales, es decir, con pendientes indeterminadas.
ordeA=Math.round(Math.random()*8-4);	ordeA=Math.round(Math.random()*8-4);	Calculo de la nueva ordenada de A: Calcula un valor aleatorio (Math.random()), que multiplica por 8 (cantidad total de números en la escala del eje vertical) y le resta 4 (cantidad total de números positivos o negativos en la escala del eje vertical), para luego redondearlo (Math.round) y asignarlo a la variable ordeA.
SetExpressionValue("ORDENADAB",ordeB);	SetExpressionValue("ORDENADAB",ordeB);	Asigna a la expresión “ORDENADAB” de la figura, el valor de la variable ordeB del script. Este es el valor de la ordenada de B en el sistema de coordenadas que aparece en la pantalla. Ese sistema de coordenadas es diferente al sistema de coordenadas que utiliza el software para determinar la posición de los puntos en la pantalla. Por eso es necesaria esta expresión para calcular la ordenada “interna” de B, calculo que está almacenado en la expresión PosicionBy (oculta en la

		figura), así: $\text{PosicionBy} = \text{OrdenadaB} * \text{FactorH} + y(P2)$
<code>SetExpressionValue("ABSCISAA",absiA);</code>	<code>SetExpressionValue("ABSCISAA",absiA);</code>	Asigna a la expresión “ABSCISAA” de la figura, el valor de la variable absiA del script. Esto para calcular la abscisa “interna” de A con la expresión PosicionAx (oculta en la figura). La expresión PosicionAx calcula: $\text{ABSCISAA} * \text{FactorH} + x(P2)$
<code>SetExpressionValue("ORDENADAA",ordeA);</code>	<code>SetExpressionValue("ORDENADAA",ordeA);</code>	Asigna a la expresión “ORDENADAA” de la figura, el valor de la variable ordeB del script. Esto para calcular la ordenada “interna” de A con la expresión PosicionAy (oculta en la figura). La expresión PosicionAy calcula: $\text{ORDENADAA} * \text{FactorH} + y(P2)$
<code>Alert("La ecuación de la recta roja es de la forma <math>y=ax+b</math>. \n Despliegue la lista de números para modificar el valor de b en esa ecuación, \n hasta lograr que la recta roja quede sobre la recta azul.");</code>	<code>Alert("La ecuación de la recta roja es de la forma <math>y=ax+b</math>. \n Mueva el punto C sobre el eje y, hasta hacer que la ecuación roja sea igual a la ecuación azul.");</code>	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.

## VI. Contadores

Este script no tiene asociado ningún botón, se debe ejecutar desde la lista de los scripts. Este script quiere dar libertad al profesor de poder borrar de la memoria los valores de las expresiones correspondientes al total de ejercicios realizados y al total de ejercicios mal.

<b><u>Tomo 1: Conversión algebraica a la gráfica</u></b>	<b><u>Tomo 2: Conversión gráfica a la algebraica</u></b>	<b>Descripción</b>
SetExpressionValue("ContadorT1",0);	SetExpressionValue("ContadorT1",0);	Cambia por cero el valor de la expresión “ContadorT1” que esta oculta en la figura. La expresión “ContadorT1” es la que almacena la cantidad de veces que el estudiante utiliza el botón verificar en la tarea 1 de los dos tomos.
SetExpressionValue("ContadorT2",0);	SetExpressionValue("ContadorT2",0);	Cambia por cero el valor de la expresión “ContadorT2” que esta oculta en la figura. La expresión “ContadorT2” es la que almacena la cantidad de veces que el estudiante utiliza el botón verificar en la tarea 2 de los dos tomos.
	SetExpressionValue("ContadorT3",0);	Cambia por cero el valor de la expresión “ContadorT3” que esta oculta en la figura. La expresión “ContadorT3” es la que almacena la cantidad de veces que el estudiante utiliza el botón verificar en la tarea 3 del segundo tomo.
SetExpressionValue("ContadorT3Parte1",0);		Cambia por cero el valor de la expresión “ContadorT3Parte1” que esta oculta en la figura. La expresión “ContadorT3Parte1” es la que almacena la cantidad de veces que el estudiante utiliza el botón verificar en la tarea 3 del primer tomo.
SetExpressionValue("ContadorT3Parte2",0);		Cambia por cero el valor de la expresión “ContadorT3Parte2” que esta oculta en la figura.

		La expresión “ContadorT3Parte2” es la que almacena la cantidad de veces que el estudiante utiliza el botón verificar en la tarea 4 del primer tomo.
SetExpressionValue("ContadorT4",0);	SetExpressionValue("ContadorT4",0);	Cambia por cero el valor de la expresión “ContadorT4” que esta oculta en la figura. La expresión “ContadorT4” es la que almacena la cantidad de veces que el estudiante utiliza el botón verificar en la tarea 4 del tomo 2 y en la tarea 5 del tomo 1.
SetExpressionValue("ContadorT5",0);	SetExpressionValue("ContadorT5",0);	Cambia por cero el valor de la expresión “ContadorT5” que esta oculta en la figura. La expresión “ContadorT5” es la que almacena la cantidad de veces que el estudiante utiliza el botón verificar en la tarea 5 del segundo tomo y en la tarea 6 del tomo 1.
SetExpressionValue("ContadorT6",0);	SetExpressionValue("ContadorT6",0);	Cambia por cero el valor de la expresión “ContadorT6” que esta oculta en la figura. La expresión “ContadorT6” es la que almacena la cantidad de veces que el estudiante utiliza el botón verificar en la tarea 6 del tomo 2 y en la tarea 7 del tomo 1.
SetExpressionValue("ContadorMT1",0);	SetExpressionValue("ContadorMT1",0);	Cambia por cero el valor de la expresión “ContadorMT1” que esta oculta en la figura. La expresión “ContadorMT1” es la que almacena la

		cantidad de ejercicios resueltos mal en la tarea 1 de los dos tomos.
SetExpressionValue("ContadorMT2",0);	SetExpressionValue("ContadorMT2",0);	Cambia por cero el valor de la expresión “ContadorMT2” que esta oculta en la figura. La expresión “ContadorMT2” es la que almacena la cantidad de ejercicios resueltos mal en la tarea 2 de los dos tomos.
	SetExpressionValue("ContadorMT3",0);	Cambia por cero el valor de la expresión “ContadorMT3” que esta oculta en la figura. La expresión “ContadorMT3” es la que almacena la cantidad de ejercicios resueltos mal en la tarea 3.del tomo 2.
SetExpressionValue("ContadorMT3Parte1",0);		Cambia por cero el valor de la expresión “ContadorMT3Parte1” que esta oculta en la figura. La expresión “ContadorMT3Parte1” es la que almacena la cantidad de ejercicios resueltos mal en la tarea 3 del tomo 1.
SetExpressionValue("ContadorMT3Parte2",0);		Cambia por cero el valor de la expresión “ContadorMT3Parte2” que esta oculta en la figura. La expresión “ContadorMT3Parte2” es la que almacena la cantidad de ejercicios resueltos mal en la tarea 4 del tomo 1.
SetExpressionValue("ContadorMT4",0);	SetExpressionValue("ContadorMT4",0);	Cambia por cero el valor de la expresión “ContadorMT4” que esta oculta en la figura. La

		expresión “ContadorMT4” es la que almacena la cantidad de ejercicios resueltos mal en la tarea 4 del tomo 2 y en la tarea 5 del tomo 1.
SetExpressionValue("ContadorMT5",0);	SetExpressionValue("ContadorMT5",0);	Cambia por cero el valor de la expresión “ContadorMT5” que esta oculta en la figura. La expresión “ContadorMT5” es la que almacena la cantidad de ejercicios resueltos mal en la tarea 5 del tomo 2 y en la tarea 6 del tomo 1.
SetExpressionValue("ContadorMT6",0);	SetExpressionValue("ContadorMT6",0);	Cambia por cero el valor de la expresión “ContadorMT6” que esta oculta en la figura. La expresión “ContadorMT6” es la que almacena la cantidad de ejercicios resueltos mal en la tarea 6 del tomo 2 y en la tarea 7 del tomo 1.

## VII. Ayuda

Este script se ejecuta con el botón “ayuda”. El botón ayuda solo aparece en el tomo 1 en las tareas de modificación del coeficiente de la variable de la ecuación de la recta.

Lo que internamente este script hace, es:

<b><u>Tomo 1: Conversión algebraica a la gráfica</u></b>	<b>Descripción</b>
t = getZC().JCM.CPs;	Este comando recupera en la variable t, todos los controles de la figura (botones, casillas, listas desplegadas, etc.). Los controles no son reconocidos como objetos de la figura en el script, razón por la que debe utilizarse este comando para posteriormente llamarlos y utilizarlos en él.

<pre>for(i=0; i&lt;t.size(); i++) {     if(t.get(i).O.getName().equals("ayuda"))         {             ayuda=t.get(i);             break;         } }</pre>	<p>Este bucle recorre la lista de los controles almacenados en la variable t, recuperando el botón ayuda, para poder ejecutar el script solo cuando se reconozca que éste esta visible en la figura.</p>
<pre>if(!ayuda.isHidden())</pre>	<p>Condicional que no permite la ejecución del script hasta que el botón “ayuda” este visible en la pantalla.</p>
<pre>{</pre>	
<pre>tarea=GetExpressionValue("Tarea")</pre>	<p>Este comando recupera y asigna a la variable tarea el valor que tiene la expresión “Tarea” en la figura. Según sea el caso de la actividad de conversión, los valores recuperados serán un número entero entre el 1 y el 7 o entre el 1 y el 6. En cada actividad los valores de la expresión corresponden con la secuencia de actividades que fue descrita en el diseño de cada tomo.</p>
<pre>switch (tarea) {     case (3) :         AyudaTarea3Parte1 ();         break;      case (4) :         AyudaTarea3Parte2 ();         break;</pre>	<p>Aquí, se ejecuta un condicional de casos posibles (switch case), con el que se evalúa el valor de la variable tarea. Cada caso (case()) tiene asociado una función que hemos denominado como AyudaTarea_ ();. En el momento en que un caso se ejecuta, el script busca el nombre de la función asociada y la ejecuta.</p>

```
case (5) :  
AyudaTarea4 ();  
break;  
}  
}
```

### Sobre las funciones AyudaTarea ()

Como ya lo señalamos, estas funciones se ejecutan a partir del condicional *switch case*. Las funciones muestran elementos que se han diseñado como ayuda para las dos partes experimentales y la parte de anticipación de las tareas de modificación del coeficiente de la variable de la ecuación. Cada tarea muestra elementos distintos, por lo que se propone mostrar una descripción detallada de cada una, así:

<b>function AyudaTarea3Parte1 ()</b>	
<b>Tomo I: tarea 3 parte 1</b>	<b>Descripción</b>
Show("OrdenadaM,TrianguloRectaAzul,TrianguloRectaRojaT3Parte1,Segmento1TrianguloRectaAzul,Segmento2TrianguloRectaAzul,Segmento1TrianguloRectaRojaT3Parte1,Segmento2TrianguloRectaRojaT3Parte1,Segmento3TrianguloRectaRojaT3Parte1,MedidaSegmento1TrianguloRectaRojaT3Parte1,MedidaSegmento2TrianguloRectaRojaT3Parte1");	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
Alert("Modifica el valor del numerador del término a de la ecuación, hasta hacer que los triángulos queden superpuestos e iguales");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.
}	

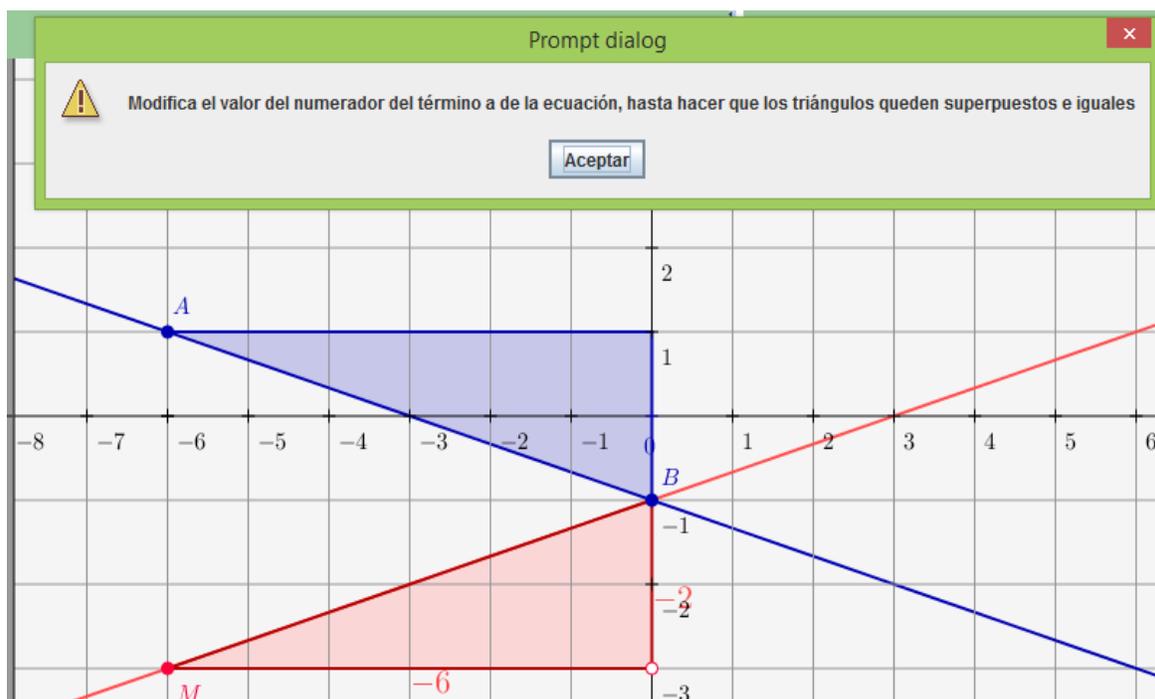


Imagen 22. determina el momento en el que la función AyudaTarea3Parte1 se ejecuta.

<b>function AyudaTarea3Parte2 ()</b>	
<b>Tomo I: tarea 3 parte 1</b>	<b>Descripción</b>
Show("OrdenadaG, TrianguloRectaAzul, TrianguloRectaRojaT3Parte2, Segmento1TrianguloRectaAzul, Segmento2TrianguloRectaAzul, Segmento1TrianguloRectaRojaT3Parte2, Segmento2TrianguloRectaRojaT3Parte2, Segmento3TrianguloRectaRojaT3Parte2, MedidaSegmento1TrianguloRectaRojaT3Parte2, MedidaSegmento2TrianguloRectaRojaT3Parte2");	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.
Alert("Modifica el valor del denominador del término a de la ecuación, hasta hacer que los triángulos queden superpuestos e iguales");	Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.

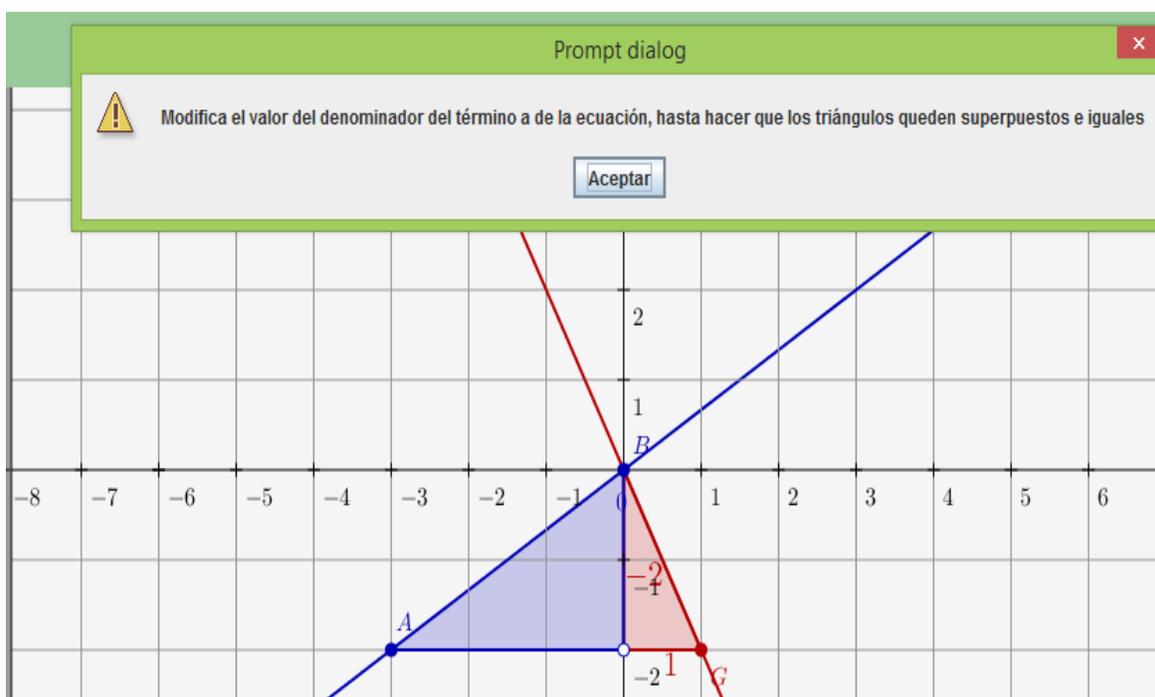


Imagen 23. determina el momento en el que la función AyudaTarea3Parte2 se ejecuta.

<b>function AyudaTarea4 ()</b>	
<b>Tomo I: tarea 3 parte 1</b>	<b>Descripción</b>
Show("OrdenadaH, TrianguloRectaAzul, TrianguloRectaRojaT4, Segmento1TrianguloRectaAzul, Segmento2TrianguloRectaAzul, Segmento3TrianguloRectaAzul, Segmento1TrianguloRectaRojaT4, Segmento2TrianguloRectaRojaT4, Segmento3TrianguloRectaRojaT4, MedidaSegmento1TrianguloRectaRojaT4, MedidaSegmento2TrianguloRectaRojaT4, MedidaSegmento3TrianguloRectaRojaT4");	Este comando muestra los objetos que en la figura se han denominado con los nombres que se muestran entre los paréntesis, separados con comas.

<pre>o1TrianguloRectaRojaT4,Segmento2TrianguloRectaRojaT4,Segmento3TrianguloRectaRojaT4,MedidaSegmento1TrianguloRectaRojaT4,MedidaSegmento2TrianguloRectaRojaT4");</pre>	<p>comas. Es importante señalar como se ha procurado asignárseles un sentido a esos nombres respecto a lo que representan. Ejemplo: TrianguloRectaRojaT3Parte1: refiere al triángulo de la recta roja de la parte1, tarea 3.</p>
<pre>Alert("Introduce en las casillas los valores del término a de la ecuación, para que los triángulos queden superpuestos e iguales");</pre>	<p>Se ejecuta el Alert; mensaje del paréntesis, que se mostrará como un cuadro de dialogo automático en la pantalla.</p>

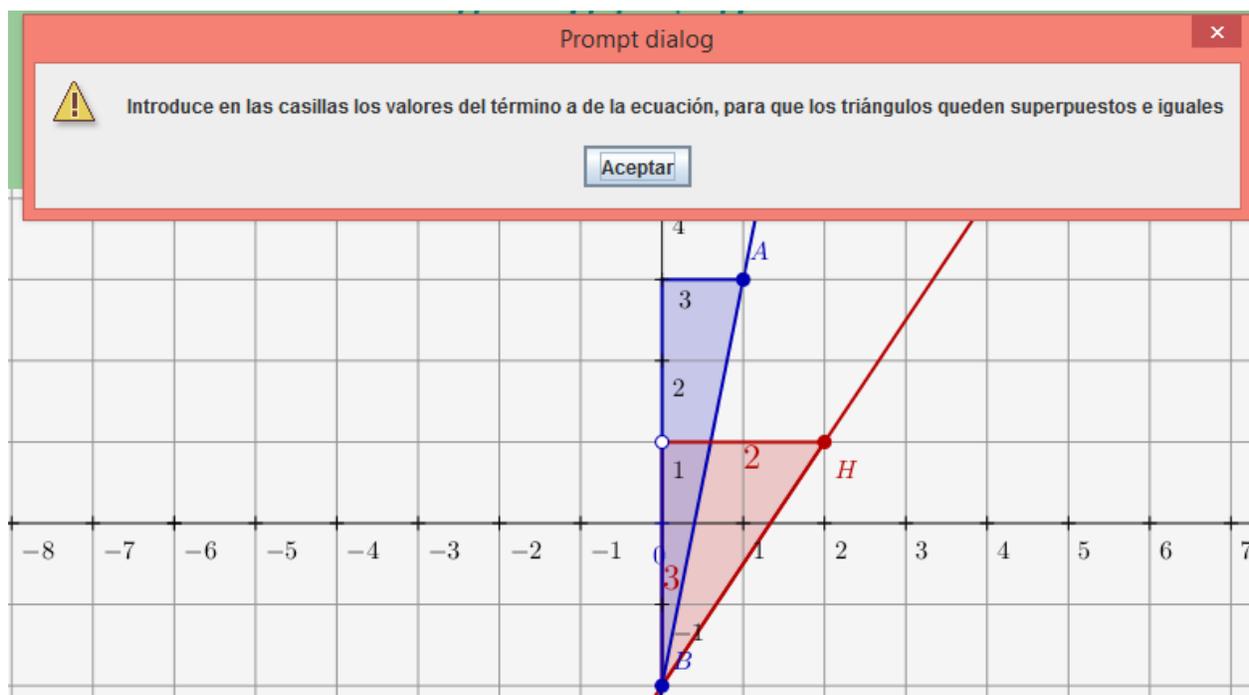


Imagen 24. Determina el momento en el que la función AyudaTarea4 se ejecuta.

### Configuración de la pantalla

En el diseño se implementó un plano independiente del zoom garantizando que las modificaciones de la ecuación de la recta pudieran ser observadas por los estudiantes en la gráfica y viceversa. Ese plano lo hemos denominado como plano flotante. Para la construcción del plano flotante se utilizaron las siguientes funciones del software:

- *Windoww*: Corresponde al ancho de la ventana.
- *Windowh*: Corresponde a la altura de la ventana.
- *Windowcx*: Corresponde a la abscisa del centro de la ventana.
- *Windowcy*: Corresponde a la ordenada del centro de la ventana.

Con estos comandos lo que primero se ubica es el punto “P2” centro de la franja blanca y origen de nuestro sistema de coordenadas. El tamaño de la franja blanca fue afectado por el ancho de la franja gris izquierda y el alto de la franja verde. Las franjas verde y gris mencionadas, fueron añadidas para ordenar lo que el estudiante verá en la pantalla<sup>3</sup> en todas las actividades. Las coordenadas entonces del punto “P2” tuvieron que ser modificadas por los tamaños de las franjas descritas.

Tras ubicar el punto “P2” se trazan rectas paralelas por éste al *eje x* y al *eje y* del sistema de coordenadas del software. Esas rectas paralelas mantienen la ortogonalidad del plano original en nuestro sistema de coordenadas. Teniendo los ejes de nuestro plano se restringen sus escalas a valores finitos y enteros; para lograrlo, definimos primero la distancia de P2 a los puntos P3 y P4. Estos puntos serán considerados como los puntos (0,1) y (1,0) de nuestro sistema coordinado. Posteriormente se emplea la herramienta simetría central para definir el resto de puntos sobre los ejes; y luego se nombran con un alias para que se vean en la pantalla con valores numéricos y no con sus nombres reales (P2, P3, P4, etc.). En CaRMetal cualquier objeto de la pantalla tiene un único nombre con el que es reconocido. Sin embargo, el software permite asignar un alias a cualquier objeto (ver imagen 25); este alias es la etiqueta que se verá del objeto en la pantalla, simulando su nombre. Los alias pueden repetirse para muchos objetos. Esta es entonces la forma como se le da nombre a todos los puntos que se han trazado sobre los ejes coordinados en la pantalla.

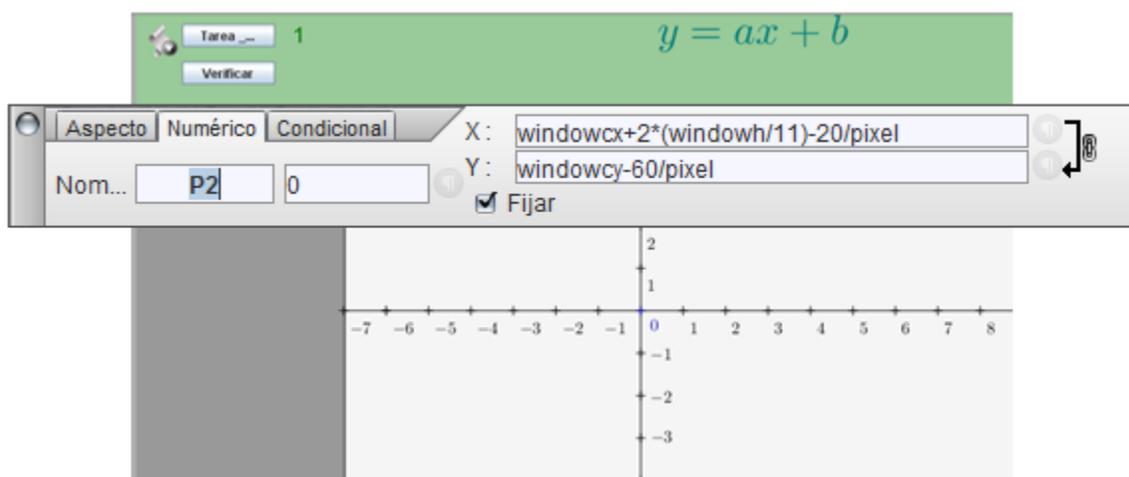


Imagen 25. alias que recibe el punto P2: origen del plano flotante, para que así se marque como nombre en la pantalla.

Teniendo el sistema de coordenadas finalizado, se crean rectas paralelas a los ejes que pasen por cada punto creado sobre éstos. De esta forma obtenemos la cuadrícula y nuestro plano que no se afecta con el zoom.

<sup>3</sup> Ver la descripción general del medio (figura de CaRMetal) que se hace en el apartado: Análisis a priori.