



Universidad Internacional de La Rioja
Facultad de Educación

Trabajo fin de máster

**El uso de herramientas de
programación gráfica en alumnos de
CFGM en sistemas
microinformáticos y redes**

Presentado por: Vicente Pulido Illana

Línea de investigación: Tecnologías de la Información y la
Comunicación (TIC)

Director/a: Esther García Tejedor

Ciudad: Olesa de Montserrat

Fecha: 16 de Enero de 2015

Resumen

El presente Proyecto Final de Máster pretende abordar la enseñanza de la programación gráfica en alumnos de grado medio de ciclos de informática. Internet nos brinda con herramientas de programación graficas,muy intuitivas, para que nuestros alumnos puedan desarrollar su creatividad, pero además se sientan motivados en lo que hacen. Estas herramientas nos proporcionan todos los ingredientes para que los educandos puedan aprender a programar jugando, sin necesidad de codificar, y siempre, con el objetivo de resolver todos los problemas que se les planteen.

No obstante, la idea se basa en los principios del constructivismo pedagógico, dónde los alumnos pueden construir sus propios proyectos, incluso modificando el de otros. Con todo esto, el alumno se siente que él es el protagonista en su proceso de aprendizaje, incentivando su autonomía y la necesidad de experimentar, con nuevos retos que se le van planteando.Nuestros alumnos necesitan realizar actividades en las que se sientan identificados, y que mejor que aprender divirtiéndose. Por ello, se presentará una propuesta práctica, en una escuela donde no se ha implantado ningún método de este tipo.

Palabras clave: herramienta gráfica, SCRATCH, TIC, constructivismo, aprendizaje significativo, creatividad.

Abstract

The following Master Final Project aims to address the teaching of programming in average degree of computing cycles. Internet provides us with graphical programming tools, very intuitive, so that our students can develop their creativity, but also be motivated in what they do. These tools provide us with all the ingredients for that learners can learn to program playing, without coding, and always with the aim of solving all the problems they pose.

However, the idea is based on the principles of constructivism teaching, where students can build their own projects, including modifying the other. With all this, the student feels it is the protagonist in the learning process, encouraging autonomy and the need to experiment with new challenges that are arising him. Our students need to perform activities in which they can identify, and what better to learn while having fun. Therefore, a practical proposal should be in this, in a school where they have not implemented any such method.

Key Words: graphical tool, SCRATCH, ICT, constructionism, meaningful learning.

ÍNDICE

1. Introducción del trabajo	4
1.1. Justificación del trabajo.....	4
1.2. Justificación del título.....	5
2. Planteamiento del problema	7
2.1. Objetivos	7
2.1.1. Objetivos generales.....	7
2.1.2. Objetivos específicos.....	7
2.2. Metodología	7
2.3. Justificación de la bibliografía utilizada.....	8
3. Marco teórico	10
3.1. La resolución de problemas en la Ley Orgánica de Educación.....	10
3.2. Las asignaturas de Tecnología e Informática en la ESO	11
3.3. Constructivismo y aprendizaje significativo	14
3.4. Herramientas de programación gráfica para ACNEE	17
3.5. La programación de computadores como habilidad para solucionar problemas	18
3.6. Solucionar problemas y programación gráfica.	20
3.7. Desarrollar la creatividad programando	23
3.8. Pensamiento algorítmico	25
3.9. Recursos web para aprender a programar	26
3.9.1. La herramienta SCRATCH	27
3.9.2. La herramienta Alice	29
3.9.3. Proyecto code.org	30
3.9.4. Ventajas y desventajas de herramientas de programación gráfica	31
4. Estudio de campo.....	34
4.1. Contexto de aplicación.....	34
4.2. Metodología y materiales	34
4.3 Resultados y análisis.....	35
4.3.1. Resultados de la encuesta realizada a los profesores.....	35
4.3.2. Resultados de la encuesta realizada a los alumnos	38
4.3.3. Resultados de las prácticas en clase	41
4.3.4. Interpretación de las encuestas realizadas a profesores	43
4.3.5. Interpretación de las encuestas realizadas a los alumnos	43

4.3.6. Interpretación global.....	44
5. Propuesta práctica	45
5.1 Introducción al pensamiento algorítmico	45
5.2. Diseñar antes de programar: el pseudocódigo y diagrama de flujo	47
5.3. Variables, constantes e identificadores	47
5.4. Estructuras básicas: secuencial, iterativa y condicional.....	49
6. Conclusiones.....	55
7. Limitaciones y líneas de investigación futuras	56
8. Bibliografía	57
8.1. Referencias bibliográficas	57
8.2. Bibliografía complementaria	59
9. Anexos.....	60

1. Introducción del trabajo

1.1. Justificación del trabajo

Los modelos educativos actuales no incentivan el uso de la creatividad en los educandos. Muchos centros educativos continúan instaurando un modelo educativo obsoleto, donde el profesor es un mero transmisor de contenidos y el alumno es el receptor pasivo. Según Ken Robinson (2009) en los tiempos actuales, donde la escuela sigue un modelo lineal, urge un cambio educativo en el que el profesor anime y fomente el desarrollo de la creatividad del alumno y no sólo se le enseñen cosas, y así cada uno pueda encontrar y desarrollar el talento que posea. Desarrollarlo dará un giro radical a nuestro entorno laboral, a nuestras relaciones y, en definitiva, a nuestras vidas.

Hoy en día la web 2.0 nos ofrecen recursos muy interesantes, donde los educandos pueden aprender metacognitivamente (aprenden a aprender), resolviendo problemas que se le plantean, encontrando ellos mismos la solución y/o los errores de otros compañeros, y no esperando que el profesor les ofrezca la respuesta.

El presente trabajo se centra en el proceso de enseñanza de entornos gráficos web 2.0 para que los educandos aprendan a programar sin necesidad de escribir complicadas líneas de código. No obstante, estas herramientas no solo nos permiten que los alumnos aprendan a programar. Los entornos gráficos de programación, que nos brinda la web 2.0, nos aportan otras ventajas muy interesantes que encajan con el paradigma constructivista. Como bien dice Sarah Macdonald (2011), las herramientas gráficas de programación para aprender a programar, como SCRATCH y otras tantas, sirven mucho más que para la programación de computadoras. Son herramientas que tratan de enseñar a los estudiantes a resolver sus propios problemas y logrando entender las cosas y descubrir la forma de resolverlas por sí mismos. Además incentivaremos en los mismos la autonomía, su crecimiento personal como estudiantes, como personas únicas e irrepetibles que son. Según Sarah Macdonald (2011), el papel del profesor está para facilitar el aprendizaje en el aula. Si nos centramos en ofrecerles las herramientas para descubrir y resolver problemas por sí mismos, los estudiantes estarán mucho más preparados para tomar las riendas de su propio aprendizaje. También cuando están “a cargo” están motivados para seguir descubriendo las distintas alternativas y soluciones.

Además no solo ofrecen acerca de la programación. Se trata de tener un objetivo, buscando soluciones, tratando con ellos, aprender de los errores y buscar nuevas maneras de resolver su problema individual. Si los estudiantes aprenden a hacer esto a través de estos recursos web 2.0, podrán aplicar estas habilidades en sus otras áreas del currículo, ya sea en inglés, arte, ciencias o

matemáticas. La resolución de problemas sin el foco en el docente le permitirá al estudiante crecer como estudiantes.

Como bien presenta la enseñanza constructivista, se trata de crear educandos independientes, pensadores autónomos. Son ellos los protagonistas de su propio proceso de aprendizaje. La Enseñanza constructivista se basa en la creencia de que el aprendizaje se produce cuando los alumnos participan activamente en un proceso de sentido y de la construcción del conocimiento en lugar de recibir pasivamente la información. Los estudiantes son los creadores del sentido y conocimiento. La enseñanza constructivista fomenta el pensamiento crítico y crea pensadores motivados e independientes. (Macdonald, 2011).

En definitiva, los educandos van a poder trabajar motivados, porque van a ser ellos, mediante la presentación de problemas, los que tendrán que solucionarlos, individualmente o de manera grupal. Pero lo importante, es que son ellos los que construyen su propia solución. El profesor les va a servir de guía en su proceso de aprendizaje, pero los educandos aprenderán por sí solos a encontrar la solución que se les plantea. Además, se sentirán completamente identificados con lo que están realizando. Uno de los mejores elementos de aprendizaje es que el alumno se divierta aprendiendo, elaborando sus propias historias o juegos.

1.2. Justificación del título

“El uso de herramientas de programación gráfica en alumnos de ciclos formativos de grado medio” es el título del presente trabajo. La justificación del título no tiene otra que demostrar que tenemos que dejar a los alumnos a crear sus propios proyectos, sus propias soluciones. Se ha demostrado que la mayoría de las personas no han estimulado la creatividad, porque se nos han explicado de forma mecánica, ofreciéndonos la solución más idónea o correcta, y como mucho, alguna que otra alternativa. Hemos aprendido de forma sistemática, lo que nos lleva siempre a encontrar una solución también sistemática. Con el título del presente trabajo, se pretende dar conocimiento de que el profesor debe ofrecer a sus educandos recursos educativos, acordes a la web 2.0. Estas herramientas gráficas permiten el trabajo colaborativo, ya que los alumnos pueden compartir sus proyectos individuales, para que otros puedan modificarlos. De esta manera, la creatividad, más arraigada en algunos alumnos, puede estimular la creatividad en otros, con una menor inteligencia creativa. La idea es que cada uno encuentre la solución que mejor crea o vea más conveniente, a un problema planteado por el profesor. Posteriormente, todos contribuirán a compartir su solución. Esto permitirá que cada uno de ellos pueda modificar la solución de otro,

aportando nuevas ideas o soluciones al problema planteado, además de ofrecer una respuesta al que no ha podido resolver el problema, o no lo ha hecho de la manera más eficiente.

El título del presente trabajo no especifica en absoluto una herramienta de programación gráfica en concreto. Esto es así, porque aunque la web 2.0 nos ofrece algunas herramientas con más éxito que otras, como puede ser el proyecto SCRATCH, desarrollado en el Media Lab del MIT (Instituto Tecnológico de Massachusetts), la web 2.0 nos ofrece muchas otras herramientas con la misma finalidad educativa, como puede ser el proyecto Code.org.

2. Planteamiento del problema

2.1. Objetivos

A continuación se presentan los objetivos que van a delimitar el presente trabajo de investigación.

2.1.1. Objetivos generales

El objetivo general es determinar qué grado de motivación y significación supone el uso de herramientas de programación gráfica como un recurso TIC para que los alumnos y alumnas de ciclos formativos de grado medio de informática aprendan a programar sin necesidad de escribir ninguna línea de código, que resulta complejo y arduo para su edad, y lo hagan divirtiéndose, además de aprender a resolver problemas que se le puedan plantear.

2.1.2. Objetivos específicos

Para la consecución del objetivo general, mencionado en el epígrafe anterior, se debería cumplir los siguientes objetivos específicos:

- Determinar el nivel de aceptación de las herramientas de programación gráfica en el profesorado y alumnado.
- Analizar el nivel de implicación y motivación del alumnado.
- Conocer las ventajas y desventajas que trae consigo la implementación de herramientas gráficas de programación.
- Analizar si su uso nos puede ayudar de base para mejorar a programar codificando.

2.2. Metodología

La metodología se ha basado en una investigación cuantitativa en su mayor grado, ya que el presente trabajo pretende explicar fenómenos a través de un enfoque de obtención de datos numéricos, mediante un instrumento de recolección de datos (encuesta). La interpretación de los datos se extraerá a partir de las encuestas. No obstante, también se realiza también una investigación cualitativa ya que se algunas de las conclusiones se desprenden de la mera observación de grupo participante objeto de estudio.

El proceso metodológico llevado a cabo en el presente trabajo ha seguido un proceso sistemático consistente en una recopilación de información o estudio bibliográfico sobre autores de relevancia y que tratan la temática actualmente. Es por esto, que la bibliografía es actual y de alto rigor científico. Posteriormente se ha realizado una propuesta práctica en el uso de herramientas de programación gráfica en alumnos de ciclos formativos de grado medio en sistemas microinformáticos y redes. No se han realizado entrevistas, pero sí se ha empleado la técnica de observación participante.

El presente trabajo de investigación ha seguido una metodología basada en las siguientes fases:

1ª. Desarrollo del marco teórico: se ha intentado enmarcar la competencia en la solución de problemas en la LOE. Seguidamente, se ha tratado las ideas del constructivismo y el aprendizaje significativo y la creatividad en relación al uso de estas herramientas. I sin obviar a los alumnos con necesidades educativas especiales, se ha expuesto como ejemplo las ventajas y desventajas de las herramientas de programación.

2ª. El estudio de campo: el estudio de campo ha consistido en realizar una encuesta basada en un conjunto de preguntas, tanto a profesores, como alumnos, para determinar si es una herramienta adecuada para enseñar programación a alumnos que no han realizado clases anteriormente, y si es motivadora y significativa para ellos, en el sentido de que se sientan identificados. También se pretende determinar si son buenas herramientas de transición y cuáles pueden ser más útiles, dependiendo del contenido específico que se quiera enseñar.

2ª. Propuesta práctica en el aula: en esta fase se desarrolla una propuesta práctica para alumnos de grado medio de sistemas microinformáticos. Se pretende desarrollar una metodología de enseñanza motivadora, para aprender a programar, jugando y divirtiéndose, mediante actividades grupales y posteriormente usando las herramientas de programación gráfica más idóneas para llevar a cabo en el ordenador lo que han aprendido en la actividad grupal.

2.3. Justificación de la bibliografía utilizada

La bibliografía se ha buscado en la Biblioteca Virtual de la UNIR, en bibliotecas locales y en la web, siempre buscando que sea lo más actual posible. Aunque de hecho, el tema tratado es de máxima actualidad.

La búsqueda de la información bibliográfica se ha basado en buscar primero las bases del constructivismo y su idea de aprendizaje significativo, ya que la metodología de enseñanza con las herramientas de programación, que se exponen en el presente trabajo, se basa en el trabajo grupal, el trabajo autónomo y aprender a aprender, mediante los errores y el trabajo de los compañeros.

Posteriormente se ha realizado un estudio bibliográfico más específico, centrándonos en el valor didáctico de las herramientas de programación que se trata en el trabajo, con lo que se ha tenido que considerar a uno de los creadores de una de las herramientas que se expone en el trabajo, como es el Dr. Resnick y su equipo de colaboradores, creadores de la plataforma SCRATCH.

Los alumnos con necesidades educativas especiales necesitan acceder y usar estas herramientas, pero muchos no pueden utilizar el ratón o el teclado convencional. Por ello se ha buscado información bibliográfica en que se realice un estudio sobre las aplicaciones que se pueden usar para salvar estos problemas. (Sánchez-Montoya, 2011) nos aporta una tabla con aplicaciones para salvar los problemas de ACNEE.

3. Marco teórico

En este epígrafe se abordará el marco teórico que nos permitirá construir los cimientos del presente trabajo de investigación. El marco teórico pretende consolidar y avalar el presente trabajo mediante estudios empíricos y teorías que tengan que ver con el tema en cuestión y que validen el presente trabajo.

No obstante, la consulta del marco teórico pretende fundamentar un tema muy delimitado y específico, como es el uso de herramientas de programación gráfica para resolver problemas, basándose en las ideas constructivistas y en el aprendizaje significativo. La resolución de problemas ya se contempla en la Ley Orgánica de Educación como habilidad o competencia inherente en el currículo (LOE 2006. Capítulo III. Artículo 23).

También se pretende que estas herramientas educativas, de enfoque constructivo, no deje de lado a los alumnos con necesidades educativas especiales (ACNEE), por lo que se dedicará un epígrafe a este colectivo.

Finalmente, se pretende que el uso de estas herramientas suponga, además de estimular el pensamiento computacional, estimular la creatividad, que puede servir de base para diferentes áreas de conocimiento, como puede ser las ciencias y las matemáticas.

3.1. La resolución de problemas en la Ley Orgánica de Educación

En la ley vigente sobre educación (LOE), concretamente, en su título preliminar y capítulo 3, se recoge entre varios elementos en el que se integra el currículo, el de las "competencias, o capacidades para aplicar de forma íntegra los contenidos propios de cada enseñanza y etapa educativa, con el fin de lograr la realización adecuada de actividades y la resolución eficaz de problemas complejos". (LOE 2006. Título preliminar. Capítulo III. 2.b).

Se pretende que el alumnado, tanto en educación primaria como en secundaria adquiera la competencia de resolver situaciones problemáticas para poder aplicarlas al mundo real.

No obstante, en la misma ley, en su capítulo III sobre educación secundaria, en su artículo 3, entre varios objetivos que se especifican para que el alumnado adquiera las capacidades de esta etapa, cabe destacar la que especifica el uso de las tecnologías de la información y la comunicación (en adelante TIC): "Desarrollar destrezas básicas en la utilización de las fuentes de información para,

con sentido crítico, adquirir nuevos conocimientos. Adquirir una preparación básica en el campo de las tecnologías, especialmente las de la información y la comunicación" (LOE 2006. Capítulo III. Artículo 23).

En el mismo capítulo, también se destaca entre varios de los objetivos que se especifican el de "Concebir el conocimiento científico como un saber integrado, que se estructura en distintas disciplinas, así como conocer y aplicar los métodos para identificar los problemas en los diversos campos del conocimiento y de la experiencia" (LOE 2006. Capítulo III. Artículo 23).

Cabe destacar este último objetivo, ya que se pretende que el alumnado adquiera la capacidad de solucionar problemas que se encuentre en la vida real, problemas que pueden abarcar muchas disciplinas y no problemas específicos de una única disciplina.

3.2. Las asignaturas de Tecnología e Informática en la ESO

Tal como se especifica en el currículo de Tecnologías de la ESO, la formación de los ciudadanos y la ciudadanas requieren actualmente una atención específica en la adquisición de conocimientos y habilidades, entre ellas, resolver problemas relacionados con las decisiones sobre el uso de objetos y procesos, con el objetivo de de aumentar su capacidad para actuar sobre su entorno y para comunicarse de manera eficaz en la sociedad del conocimiento actual.

Las TIC influyen positivamente en el rendimiento escolar si se hace un uso adecuado de las mismas y suministran al alumnado una herramienta que le permitan explorar todas las materias curriculares, explorar y simular situaciones nuevas que le ayuden a aprender a aprender.

Entre las competencias propias de la materia de tecnologías, cabe destacar el de tratamiento de la información y competencia digital. El tratamiento específico de las TIC, integrado en esta materia, proporciona una oportunidad especial para desarrollar esta competencia, asociada al desarrollo de contenidos que permiten localizar, procesar, elaborar, almacenar, presentar y difundir la información a través de las TIC (Decreto 143/2007, de 26 de Junio).

No obstante, la materia de tecnología colabora en el desarrollo de otras competencias básicas, como en el desarrollo de la competencia matemática, por medio del uso de herramientas matemáticas, además proporciona situaciones de aplicabilidad a diversos campos.

Uno de los campos que no se especifica explícitamente en el currículo de tecnología y informática, es el uso de herramientas gráficas de programación para resolver problemas sin necesidad de aprender programación mediante el uso de código en lenguajes de programación específicos. Las herramientas matemáticas, presentes en la materia, están relacionadas con la resolución de problemas prácticos del entorno: medida y cálculo de magnitudes básicas, el uso de escalas, la lectura y la interpretación de gráficos, etc. Pero se ha de decir, que estas herramientas pretenden resolver problemas específicos relacionados con las matemáticas. No obstante, la incorporación de herramientas gráficas de programación visual, en el currículo de tecnología y/o informática, pretende resolver problemas generales, al igual que lo hace la programación profesional. La finalidad de esta es pretender dar una respuesta a una necesidad que la sociedad le ha planteado. En concreto, la necesidad que transmite un cliente a una empresa que le va a ofrecer la solución, una necesidad que se transforma en un problema. La diferencia es que el resultado o la solución a esta necesidad es un programa o aplicación, un objeto intangible, pero que no deja de ser un resultado o solución a la necesidad o problema.

Los proyectos que crean productos u objetos tangibles, dando respuesta a una necesidad que la sociedad le plantea, lo hacen siguiendo una secuencia o ciclo de vida para solucionar el problema: obtención de requisitos, planificación, diseño, implementación, prueba del resultado o producto obtenido. El diseño de un programa, requiere de los mismos pasos o etapas para obtener el objeto. La diferencia, como se ha dicho antes, es que el resultado es intangible. Es importante destacar este hecho, ya que los alumnos de la ESO pueden adquirir esta competencia, pueden aprender a recoger requisitos, planificar y diseñar, antes de implementar la solución al problema planteado. La diferencia es que harán uso de herramientas más asequibles y comprensibles a su edad, con un carácter más motivador y en el que se van a sentir más identificados.

No obstante, las materias de tecnología e informática contribuyen también a la adquisición de competencias comunicativas, ya que ha de utilizar un lenguaje específico que ha de ser utilizado en los procesos de investigación, análisis, selección, síntesis y comunicación de la información. También contribuyen estas materias a la adquisición de la competencia de aprender a aprender, que se contribuye por medio del desarrollo de estrategias de resolución de problemas tecnológicos, y la competencia de autonomía e iniciativa personal, que implica abordar los problemas tecnológicos de manera reflexiva y plantear alternativas y soluciones de manera autónoma y creativa.

Entre los objetivos de la materia de tecnologías de educación secundaria cabe destacar los siguientes, y que especifican las capacidades que se tienen que adquirir, una vez el alumnado ha finalizado la etapa (Decreto 143/2007, de 26 de Junio):

5. Expresar y comunicar ideas y soluciones técnicas, razonando su viabilidad, y utilizando recursos gráficos e informáticos, la terminología y la simbología adecuadas.

6. Trabajar de forma autónoma, responsable y creativa en la toma de decisiones, en la ejecución de tareas y en la búsqueda de soluciones, todo mostrando una actitud dialogante y de respeto en el trabajo en equipo.

7. Utilizar los diferentes recursos que nos ofrecen las TIC e internet como herramientas de trabajo habituales así como gestionar, de forma correcta y con seguridad, la información, los sistemas operativos y los programas informáticos adecuados para la resolución de un problema concreto.

Con estos objetivos se quiere hacer hincapié en la importancia de que los alumnos y alumnas de la ESO adquieran la destreza de buscar soluciones a problemas planteados o resolver un problema concreto, haciendo uso de herramientas informáticas y programas informáticos, pero trabajando con cierta autonomía y siendo responsable en la toma de decisiones y en la búsqueda de soluciones.

Cabe destacar que, solo en el tercer curso de la ESO, se hace mención entre los contenidos de los proyectos tecnológicos, el de "identificación de problemas tecnológicos y de las fases del proceso de búsqueda de soluciones" y el del "uso de aplicaciones informáticas para la búsqueda de información, la resolución de problemas y la presentación de la memoria (Decreto 143/2007, de 26 de Junio).

No obstante, ni en el resto de cursos, ni en informática en cuarto de ESO (se presenta como una asignatura optativa) se menciona entre sus contenidos aquellos que tengan alguna relación con la identificación de problemas y la búsqueda de soluciones, o el uso de aplicaciones informáticas para resolver problemas.

El presente trabajo pretende, además de trabajar la habilidad de adquirir la competencia de resolver problemas, haciendo uso de herramientas de programación gráfica e interactivas, también pretende darle una importancia especial a esta habilidad, proponiendo la inclusión del uso de la programación con entornos de trabajo gráfico, entre los contenidos de tecnología e informática. No obstante, la inclusión de la programación en la asignatura de tecnología e informática, y en toda la etapa educativa, no sería un disparate, ya que sería muy útil que los alumnos y alumnas adquieran la destreza y la habilidad de resolver problemas mediante la programación gráfica, pudiendo aumentar la complejidad de los problemas, conforme se va avanzando entre los cursos que compone la etapa educativa.

Sin menospreciar lo dicho anteriormente, en informática de cuarto de ESO, que se presenta como una asignatura optativa, no menciona entre los contenidos nada que tenga relación con la programación informática, hecho que debería incorporar, ya que se considera una rama básica en la informática, además que puede servir para resolver problemas generales que se presenten en otras disciplinas que no tengan que ver con la informática.

La informática aplicada está en auge, y es posible que en un futuro próximo se demanden miles de bioinformáticos, una disciplina que trabaja la biología junto a la informática. Esta profesión va a requerir trabajar mano a mano con los biólogos y médicos, para poder ordenar la gran cantidad de datos que está generando la biomedicina. Esta ordenación sobre este conjunto de datos no se va poder llevar a cabo sino implementamos algoritmos eficientes y eficaces que resuelvan los problemas complejos a los que se van a tener que enfrentar los bioinformáticos. No podremos dar una respuesta acertada a este problema anterior, sino enseñamos a los niños y niñas, y adolescentes también, a que muchos de los problemas que se van a enfrentar se van a tener que resolver implementando una aplicación o programa. Un programa que se va a ejecutar en un sistema ubicado en una institución pública o empresa privada, y que va a ofrecer una solución al problema, un problema que ellos han generado como institución o empresa.

Como ejemplo, y volviendo a la bioinformática, es posible que muchas aplicaciones o programas informáticos resuelvan muchas enfermedades consideradas actualmente incurables, por la imposibilidad que los humanos puedan ordenar, manualmente, la enorme cantidad de datos que genera la genética o ciertas proteínas.

3.3. Constructivismo y aprendizaje significativo

Según Delval (1997) podemos encontrar en algunos pensadores como Darwin o Kant elementos del constructivismo. Estos autores, junto a los actuales exponentes del constructivismo piensan que el ser humano ha desarrollado su propia cognición gracias a su capacidad para adquirir conocimientos y reflexionar sobre sí mismos. Esta capacidad le ha permitido explicar, controlar y predecir la naturaleza de las cosas y construir su propia cultura.

Algunos autores basan sus estudios, sobre el constructivismo, en el entendimiento sobre cómo funciona y que contenido tiene la mente, como por ejemplo el constructivismo psicogenético de Piaget. Aunque para la mayoría, el foco de interés se centra en el desarrollo de habilidades de origen social, como argumenta Vigotsky en su constructivismo social.

Según Díaz, F. y Hernández, G. (2002) la manera que concibe el movimiento constructivista el aprendizaje escolar y la intervención educativa constituye el punto central de atención de diversos enfoques psicológicos a problemas como:

- El individuo desarrolla su psicología conforme va aprendiendo en la escuela.
- Podemos identificar y poner atención a las necesidades y motivaciones que tienen los alumnos en su proceso de enseñanza-aprendizaje.
- Replantear los contenidos curriculares hacia contenidos más motivadores y significativos para el alumnado.
- Reconocer que existen diversos tipos y modalidades de aprendizaje escolar, dando una atención más integrada a los componentes intelectuales, efectivos y sociales.
- Debemos de promover la interacción entre el docente y sus alumnos, así como entre los mismos alumnos, empleando el aprendizaje cooperativo.
- Valorar el papel del profesor, no sólo como un mero transmisor de conocimiento, sino como mediador del mismo, pero también de valores.

A pesar de los diversos enfoques entre los autores del constructivismo, la mayoría comparten "el principio de la importancia de la actividad mental constructiva del alumno en la realización de los aprendizajes escolares. Dicho principio explicativo básico es lo que Coll denomina la idea-fuerza más potente y también la más ampliamente compartida" (Díaz, F. et al, 2002).

La idea es que el aprendizaje escolar de los educandos se construye con su propio conocimiento a partir de los conocimientos y de las experiencias previas, pero también de la enseñanza que transmite el profesor, como mediador en este proceso de enseñanza-aprendizaje.

La concepción constructivista del aprendizaje escolar se sustenta en la idea de que la finalidad de la educación que se imparte en las instituciones es promover los procesos de crecimiento personal del alumno en el marco de la cultura del grupo a que pertenece. Estos aprendizajes no se producirán de manera satisfactoria a no ser que se suministre una ayuda específica mediante la participación del alumno en actividades intencionales, planificadas y sistemáticas, que logren propiciar en éste una actividad mental constructivista (Coll, 1988).

Desde la postura constructivista, el alumno no se considera un mero receptor de los saberes, tampoco se acepta que el alumno desarrolle su conocimiento por una simple acumulación de conocimientos específicos. La filosofía constructivista implica que las instituciones educativas deben permitir a los educandos construir su propia identidad personal en un contexto social y cultural determinado. Según Coll (1988), la finalidad de estas instituciones es que los alumnos

desarrollen la capacidad de realizar aprendizajes significativos por sí solo en una amplia gama de situaciones y circunstancias (aprender a aprender).

De acuerdo con Coll (1990), la concepción constructivista se organiza en torno a tres ideas fundamentales:

- El alumno es el responsable de su propio proceso de aprendizaje. Él es quien construye (o más bien reconstruye) los saberes de su grupo cultural, y éste puede ser un sujeto activo cuando manipula, explora, descubre o inventa, incluso cuando lee o escucha la exposición de los otros.
- El alumnado no tiene en todo momento que descubrir o inventar en un sentido literal todo el conocimiento escolar. Debido a que el conocimiento que se enseña en las instituciones escolares es en realidad el resultado de un proceso de construcción a nivel social, los alumnos y profesores encontrarán ya elaborados y definidos una buena parte de los contenidos curriculares.
- La función del profesor es orientar y guiar explícitamente la actividad mental constructiva de sus alumnos.

Podemos decir que los alumnos construyen su conocimiento mediante un proceso de elaboración, dónde el alumno selecciona, organiza y transforma la información que recibe de muy diversas fuentes, estableciendo relaciones con dicha información y sus ideas o conocimientos previos.

No obstante, las instituciones educativas tienen que promover prácticas a sus educandos que tengan cierta relevancia cultural, que sean prácticas auténticas (cotidianas, significativas, que tengan relevancia en su cultura). No podemos trabajar con los alumnos ofreciéndoles prácticas fuera de su contexto cultural y social, porque entrarían en contradicción con la vida real a la que se van a enfrentar. También, las instituciones tienen que promover ciertos valores, si queremos que se considere una educación de calidad: enseñar el autodidactismo, la capacidad de pensamiento crítico, la motivación y la responsabilidad por el estudio, disposición para aprender significativamente y por cooperar con el grupo, la atención a la diversidad de los alumnos que aprenden, etc.

Por poner un ejemplo, de prácticas auténticas, podíamos mencionar las prácticas o formación en centros de trabajo que realizan los alumnos en los Ciclos Formativos de Grado Medio y Superior. Este hecho debería también implantarse en la ESO. Es importante que el alumnado sienta que lo que está aprendiendo tiene una relevancia cultural, es un aprendizaje que lo va a poder poner en práctica en el mundo real. Tenemos que procurar que el alumno siempre se sienta dentro del

contexto en el que está aprendiendo. Muchos alumnos de Ciclos realizan prácticas en centros de trabajo (empresas privadas e instituciones escolares en su mayoría) donde ponen en práctica ya no solo conocimientos que han adquirido, sino prácticas a veces descontextualizadas. Es en estos centros de trabajo donde van a poder relacionar muchas ideas inconexas que tenían y donde van a encontrar un significado, a sus conocimientos, más auténtico, más real. Los mismos, podrán solucionar problemas reales. Podemos decir que las prácticas en empresa, supone prácticas diarias contextualizadas y reales, y que les van a ayudar como conector de ideas y/o conocimientos inconexos que tuvieran.

3.4. Herramientas de programación gráfica para ACNEE

Los alumnos con necesidades educativas especiales requieren del uso de recursos especiales para aprender a programar. Es por esto que (López y Sánchez, 2012) reflexionan sobre este tema para que las herramientas de programación sean accesibles para todos. En este caso se trabaja con SCRATCH, aunque se puede extrapolar a muy diversas herramientas con el mismo propósito. Según López et al. (2012) el trabajo con SCRATCH ofrecerá a los estudiantes con NEE plantear dudas, trabajar para resolver dudas, se igualarán a otros en el error, aprendiendo que los errores tienen sentido y que sirven para aprender, y que se pueden corregir. Todo esto les permitirá un aprendizaje activo y significativo.

Según López et al. (2012) la interfaz de SCRATCH no estaba diseñada para realizar un aprendizaje universal (en adelante DUA). Es por ellos que estos autores proponen es tecnología asistida para salvar este problema, ya que los ratones, teclado o pantalla no son accesibles a las necesidades de estos alumnos. Estos autores proponen la tabla de Rampas Digitales o tecnología asistida presentada en la VII Jornadas de Cooperación Educativa con Iberoamericana sobre Educación Especial e Inclusión Educativa, en Montevideo (Sánchez-Montoya, 2011):

Producto	Windows	GNU/ Linux
Adaptar teclado y ratón	Escritorio, Opciones de accesibilidad, <i>Emuclit</i> ¹	Escritorio de Gnome,
Webcam, el ojo que todo lo ve	HeadDev ² , CameraMouse ⁴ , HeadMouse ⁵	FacialMouse ³ ,
Joystick sustituye al ratón	MouseJoystick ⁶	
Reconocimiento de voz	Dragon Dictate ⁷	
Teclado sustituye al ratón	MouseKeys ⁸	
Leer con los oídos	Jaws ⁹ , NVDA ¹⁰	Orca ¹¹

¹ Gratuito : www.antoniosacco.com.ar/soft/emuclic16_inst.exe

² Gratuito : www.integraciondiscapacidades.org/index.php?m=Descargas&top=descargar&did=3

³ Gratuito : www.capacidad.es/FacialMouse-v2.2

⁴ Gratuito : www.cameramouse.org/downloads/CameraMouse2010Installer.exe

⁵ Gratuito : <http://robotica.udl.es/headmouse/headmouse2/download/HeadMouse2Installer.exe>

⁶ Gratuito : www.xtec.cat/~jlagares/download/mousejoystick.zip

⁷ Más información : <http://www.nuance.com/naturallyspeaking/products/default.asp>

⁸ Gratuito: www.microsoft.com/spain/accesibilidad/training/windowsxp/mousekeys.aspx

⁹ Demo: <ftp://ftp.freedomscientific.com/users/hj/private/WebFiles/JAWS/J11.0.756-32bit.exe>

Tabla 1. Ejemplo de tecnología asistida. Fuente (Sánchez-Montoya, 2011)

3.5. La programación de computadores como habilidad para solucionar problemas

Sentimos la necesidad de que los alumnos adquieran la destreza de solucionar problemas. Es por esto que, se han de elaborar estrategias efectivas para ayudar a que los estudiantes desarrollen dicha estrategia. Como una buena alternativa para atender esta necesidad, contamos con la programación visual, pero siempre y cuando la enfoquemos como logro para adquirir la destreza de solucionar problemas. Es importante por ello, no enseñar a los educandos de educación primaria y secundaria la programación que se enseña en los niveles superiores, sino más bien entornos más amigables como Alice o SCRATCH, Micro Mundos, KPL o proyectos como Code.org. Según Stager (2003) la programación gráfica posibilita no solo activar una amplia variedad de estilos de aprendizaje sino desarrollar el pensamiento algorítmico: decidir sobre la naturaleza del problema, seleccionar una representación que ayude a resolverlo y aprender a resolverlos junto a sus estrategias.

Otra habilidad que se puede desarrollar, además del pensamiento algorítmico, es la creatividad. La creatividad, forma parte de las prioridades en los sistemas educativos de varios países, entre ellos Inglaterra y EE.UU, que buscan promover en la educación el desarrollo de habilidades de pensamiento que conduzcan la formación de personas orientadas a la creatividad y a la innovación.

Hoy en día, la sociedad de la información ha reemplazado a la sociedad industrial, demandando a su vez a talento y creatividad. Cada vez es mayor el número de empresas que fundamentan su

modelo de negocio en la creatividad y la innovación. Además de los conocimientos para ejercer el cargo, se busca innovación y creatividad. Pero es importante que estas habilidades se inicien desde una edad temprana, por esto, las instituciones educativas deben de asumir su parte de responsabilidad en su nueva tarea.

Según López, J.C. (2009) la programación gráfica también puede colaborar a una mejor comprensión de ciertos contenidos en las matemáticas y ciencias naturales:

- Concepto de variable y de función (entrada de parámetros, procesamiento y resultado final al igual que las funciones algorítmicas).
- Ecuaciones y representación gráfica.
- Realizar proyectos cuyo resultado final represente resultados obtenidos con la experimentación en ciencias naturales:
 - Representar una explicación de la función de una célula.
 - Exposición de órganos y otros sistemas del cuerpo humano.
 - Clasificación de los seres vivos en plantas, animales, microorganismos.
 - Descripción y comparación de diversos tipos de neuronas.
 - Función de componentes de un circuito eléctrico.
 - Descripción de los principales elementos del sistema solar.

En anexos (anexo A) se adjunta un problema de ciencias naturales que se puede desarrollar con una herramienta de programación gráfica.

Tanto en las matemáticas como en las ciencias naturales, los estudiantes tienen que solucionar el problema pensando en dos pasos: primero, plasmar el problema en papel y luego se automatiza en el computador, y en su caso, una previa comprensión del tema.

No obstante, dicho lo anterior, algunos autores afirman que es posible resolver problemas de carácter general en muchas áreas mediante las siguientes estrategias: ensayo y error, iluminación, heurística, algoritmos, análisis de medios y fines, razonamiento analógico, pensamiento lateral. Estos autores también afirman que los profesores deben ayudar a los estudiantes a desarrollar habilidades para solucionar problemas López, J.C. (2009).

Según López, J.C. (2009) entre estas habilidades destacamos las siguientes:

- Presentar diversas estrategias de solución de problemas.
- Asignar problemas que tengan una aplicación en la vida diaria.
- Ofrecer experiencias que estimulen al estudiante en la investigación.
- Permitir que los estudiantes adopten la iniciativa al plantear preguntas.
- Plantear problemas contextualizados para que aprendan conceptos y habilidades.
- Utilizar representaciones visuales que mejoren la comprensión de conceptos, como los diagramas de flujo.
- Ofrecer una retroalimentación por su esfuerzo.
- Incentivar el trabajo colaborativo estudiante-estudiante y docente-estudiante.
- Realizar actividades que permitan a los estudiantes formular preguntas y llegar a conclusiones o defender ideas.

3.6. Solucionar problemas y programación gráfica.

La programación gráfica es una potente herramienta para solucionar problemas ya que nos brinda de habilidades como encontrar diferentes caminos para solucionar el mismo problema, anticipar errores y evaluar diferentes escenarios antes de tomar una decisión. Por ejemplo, aplicaciones como Logo, Micro mundos o Catch, entre otros, nos permite explorar y ver que ejecutará cada instrucción que hemos creado para solucionar un problema planteado.

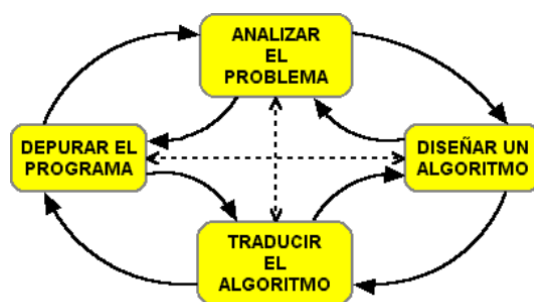


Ilustración 1. Fases para elaborar un programa de computador. Fuente: López, J.C. (2009)

Como podemos apreciar en la ilustración2 los estudiantes aprenden a dibujar un rectángulo, pero lo pueden extrapolar a cualquier problema general ya que tratan las dimensiones de la figura como variables, no como constantes. En el mundo real es más lógico encontrarnos con problemas donde

los objetos que intervienen varíen de valor y no sean valores constantes. Además obliga al estudiante a definir un lenguaje concreto que solo entiende el ordenador, construyendo conocimiento intuitivo acerca de la definición de esta figura geométrica.

EJEMPLO

Pedir a los estudiantes que escriban un procedimiento para dibujar un rectángulo con unas medidas determinadas (Lado1= 80; Lado2=120), implica que ellos deben pensar en algo muy parecido a lo siguiente (y escribirlo):

MicroMundos	Scratch
<p>para rectángulo</p> <p>cp</p> <p>adelante 80</p> <p>derecha 90</p> <p>adelante 120</p> <p>derecha 90</p> <p>adelante 80</p> <p>derecha 90</p> <p>adelante 120</p> <p>Fin</p>	 <p>al presionar </p> <p>bajar lápiz</p> <p>mover 80 pasos</p> <p>girar 90 grados</p> <p>mover 120 pasos</p> <p>girar 90 grados</p> <p>mover 80 pasos</p> <p>girar 90 grados</p> <p>mover 120 pasos</p>

Ilustración 2. Procedimiento para escribir un rectángulo en Micro mundos y SCRATCH. Fuente: López, J.C. (2009).

Además, la programación de computadores permite a los estudiantes sobre la naturaleza del problema, seleccionar una representación que le ayude a resolverlo, y coger la costumbre de resolver un problema con rigor y sistemáticamente.

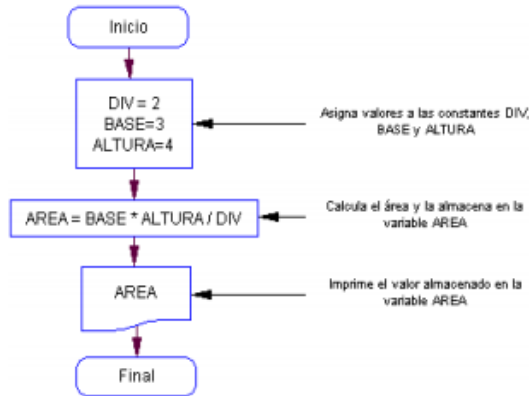
Numerosos autores de libros de programación, coinciden que para solucionar un problema general, como uno específico (un problema matemático), y que concuerdan con las operaciones mentales para resolver un problema, se han de plantear cuatro fases:

Analizar el problema (entender el problema): se trata de formular el problema, especificar los resultados que se desean obtener, identificar la información disponible, determinar las restricciones y definir los procesos necesarios para convertir los datos disponibles en resultados.

Diseñar un algoritmo (trazar un plan): mediante un diagrama de flujo (representación gráfica mediante símbolos geométricos) o mediante una secuencia de pasos en pseudocódigo.

ALGORITMO EN SEUDOCÓDIGO

Paso 1: Inicio
 Paso 2: Asignar el número 2 a la constante "div"
 Paso 3: Asignar el número 3 a la constante "base"
 Paso 4: Asignar el número 4 a la constante "altura"
 Paso 5: Guardar en la variable "área" el resultado de $\text{base} \times \text{altura} / \text{div}$
 Paso 6: Imprimir el valor de la variable "área"
 Paso 7: Final



ALGORITMO EN DIAGRAMA DE FLUJO

Ilustración 3. Diseño de algoritmo con pseudocódigo y diagrama de flujo. Fuente: López, J.C. (2009).

Traducir el algoritmo a un lenguaje de programación (ejecutar el plan): cada lenguaje posee sus propias reglas gramaticales, por lo que es importante que los estudiantes dominen la sintaxis del lenguaje de programación.

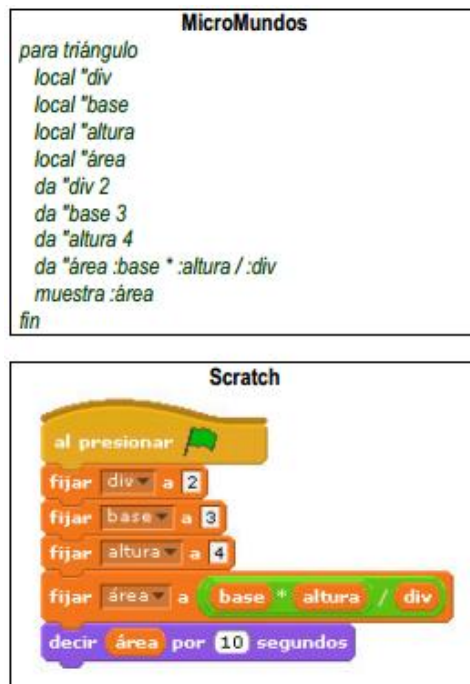


Ilustración 4. Traducción o codificación del algoritmo de la figura 3. Fuente: López, J.C. (2009)

Depurar el programa (revisar): probar y validar los resultados del programa. La depuración contribuye a mejorar la capacidad de los estudiantes para resolver problemas, estimula en los estudiantes la curiosidad, la perspectiva, la comunicación y promueve valores como la responsabilidad, fortaleza, laboriosidad, paciencia y perseverancia. La programación facilita un diálogo interior en el cual la retroalimentación constante y el éxito gradual empujan a los alumnos a ir más allá de sus expectativas (Stager, 2003).

3.7. Desarrollar la creatividad programando

Como bien dice Stenberg(1997), la creatividad no es solo una capacidad, sino un proceso en el que intervienen tres tipos de inteligencia: creativa (ir más allá de lo dado y engendrar ideas nuevas e interesantes), analítica (analizar y evaluar ideas, resolver problemas y tomar decisiones) y práctica (traducir teorías abstractas en realizaciones efectivas. Por su parte, Gardner (1993) define a la persona creativa como alguien que resuelve problemas y genera productos en un nuevo dominio que al final será aceptado por un grupo.

Según el Comité Consultivo Nacional para la Educación Creativa y Cultural de Inglaterra (NACCCE, por su sigla en Inglés), la creatividad se define como la actividad imaginativa que tiene como objetivo producir resultados tanto originales como generadores de valor (Robinson, 1999).

Son muchas las definiciones que intentan explicar el concepto de creatividad. Lo que se pretende con este epígrafe es intentar ver que el pensamiento algorítmico y la creatividad van cogidos de la mano. En el mundo actual de continuo cambio, los alumnos tienen que estar preparados para enfrentarse a improvisaciones y a resolverlas de manera creativa aportando diferentes alternativas a la solución del problema. Lo mismo que el pensamiento creativo lo podemos ir desarrollando desde edades tempranas, con la creatividad pasa lo mismo. Según algunos autores, se puede enseñar en edades tempranas, técnicas de pensamiento, que facilitan el desarrollo de la creatividad. Estas técnicas son multidisciplinarias (las podemos implementar en múltiples asignaturas): plantear problemas inesperados, formular diferentes alternativas a los problemas planteados, proponer e implementar diseños, utilizar metáforas y analogías, etc.

Las técnicas más apropiadas, entre las citadas anteriormente, para implementar un curso de programación gráfica pueden ser: planteamiento de problemas y formulación de alternativas. El planteamiento de problemas. El planteamiento de problemas inesperados complementa el enfoque de solución de problemas, pero tiene el inconveniente, que los alumnos no sepan salir de la metodología para resolverlos, reduciendo su creatividad.

En este sentido, entornos de programación como SCRATCH y Micro Mundos, comprometen a los estudiantes en la búsqueda de soluciones innovadoras a problemas inesperados; no se trata solamente de aprender a solucionar problemas de manera predefinida, sino de estar preparado para definir nuevas soluciones a medida que los problemas se presentan (Resnick, 2007).

Desde el punto de vista de la creatividad, formular alternativas no se trata de formular la mejor alternativa o la más satisfactoria, se trata de que los alumnos formulen el mayor número de alternativas.

Resnick(2007) propone utilizar la Espiral del Pensamiento Creativo para el desarrollo de la creatividad en los alumnos. En esta espiral, los estudiantes primero imaginan lo que quieren hacer; crean un proyecto basado en sus ideas; juegan con sus ideas y creaciones; comparten sus ideas y creaciones con otros y reflexionan sobre sus experiencias. Estas reflexiones les llevan a imaginar nuevas ideas y nuevos proyectos. La espiral, vuelve al punto de partida, generando un proceso indefinido de mejora continua.



Ilustración 5. Espiral del pensamiento creativo diseñada por el Dr. Resnick. Fuente: López, J.C. (2009)

En un principio, el proceso anterior lo debe planear y dirigir el docente, pero a medida que los estudiantes se habitúan al método y lo interiorizan, pueden trabajar la espiral de manera más autónoma. No obstante, es muy importante que los docentes creen ambientes idóneos para que los estudiantes puedan trabajar las distintas fases de la espiral de la creatividad. El docente ha de valorar más la cantidad de alternativas que generan los estudiantes a un problema, que las respuestas correctas. Por tanto, en la retroalimentación se ha de evitar los juicios de valor negativos. También es importante que en las diversas fases de la espiral se planteen problemas inesperados, solicitando a los estudiantes que aporten alternativas a un problema.

No obstante los docentes deben de promover otros valores para promover la sociabilización del alumnado: escuchar activamente a los compañeros y compañeras, reconocer puntos de vista diferentes y compararlos con los propios, reconocer y aceptar el escepticismo de los demás compañeros, se responsables con las funciones asignadas, respetar los seres vivos y objetos presentes en el entorno.

3.8. Pensamiento algorítmico

No parece difícil explicarles a los alumnos que el pensamiento algorítmico está por todas partes, en sus vidas cotidianas. Si le decimos que cuando se quiere lavar la ropa en una lavadora o cuando se van a lavar los dientes, están realizando un procedimiento realizable (acaba en una cantidad finita de pasos), comprensible (debe ser claro lo que hace) y preciso (el orden de estos pasos deben de estar perfectamente indicados) le podemos decir que estos procedimientos lo pueden determinar mediante un algoritmo, ya que pueden describir los pasos uno a uno para conseguir el resultado que se espera: ropa lavada y dientes lavados.

Describir los pasos para lavarse los dientes no parece muy complejo, pero nos podemos encontrar con problemas más complejos, en los que necesitemos descomponerlos en problemas más sencillos o por funciones, en los que tengamos que repetir ciertos pasos un número determinado de veces o mientras no se cumpla una condición, y además, nos tengamos que servir de estructuras de datos para organizarlos y/o ordenarlos. Estos problemas pueden ser transferibles a otros ambientes diferentes a la programación. En definitiva, la programación gráfica con computadores aporta a los estudiantes un laboratorio que les permite desarrollar habilidades indispensables en la actual sociedad de la información del S.XXI.

Es muy importante que los alumnos desarrollen el pensamiento algorítmico, ya que les ayudará a solucionar diferentes situaciones problemáticas pero que requieren la misma técnica o algoritmo para solucionar el problema.

Por poner un ejemplo, si nos fijamos en la mayoría de sitios webs relacionados con la búsqueda de la rutas para viajar, y que nos permiten buscar la ruta más corta o la más económica, tienen detrás el mismo algoritmo, ya que la idea siempre es la misma: explicar con un conjunto de pasos el camino más corto, desde un punto A hasta B, cuando tenemos diferentes caminos para llegar al punto B; o ir por la ruta más rápida o económica desde A hasta B. En resumen, cada web implementa de una manera diferente el mismo problema.

3.9. Recursos web para aprender a programar

Una vez más, Internet nos brinda con numerosos recursos web para que los niños, y no tan niños, aprendan a programar de una manera significativa. La filosofía de estas plataformas web es siempre la misma: que los niños y adolescentes puedan aprender el lenguaje algorítmico, desarrollen el pensamiento algorítmico, construyendo sus propios proyectos, jugando con los mismos y compartiéndolos. Las aplicaciones de programación gráfica son útiles para una amplia variedad de edades: desde los 7 o 8 años hasta los 14-16 años. Pero también pueden servir como base para alumnos de CFGM que se quieren iniciar a la programación. La idea de que los alumnos de primaria, ESO y CFGM se inicien en la programación con estas herramientas y no con los complicados lenguajes de programación, tiene más de una finalidad:

- Que los alumnos no vean la programación como algo complicado sino como algo motivador y divertido para ellos.
- Reducir la alta tasa de abandonos de alumnos que se inician en la programación.
- Promover un aprendizaje significativo: los alumnos se sienten más identificados con estas herramientas que con los rudos lenguajes de programación.
- Llevar la programación a edades más tempranas para que los educandos aprendan a resolver problemas e incentivar su creatividad.

Son muchas las herramientas de programación gráficas que podemos encontrar por la web. En este apartado se especificarán un reducido conjunto de ellas, aunque sí son las más aceptadas por la comunidad y las que más éxito están teniendo.

No se va a hacer un análisis exhaustivo de cada una de las herramientas, ya que está fuera del alcance del presente proyecto. No obstante, se explican de manera resumida cada una de ellas, para que el lector se haga una pequeña idea de las funcionalidades, semejanzas y diferencias que podemos encontrar entre las mismas.

3.9.1. La herramienta SCRATCH

SCRATCH es un entorno de programación, de código abierto, desarrollado por un grupo de investigadores del Instituto Tecnológico de Massachusetts (MIT), una de las Universidades más importantes del mundo, bajo la dirección del Dr. Mitchel Resnick. SCRATCH hace que la programación sea más divertida para todo aquel que se enfrente por primera vez a aprender a programar.

Según sus creadores, fue diseñado como medio de expresión para ayudar a niños y jóvenes a expresar sus ideas de forma creativa, inventando sus propias historias, animaciones, música, juegos y más; al tiempo que desarrollan habilidades de pensamiento lógico y de aprendizaje del Siglo XXI.

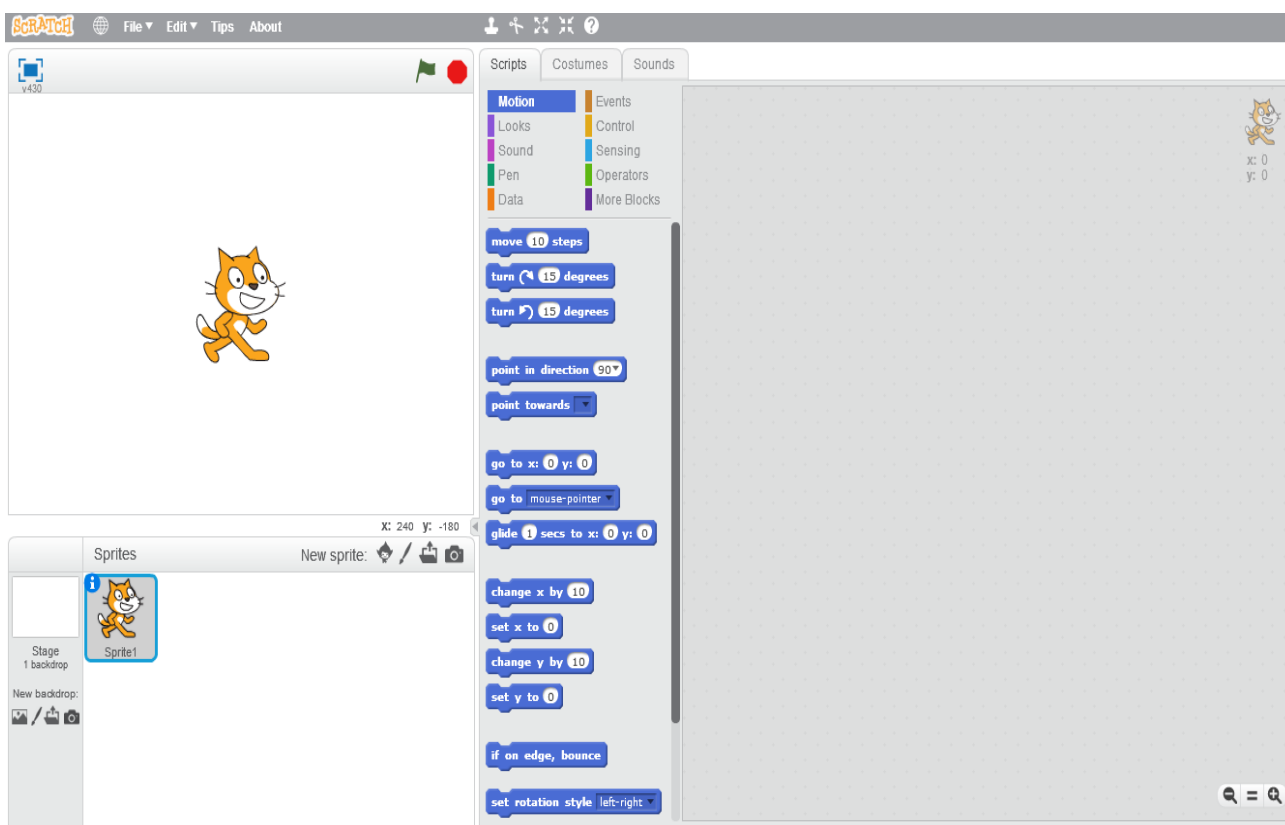


Ilustración 6. Entorno de trabajo SCRATCH versión 2. Fuente: Elaboración propia.

Es muy importante que la enseñanza y el aprendizaje de tecnologías informáticas se inicien a edades muy tempranas; especialmente en los niños en los primeros años de la escuela primaria. Esto les dará una base muy importante para seguir aprendiendo al mismo tiempo que la tecnología va avanzando.

SCRATCH, como el resto de herramientas que se explican en el presente trabajo, proporcionan a estos estudiantes conceptos básicos sobre la programación, explicados de una manera clara y sencilla, siendo al mismo tiempo divertida para los niños.

SCRATCH es un entorno de trabajo que cuenta con su propio lenguaje de programación. Lo interesante, es que este lenguaje no cuenta con instrucciones escritas de código, hecho que desmotiva a estos alumnos, además de su cierta complejidad. SCRATCH, por el contrario, cuenta con bloques que representan instrucciones, los cuales se arrastran desde una paleta de bloques hasta el área de información del objeto. Dichos bloques de instrucciones se pueden unir formando una secuencia de instrucciones de diferentes bloques, los cuales al iniciar la animación se ejecutan en forma secuencial. Estos bloques tienen instrucciones predeterminadas que al ser seleccionadas y arrastradas en forma secuencial, de arriba hacia abajo, harán que el objeto o carácter actúe, se mueva o suene como tú quieras.

SCRATCH tiene su sitio en internet, donde se puede obtener información adicional del proyecto, y donde permite descargar el programa para ejecutarlo en local sin necesidad de estar conectado a internet. El sitio web te permite crear una cuenta para poder compartir proyectos, así como poder ver los proyectos que hacen otros usuarios. Además de crear programas interactivos, SCRATCH tiene la opción de recibir respuestas del medio externo a través de dispositivos de tipo robótico que funcionan como sensores (leen datos analógicos que se envían por un puerto determinado para que pueda ser leído por el programa).



PicoBoard: Dispositivo provisto de sensores que permite a los proyectos de SCRATCH, responder a la luz, sonido y pushbuttons.



LEGO® WeDo™. Combina motores y sensores con tus animaciones en SCRATCH.

Ilustración 7. Dispositivos externos que interactúan con CATCH. Fuente: López, J.C. (2009)

3.9.2. La herramienta Alice

Alice es un entorno de programación en 3D innovador que nos permite crear y contar una historia, jugando un juego interactivo, o un video para compartir en la web. Alice es una herramienta de enseñanza libre diseñada para que los estudiantes se inicien en la programación orientada a objetos (en adelante POO). Alice permite que los estudiantes aprendan conceptos fundamentales de programación, en el contexto de creación de películas animadas y videojuegos simples. En este entorno, objetos 3D, como pueden ser personas, animales y vehículos) son incorporados en un entorno virtual, donde los estudiantes crean un programa para animar los objetos anteriores.

En la interfaz interactiva de Alice, los estudiantes arrastran y sueltan, en una zona de trabajo, bloques de instrucciones gráficas para crear un programa (al igual que SCRATCH), donde las instrucciones corresponden a declaraciones estándar en un lenguaje de POO, como Java, C++ y C#. Alice permite a los estudiantes ver de inmediato, al ejecutar sus programas, la relación entre las declaraciones de programación y el comportamiento de sus objetos animados. Mediante la manipulación de los objetos en su mundo virtual, los estudiantes adquieren experiencia con todas las construcciones programadas por ellos.



Ilustración 8. Entorno de trabajo educacional en 3D. Alice versión 3.1.93. Fuente: Elaboración propia

Al igual que SCRATCH, el entorno de programación Alice se debería enseñar a edades muy tempranas, ya que sirve de base para que los niños se inicien en el pensamiento algorítmico, pero a la vez motivador y que les aporta una enseñanza significativa. Los estudiantes se sienten identificados con lo que hacen, creando, jugando y compartiendo. No obstante, Alice es una buena plataforma que va a servir como tema de introducción a la programación en las asignaturas de tecnología e informática en la ESO, así como en CFGM de las ramas de informática.

Alice, al igual que SCRATCH, cuenta con un proyecto en internet, donde se puede obtener tutoriales para los profesores y alumnos, y donde permite descargar el programa para ejecutarlo en local para múltiples plataformas. Si bien, los requisitos de memoria son mayores en Alice que en SCRATCH.

3.9.3. Proyecto code.org

El proyecto code.org es una herramienta web educativa creada con el propósito de enseñar a estudiantes de todas las edades y cualquier persona que sienta interés, sobre la programación de computadoras. La organización está impulsada por grandes programadores como Mark Zuckerberg o Drew Houston, que apoyan que la programación sea parte esencial de la educación actual, ya que se considera que programar es la correcta forma de pensar; pensar el lenguaje algorítmico es pensar para solucionar problemas.

La justificación de los grandes programadores de las industrias tecnológicas, es la falta de conocimiento que hay sobre el tema, en la educación primaria (que es la edad en la que se podía ya enseñar a programar con esta herramienta) y la enorme demanda y plazas vacantes que pueden quedar libres en el futuro, provocando tal escasez de programadores que podría frenar la producción de tecnología, lo cual afectaría el negocio.

Esta organización sin ánimo de lucro persigue ampliar la educación de las Ciencias de la Computación a través de la incorporación de esta disciplina en el currículo de los estudiantes. Ya en la LOE, se impone la necesidad de incorporar el uso de las TIC, no solo en las asignaturas de informática y tecnología, sino en todo el currículo educativo de los educandos. El proyecto code.org pretende que los estudiantes en cada escuela tengan la oportunidad de aprender el lenguaje de programación, pretende conseguir que la informática sea parte del plan de estudios en la educación, junto con otras asignaturas como las matemáticas, la biología, la física, la química y álgebra.

El proyecto code.org, a diferencia de otros proyectos como SCRATCH o Alice es un proyecto totalmente en línea, y que a diferencia de estos anteriores, presenta vídeos tutoriales en subtítulos en diferentes idiomas, cuatro cursos elementales para principiantes (para edades de más de 4 años, más de 6 años, más de 8 años y más de 10 años respectivamente), la hora de código (los estudiantes pueden probar los conceptos básicos de la informática con personajes divertidos), un proyecto para crear tu propio juego en 10 minutos con un tutorial, e incluso un curso de introducción a la informática para la ESO (entre 15 a 25 horas).

3.9.4. Ventajas y desventajas de herramientas de programación gráfica

Podríamos nombrar un sinnúmero de ventajas inherentes en las herramientas de programación gráfica que se han expuesto en los epígrafes anteriores. En este apartado se va exponer las ventajas más significativas, matizando aquellas herramientas que aportan algunas funcionalidades o características que no encontramos en otras.

No obstante, estas herramientas llevan por sí mismas ciertas desventajas, que se han de referir y que se han de tener en cuenta. Además estas herramientas suponen una pequeña muestra de un número mayor de herramientas que encontramos en la web y que aportan un valor educativo semejante. Con esto se quiere hacer hincapié que las ventajas e inconvenientes aquí referidas no solo se refieren a las anteriores herramientas expuestas, sino a un conjunto de herramientas con semejante finalidad y objeto educativo.

Entre las numerosas ventajas cabe destacar las siguientes:

- Se ha comprobado que su interfaz es sencilla e intuitiva. Además son programas gratuitos de software libre; y están disponibles para varios sistemas operativos: Windows, Mac y Linux. I suelen ser la mayoría multilinguaje.
- Las herramientas de programación gráfica se consideran software educativo. Son herramientas que están diseñadas únicamente para enseñar a programar.
- Algunas herramientas, como Alice, ayudan a entender la POO y el modelo basado en el paso de mensajes y métodos, e influyen positivamente en la enseñanza de la POO,
- Son entornos colaborativos que permiten compartir proyectos, pudiendo ser descargados, y utilizados por otras personas para ser modificados. Los usuarios de estas aplicaciones

pueden compartir sus trabajos con compañeros de su clase y el resto de usuarios en general. Al compartirlos, otros pueden reelaborar los proyectos agregando nuevas funcionalidades y creatividad al proyecto. Este trabajo colaborativo hace que unos aprendan de otros, aportando ideas nuevas y creativas.

- Son plataformas idóneas para aprender e iniciarse en la programación. Los alumnos entienden muy bien las estructuras básicas de programación.
- Tanto los profesores como los educandos las consideran herramientas atractivas y motivadoras.
- Se pueden utilizar en otras áreas, como las matemáticas y ciencias naturales
- Los estudiantes se entusiasman con los retos de manera creativa.
- Evitan, en muchas ocasiones, el aprendizaje frustrado de los lenguajes de programación, ya que el aprendizaje es arduo y complejo.
- Los alumnos adquieren otras habilidades, como el aprendizaje autónomo y muestran mucho más interés, ya que se sienten identificados con lo que hacen.
- Estas herramientas pretenden que los estudiantes aprendan realizando juegos, dibujando, creando historias, animaciones. Supone una diversión para ellos, y encima, aprenden.

Aunque no son muchas las desventajas, a continuación se muestran las siguientes:

- En algunas plataformas, como SCRATCH, es necesario tener instalado Java para su uso.
- Algunas operaciones pueden resultar lentas, sobre todo si las ejecutamos en línea, aunque se ha comprobado que algunas, como SCRATCH y Alice, se pueden descargar en local.
- Cuando creamos aplicaciones más complejas, ocupan más memoria, lo que lleva a que debamos ejecutarlas en local y no en línea. Las aplicaciones, como SCRATCH y Alice permiten esta ventaja.

- Puede pasar que los alumnos creen que se programa de esta manera, con lo que el salto al aprendizaje en un lenguaje de programación puede resultar frustrante. En la presente investigación se ha encontrado que hay herramientas de aprendizaje que se encuentran entre las herramientas de programación gráfica y los lenguajes de programación, para que el salto de una a otra no sea muy pronunciado.

4. Estudio de campo

En el capítulo dedicado al marco teórico hemos mencionado cómo el uso de herramientas de programación aumenta la motivación, mejora la autonomía y fomenta la creatividad de los estudiantes, además que los prepara para un mercado laboral que demanda, cada vez más, profesionales en el área TIC, un mercado que requiere solucionar problemas de diferente envergadura.

A continuación se desarrolla un estudio de campo en un instituto de secundaria, concretamente en el instituto Daniel Blanxart. El objetivo del estudio de campo es analizar e investigar el uso de las herramientas de programación gráfica, como una metodología de enseñanza-aprendizaje novedosa y motivadora, para alumnos de grado medio de sistemas microinformáticos a los que les resulta complicado entender el lenguaje algorítmico.

4.1. Contexto de aplicación

La metodología que se propone pretende realizarse en un contexto donde se observa un brecha entre los profesores de enseñanzas en el área de informática y tecnología, que imparten enseñanzas academicistas y clásicas, y los alumnos de hoy en día, llamados “nativos digitales”, que se sienten lejanos a la metodología de enseñanza actual, desmotivándolos en la mayoría de los casos.

No obstante, el trabajo de campo se centra en un instituto de secundaria público, pero que imparte por las tardes ciclos formativos de grado medio en Administración y empresas, actividades deportivas y sistemas microinformáticos y redes. Concretamente, los alumnos de este último ciclo no han comenzado todavía ninguna asignatura de programación pura. Aunque sí se han introducido en un lenguaje específico para programar páginas web y se han encontrado con la dificultad que muchos tienen a la hora de empezar un lenguaje de programación con una metodología clásica. El presente trabajo de campo se centra en estos alumnos.

4.2. Metodología y materiales

La metodología se ha basado en recoger, primeramente, información bibliográfica para fundamentar el presente trabajo de investigación. Seguidamente se ha llevado a cabo la propuesta práctica, que se ha desarrollado en 7 sesiones de una hora, impartida por un profesor de CFGM de

sistemas microinformáticos del Instituto Daniel Blanxart. La propuesta práctica se basa en el aprendizaje de principios básicos de algoritmia, estructuras secuenciales, iterativas, condicionales, diseño de un algoritmo y variables. Se ha realizado una observación sobre la participación de los estudiantes y el profesor que ha colaborado en la investigación.

Finalmente se ha realizado la recolección de datos, mediante una encuesta dirigida a profesores de informática (4 participantes) y a 15 alumnos (todos chicos). El contenido de las encuestas, tanto de profesores y alumnos lo podemos encontrar en Anexos (Anexo B y C).

No obstante, los materiales que se emplearon para el presente estudio de investigación han sido: las herramientas de programación SCRATCH, plataforma code.org y Alice. Un aula de informática con 15 ordenadores y la plataforma MOODLE donde se pueden compartir recursos y pruebas.

Además, se contará con un instrumento de recolección de datos, como puede ser una encuesta, para posteriormente hacer una tabulación de los datos recogidos con la aplicación Excel y graficar los resultados obtenidos. Finalmente, se exponen los resultados y se realiza un análisis de las dos encuestas, para llegar a una interpretación individual y global.

4.3 Resultados y análisis

Los resultados más significativos obtenidos del presente trabajo de campo han sido los siguientes.

4.3.1. Resultados de la encuesta realizada a los profesores

En referencia a los resultados de la encuesta emitida a los profesores se han obtenido los siguientes resultados:

En la 1ª pregunta, refiriéndose a si las herramientas de programación gráfica eran adecuadas para aprender a programar, el 50 % piensa que son muy adecuadas, mientras que un 50 % cree que son adecuadas.

En la 2ª pregunta, se consideraba si estas herramientas son idóneas antes de empezar un curso de programación clásico. El 100 % de los profesores creen que sí son buenas herramientas transitorias.

En la 3ª pregunta, se consideraba si estas herramientas suponen un buen elemento constructivo y significativo para el proceso de aprendizaje. Con lo que un 75 % han respondido que mucho, mientras que un 25 % han respondido que bastante.

En referente a la 4ª pregunta, considerando si es importante la elección de un tipo de herramienta de programación según el contenido que se quiere enseñar, el 25 % piensa que son muy importantes, mientras que 75 % piensa que es importante.

En la 5ª pregunta, se formulaba si el uso de estas herramientas puede extrapolarse a las matemáticas o las ciencias naturales. La respuesta ha sido rotunda; un 100 % del profesorado cree que las herramientas de programación gráfica pueden enseñarse en estas materias.

En la 6ª pregunta, se formulaba si el uso de estas herramientas supone un elemento motivador para el alumnado. El 50 % piensa que su uso es muy motivador, mientras que 50 % piensa que es motivador.

Refiriéndonos a la 7ª pregunta, si el aprendizaje de estas herramientas es intuitivo o sencillo, o de lo contrario, difícil; el 100 % del profesorado afirma que son herramientas intuitivas y cuyo aprendizaje es sencillo.

En la 8ª pregunta, se preguntaba si la herramienta de programación gráfica Alice era de la más adecuada para aprender la base en POO. La respuesta ha sido que el 25 % piensa que es una herramienta muy adecuada, mientras que 75 % piensa que es bastante adecuada.

En la 9ª pregunta se pedía a los profesores que explicaran cómo fue su experiencia en el aula con el uso de alguna herramienta de programación gráfica. La recopilación de las respuestas se detalla a continuación:

- ❖ Le motivaron los retos a los alumnos
- ❖ Aun incrementando el nivel de dificultad les motivaron los retos
- ❖ Los alumnos entienden conceptos de programación difíciles de comprender
- ❖ Muy positiva
- ❖ Bastante satisfactoria
- ❖ La experiencia fue buena

En el siguiente gráfico de sectores se reflejan las opiniones de los profesores:

Experiencia en el aula con el uso de herramientas de programación gráfica

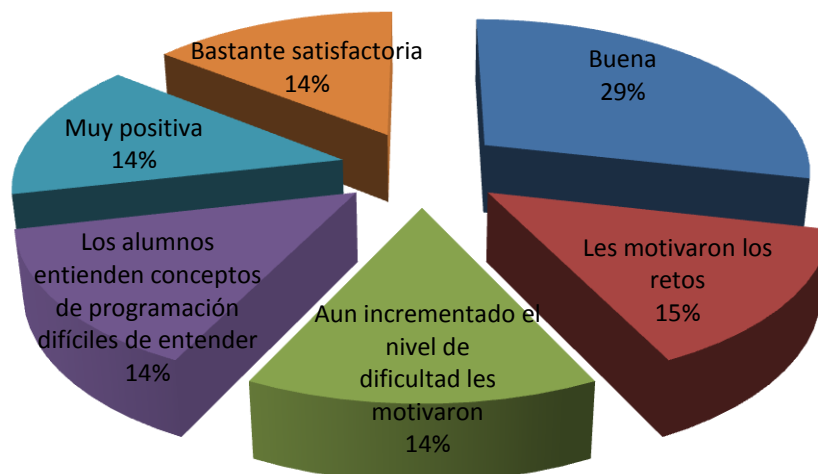


Gráfico 1. Opiniones sobre el uso de herramientas de programación gráfica. Fuente: Elaboración propia.

En la última pregunta, se solicitaba que se explicara qué herramienta ha utilizado para sus clases. La mayoría utilizaron SCRATCH, mientras que un profesor afirma que usó APPInventor y otro la plataforma LEGO mindstorms, con el propósito de tutorizar a un alumno que realizaba un proyecto. En el siguiente gráfico de sectores se reflejan las herramientas utilizadas por los profesores:

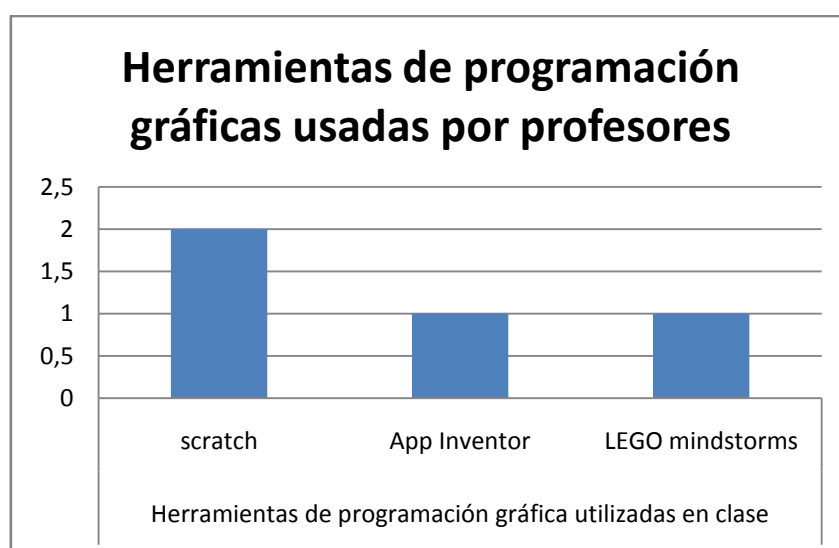


Gráfico 2. Herramientas de programación gráfica usadas en clase. Fuente: Elaboración propia.

4.3.2. Resultados de la encuesta realizada a los alumnos

En referencia a los resultados de la encuesta emitida a los alumnos se han obtenido los siguientes resultados:

En la 1ª pregunta, refiriéndose a si las herramientas de programación gráfica eran adecuadas para aprender a programar, un 26,7 % piensa que son muy adecuadas, mientras que un 73,3 % piensa que son adecuadas. Nadie cree que sean poco o nada adecuada.

En la 2ª pregunta, se consideraba si estas herramientas son idóneas antes de empezar un curso de programación clásico. El 80 % piensa que sí son buenas herramientas transitorias, mientras que un 20 no opinan.

En la 3ª pregunta, se consideraba si con estas herramientas se sienten más identificados que los métodos clásicos basados en la comprensión de un lenguaje de programación. Con lo que un 20 % han respondido que se sienten muy identificados, mientras que un 80 % piensa que bastante identificado. Nadie responde que no se sientan identificados con estas herramientas.

En referente a la 4ª pregunta, considerando si el uso de una herramienta de programación es un elemento motivador para su proceso de enseñanza-aprendizaje, el 13,3 % piensa que es muy motivador, mientras que 86.67 % piensa que es bastante motivador. Nadie cree que sea poco o nada motivador.

Refiriéndonos a la 5ª pregunta, si el aprendizaje de estas herramientas es intuitivo o sencillo, o de lo contrario, difícil; el 60 % considera intuitivo y sencillo el aprendizaje, mientras que 40 % cree que su aprendizaje no es difícil ni sencillo, lo considera en un término medio. Curiosamente, nadie cree que sea difícil.

En la 6ª pregunta, se preguntaba si habían recibido clases con alguna herramienta de programación. La respuesta ha sido que un 20 % afirma haber recibido muchas veces, mientras que un 33,3 % afirma que alguna vez. Finalmente, un 46,7 % afirma que alguna vez.

En la 7ª pregunta se pedía a los alumnos que explicaran si el uso de estas herramientas le había ayudado a entender las estructuras básicas de programación. La recopilación de las respuestas se detalla a continuación, junto a las personas que respondieron dichas respuestas:

- ❖ Muy útiles para entender las estructuras (4).
- ❖ Me ayudó bastante para entenderlas (3).
- ❖ Aprendes qué debe ir con qué para que la programación funcione (1).
- ❖ Me ha ayudado un poco (3).
- ❖ Es mucho más fácil entender la programación y sus estructuras (3).
- ❖ Me ayudó mucho AppInventor (parecido a SCRATCH) para entender la programación y sus estructuras (1).

En el siguiente gráfico de barras se reflejan las opiniones y número alumnos, de si fue de ayuda para entender las estructuras básicas de programación:

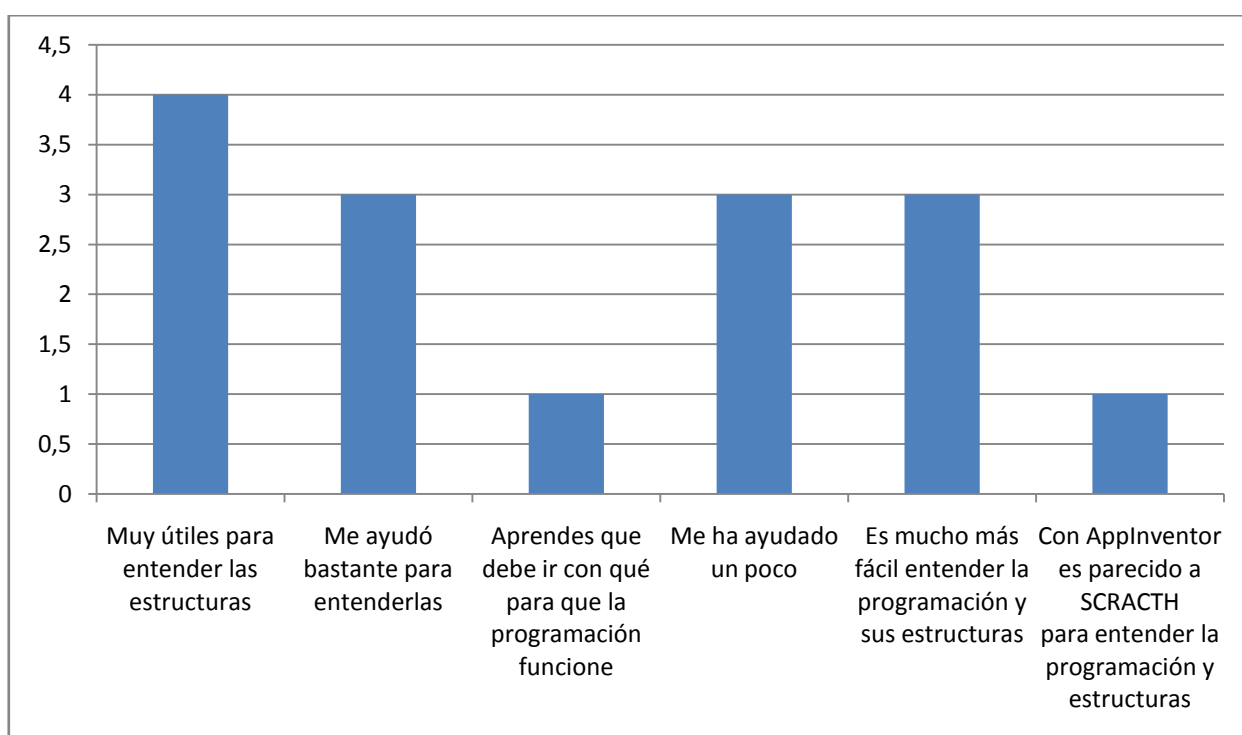


Gráfico 3. Opiniones alumnos sobre la comprensión de estructuras básicas. Fuente: Elaboración propia.

En la última pregunta, se solicitaba que se explicaran las ventajas y desventajas de estas herramientas. La recopilación de las ventajas junto a las personas que han respondido se muestra a continuación:

- ❖ De manera visual se entienden conceptos de programación (4).
- ❖ Intuitivas y prácticas para enseñar a programar (2).
- ❖ Sencillas y fáciles de usar (2).
- ❖ Aprendes practicando y no con teoría (1).
- ❖ Te entretienes más y aprendes más rápido (2).

En cuanto a las desventajas, estas han sido:

- ❖ Si lo utilizas sin saber nada no sabes hacerlo (1).
- ❖ Es poco potente y que esté solo disponible en inglés (1).
- ❖ Necesitas a alguien que te enseñe a utilizarlas (1).
- ❖ Tiene algunos bloques difíciles de entender (1).

En el siguiente gráfico de barras se reflejan las ventajas que expusieron los alumnos en el uso de las herramientas de programación gráfica:

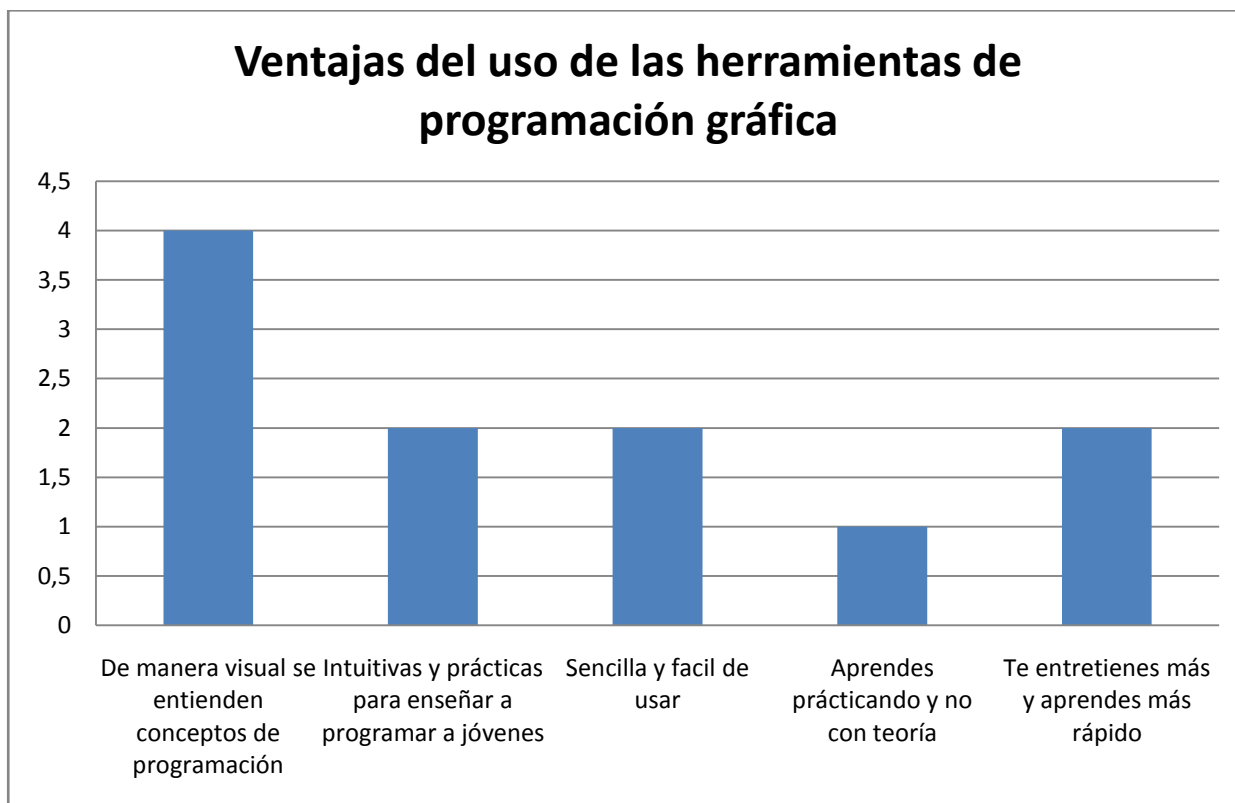


Gráfico 4. Ventajas del uso de las herramientas de programación gráfica. Fuente: Elaboración propia.

En el siguiente gráfico de barras se reflejan las desventajas que expusieron los alumnos en el uso de las herramientas de programación gráfica:

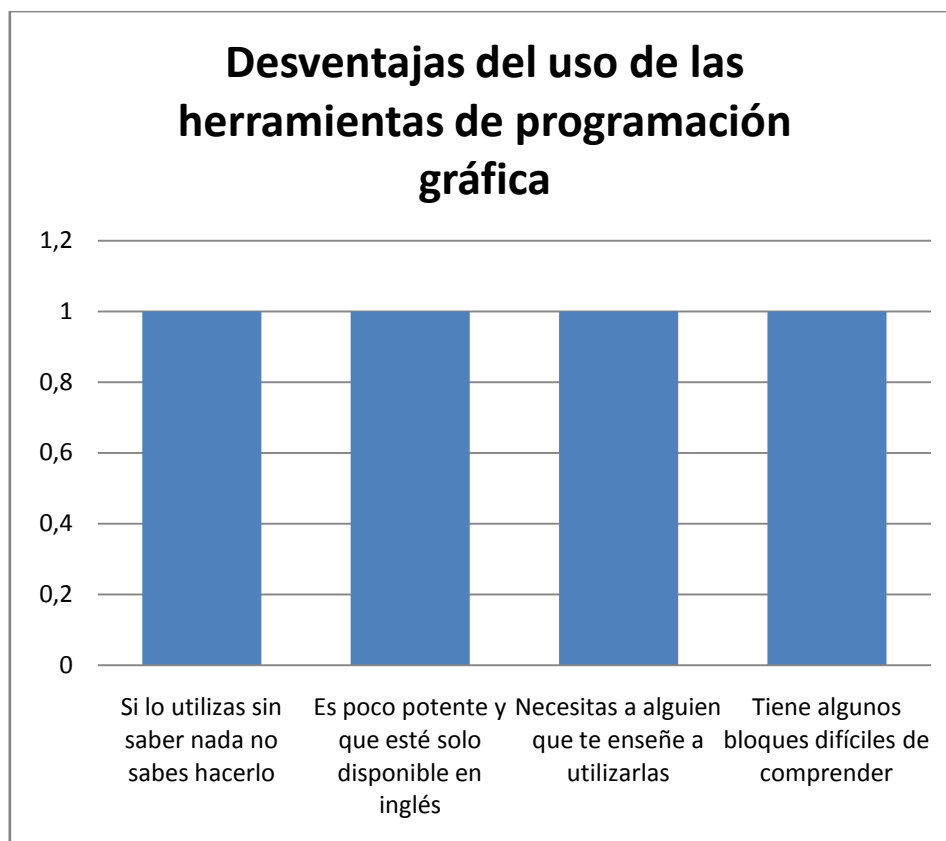


Gráfico 5.Desventajas del uso de las herramientas de programación gráfica. Fuente: Elaboración propia.

4.3.3. Resultados de las prácticas en clase

En el estudio de campo se incluye una propuesta práctica de taller, llevada a cabo por 15 alumnos del segundo curso de grado medio de sistemas microinformáticos y redes. Las actividades se realizaron en un aula taller con 15 ordenadores, ya que la clase de segundo está formada por 15 alumnos. El coordinador de informática, en horas de libre disposición, se ofreció a realizar las actividades con los alumnos. La temporización de las actividades queda reflejada en la siguiente tabla:

Sesión	Tema	Actividad
1ª sesión	El lenguaje preciso es el que entienden los ordenadores	Actividad 1 – Actividad con Google Actividad 2 – Actividad con code.org
2ª sesión	El algoritmo es un conjunto de pasos lógicos y ordenados	Actividad 1 – Ordenar una serie de pasos desordenados Actividad 2 – Elaborar una pajarita de papel trabajando colaborativamente
3ª sesión	Diseñar antes de programar	Actividad 1 – Diseñar en pseudocódigo y en diagrama de flujo cálculos de áreas y perímetros de figuras geométricas
4ª sesión	Variables, constantes e identificadores	Actividad 1 – Entender las variables y constantes: pegatinas y rotulador Actividad 2 – Trabajar las variables con SCRATCH calculando el área de un rectángulo
5ª sesión	Estructura básica secuencial	Actividad 1 – Dibujar figuras geométricas con SCRATCH
6ª sesión	Estructura básica iterativa	Actividad 1 – Suma de n números mediante una caja de cartón y papeles Actividad 2 – Suma de n números con SCRATCH Actividad 3 – Dibujar el cuadro de la sesión 1 Cuatro veces con menos instrucciones
7ª sesión	Estructura básica condicional	Actividad 1 – Jugar al guardia de seguridad Actividad 2 – Sistema de evaluación de una asignatura con SCRATCH

*Tabla 2.*Secuencia de actividades. Fuente: Elaboración propia.

La propuesta en práctica ha sido muy exitosa, ya que los alumnos se han sentido motivados en las actividades que han realizado. Muchas de estas actividades constaban de juegos grupales para que entendieran las estructuras, variables y secuencias. Una vez se realizaba la actividad grupal y los alumnos habían asimilado los conceptos, los completaban con las aplicaciones de programación como SCRATCH y la plataforma code.org

Los alumnos han trabajado en conjunto para entender los problemas que se les planteaban, para luego implementarlos con las herramientas. La asimilación de estas herramientas ha sido muy buena, ya que el profesor, mediante el proyector les presentaba ejemplos ya elaborados por el autor del presente trabajo.

4.3.4. Interpretación de las encuestas realizadas a profesores

De los resultados obtenidos de las encuestas que se realizaron a los profesores, se interpreta que son unas herramientas muy adecuadas para aprender a programar. De hecho nadie piensa que no lo son. Además todos los encuestados afirman que es necesario el aprendizaje con estas herramientas antes de empezar un curso de programación clásico, además de que es un elemento constructivo y significativo en el aprendizaje de su alumnado, y verdaderamente motivador. Curiosamente el 100 % de los encuestados afirman que también se pueden utilizar para el aprendizaje de las matemáticas y las ciencias naturales, siendo una herramienta intuitiva y sencilla de usar. En concreto, para aprender POO, la mayoría piensa que Alice es una herramienta bastante adecuada.

No obstante, su experiencia en el uso de herramientas de programación ha sido muy positiva y bastante satisfactoria, observando, en su mayoría, que los retos que se les planteaba a los alumnos les motivaban. Incluso si se le aumentaba la dificultad, también asumían los nuevos retos. Observan que los alumnos, usando estas herramientas, entienden conceptos de programación, difíciles de entender si se les enseña con metodologías clásicas, basadas en la comprensión de un lenguaje de programación.

Finalmente, la mitad usaron la herramienta SCRATCH para enseñar, solo dos profesores usaron otras herramientas, pero con la misma finalidad.

4.3.5. Interpretación de las encuestas realizadas a los alumnos

De los resultados obtenidos de las encuestas que se realizaron a los alumnos, se interpreta que son unas herramientas adecuadas para aprender a programar. Ninguno piensa que no lo son. Además, y en la misma opinión que los profesores, la mayoría afirma que es necesario el aprendizaje con estas herramientas antes de empezar un curso de programación clásico. Solo 3 alumnos no contestan. También afirman que se sienten bastante identificados con estas aplicaciones y que es

un elemento bastante motivador. De hecho ninguno afirma que no se siente. Ganan los que dicen que son herramientas intuitivas y sencillas, aunque un 40 % considera que no son difíciles de aprender, pero tampoco fáciles. No hay nadie que haya dicho que no ha utilizado alguna vez una herramienta de programación. Sus opiniones en cuanto al uso de estas herramientas fueron muy diversas. La mayoría opinan que son muy útiles y que les ayudó bastante a la hora de entender las estructuras básicas de programación. De hecho, algunos opinan que son fáciles de entender. Solo un 20 % opina que le ayudó un poco. El dato curioso, es que nadie dijo que no le ayudó.

En referencia a las ventajas y desventajas, la mayoría expuso muchas ventajas. Solo un 26 % formuló alguna desventaja. Muchos piensan que con la programación gráfica se entienden los conceptos de programación, siendo fácil y sencillo de usar, intuitivas y prácticas, ya que aprendes practicando.

4.3.6. Interpretación global

Tanto de las encuestas realizadas a los profesores como a los alumnos, incluyendo las actividades taller realizadas en clase, se llega a la conclusión de que la experiencia ha sido más positiva de lo que a priori se esperaba. Los profesores ven que sus alumnos se sienten muy identificados con el uso de estas herramientas, aprenden divirtiéndose, y se motivan en todo momento, ya que los retos son pensar en la solución del problema, implementarlo visualmente, y afrontar retos cada vez más complicados. Los alumnos observan que su trabajo anterior se puede realizar de manera más eficiente empleando otras estructuras más complejas. Tanto los profesores como los alumnos están de acuerdo que las herramientas de programación gráfica son adecuadas para aprender a programar y sirven de paso previo antes de adentrarse con otro tipo de metodología, como es la que se implanta en la mayoría de las universidades. No obstante, era de esperar que esta generación que ha nacido con la tecnología debajo de sus brazos, se identificara más con herramientas que emplean personajes de juegos, que ellos conocen. Además son conscientes que están aprendiendo algo útil y con una enorme salida laboral, ya que saben que, tanto para crear videojuegos como aplicaciones para móviles, se necesita aprender programación y entender las estructuras básicas, inherentes en todos los lenguajes de programación.

5. Propuesta práctica

5.1 Introducción al pensamiento algorítmico

Antes de abordar las estructuras básicas del lenguaje algorítmico (secuencial, decisión y repetitiva), los alumnos deben de aprender conceptos básicos de programación. Los contenidos son los siguientes:

1. El lenguaje preciso es el que entienden los computadores.

El objetivo es que los alumnos entiendan que los computadores necesitan un lenguaje preciso. Nosotros los humanos, si escribimos una frase y se nos olvida una letra o nos olvidamos de una tilde, vamos a poder entender el significado de la misma, pero las computadoras no. Para que los alumnos asimilen correctamente este contenido se proponen las siguientes tipos de actividades:

Primera actividad

Una actividad muy sencilla pero muy aclaradora es invitarles a los alumnos a que escriban palabras en el navegador (como <http://www.google.es>) con acento y verán que no consiguen ningún resultado. Verán que los programas no entienden lo mismo Google que Góogle, por eso no se le muestra el recurso web.

Segunda actividad

En esta segunda actividad, los alumnos son conscientes que los programas no entienden todo lo que le escribes. Lo mismo que no entienden todo lo que le escribes, tampoco entienden nuestras órdenes si no lo hacemos de manera correcta y en el orden correcto.

Una propuesta muy constructiva y significativa es el uso de la herramienta code.org. Cualquier actividad es buena para que entiendan que si no ejecutan las órdenes correctas y en el orden correcto no llegarán a cumplir su objetivo. No obstante, la presente actividad sirve como tema introductorio para el tema siguiente "El algoritmo es un conjunto de pasos lógicos".



Ilustración 9. Usando la herramienta code.org. El personaje no ha podido dibujar el cuadro porque no se han especificado las órdenes correctas. Fuente: Elaboración propia.

2. El algoritmo es un conjunto de pasos lógicos y ordenados

A diferencia de los seres humanos que realizan actividades sin detenerse a pensar en los pasos que deben de seguir, los computadores son muy ordenados y necesitan que el programador les especifique cada uno de los pasos necesarios y su orden lógico de ejecución.

Las actividades que se proponen son las siguientes:

Primera actividad

Según López, J.C. (2009) podemos listar una serie de pasos para realizar una tarea y presentarlos a los estudiantes de forma desordenada para que ellos los ordenen. En Anexos (Anexo D) podemos encontrar una actividad de este tipo.

Segunda actividad

El profesor traerá a clase un papel cuadriculado y propondrá a los estudiantes que realicen una pajarita de papel entre todos, adoptando cada uno un rol. Un alumno hará de procesador (realizará el paso concreto que se especifique en una hoja de papel que recibirá), otro hará de reloj bajando una banderilla en intervalos de tiempo determinados, un tercero hará de transmisor de la orden (en un papel llevará escrito el primer paso, luego el segundo, y así consecutivamente para que el procesador realice la pajarita). El alumno que hace de transmisor de la orden recibirá el papel de un cuarto alumno, que será el programa (el tiene todos los pasos o papeles donde se especifica

cómo se tiene que hacer una pajarita de papel). Esta actividad se basa en una actividad desarrollada por López, J.C. (2009), pero la creatividad es de elaboración propia.

Seguidamente, el profesor ordenará al alumno, que hace de programa, que entregue ahora los papeles, pero desordenados, para que entiendan que cuando cambiamos el orden de pasos lógico para realizar nuestra pajarita, no la podremos hacer ahora. Pero también a que la consecución del objetivo se consigue mediante la colaboración de diferentes elementos que trabajan sincronizados y con funcionalidades específicas

5.2. Diseñar antes de programar: el pseudocódigo y diagrama de flujo

El objetivo es que los alumnos entiendan que podemos expresar mediante proposiciones detalladas las instrucciones que llevan desde un estado inicial (problema) hasta la solución del problema. Las técnicas que utilizan formas gráficas para representar ideas e información ayudan a los estudiantes a clarificar su pensamiento, y a procesar, organizar y priorizar nueva información. Los diagramas de flujo revelan patrones, interrelaciones e interdependencias además de estimular el pensamiento creativo.

Primera actividad

Según López, J.C. (2009) el profesor puede exponerles a los estudiantes ejemplos de cómo elaborar un algoritmo para calcular el área de un rectángulo, tanto en pseudocódigo como en un diagrama de flujo. Posteriormente, les expondrá otros problemas, para que los realicen ellos: Hallar el perímetro de un cuadrado de lado n , hallar el área de un cuadrado de lado n , etc.

En Anexos (Anexo E) podemos encontrar la simbología que se le enseñará a los estudiantes para elaborar un diagrama de flujo.

5.3. Variables, constantes e identificadores

Los alumnos tienen que aprender que en los programas se utilizan datos que varían, pero también podemos utilizar datos constantes como el número π . Todos estos datos se guardan en identificadores.

Primera actividad

Antes de empezar a trabajar el uso de las variables con una herramienta de programación (en este caso concreto se recomienda SCRATCH), se puede trabajar en clase actividades muy interesantes para entender las variables, constantes e identificadores.

Por ejemplo, con formas que representen los identificadores, pegatinas amarillas que las pegaremos encima de las formas, pero que al ser pegatinas las podemos quitar y volver a poner otras (representan las variables), y formas en las que escribiremos en su superficie, y con rotulador, su valor (representan las constantes que siempre tienen el mismo valor). También podemos trabajar con variables que adoptan un valor temporal o transitorio. Una forma que representa el concepto temporal se utilizaría para este fin.

Segunda actividad

Ahora podemos trabajar con SCRATCH el uso de las variables. En la siguiente imagen se muestra como los alumnos pueden trabajar el uso de las variables calculando el área de un rectángulo. Al presionar el gato volverán a tener las variables a cero. Pueden trabajar con una variable temporal para que entiendan todos los pasos intermedios.

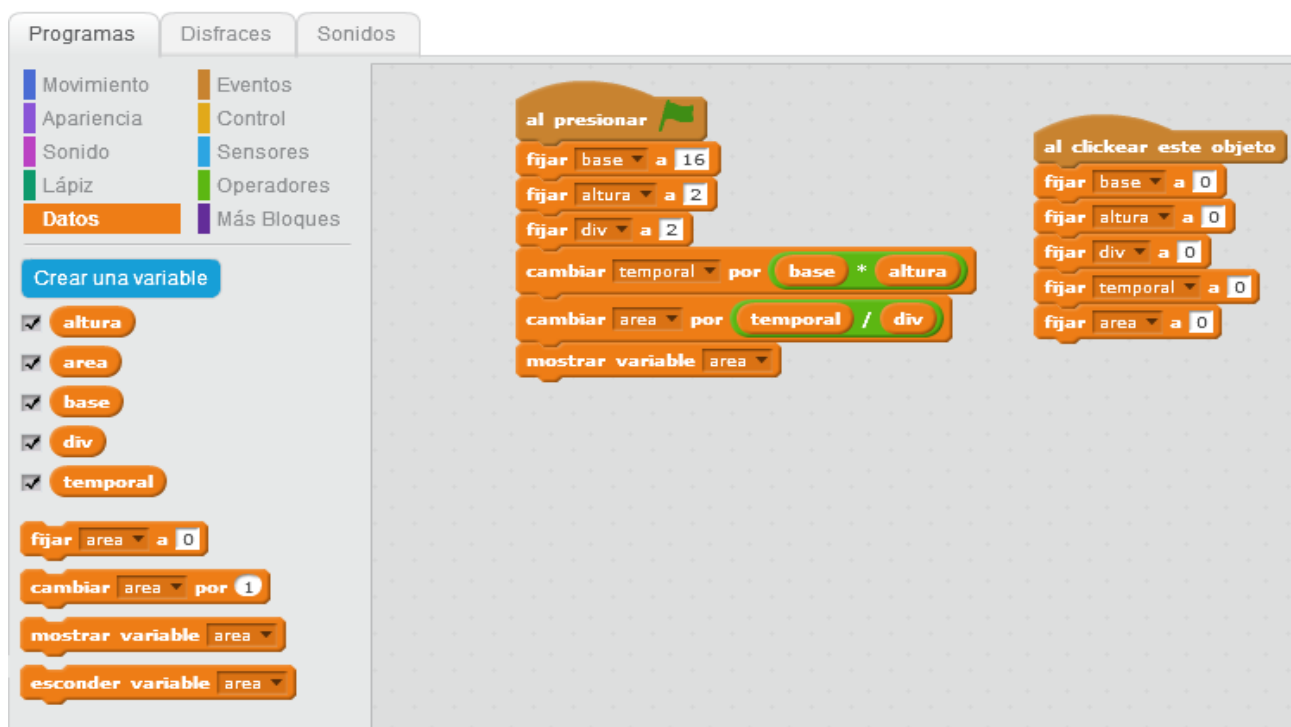


Ilustración 10. Captura de SCRATCH. Trabajando con variables para calcular el área de un rectángulo.
Fuente: Elaboración propia

5.4. Estructuras básicas: secuencial, iterativa y condicional

Los alumnos tienen que entender que la estructura de control más sencilla es la secuencial o estructura lineal. Son instrucciones que deben de ejecutarse de forma consecutiva, una detrás de otra, siguiendo una línea de flujo. Esta estructura básica convive con las estructuras iterativas y condicionales para formar programas más complejos.

1. Las estructuras secuenciales o lineales

Primera actividad

Actividad interesante y muy didáctica, que sirve de base para entender las estructuras que se abordarán después. Estas, no son más que estructuras que optimizan las secuenciales, haciendo el programa más eficiente, ya que los alumnos se darán cuenta que con menos instrucciones consiguen el mismo resultado, la misma solución. Trabajarán con SCRATCH, aunque también puede ser interesante trabajar con code.org, aunque más enfocado para alumnos de primaria. Según López, J.C. (2009), como actividad práctica pueden dibujar de manera secuencial: cuadros, triángulos y rectángulos o una combinación de las anteriores figuras.

El profesor, a su vez, puede prestarles una ayuda, que consiste en que les enseñará a que realicen una secuencia de instrucciones, que al presionar la barra espaciadora, el gato vuelva a su punto de origen y se borre todo lo que se ha dibujado hasta el momento. De esta manera, al alumno le resultará más fácil corregir los errores. Esta pequeña ayuda, motiva al alumno, ya que si ejecuta el programa más de una vez, el gato dibujará más cuadros, despistando al alumno el objetivo de su tarea (dibujar solo dos cuadros).

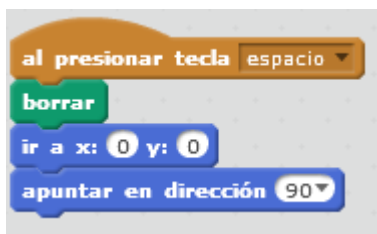



Ilustración 11. Captura de SCRATCH. Trabajando con estructuras secuenciales. Sección de instrucciones que trabaja en paralelo con las instrucciones que dibujan las figuras. Al presionar la barra de espacio se borra todo y se vuelve a empezar. Fuente: Elaboración propia.



The image shows the Scratch 2.0 interface. On the left, a cat sprite is on a stage with a blue rectangle. The 'Programas' palette is open, showing 'Movimiento' blocks. The 'al presionar' event block is selected, and the 'bajar lápiz' block is being dragged into the script area.

2. Las estructuras iterativas (instrucciones que se repiten)



Three yellow Scratch 'repeat' blocks are shown. The first block is labeled 'repetir 10'. The second block is labeled 'por siempre'. The third block is labeled 'repetir hasta que' and has a small grey block with a question mark in its condition slot.

50

Primera actividad

Podemos realizar una actividad de suma de n números. Por ejemplo sumar los 10 primeros números naturales: $1 + 2 + 3 + 4 + \dots + 10$. Se puede proponer a los alumnos que en una caja de vacía (representa la variable suma y iniciada a 0) introduzcamos papeles con los números 1, 2 y así hasta 10. Cuando introducimos el primer valor, el papel con el número 1, el valor de la caja (variable suma) es 1, pero cuando introducimos el 2, ahora el nuevo valor de la caja es la suma del número que ya está dentro de la caja (el 1) más en número 2 (ahora suma vale 3) y así sucesivamente. Otro alumno puede hacer de contador, para que entiendan el significado de esta variable, muy utilizada en la programación. El alumno incrementará en 1 cada vez que se introduzca un papel. Como tenemos 10 papeles, contador llegará a 10.

Segunda actividad

Implementar la actividad anterior pero con alguna herramienta de programación, como SCRATCH. En caso que algunos alumnos tengan algún problema con estas estructuras repetitivas, podemos invitarte, a que previamente realice actividades en la plataforma code.org para una mayor comprensión de la estructura. Como podemos observar en la ilustración que se muestra a continuación, el resultado 55 es que tendrá que tener la caja cuando se repita la suma 10 veces



Ilustración 14. Captura de SCRATCH. Estructuras iterativas en SCRATCH. Suma de n números. Fuente: Elaboración propia.

Tercera actividad

Los alumnos entenderán que cuando se repite una secuencial de instrucciones de una manera lógica, la podemos insertar dentro de una estructura repetitiva. Es importante insertar un determinado tiempo para que los alumnos vean que hace cada instrucción y en cada repetición. En esta actividad podrán dibujar más cuadros y con menos instrucciones que la estructura secuencial.

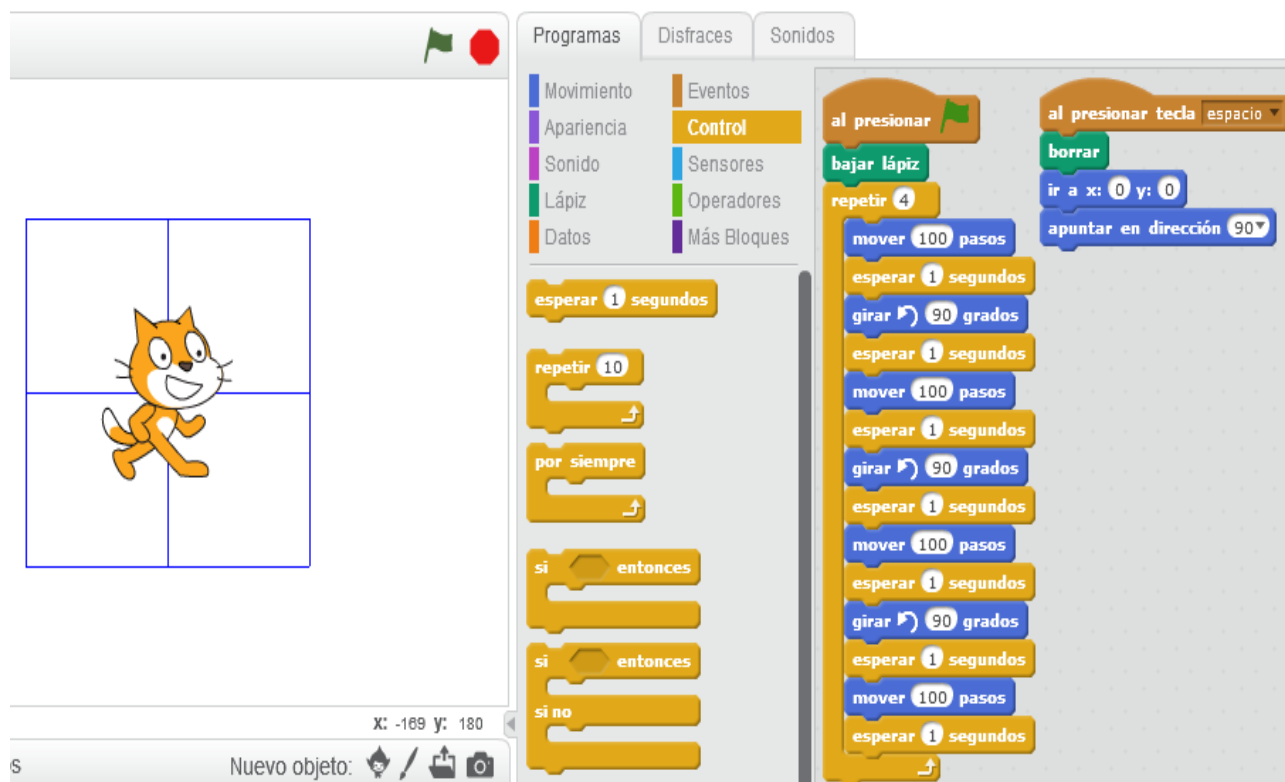


Ilustración 15. Captura de SCRATCH. Estructuras iterativas en SCRATCH. Dibujo de 4 cuadros. Fuente: Elaboración propia.

3. Las estructuras condicionales

Los alumnos tienen que entender que en la programación también hay decisiones. El profesor puede presentarle un ejemplo muy gráfico: Cuando nos encontramos frente a un control de seguridad, es probable que se nos solicite una acreditación (acreditación de la empresa, DNI, etc.) para mostrar nuestra identidad. El personal de seguridad evaluará si nuestra acreditación cumple la condición para entrar (somos la persona que figura en el DNI o tarjeta de empresa). Si se cumple la condición, el personal de seguridad te dejará entrar; sino, no te dejará entrar.

Lo mismo que ocurre en esta situación en el mundo real, las empresas emulan la misma situación en sus sistemas informáticos. Cuando accedemos a una zona restringida en una web (banco, universidad, mail, etc.) nos debemos acreditar de manera virtual. En este caso, nos evaluará un programa (no un agente de seguridad). La condición que evaluará el programa será nuestro nombre de usuario y nuestra contraseña. Si coincide el usuario con la contraseña, nuestra autenticación será válida y el programa nos dejará entrar al recurso restringido; sino, (si no coincide el usuario y/o la contraseña) no accederemos al sistema.

Primera actividad

Los alumnos tienen que entender que no siempre lo que se evalúa ha de ser igual a algo. Por ejemplo, en los parques de atracciones, los niños tienen que medir, como mínimo, una determinada altura, lo que significa que entrará a la atracción el niño o niña que tenga esa altura o mayor. Si la altura mínima es de 1.50 m o mayor entrarán (altura ≥ 1.50). En otros casos, tenemos que cumplir un intervalo de edades.

Por ello se puede realizar el siguiente ejercicio. Un alumno tendrá el rol de guardia de seguridad y tendrá varios papeles donde se le escribirá unas condiciones simples, y que en el caso que las cumpla, se le colocará en un lugar concreto de la clase, sino cumple la condición se le dejará sentado.

Segunda actividad

Se propone a los alumnos que simulen el sistema de evaluación de la programación de su asignatura, con la herramienta SCRATCH. La calificación final de su nota depende de 1 trabajo práctico, y de examen de control sobre la unidad didáctica. Pero para ser evaluado, y que se les pueda hacer la media, han de cumplir las siguientes condiciones:

- En el trabajo práctico han de obtener un mínimo de 4
- En el examen un mínimo de 5
- En caso que no cumpla alguna de estas condiciones se le mostrará suspenso



Ilustración 15. Captura de SCRATCH. Estructura condicional en SCRATCH. Cálculo de la nota media. Fuente: Elaboración propia.

6. Conclusiones

Una vez se ha realizado un estudio bibliográfico y un marco teórico que cimienta el presente trabajo, además de realizar un trabajo de campo (basado en actividades grupales e individuales y la recolección de datos mediante dos encuestas realizadas), se llega a la conclusión de que los profesores que usan herramientas programación gráfica en su proceso de enseñanza-aprendizaje y los alumnos que reciben el proceso, se sienten fuertemente agradecidos y motivados con el uso de estas herramientas. Podemos decir que la experiencia ha sido fuertemente aceptada tanto por unos como por otros. Las observaciones llevadas a cabo y los resultados de las encuestas confirman este hecho.

Al principio del presente trabajo se plantearon cuatro objetivos específicos que se debían de cumplir para conseguir el objetivo general. En referencia al primer objetivo específico, se llega a la conclusión de que el nivel de aceptación de las herramientas de programación gráfica, tanto por profesores y alumnos, ha sido muy alto. Se puede afirmar que en su totalidad aceptan estas herramientas como metodología de enseñanza-aprendizaje innovadora y motivadora. Tanto profesores como alumnos se han implicado en la enseñanza y en el interés por aprender respectivamente, llegando a cumplir el segundo objetivo. No obstante, la mayoría de los encuestados han expuesto muchas ventajas y prácticamente pocos alumnos han expuesto desventajas. Incluso un alumno expone una desventaja por falta de conocimiento, ya que muchas herramientas incorporan varios idiomas. Finalmente, se llega a la conclusión que las herramientas de programación gráfica, objeto de estudio del presente trabajo, sirve de base para aprender a programar, además de que se consideran herramientas adecuadas para realizar una transición antes de adentrarse en metodologías clásicas para aprender a programar.

Es por lo dicho anteriormente por lo que el objetivo general se ha cumplido, ya que para los alumnos de ciclo formativo de grado medio en sistemas microinformáticos, ha significado aprender de manera significativa y motivadora, viéndose en todo momento identificados en las actividades que realizaban. Las ideas constructivistas de aprender a aprender y con cierta autonomía se han visto en las observaciones realizadas en las actividades desempeñadas por el colectivo de alumnos. También, el papel de guía que ha de desempeñar el profesor (más que un mero transmisor de información) se ha visto relucir asombrosamente en este tipo de actividades.

7. Limitaciones y líneas de investigación futuras

Las principales limitaciones con que nos encontramos se centran en el desconocimiento de los sistemas de trabajo que hemos presentado. La falta de formación y de conocimiento suficiente tanto por parte de los alumnos como por el profesorado incide en la baja utilización de unos recursos cuya eficacia potencial se ha pretendido demostrar en este trabajo. No obstante, dada la temática y los resultados de las encuestas, se puede afirmar que estos resultados son altamente extrapolables, dadas las características habituales de la población de muestra de la investigación y la generalización de la relación con las nuevas tecnologías por parte de dicha población, ya que su conocimiento y manejo varía más por factores de edad (que en este trabajo se pueden extrapolar) que por otros factores como podrían ser los geográficos.

Como línea de investigación futura se propone realizar una segunda fase, en el que se tendría en cuenta un segundo grupo de control. Dicho grupo de control estaría compuesto por un grupo de alumnos que jamás se le haya impartido clases con herramientas de programación gráfica. El primer grupo, que sería el grupo experimental, sería el grupo que se ha tenido en cuenta en el presente trabajo de investigación. Este grupo recibiría más sesiones grupales e individuales basadas en las actividades que se exponen en el presente trabajo. Se añadirían actividades para aprender POO como herramientas específicas. Una vez acabadas las actividades, se propondría la realización de una prueba de lógica, pensamiento abstracto y resolución de problemas generales. Con la presente prueba, en la que recogeríamos las calificaciones, tanto de un grupo como del otro, se podría determinar si hay una relación directa entre el aprendizaje de la programación mediante las herramientas de programación gráfica y una mejor comprensión para solucionar problemas de carácter general.

8. Bibliografía

8.1. Referencias bibliográficas

Coll, C. (1988). *Infancia y Aprendizaje*. Barcelona: Universidad de Barcelona.

Coll, C. (1990). Capítulo 23. Un marco de referencia psicológico para la educación escolar: la concepción constructivista del aprendizaje y de la enseñanza. En Coll, C., Palacios, J. y Marchesi, A. (comp.) *Desarrollo psicológico y educación II*. Madrid: Alianza.

Delval, J. (1997). *Tesis sobre Constructivismo*. Barcelona: Paidós.

Díaz, F. y Hernández, G. (2002). Capítulo 2. Constructivismo y Aprendizaje significativo. En Díaz, F., Hernández, G. (2ª Ed). *Estrategias docentes para un aprendizaje significativo. Una interpretación constructivista* (pp.23-61). México. Mc Graw Hill.

Gardner, H. (1993). *Creating minds: An anatomy of creativity seen through the lives of Freud, Einstein, Picasso, Stravinsky, Eliot Graham and Gandhi*. Nueva York: Basic Books.

Ley orgánica 2/2006, de 3 de mayo, de Educación. Boletín Oficial del Estado, 106, de 4 de mayo de 2006.

López, J.C. (2009). *Algoritmos y Programación (guía para docentes)*. Material no publicado. Recuperado el 12 de Diciembre de 2014, de <http://www.eduteka.org/pdfdir/AlgoritmosProgramacion.pdf>.

López, C. y Sánchez, R. (2012). *Scratch y Necesidades Educativas Especiales: Programación para todos*. Revista de Educación a Distancia, Número 34. Recuperado el 6 de Enero de 2015, de <http://um.es/ead/red/34>.

Macdonald, S. (2011). *Beyond-programming-scratch-constructivist-learning-environment*. SCRATCHED. Recuperado el 6 de Enero de 2015 de: <http://scratched.gse.harvard.edu/stories/beyond-programming-scratch-constructivist-learning-environment>.

MacMillan, J. y Schumacher, S. (2005). *Investigación educativa*. Madrid: Pearson Addison Wesley.

Real Decreto 143/2007, de 26 de junio, por el que se establece la ordenación de las enseñanzas de la educación secundaria obligatoria. Diario Oficial de la Generalitat de Catalunya, 4915, de 29 de junio de 2007.

Resnick, M. (2007). *Sembrando las semillas para una sociedad más creativa*. Laboratorios de medios de MIT, Massachussets. Eduteka. Recuperado el 2 de Enero de 2015 de: <http://www.eduteka.org/modulos.php?catx=9&idSubX=277&ida=914&art=1>.

Robinson, K. (1999). *All Our Futures: Creativity, Culture and Education*. *Imagination in the source of all human achievement*. SIR KEN ROBINSON. Recuperado el 23 de Diciembre de 2014 de: <http://sirkenrobinson.com/pdf/allourfutures.pdf>.

Robinson, K. (2009). I Foro Mundial sobre el Talento en la Era del Conocimiento. Ágora Talentia. Pamplona.

Sánchez-Montoya, R. (2011). *¿Más avance tecnológico implica mayor inclusión?*. VII Jornadas de Cooperación Educativa con Iberoamericana sobre Educación Especial e Inclusión Educativa. Octubre, 2011, Montevideo, Uruguay.

Stager, G. (2003). *El pro de los computadores*. Eduteka. Recuperado el 19 de Diciembre del 2014 de: <http://www.eduteka.org/ProComputadores.php>.

Stenberg, R. (1997). *Inteligencia Exitosa. Cómo una inteligencia práctica y creativa determina el éxito de la vida*. Barcelona: Paidós.

8.2. Bibliografía complementaria

Ezquerro, M, (2014). *Uso de Geogebra en enseñanza de geometría analítica en 4º de ESO*. (Proyecto Final de Máster). Universidad Internacional del la Rioja, Logroño. Recuperado de <http://www.unir.net>.

Grupo Lifelong Kindergarten del Laboratorio de Medios del MIT. *Scratch - Imagina, Programa, Comparte*. Recuperado el 09 de Diciembre de 2014 de: <http://scratch.mit.edu>.

López, J.C. (2011). *Programación con Scratch*. Material no publicado. Recuperado el 12 de Diciembre de 2014, de <http://www.eduteka.org/pdfdir/AlgoritmosProgramacionCuaderno1.pdf>

Serrano, G. (2012). *Programación Scratch para Niños*. [Sitio de internet]. Recuperado el 10 de Diciembre de 2014 de: <http://www.programaciónScratch.com>

9. Anexos

Anexo A. Problema de la cadena alimenticia

EJEMPLO

Proyecto: La cadena alimentaria

Estándares MEN que se cubren (Colombia): Explico la dinámica de un ecosistema teniendo en cuenta las necesidades de energía y nutrientes de los seres vivos (cadena alimentaria).

Descripción: En este proyecto los estudiantes deben representar el comportamiento de varios seres vivos en su respectivo ecosistema, teniendo en cuenta tanto necesidades como cantidades disponibles de energía y nutrientes (cadena alimentaria). Para ello, deben elaborar una simulación, en MicroMundos o en Scratch, de una cadena alimentaria teniendo en cuenta seres productores, herbívoros, carnívoros y omnívoros.

Fases del proyecto:

1. Los estudiantes deben imaginar un ecosistema que contenga por lo menos cuatro seres vivos. Luego deben dibujar o importar los seres vivos que imaginaron. Además, deben dibujar el escenario que representa el ecosistema.
En este punto, el docente debe estimular la reflexión para que ellos verifiquen que el ecosistema que dibujaron corresponde con los seres vivos que en la simulación incluyeron en este. Un problema inesperado puede plantearse mediante la pregunta ¿todos los seres vivos que representaste viven en ese ecosistema? Si la respuesta es negativa, deben plantear por escrito al menos tres alternativas de solución (por ejemplo: cambiar el ecosistema, cambiar alguno de los seres vivos, cambiarlo todo, etc).
Además, cada ser vivo debe tener un tamaño proporcional en relación a los otros seres y elementos del ecosistema. En caso de ser necesario, se debe destinar un lapso de tiempo de la clase para investigar, en Internet o en la Biblioteca Escolar, qué seres vivos habitan en el ecosistema que dibujaron.
2. Esta fase inicia con otro problema inesperado: "La tarea quedó mal planteada y hay que corregirla, de los cuatro seres vivos que se crearon en la fase anterior, debe haber por lo menos un ser vivo de cada tipo: productor, herbívoro, carnívoro y omnívoro".
Los estudiantes deben investigar qué seres vivos de cada tipo habitan en el ecosistema que dibujaron. Luego, dibujar o importar los seres vivos correctos para que se cumpla la condición planteada en la tarea rectificada. No es necesario que eliminen los seres vivos que habían creado en la fase 1, siempre y cuando correspondan al ecosistema.
3. A continuación, deben programar el desplazamiento de los seres vivos por todo el espacio disponible y de manera aleatoria (se pueden utilizar las instrucciones "rebotar si está tocando un borde" y "número al azar entre 1 y 15 grados" como parámetro de la instrucción girar).
Como problema inesperado pedir que se restrinja el movimiento de manera que se aproxime al comportamiento real de cada ser vivo. Por ejemplo, si el fondo tiene tierra y firmamento, entonces un ser vivo que no vuela, no se puede mover en el área de la pantalla que representa el firmamento. Agregar variables para controlar la velocidad a la que se desplaza cada ser vivo. Algunos, como las plantas, tendrán velocidad 0.
4. Los estudiantes comparten el trabajo realizado con el resto de la clase y reciben retroalimentación tanto de sus compañeros, como del docente.
5. Atender la retroalimentación suministrada. Hacer la programación correspondiente para que cuando a un ser vivo lo toque otro al que le sirve de alimento, el primero desaparezca (como si se lo hubiera comido).
6. El problema inesperado ahora es que debe programarse la aparición de varios seres vivos iguales, ubicados en diferentes posiciones de la pantalla (por ejemplo, si uno de los seres vivos de la animación es un conejo, copiar entonces el objeto conejo, al menos tres veces, pues en un ecosistema rara vez se encuentra un solo animal de cada especie). Solicitar a los estudiantes, al menos dos alternativas, para realizar esta tarea. Las apariciones deben hacerse de acuerdo a una tasa de reproducción establecida para cada uno de los seres vivos. Por ejemplo, se reproducen más rápidamente los conejos que los zorros.
Se puede destinar un tiempo de la clase para investigar la tasa de reproducción de cada uno de los seres vivos que se incluyen en la animación.
7. Compartir con el resto de la clase el trabajo realizado y recibir retroalimentación de los compañeros.

Ilustración 16. Problema de la cadena alimenticia para ciencias naturales para implementar en SCRATCH.

Fuente: López, J.C. (2009)

Anexo B. Encuesta para los docentes:

Pregunta 1: ¿Usted como docente en el área de programación, considera que las herramientas de programación gráfica son adecuadas para aprender a programar?

- A – Muy adecuada
- B – Adecuada
- C – Poco adecuada
- D – Nada adecuada

Pregunta 2: ¿Usted como docente en el área de programación, considera que las herramientas de programación gráfica son buenas herramientas transitorias antes de empezar un curso de programación clásico?

- A – Sí
- B – No
- C – NS/NC

Pregunta 3: ¿Cree que el uso de herramientas de programación gráfica supone un buen elemento constructivo y significativo para el proceso de aprendizaje de sus alumnos?

- A – Sí, mucho
- B – Bastante
- C – Poco
- D – Nada

Pregunta 4: ¿Considera que es importante la elección de un tipo de herramienta de programación según el contenido que se quiere enseñar?

- A – Sí, muy importante
- B – Si, importante
- C – Poco importante
- D – Nada importante

Pregunta 5: ¿Cree que el uso de herramientas de programación gráfica puede extrapolarse al aprendizaje de otras asignaturas que requieran solucionar problemas, como las Matemáticas o las ciencias naturales?

- A – Sí, a estas dos asignaturas
- B – Si, solo a las matemáticas
- C – A ninguna de ellas

Pregunta 7: ¿Considera el uso de herramientas de programación gráfica un elemento motivador para su alumnado?

- A – Sí, muy motivador
- B – Bastante motivador
- C – Poco motivador
- D – Nada motivador

Pregunta 8: ¿Considera el aprendizaje de estas herramientas intuitivas y sencillas o difíciles?

- A – Lo considero difícil
- B – Lo considero intuitivo y sencillo
- C – Normal

Pregunta 9: ¿Considera la herramienta Alice una de las más adecuadas para aprender la base en la Programación Orientada a Objetos?

- A – Sí, muy adecuada
- B – Bastante adecuada
- C – Poco adecuada
- D – Nada adecuada

Pregunta 10: ¿Explica cómo fue su experiencia en su aula con la utilización de alguna herramienta de programación gráfica?

Pregunta 11: ¿Explica qué herramienta de programación gráfica ha utilizado para sus clases?

Anexo C. Encuesta para los alumnos

Pregunta 1: ¿Consideras que las herramientas de programación gráfica son adecuadas para aprender a programar?

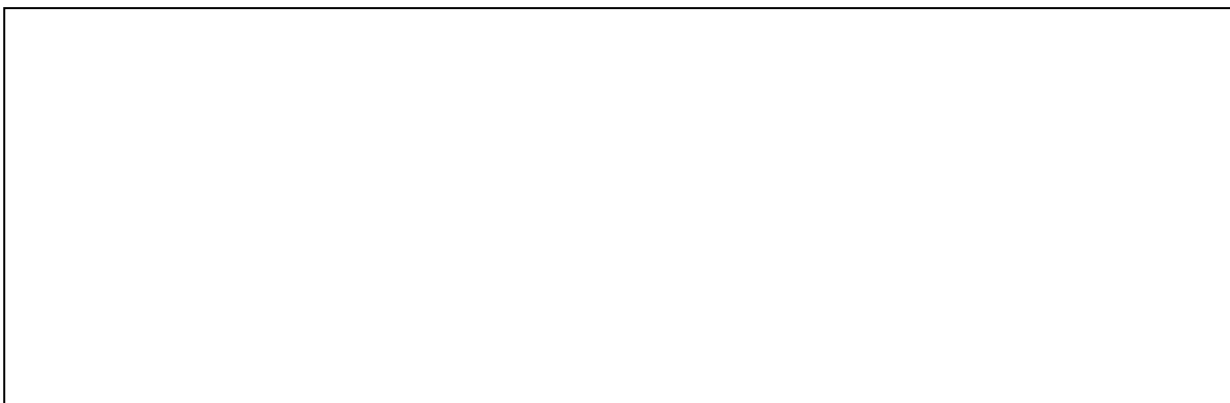
- A – Muy adecuada
- B – Adecuada
- C – Poco adecuada
- D – Nada adecuada

Pregunta 2: ¿Consideras que las herramientas de programación gráfica son buenas herramientas transitorias antes de empezar un curso de programación clásico?

- A – Sí
- B – No
- C – NS/NC

<p>Pregunta 3: ¿Te sientes más identificado con estas herramientas de programación para aprender a programar que mediante los métodos clásicos basados en la comprensión de un lenguaje de programación?</p> <p>A – Sí, muy identificado B – Sí, bastante identificado C – Poco identificado</p>
<p>Pregunta 4: ¿Considera el uso de herramientas de programación gráfica un elemento motivador para su proceso de enseñanza-aprendizaje?</p> <p>A – Sí, muy motivador B – Bastante motivador C – Poco motivador D – Nada motivador</p>
<p>Pregunta 5: ¿Considera el aprendizaje de estas herramientas intuitivas y sencillas o difíciles?</p> <p>A – Las considero difíciles B – Sí, intuitivas y sencillas C – Normal</p>
<p>Pregunta 6: ¿Has recibido clases con alguna herramienta de programación gráfica?</p> <p>A – Sí, muchas veces B – Sí, alguna vez C – Pocas veces D – Ninguna vez</p>
<p>Pregunta 7: Explica si con el uso de estas herramientas te ha ayudado a entender la estructuras básicas de programación.</p> <div style="border: 1px solid black; height: 100px; width: 100%;"></div>

Pregunta 8: Explica las ventajas y desventajas que has encontrado con estas herramientas de programación.



Anexo D. Actividad para entender que los algoritmos siguen unos pasos lógicos

ACTIVIDAD

A diferencia de los seres humanos que realizan actividades sin detenerse a pensar en los pasos que deben seguir, los computadores son muy ordenados y necesitan que el programador les especifique cada uno de los pasos necesarios y su orden lógico de ejecución.

Listar una serie de pasos para realizar una tarea y presentarlos a los estudiantes en forma desordenada para que ellos los ordenen.

Por ejemplo, ordenar los pasos para pescar:

- ___ El pez se traga el anzuelo.
- ___ Enrollar el sedal.
- ___ Tirar el sedal al agua.
- ___ Llevar el pescado a casa.
- ___ Quitar el Anzuelo de la boca del pescado.
- ___ Poner carnada al anzuelo.
- ___ Sacar el pescado del agua.

Ilustración 17. Problema para desarrollar en la actividad del tema "El algoritmo es un conjunto de pasos lógicos" Fuente: López, J.C. (2009)

Anexo E. Simbologías de los diagramas de flujo

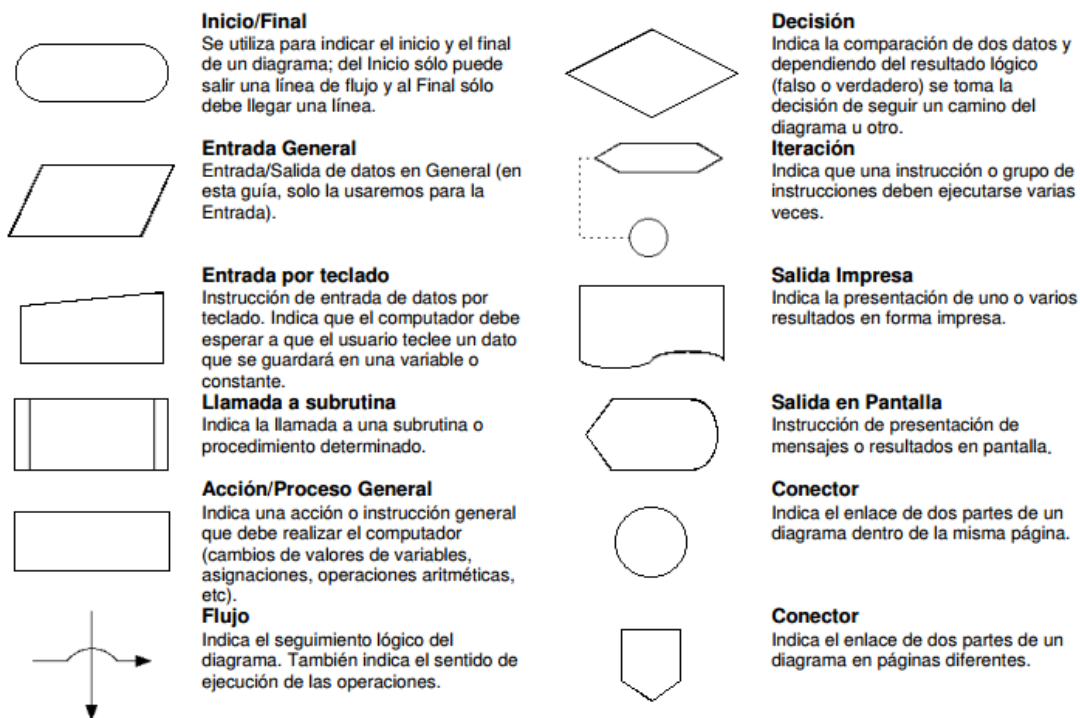


Ilustración 18. Simbología de los diagramas de flujo. Fuente: López, J.C. (2009)