

ULL

Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática



Trabajo de Fin de Grado

Desarrollo y evaluación del
pensamiento computacional: una
propuesta metodológica y una
herramienta de apoyo

*Development and assessment of computational
thinking: a methodological proposal and a support tool.*

Alexis Daniel Fuentes Pérez

La Laguna, 5 de junio de 2017

D. **Gara Miranda Valladares**, con N.I.F. 78.563.584-T profesora Ayudante Doctor adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

I N F O R M A

Que la presente memoria titulada:

“Desarrollo y evaluación del pensamiento computacional: una propuesta metodológica y una herramienta de apoyo.”

ha sido realizada bajo su dirección por D. **Alexis Daniel Fuentes Pérez**, con N.I.F. 79.064.257-D.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 5 de junio de 2017

Agradecimientos

Agradecerle a mi tutora, Gara Miranda Valladares, por haberme guiado durante la realización de este trabajo, por implicarse en la difusión del mismo, y por interesarse en mi desarrollo profesional como Ingeniero Informático.

Agradecerle a mi familia, por haber estado a mi lado apoyándome siempre, por animarme en la persecución de mis logros, por ayudarme en el día a día, y por todo el afecto que he recibido de ellos.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

Resumen

El objetivo de este trabajo ha sido la definición de una metodología para el desarrollo y medición del pensamiento computacional, y el posterior diseño e implementación de una plataforma web para su puesta en marcha.

Para la propuesta metodológica se ha realizado una investigación de los antecedentes y estado actual del tema, búsquedas de estudios realizados alrededor del mundo cuyos objetivos han sido el desarrollo del pensamiento computacional, la respuesta cognitiva y/o emocional en niños al enseñarles pensamiento computacional, y de diferentes enfoques o maneras de medir el pensamiento computacional. Así mismo se ha hecho un análisis de los diferentes métodos encontrados para poder evaluar el pensamiento computacional, pues no es algo trivial, y hubo que establecer los pasos a seguir.

Además se ha buscado desarrollar una plataforma web que englobe todas las herramientas necesarias para el desarrollo y medición del pensamiento computacional, y que permita la puesta en marcha de la metodología previamente comentada. Esta plataforma debe ser fácil de comprender y no debe requerir conocimientos previos para uso, ya que está dirigida para alumnos en edades tempranas (menores de 18 años) y profesorado que no tiene porqué estar relacionado con asignaturas de Informática, y uno de nuestros objetivos es lograr la difusión de esta plataforma para fomentar el desarrollo del pensamiento computacional. Por último, la plataforma debe poder generar informes del progreso de los cursos de forma automática, donde el profesorado pueda obtener resultados de los avances que consigue así como datos estadísticos de progreso por edades o género, entre otros.

Palabras clave: Pensamiento computacional, metodología para el desarrollo y la evaluación, educación pre-universitaria, plataforma web, Abstractly.

Abstract

The objective of this work has been the realization of a methodology for the development and measurement of computational thinking, and the design and realization of a web platform for its implementation.

To obtain the methodology, I have done an investigation about the background and the current status of the subject, searches of studies carried out around the world whose objectives have been the development of computational thinking, the cognitive and emotional response in children when teaching them computational thinking, and different approaches or ways of measuring computational thinking. Also I have done an analysis of the different methods found to evaluate computational thinking, because it is not trivial, and I have to establish which steps would follow our methodology.

In addition, we have developed a web platform that has all the necessary tools for the development and measurement of computational thinking, and that allows the implementation of the previously mentioned methodology. This platform should be easy to use, since it is aimed at students at an early age (under 18 years old) and teachers that does not have to be related to IT subjects, because one of our objectives is to achieve the dissemination of this platform to encourage the development of computational thinking. Finally, the platform should be able to automatically generate reports of the progress of the courses, where teachers can obtain results of the progress achieved as well as statistical data of progress by age or gender, like others.

Keywords: *Computational thinking, methodology for development and assessment, pre-university education, web platform, Abstractly.*

Índice general

1. Introducción	1
1.1. Pensamiento computacional	1
1.2. Antecedentes y estado actual	2
1.2.1. Iniciativas	3
1.3. Objetivos	4
2. Desarrollo del pensamiento computacional	6
2.1. Proceso de desarrollo	6
2.1.1. Cómo desarrollarlo	6
2.1.2. Formas de aprendizaje	6
2.2. Herramientas para su desarrollo	7
2.2.1. Scratch	7
2.2.2. Snap!	7
2.2.3. Blockly	8
2.2.4. MIT App Inventor	8
2.2.5. Gamefroot	8
3. Medición del Pensamiento Computacional	9
3.1. Tests	9
3.1.1. Test de Marcos Román	10
3.1.2. Test de la CACM	11
3.2. Enfoques de Brennan & Resnick	11
3.3. Estudio por grupos de alumnado	12
3.4. Comparación de experiencias	14
4. Requisitos y tecnologías	16
4.1. Requisitos fundamentales	16
4.2. Búsqueda de un Framework	16
4.3. Elección de un Framework	18
4.4. Búsqueda y desarrollo de una metodología	19
5. Plataforma Web	20
5.1. Descripción	20
5.2. Configuración	20

5.2.1.	Metodología y despliegue	20
5.2.2.	Base de datos	21
5.2.3.	Gemas	21
5.3.	Diseño	22
5.4.	Funcionamiento	22
5.4.1.	Inicio	22
5.4.2.	Creación de usuarios	23
5.4.3.	Navegación	24
5.4.4.	Perfil	26
5.4.5.	Creación de cursos	27
5.4.6.	Matriculación en un curso	29
5.4.7.	Vista del curso	29
5.4.8.	Organización al alumnado	32
5.4.9.	Añadir actividades al curso	32
5.4.10.	Registrar una actividad	33
5.4.11.	Generación de informes	35
5.4.12.	Creación de actividades	37
5.4.13.	Actividades de medición	38
5.4.14.	Administración de usuarios	39
5.4.15.	Administración de actividades	40
5.4.16.	Alojamiento de archivos	41
6.	Conclusiones y líneas futuras	42
7.	Summary and Conclusions	44
8.	Presupuesto	46
8.1.	Presupuesto	46
	Bibliografía	46

Capítulo 1

Introducción

1.1. Pensamiento computacional

El pensamiento computacional, *computational thinking*, es el razonamiento llevado a cabo en la formulación de un problema y en la expresión de su solución de forma que pueda ser llevado a cabo eficientemente por un agente que procesa información.

Este término apareció por primera vez en 2006 cuando **Jeannete M. Wing** publicó un artículo [27] en el que lo dio a conocer y explicó la manera de pensar de un científico en computación cuando se enfrenta a un problema. En este artículo también se expone la importancia que tendría la adquisición de esta ‘habilidad’ en el resto de personas, ya que supondría una mejora en el procedimiento de resolución de problemas en la vida cotidiana. Además J. M. Wing presenta un gran interés porque este nuevo concepto sea difundido en la educación, e invita al profesorado a que se involucre a motivar y enseñar a sus alumnos por las Ciencia de la Computación.

La computación es un sinónimo de abstracción y automatización, ambos bases del pensamiento computacional.

La **abstracción** es el proceso de aislar un elemento de su contexto. En la computación, el nivel de abstracción es diferente a como se realiza en otras ciencias. Se realiza una abstracción simbólica, lo que la hace mucho más general, no se entra a especificar detalles. Un ejemplo de esto son los algoritmos, que no son más que una abstracción de un procedimiento paso a paso que recibe una entrada y da una salida. Además, esta abstracción también se caracteriza por el hecho de que se tiene en cuenta la posibilidad de fallo o errores que puedan ocurrir. Con el uso de la abstracción, van apareciendo diferentes capas de abstracción. Estas capas deben tener una interfaz bien definida entre ellas que nos permitirá realizar sistemas más complejos.

La computación es la **automatización** de nuestras abstracciones, y se necesita una ‘maquina’ capaz de procesar dicha abstracción. Es común pensar que

para computar es necesario la utilización de un ordenador, una máquina, pero esto no es del todo correcto. Los humanos somos capaces de procesar información, por lo que no es necesario disponer de un ordenador para hablar de computación.

El pensamiento computacional está influenciando a **muchas disciplinas**, tanto de ciencias como de humanidades [21]. Un ejemplo de esto lo podemos observar en algoritmos de aceleración de la secuenciación del genoma humano, en el descubrimiento del papel de 12 genes en la levadura por el robot Adam, cálculos atómicos para la exploración de fenómenos químicos, etc.

Debido a esto, si nos detenemos a observar a nuestro alrededor podemos encontrar el pensamiento computacional en nuestra vida cotidiana, y no necesariamente en campos de informática y computación. Es una habilidad que es posible desarrollar sin importar la edad en la que uno se encuentre, y cuya obtención debería ser un ‘must’ en esta actualidad en la que vivimos, no ya solo por su aparición en múltiples campos y disciplinas, sino por la cantidad de beneficios que aporta, tanto en la resolución de problemas complejos como en otros mas sencillos de nuestro día a día.

1.2. Antecedentes y estado actual

A partir de 2006 se ha comenzado a trabajar en la búsqueda de una definición operativa que describa con precisión el pensamiento computacional. La iniciativa más conocida en este ámbito es la promovida por la *Sociedad Internacional de la Tecnología en la Educación* (ISTE) y la *Asociación de Profesores de Informática* (CSTA) quienes han definido el pensamiento computacional como el proceso de resolución de problemas que incluye:

- Formular problemas de forma que un ordenador u herramienta pueda ayudar a resolverlo.
- Organizar y analizar lógicamente la información.
- Representar la información a través de abstracciones.
- Automatizar soluciones haciendo uso del pensamiento algorítmico, es decir, estableciendo una serie de pasos ordenados para llegar a la solución.
- Generalizar y transferir este proceso de resolución de problemas para ser capaz de resolver una gran variedad de familias de problemas.

Las líneas principales de trabajo en el ámbito del pensamiento computacional se han centrado en la enseñanza de programación a estudiantes, ya que es

la manera en la que los científicos en computación o ingenieros informáticos aprenden e interiorizan esta habilidad, y en la incorporación de especialistas en Informática a equipos de trabajo multi-disciplinarios para que así sus conocimientos puedan ayudar en otros ámbitos o disciplinas. En cuanto al análisis de cómo y en qué medida ayuda el pensamiento computacional a desarrollar habilidades que mejoren la resolución de problemas en la vida cotidiana, la creatividad o la lógica se encuentra poco trabajo útil. Debido a esto, este trabajo se centrará en proporcionar una herramienta que de soporte a la hora de evaluar el pensamiento computacional en aquellos estudiantes que realicen actividades de programación o de resolución de problemas.

1.2.1. Iniciativas

Code.org

Code.org [4] es una organización no gubernamental fundada en 2012 por Hadi y Ali Partovi. Presenta numerosos objetivos entre los que se destacan la inserción de informática como una asignatura obligatoria en los colegios, y la inclusión de la mujer en la computación. Además, es el organizador del movimiento global **Hora del Código** [9] dónde se pretende que todas las personas y estudiantes de cualquier edad puedan ver, al menos una vez, una hora de código en su vida.

En su página web puedes encontrar de forma gratuita cursos y actividades para alumnos que quieran aprender, organizadas por edades y capacidades, permitiendo de esta manera que niños que no saben leer también puedan participar. Además, también presenta cursos para profesores que quieran aprender como enseñarles a sus alumnos, y les proporcionan facilidades para crear ellos mismos actividades.

Por último, destacar que puedes encontrar numerosas actividades propuestas para Hora del Código tanto en su página web como en la propia de *Hour of Code*.

Computational Thinking for Educators

Computational Thinking for Educators [24] es un curso gratuito desarrollado por Google para que el profesorado aprenda en qué se basa y qué es exactamente el pensamiento computacional y cómo incluirlo en las diferentes asignaturas que ellos imparten, ya sea Literatura, Matemáticas, Química, etc. Además, les enseña cómo añadir el Pensamiento Computacional en sus propios currículums.

El contenido de estos cursos está compuesto por:

1. “*Introducción al Pensamiento Computacional: ¿Qué es?*”

2. *“Exploración de Algoritmos”*
3. *“Encontrar Patrones”*
4. *“Desarrollar Algoritmos”*
5. *“Proyecto Final: Aplicar el Pensamiento Computacional”*

Programamos

Programamos [13] es una asociación sin ánimo de lucro que promueve el desarrollo computacional en edades tempranas a través de la programación, tanto de videojuegos como de aplicaciones móviles.

Entre sus características destacables está la disposición del material necesario para el profesorado, la formación a docentes, la organización y realización de eventos presenciales destinados al desarrollo del pensamiento computacional, y la investigación del impacto que tiene este aprendizaje.

1.3. Objetivos

El objetivo de este trabajo es la realización de una propuesta metodológica que incluya tanto tareas para desarrollar el pensamiento computacional, como pruebas o tests que permitan evaluar en qué medida se ha desarrollado el pensamiento computacional o habilidades relacionadas con la resolución de problemas, y la posterior realización de una aplicación informática que incorpore dicha metodología, y que presente una interfaz sencilla, y atractiva. Además, esta aplicación informática también deberá ser capaz de generar informes en los que se puedan observar la evolución del pensamiento computacional y en cómo mejora las capacidades del alumnado para la resolución de problemas.

Para el estudio del grupo y el desarrollo de las actividades el alumnado será dividido en dos grupos, uno experimental y otro de control. Al comienzo del curso todos los alumnos realizarán un pre-test que permita obtener una aproximación inicial del estado las habilidades de resoluciones de problemas en el alumnado, y al finalizarlo deberán realizar un post-test. De esta manera se pretende encontrar una diferencia entre ambos grupos de alumnado, una diferencia de la evolución del desarrollo del pensamiento computacional entre sexos, y una relación entre la satisfacción y dificultad encontrada para las actividades de la metodología.

Los test, las actividades, y los cursos estarán ajustados y diferenciados por un rango de edad. Así mismo, no todos los cursos tendrán la misma duración y esta será escogida por el profesor. De esta manera, la calidad de los resultados

obtenidos para cursos de edad similar puede ser diferente, obteniendo teóricamente una mejor experiencia con aquellos alumnos que sean expuestos a la metodología durante un mayor tiempo.

Capítulo 2

Desarrollo del pensamiento computacional

2.1. Proceso de desarrollo

2.1.1. Cómo desarrollarlo

El pensamiento computacional puede desarrollarse con la práctica de principios básicos de la computación, como puede ser la secuenciación de tareas, la utilización de bucles, la abstracción, la modularización, el reconocimiento de patrones, etc. Debido a esto la programación es uno de los medios existentes para dicho desarrollo, pero no el único.

Como bien dijimos en apartados anteriores, para hablar de computación no es necesario hablar de ordenadores, y por lo tanto, para desarrollar el Pensamiento Computacional tampoco. Existen una gran variedad de ejercicios que se utilizan para promover el Pensamiento Computacional en aulas, los cuales no necesitan de un ordenador para ejecutarlos. En ellos los alumnos deben reconocer patrones en imágenes, o establecer el orden de ejecución de tareas para lograr un objetivo, entre otros.

2.1.2. Formas de aprendizaje

En 2012 se hizo un **estudio de pensamiento computacional**, de mano de Karen Brennan y Mitchel Resnick [20] y se presentó en el congreso de la Asociación de Investigación Educativa Americana. Los estudiantes realizaron prácticas de pensamiento computacional, las cuales estaban enfocadas en procesos de pensamiento y aprendizaje, y donde se le daba más importancia al cómo que al qué se estaba aprendiendo. En dicho estudio se observó que la forma de actuar de los estudiantes podía ser agrupada en diferentes grupos.

- Incremental e iterativo. Los estudiantes describían iteraciones de imagi-

nación y construcción. Empezaban por pequeñas ideas, las realizaban, y luego desarrollaban más, continuaban mejorándolas o implementándoles cosas nuevas.

- Probando y depurando. Desarrollos a base de pruebas y errores, aprendiendo de otros proyectos o de otras personas.
- Reutilizando y remezclando. Es uno de los objetivos de la comunidad online de Scratch. Desarrollaba las capacidades relacionadas con lecturas de código y al razonamiento de preguntas de cómo se puede adaptar y temas de propiedad.
- Abstrayendo y modularizando. Separaban diferentes comportamiento o funcionalidades en grupos de código, lo que permitía llegar a algo grande uniendo muchas cosas pequeñas.

Se les preguntó a los niños su opinión sobre este proceso de aprendizaje, y su respuesta fue que lo veían como un medio diferente a lo demás, algo que les permitía expresarse y crear lo que quisieran.

2.2. Herramientas para su desarrollo

2.2.1. Scratch

Scratch [14] es un entorno de programación visual gratuito diseñado por un equipo de investigación del MIT, que se encuentra bastante difundido en la enseñanza ya que permite programar sin saber código y permite desarrollar el pensamiento computacional. Además, cuenta con una comunidad online denominada *Scratch online community*, donde cientos de jóvenes desarrolladores comparten sus proyectos e interactúan con otros usuarios.

Esta herramienta presenta una interfaz amigable donde el usuario puede crear bucles, condiciones, eventos, estructuras de datos, realizar operaciones, etc. simplemente arrastrando “piezas” como si fuera un puzzle. Esto hace que sea bastante intuitiva y fácil de usar para alumnos de todas las edades, y les enseña conceptos de la computación sin la necesidad de saber como programar.

2.2.2. Snap!

Snap! [15] es una modificación avanzada de Scratch, es decir, incluye todo el contenido base de Scratch pero además incorpora muchas nuevas características y utilidades como lambdas, procedures, recursión, debugger, listas multi-dimensionales, clonning, comprobación de tipos, etc.

A partir de su versión 4.0 fue reescrita en JavaScript y no se le considera más una ‘modificación de Scratch’, sino una herramienta más.

2.2.3. Blockly

Blockly [3] es una librería para web o aplicaciones Android que añade un editor visual de código. Está desarrollado por Google y permite traducir en tiempo real la ‘programación visual en bloques’ a código sintácticamente correcto en JavaScript, Python, PHP, Lua, o Dart.

Entre sus características destaca que el código es exportable, es open source (por lo que puede modificarse o utilizarse en cualquier web/aplicación), es extensible, permite la creación de códigos potentes, y que está disponible en más de cuarenta idiomas.

2.2.4. MIT App Inventor

MIT App Inventor [10] es una herramienta diseñada para los principiantes en la programación y creación de aplicaciones Android. Se basa en la programación visual y en la construcción Drag & Drop de bloques. Está desarrollada por Google, el Programa de Formación de Maestros Scheller (STEP), y el MIT. Su uso está dirigido a la educación.

Su misión es fomentar el traspaso de los usuarios de ‘consumidores’ de apps a ‘desarrolladores’, y entre sus objetivos se encuentra la promoción de la educación en ciencias de la computación y en la realización y apoyo a la investigación comunitaria.

2.2.5. Gamefroot

Gamefroot [7] es una plataforma web que permite crear y compartir videojuegos en 2D. Se basa la colocación de elementos “pre-programados” en la escena, y permite la creación de scripts personalizados mediante programación visual Drag & Drop de bloques.

Ofrece una experiencia muy realista y completa para la realización de un videojuego completo, desde niveles, enemigos, objetos... Además posee una comunidad online para compartir experiencias, pedir ayuda, o trabajar en equipo.

La plataforma ofrece talleres para profesores y estudiantes, y cursos online que aportan contenido profesional para tu currículum.

Capítulo 3

Medición del Pensamiento Computacional

En este capítulo veremos qué utilizaremos para medir el pensamiento computacional, y qué experiencias y estudios se han realizado ya, así como los resultados que obtuvieron.

Uno de los temas más discutibles del pensamiento computacional es cómo se debe medir, ya que no es algo trivial y no existe un método específico ni correctamente aceptado, ya que tiene que ver con la lógica, la creatividad y la resolución de problemas.

La actividad más extendida consiste en la realización de tests cuyas preguntas abarquen todas aquellas áreas de interés, pero también existen enfoques cuyos resultados son más cualitativos que cuantitativos, ya que se centran en observar cómo se abarca la resolución de un problema por un determinado individuo.

A continuación, se describirán diferentes métodos utilizados por el momento y la complejidad que cada uno de ellos abarca.

3.1. Tests

Los tests de pensamiento computacional son unas de las herramientas más extendidas y utilizadas para llevar a cabo la medición del pensamiento computacional. Consisten en test de múltiple respuesta con una extensión definida que deben ser realizados en un tiempo determinado.

El diseño y la creación de estos tests es bastante complejo, ya que primero debe realizarse un estudio tanto del contenido que este debe abarcar, como del público al que está dirigido. Además, estos tests deben pasar por un proceso de evaluación mediante el cual diferentes expertos lo califican y proponen cambios o mejoras a tener en cuenta para su efectividad a la hora de obtener una medición real.

La forma de actuación con estos tests, por lo general, suele ser la realización de un test antes de que los conocimientos de pensamiento computacional sean desarrollados en el alumno (pre-test), y la realización de otro al finalizar los mismos (post-test). De esta manera, es posible comparar los diferentes resultados obtenidos por el alumno en los aspectos de la computación que se estudiaban.

3.1.1. Test de Marcos Román

Marcos Román González realizó en 2014 un test que permite medir el Pensamiento Computacional [25]. Este fue evaluado por 20 expertos.

Inicialmente el test presentaba los siguientes principios:

- Objetivo: Medir el nivel de PC del sujeto
- Contenido: Principios de la computación como bucles, condiciones, funciones, variables, etc.
- Dirigido a: Estudiantes españoles entre 12 y 13 años
- Herramienta: Test de múltiple respuesta con 4 opciones
- Tamaño: 40 preguntas
- Tiempo: 45 minutos

En cuanto a las preguntas de este test se tuvo que estas debían estar en orden creciente de dificultad; podían estar todas en la misma interfaz o, incluso, en diferentes interfaces; podía existir anidamiento entre ellas; las tareas a realizar podían ser ordenar, completar o detectar un error; y que las respuestas podían representarse mediante flechas, bloques o texto.

Tras la evaluación de los expertos se refinó la primera versión y se obtuvieron las siguientes mejoras:

- Eliminación de funciones con parámetros debido a su dificultad
- Es más apropiado utilizar las respuestas visuales que las textuales
- Las pruebas de debugging debían mostrar solo una vez el código
- Evitar anidamiento excesivo
- Mostrar ejemplos al inicio del test
- Disminución del tamaño a un total de 28 preguntas

3.1.2. Test de la CACM

En 2008 en CACM tenían un proyecto llamado *Making Games in Schools* para promover el Pensamiento Computacional en las aulas. La idea era que los niños aprendieran importantes conceptos de computación mientras desarrollaban sus juegos, pero ¿cómo podían medir este progreso?

En un inicio, se propuso utilizar pre-test y post-test con los contenidos actuales de Pensamiento Computacional para la educación, pero llegaron a disputas entre que debería abarcar y que no. Descubrieron un proyecto similar en el cual se desarrolló un juego donde los niños debían responder preguntas al jugarlo (idea de *Keiron Nicholson*) y que servía para ver cómo expresaban estos niños las construcciones de programación con lenguaje natural.

Al final se desarrolló otra versión del anterior juego donde estas preguntas eran de tipo multirespuesta, y era un test dirigido para niños de entre 11 y 12 años. Este test cubría:

- Abstracción y categorización
- Identificación de reglas y secuencias
- Identificación de condiciones IF, ELSE
- Operaciones básicas AND, OR, NOT
- Seguimiento de estados (para eventos)

Como conclusiones del proyecto, aparecieron dudas acerca de si realmente estaban midiendo el Pensamiento Computacional y preguntas acerca de si el test multirespuesta medía la capacidad de identificar reglas pero no la de realizarlas.

3.2. Enfoques de Brennan & Resnick

En el estudio que realizaron Brennan & Resnick en 2012 [20], utilizaron Scratch, y para evaluar el aprendizaje dieron varios enfoques:

- **Análisis de la carpeta de proyectos.** Se utilizó una aplicación llamada Scrape, en concreto la herramienta “*User Analysis*”, para analizar la carpeta de proyectos de un usuario y generar una representación visual de los bloques utilizados y no utilizados de cada proyecto. Se obtenía un gráfico donde cada columna era un proyecto, y cada fila un tipo de bloque de Scratch. La columna del final identifica los bloques que nunca habían sido utilizados. Esta herramienta les permitía analizar el avance de un estudiante a lo largo del tiempo (tiempo en el que formó su carpeta de

proyectos). El inconveniente de esto era que estaba muy enfocado al propio contenido de Scratch, y no proporcionaba mucha información de las prácticas de pensamiento computacional personales del usuario.

- **Entrevistas.** Se entrevistaron a 31 Scratchers que representaban diferentes rangos de edades, de diferentes regiones geográficas, algunos seleccionados al azar y otros que pidieron participar. La entrevista duraba entre 1 y 2 horas, y trataba diferentes bloques: ‘¿Qué es Scratch?’, ‘Creación de proyectos’, ‘Comunidad Online’, y ‘Valoración personal’. La sección de ‘Creación de proyectos’ es la que más ofrecía para evaluar el aprendizaje. Se le hacía escoger al alumno dos proyectos que le hubieran llamado bastante la atención. Luego se le preguntaba por su motivación para hacerlo, se ejecutaba, y se pasaba a preguntar cómo lo hizo, con qué problemas se encontró y como los solucionó. Al final se les preguntaba que cambiarían, si estaban orgullosos del resultado obtenido, etc. Con este enfoque se pudo observar que en algunos proyectos los niños no sabían exactamente cómo funcionaban partes, ya que habían copiado otros modelos parecidos hasta lograr algo funcionando. Por lo tanto, que el usuario haya utilizado una determinada cantidad de bloques Scratch no demostraba que hubiera adquirido dicho conocimiento.
- **Diseño de escenarios.** Este enfoque se realizaba en aulas y con un número más reducido de alumnos. Se les hacía una entrevista y se les daba a escoger entre diferentes proyectos ya terminados. A continuación tenían que decir qué hacía el proyecto escogido, cómo lo mejorarían, y arreglar un fallo. Estas actividades estaban preparadas por ser las que se debían haber desarrollado en los talleres. Este enfoque permitía estudiar el nivel de conocimiento de cada una de las áreas de interés y ver cómo reaccionaba el estudiante en vivo.

Como acabamos de ver los resultados que obtenemos de estos enfoques son más cualitativos que cuantitativos, ya que se basan en la observación y en la percepción de cómo los alumnos afrontan la resolución de un problema, y cómo actúan ante ella, por lo que la valoración del desarrollo puede variar dependiendo del “observador”.

3.3. Estudio por grupos de alumnado

Otras de las prácticas a la hora de evaluar el Pensamiento Computacional consiste en dividir al alumnado en dos grupos: un grupo denominado “experimental” el cual realiza actividades para el desarrollo del PC, y otro denominado “control” que realiza actividades totalmente diferentes y, de esta manera, poder observar una diferencia en el progreso de los resultados de ambos grupos

utilizando otras de las herramientas de evaluación, como por ejemplo los test, unidos a la estadística.

Un ejemplo de esto lo podemos ver un estudio de Pensamiento Computacional que se realizó en 2016 en el Instituto de Secundaria de la Comunidad Valencia, de mano de *Juan-Jose Boix-Tormos*[19]. En él se le enseñó programación de alto nivel a una clase de 1º de bachillerato en la optativa de Informática durante un periodo de dos meses.

Los alumnos se dividieron en dos grupos, el grupo experimental formado por 14 alumnos (12 de ciencias, 2 de humanidades) que cogieron la optativa de informática, y un grupo de control formado por 13 alumnos (6 de ciencias, 7 de humanidades) que no cogieron dicha optativa.

Para llevar a cabo un control de la evolución del Pensamiento Computacional en los alumnos se propuso un pre-test y un post-test. El nivel de ambos test era similar y el contenido estaba relacionado con las habilidades a estudiar:

- Análisis. Interpretación de la información y detección de patrones en los datos aportados.
- Abstracción. Ser capaz de reconocer datos de la vida real y diferenciar datos relevantes.
- Secuenciación. Definición de algoritmos, ser capaz de reconocer secuencias.

Para el desarrollo y formación se utilizó Gambas [6], un lenguaje de programación libre derivado de BASIC. Una de las razones por las que se escogió fue por la facilidad que presta para realizar programas sin saber de programación. Se iban realizando ejercicios prácticos de programación de forma creciente, y la formación duró un total de 18 horas.

Se hizo un estudio probabilístico de ambos grupos correspondientes a los resultados de la 2ª evaluación (antes de empezar el estudio) y los obtenidos con la 3ª evaluación (finalización del estudio). No se obtuvieron resultados muy significativos en las medias de 3ª evaluación de ambos grupos, pero se observó un aumento de porcentaje del 13 % del grupo experimental, y una disminución del 3 % del grupo de control.

No se pudo realizar el post-test debido a imprevistos ocurridos con las actividades de fin de curso.

3.4. Comparación de experiencias

Se ha realizado una búsqueda de diferentes experiencias relacionadas con el desarrollo del pensamiento computacional y la medición de resultados realizadas con niños en edades tempranas. A continuación, se muestra una tabla en la que se comparan las experiencias estudiadas en cuanto a edades, duración, herramientas utilizadas, actividades realizadas, etc.

R.	Experiencia	Duración	Edad	Herramientas	Actividades	Medición
[20]	Entrevista	1m - 4a	8 - 17	Scratch	Scratch	Enfoques
[19]	Instituto	18h	16 - 19	GAMBAS	Progr. Visual	Grupo, Estadística
[22]	Colegios	2a	10 - 14	Alice	Drag&Drop	Pre/Post-test, Tareas
[12]	Instituto	9h	8 - 9	Scratch	Animacion	Análisis de Tareas
[26]	Colegios	1h	12 - 14	Scratch	Scratch	Test estadístico
[17]	Colegios	2.5h	15	Scratch, Lego	Varias	Test, Grupos, Tareas, Estadística
[11]	Colegios	20h	11 - 13	Scratch, Pico-board, Raspberry Pi	Prog. Visual	Grupo, Análisis descriptivo, Estadística
[18]	Colegio	12h	9 - 10	Talleres T/P	Hab. de PC	Grupo, Test

Capítulo 4

Requisitos y tecnologías

Para implementar la plataforma online de este trabajo se necesitó la utilización de un framework. Aquí se detallará cómo fue el proceso de selección y la realización de la plataforma.

4.1. Requisitos fundamentales

Se quiere desarrollar una plataforma web que sirva de herramienta para facilitarle al profesorado la impartición cursos de pensamiento computacional y la obtención informes de evaluación de su alumnado, sin necesidad de poseer grandes conocimientos en ciencias de la computación. Esta interfaz deberá ser sencilla e intuitiva tanto para el profesorado como para el alumnado, ya que se pretende fomentar el desarrollo del pensamiento computacional en edades tempranas.

Además la plataforma deberá ofrecer la posibilidad de generar cursos de diferente duración, con actividades orientadas a una edad de alumnado específico, y ayudar al profesor a llevar un seguimiento del transcurso del mismo. De forma opcional, el profesorado podría crear actividades propias y compartirlas con la comunidad.

Por último, el profesor debería poder clasificar a sus alumnos en dos grupos, un grupo experimental y otro de control, de manera que permita obtener un estudio comparativo al final del curso.

4.2. Búsqueda de un Framework

Se realizó una búsqueda de numerosos framework que fueran tendencia en nuestra actualidad, y se investigó las ventajas de proporcionaban cada uno.

- **Ruby on Rails.** Entorno de desarrollo web de código abierto que permite crear aplicaciones web que acceden a bases de datos.

- Utiliza el MCV (Modelo Vista Controlador)
 - Evita repetición de código
 - Acceso a librerías de bases de datos
 - Plugin de seguridad
 - Framework de validación de formularios
 - Framework de manejo de sesiones
 - Debuging
 - Pocas dependencias
- **CakePHP.** Framework de desarrollo rápido de aplicaciones con MCV para PHP inspirado en Ruby on Rails.
- Construye aplicaciones de forma simple y rápida
 - Capa de acceso flexible a bases de datos
 - Herramientas de validación
 - Protección CSRF
 - Protección de manipulación en formularios
 - Prevención de inyección de SQL
 - Auententicaciones
 - Validaciones
- **Zend Framework.** Framework orientado a objetos para PHP5 diseñado para eliminar detalles tediosos de código.
- Diseño MCV
 - Pocas interdependencias
 - Soporta múltiple bases de datos como MySQL, Oracle, Microsoft SQL Server, etc.
 - Composición y entrega de correo electrónico
 - Validación de formularios
 - Almacena contraseñas con bcrypt y encripta con AES-256
- **Django.** Framework de aplicaciones web Python de alto nivel que sigue el MCV.
- Mapeado relacional de objetos (ORM) donde describe el diseño de la base de datos en lenguaje Python.
 - Interfaz de administración dinámica automática

- Sistema de serialización y validación de formularios
 - Framework de testing
 - Migración de BD
 - Ligero, Standalone de servidor web
 - Rapido
 - Seguro
- **Yii.** Framework PHP para el desarrollo de aplicaciones web que promueve el principio DRY.
 - Acceso de bases de datos
 - AJAX
 - Autenticación
 - Inyección de SQL
 - Pruebas unitarias y de funcionalidad
 - Entrada y validación de formularios
 - Control de accesos basados en roles
 - **Pylons.** Framework Web para Python que combina las mejores ideas de Ruby, Python y Perl para obtener gran flexibilidad.
 - AJAX
 - Controlador para MCV
 - Marco de pruebas con ruido
 - Promueve reusabilidad y separa funcionalidades en módulos
 - Desarrollo predecible y productivo

4.3. Elección de un Framework

Tras realizar un estudio de los diferentes framework que podría utilizar se decidió que la elección estaría entre Ruby on Rails y Django. A continuación, compararemos ambos y sacaremos una conclusión:

Framework	Lenguaje	MVC	Escritura	Librería estándar	Aplicaciones reales
Ruby on Rails	Ruby	Sí	Permite escritura ágil y cómoda	Moderna	Twitter, Soundcloud, Twitch
Django	Python	Sí	Se escribe para que sea legible	Basada en Java	Instagram, Pinterest, Nasa Science

Como vemos ambos guardan similitudes, y ambos han permitido la creación de plataformas webs de gran importancia hoy en día. Sin embargo, debido a su comodidad la hora de escritura de código y al mayor conocimiento en el lenguaje de programación Ruby, la elección final será **Ruby on Rails**.

4.4. Búsqueda y desarrollo de una metodología

La metodología que se propone para el desarrollo y la evaluación del pensamiento computacional es la siguiente:

- Cada curso está definido por una duración y una dificultad.
- Separar al alumnado en dos grupos: grupo experimental y grupo de control
- Al comienzo de cada curso se realizará un test que evalúe el pensamiento computacional para tener una medida de como están los alumnos, y se realizará otro test al finalizar el curso para observar los resultados. Estos test serán iguales para ambos grupos.
- Las actividades serán diferente para ambos grupos. El grupo experimental tendrá actividades destinadas al desarrollo computacional, mientras que el grupo de control tendrá actividades que no tengan que ver con el pensamiento computacional.
- Cada curso tendrá actividades acorde a su dificultad, pero al inicio del mismo tendrá algunas de dificultad menor, y al final de dificultad superior.

Por último, tras numerosas investigaciones y lecturas se ha decidido que las actividades para el grupo experimental serán obtenidas de las siguientes plataformas debido a la calidad de las mismas, y la variedad de contenido que presentan:

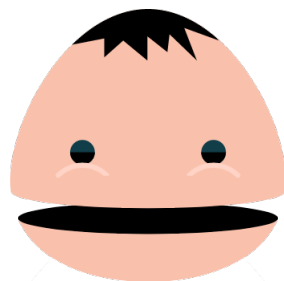
- Code.org [4]
- Teaching London Computing [16]
- CSUnplugged [5]

Capítulo 5

Plataforma Web

5.1. Descripción

Abstractly, nombre que toma la plataforma web, está destinado a promover y desarrollar el pensamiento computacional en edades tempranas dentro de las aulas. Proporciona al profesorado todas las herramientas necesarias para poder impartir un curso desde cero, sin necesidad de poseer conocimientos específicos, y además les permite gestionar tanto a sus alumnos como a las actividades que realizan, llevar un seguimiento del progreso, y obtener informes estadísticos día a día.



5.2. Configuración

5.2.1. Metodología y despliegue

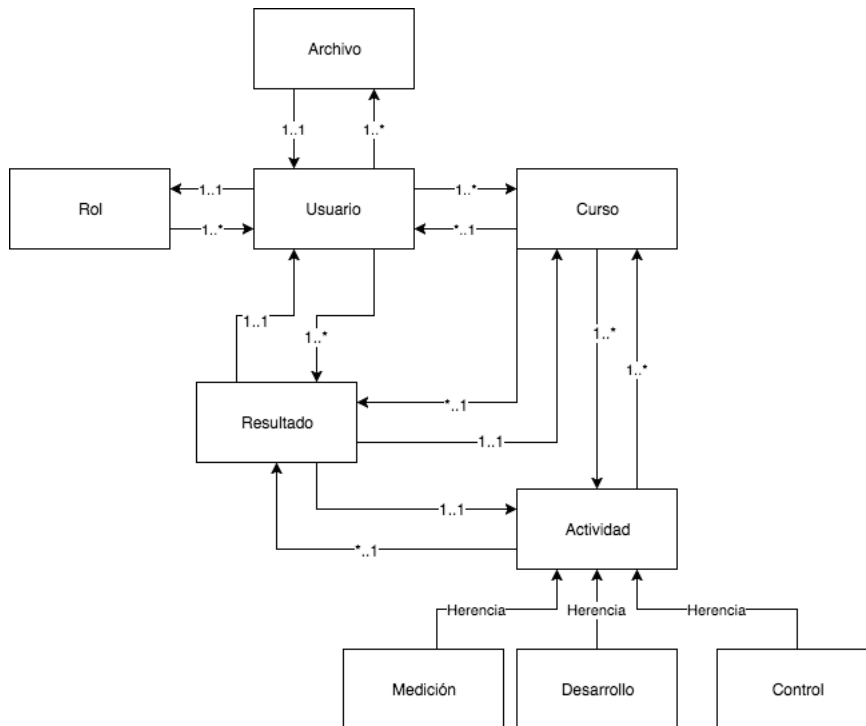
Abstractly ha sido desarrollada con Ruby on Rails, siguiendo las directrices del Modelo-Vista-Controlador (MVC), y algunas de sus funcionalidades con JavaScript.

Actualmente se encuentra desplegada en Heroku [8], una plataforma de computación en la nube, con un Dynos gratuito. Su enlace es el siguiente:

<http://abstractly.herokuapp.com/>

5.2.2. Base de datos

En un principio se quiso utilizar MySQL como base de datos, pero debido a las restricciones que presentaba un Dynos gratuito, se tuvo que utilizar PostgreSQL.



5.2.3. Gemas

La programación de la plataforma se basa en Ruby, y una de las características y funcionalidades que presenta este lenguaje es la posibilidad de la incorporación de gemas a nuestros proyectos. Una gema es un paquete que se instala y está listo para ser utilizado.

En Abstractly se han incluido las siguientes gemas:

- **PG**. Permite la utilización de PostgreSQL como base de datos para el Active Record.
- **Bcrypt**. Facilita la encriptación de contraseñas de los usuarios de la plataforma.
- **Seed Dump**. Permite la exportación de la base de datos como semillas.
- **CarrierWave**. Facilita la subida de archivos y su inclusión dentro de los modelos de Active Record.
- **Sass-Rails**. Permite usar SCSS para las hojas de estilos.

- **Uglifier**. Permite comprimir los archivos de JavaScript.
- **JQuery-Rails**. Inclusión de JQuery en la plataforma.
- **Bootstrap-Sass**. Permite la utilización de Bootstrap.

5.3. Diseño

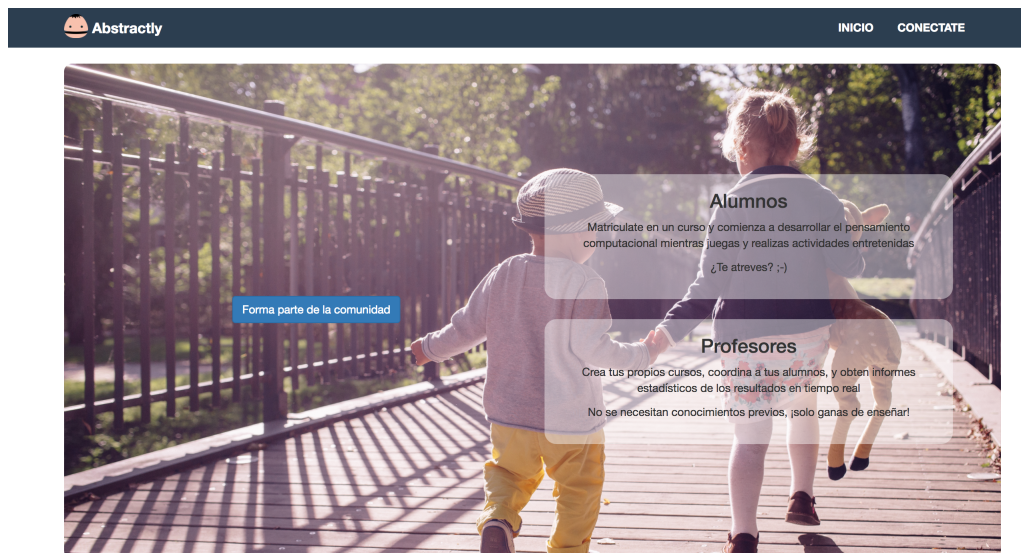
Uno de los puntos fuertes que tiene la plataforma es su diseño. Abstractly presenta un diseño sencillo e intuitivo, ya que está destinada principalmente a niños de temprana edad, los protagonistas de nuestra plataforma, y por ello se ha querido utilizar un diseño colorido y minimalista, con botones y paneles que permiten observar todo de una manera mas clara, limpia, e interactiva.

Además, en segundo plano pero no por ello menos importante, Abstractly también esta destinada a ser manejada por profesores que no tienen por qué tener grandes conocimientos informáticos, y cabe recordar que el propósito de la misma es facilitar la difusión del pensamiento computacional. Es por esto que se han incorporado técnicas que faciliten uso como Drag-and-Drop para la organización del curso, gráficos personalizables para la generación de informes, una vista general del curso dónde encontrar todo lo necesario para su administración, y la posibilidad que el profesorado se pueda involucrar en la iniciativa y dejar su semilla, permitiéndoles proponer actividades propias para la plataforma con pocos clicks de ratón.

5.4. Funcionamiento

5.4.1. Inicio

La página principal de Abstractly presenta en unas pocas líneas la misión principal de esta plataforma e invita al usuario a participar en la experiencia, ya sea en el rol de estudiante o profesor. Podemos apreciar desde el inicio el diseño minimalista e intuitivo que encontraremos a lo largo de a la plataforma.



5.4.2. Creación de usuarios

La creación de un usuario en la plataforma es bastante sencilla, consta de un formulario en el que se pide unos pocos datos, como su nombre completo, su fecha de nacimiento (la cual se puede introducir con un calendario interactivo), y su sexo (niño o niña).

Se ha decidido utilizar como método de identificación un “nick”, un apodo, para agilizar de esta manera el registro de los usuarios, ya que la utilización de un correo electrónico requeriría de verificación del mismo y es algo bastante engorroso, tanto para el alumnado como para el profesorado, y la política de Abstractly es facilitar en todo momento su interacción con la plataforma. No obstante, permitimos la inserción de un correo electrónico como información opcional, por si se desea que los alumnos o otros usuarios puedan ponerse en contacto con el profesor/administrador por ejemplo.

Por último, también es posible subir una imagen para asociar a un perfil de usuario y, por supuesto, es opcional.

Por defecto, todos los usuarios obtienen el rol de usuario “básico” en el momento de crearse una cuenta. Existen diferentes maneras de cambiar este rol:

- Rol “profesor”. Los usuarios que quieran ser profesores deberán pedirlo a la plataforma y posteriormente les será concedido. Se ha querido que esto sea así para evitar que cualquier usuario pueda crearse una cuenta con privilegios de profesor y perjudique a la plataforma, ya sea con la creación de cursos innecesarios o sin alumnado, proponiendo constantemente actividades inapropiadas, o cualquier otra situación que implique permisos de profesor.

Abstractly INICIO CONECTATE

Regístrate

Usuario

Contraseña

Confirmación de contraseña

Nombre completo

Fecha de nacimiento

Género
 Niño Niña

Correo electrónico (opcional)

Imagen de perfil (opcional)
 Ningún archivo seleccionado

Abstractly INICIO CONECTATE

Conectate

Usuario

Contraseña

[¿Aún no tienes cuenta? Regístrate ahora](#)

Computational Thinking Platform by Alexis Daniel Fuentes Pérez GitHub

- Roles de estudiante “experimental” o “control”. Los otorga el profesor de un curso a sus alumnos matriculados en el mismo. Más adelante se explicará de forma detallada el proceso.

Destacar aquí que los permisos de administrador de la plataforma no cuentan como roles por motivos de seguridad, y estos solo pueden otorgarse desde el propio servidor. Además, esta característica permite a los propios administradores alternarse entre los cuatros roles diferentes que existen (básico, profesor, experimental, y control) sin perder sus permisos de administrador, con el fin para probar nuevas funcionalidades en producción o bugs existentes por ejemplo.

5.4.3. Navegación

La navegación dentro de la plataforma es bastante intuitiva y fácil de entender. Consiste en un menú superior con diferentes secciones.

Sin login en la plataforma:

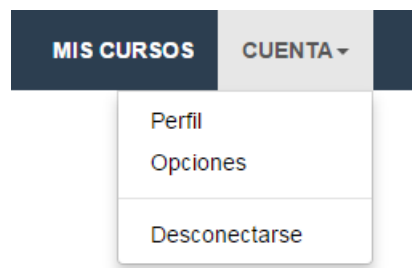
- **Inicio.** Página principal de la plataforma.
- **Conéctate.** Sección de acceso a la plataforma.

- **Quiénes somos.** Información sobre el autor y la plataforma.



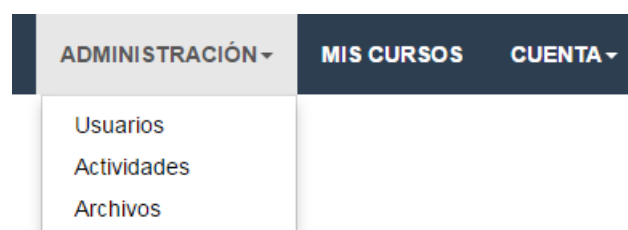
Con sesión iniciada en la plataforma:

- **Mis Cursos.** Desde aquí se realiza el proceso de matriculación, la creación de cursos, y el acceso a los cursos a los que perteneces.
- **Cuenta.** Menú desplegable con interacciones relacionadas con tu cuenta.
 - **Perfil.** Acceso a tu perfil de usuario.
 - **Opciones.** Acceso a las opciones de tu cuenta.
 - **Desconectarse.** Permite cerrar sesión en la plataforma.



Además, los usuarios con permisos de administrador dispondrán de una sección extra:

- **Administración.** Conjunto de herramientas para administrar la plataforma.
 - **Usuarios.** Administración de usuarios.
 - **Actividades.** Administración de actividades.
 - **Archivos.** Almacenamiento de archivos en la plataforma.

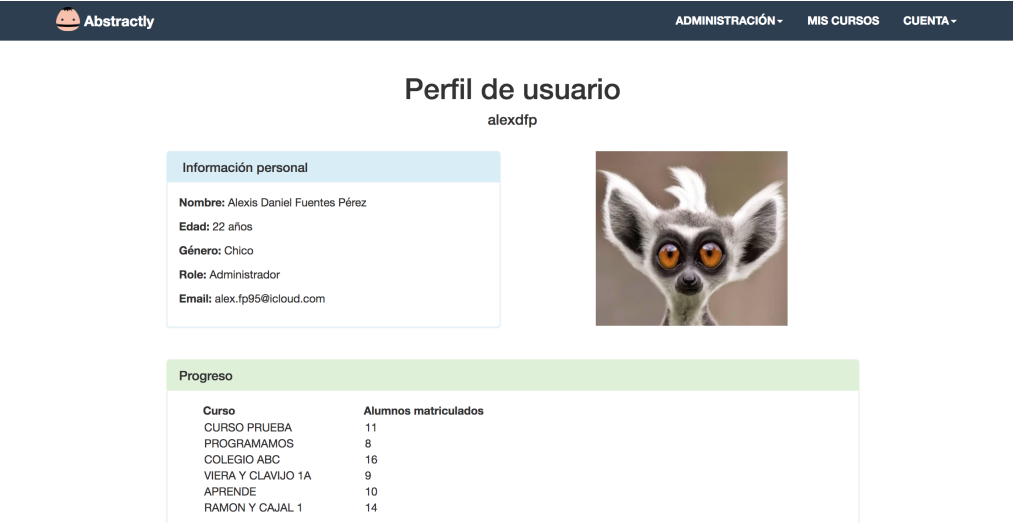


5.4.4. Perfil

Cada usuario tiene su propio perfil. En él se puede observar su información personal, como su nombre, edad, o rol actual en la plataforma. También se podrá ver su correo electrónico, si el usuario decide establecer, así como una imagen que es de carácter opcional.

Destacar que también se puede observar un pequeño resumen de su actividad en Abstractly, y aquí se diferenciarían dos tipos de perfil diferentes.

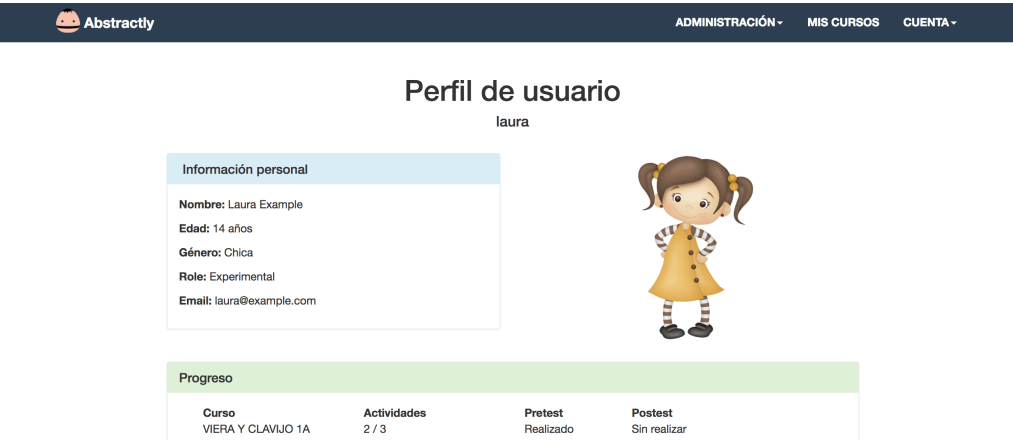
- Perfil de **alumnos**. Se muestra un listado de los cursos en los que ha estado, si realizó las pruebas de medición, y el porcentaje de actividades realizadas entre las totales de ese curso.



The screenshot shows the user profile for 'alexdfp' on the Abstractly platform. The header includes the Abstractly logo and navigation links for 'ADMINISTRACIÓN', 'MIS CURSOS', and 'CUENTA'. The profile title is 'Perfil de usuario alexdfp'. It features a 'Información personal' section with the following details: Nombre: Alexis Daniel Fuentes Pérez, Edad: 22 años, Género: Chico, Role: Administrador, and Email: alex.fp95@icloud.com. To the right is a profile picture of a lemur. Below this is a 'Progreso' section with a table of courses and the number of enrolled students.

Curso	Alumnos matriculados
CURSO PRUEBA	11
PROGRAMAMOS	8
COLEGIO ABC	16
VIERA Y CLAVIJO 1A	9
APRENDE	10
RAMON Y CAJAL 1	14

- Perfil de **profesor**. Se muestra un listado de los cursos que ha dado, y la cantidad de total de alumnos que tuvo en cada uno de ellos.



The screenshot shows the user profile for 'laura' on the Abstractly platform. The header includes the Abstractly logo and navigation links for 'ADMINISTRACIÓN', 'MIS CURSOS', and 'CUENTA'. The profile title is 'Perfil de usuario laura'. It features a 'Información personal' section with the following details: Nombre: Laura Example, Edad: 14 años, Género: Chica, Role: Experimental, and Email: laura@example.com. To the right is a profile picture of a cartoon girl. Below this is a 'Progreso' section with a table showing course progress.

Curso	Actividades	Pretest	Postest
VIERA Y CLAVIJO 1A	2 / 3	Realizado	Sin realizar

Por último, los alumnos solo pueden ver su perfil, pero el profesorado (y los administradores) pueden ver su perfil y también el del alumnado.

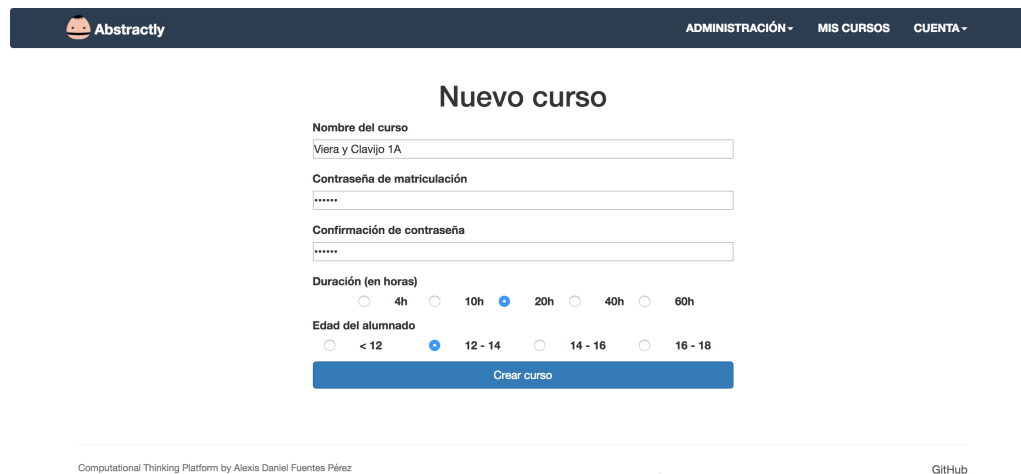
5.4.5. Creación de cursos

La creación de cursos está disponible para el profesorado desde la sección “*Mis Cursos*”. Un profesor puede crear y realizar distintos cursos al mismo tiempo con diferente alumnado.



Durante la creación del curso el profesor debe darle un nombre y una contraseña, que serán utilizados por su alumnado para el proceso de matriculación, y establecerá la configuración del curso: la duración y la dificultad del mismo, que no es más que a la edad a la que estará dirigido.

La duración limitará la cantidad de actividades que se realizarán en el curso, y la dificultad fijará las pruebas de medición (pre-test y post-test) y el tipo de actividades, aunque estas últimas pueden ser eliminadas del curso, y ser sustituidas por otras a elección del profesor siempre que estas nuevas no superen el tiempo total de duración estimado. Esto será explicado más adelante.



Una vez el profesor haya enviado los datos, se procederá a la creación del curso. Abstractly de forma automática rellenará el curso con las pruebas de medición y actividades correspondientes a los parámetros indicados, y se aplicará la metodología de la siguiente manera para ambos grupos, experimental y control:

■ **Curso: < 12 años**

- 70 % de la duración*: actividades correspondientes a “< 12 años”
- 30 % de la duración*: actividades correspondientes a “12 - 14 años”

■ **Curso: 12 - 14 años**

- 20 % de la duración*: actividades correspondientes a “< 12 años”
- 60 % de la duración*: actividades correspondientes a “12 - 14 años”
- 20 % de la duración*: actividades correspondientes a “14 - 16 años”

■ **Curso: 14 - 16 años**

- 20 % de la duración*: actividades correspondientes a “12 - 14 años”
- 60 % de la duración*: actividades correspondientes a “14 - 16 años”
- 20 % de la duración*: actividades correspondientes a “16 - 18 años”

■ **Curso: 16 - 18 años**

- 30 % de la duración*: actividades correspondientes a “14 - 16 años”
- 70 % de la duración*: actividades correspondientes a “16 - 18 años”

(*): Duración total del curso - duración de las actividades de medición

Las actividades de medición ya se encuentran asignadas a cada grupo de edad, y ya están conformadas por preguntas de diferentes dificultad. Esto será explicado mejor mas adelante, en un apartado específico.

The screenshot shows the course management interface for 'VIERA Y CLAVIJO 1A' on the Abstractly platform. At the top, there is a navigation bar with the Abstractly logo and user options: ADMINISTRACIÓN, MIS CURSOS, and CUENTA. Below the course title, there are four action buttons: 'Cambiar vista', 'Ver informes', 'Gestionar alumnos', and 'Añadir actividades'. The main content area displays the course progress, indicating that 0.0 hours of the 10-hour course have been completed. There are two sections: 'Medición [Pre-test]' and 'Actividades [Experimental]'. The 'Medición' section contains a 'PRE-TEST' activity with an 'Activar' button. The 'Actividades' section contains two activities: 'PUZZLE: APRENDER DRAG & DROP' and 'MAPAS FELICES', each with 'Activar' and 'Quitar del curso' buttons.

5.4.6. Matriculación en un curso

Para matricularte en un curso en Abstractly basta con ir a “Mis Cursos”, dónde te saldrá un listado con aquellos cursos en los que estás matriculado, y a continuación acceder a “Buscar Cursos”.

Te saldrá un listado con los cursos activos en este momento. Para matricularte en uno, basta con entrar en el que te interesa e introducir la contraseña de matriculación. Esta deberá ser proporcionada por el profesor a su alumnado.



The screenshot shows the Abstractly user interface. At the top, there is a dark blue navigation bar with the Abstractly logo on the left and three menu items: 'ADMINISTRACIÓN', 'MIS CURSOS', and 'CUENTA'. Below the navigation bar, the page title is 'Matricula' followed by the course name 'VIERA Y CLAVIJO 1A'. Underneath, there is a label 'Contraseña de matriculación' above a white text input field. At the bottom of the form is a blue button labeled 'Matricularme'.

Una vez dentro te saldrá el mensaje de bienvenida a un curso, y se te redirigirá al mismo.

Cabe destacar que un alumno no puede matricularse en más de un curso activo al mismo tiempo, ya que esto “falsearía” los datos de medición que se obtendría en dichos cursos ya que dicho alumno estará realizando más actividades de lo que está establecido para su grupo.

5.4.7. Vista del curso

Los cursos son el objeto más importante dentro de la plataforma, ya que es a través de ellos donde se realiza la organización del alumnado, el desarrollo del pensamiento computacional, la obtención de informes, la organización de las actividades, etc. Es por esto que se ha querido que en la vista del curso, tanto el profesorado como el alumnado, tenga a su alcance todo lo necesario para su realización y, al mismo tiempo, que se muestra de forma sencilla e intuitiva.

The screenshot shows the course management interface for 'COLEGIO LA LAGUNA' on the Abstractly platform. At the top, there is a navigation bar with the Abstractly logo, 'MIS CURSOS', and 'CUENTA'. Below this, the course title 'COLEGIO LA LAGUNA' is displayed. A toolbar contains four buttons: 'Cambiar vista' (blue), 'Ver informes' (orange), 'Gestionar alumnos' (blue), and 'Añadir actividades' (green). The main content area is titled 'Progreso del curso' and features a green progress bar at 56.25%. Below the progress bar, it states 'Se han realizado 2.25 horas de las 4 horas que conforman el curso'. The interface is divided into three sections: 'Medición [Pre-test]' (purple header) containing a 'PRE-TEST' box; 'Actividades [Experimental]' (blue header) containing a list of activities with their respective action buttons; and 'Medición [Post-test]' (purple header) containing a 'POST-TEST' box with an 'Activar' button.

Actividades [Experimental]	
PUZZLE: APRENDER DRAG & DROP	Calificar actividad
NUMEROS BINARIOS I	Calificar actividad
ACTIVIDAD PROPIA	Proponer Calificar actividad
MUÉVELO, MUÉVELO	Cerrar actividad
MAPAS FELICES	Activar Quitar del curso
LABERINTO: SECUENCIA	Activar Quitar del curso

Como se puede observar en la imagen anterior, en la parte superior del curso se encuentra una botonera de acciones para el profesorado (únicamente visible y accesible para ellos).

- **Cambiar vista.** Permite cambiar la vista del curso entre ‘Grupo Experimental’ y ‘Grupo de Control’. Más adelante se explicará en qué consisten estos grupos.
- **Ver informes.** Permite acceder a la generación automática de informes del curso.
- **Gestionar alumnos.** Permite administrar al alumnado matriculado.

- **Añadir actividades.** Abre la herramienta para añadir más actividades al curso.

En la cabecera aparece el “**Progreso del curso**”, donde se puede apreciar para cada vista del curso el transcurso del mismo y permite observar si se va bien de tiempo, ya que no es lo mismo el tiempo estimado que se tiene para la realización de cada actividad, que el tiempo real que se tardó en realizarla. Ese tiempo lo introduce el profesor con el registro de cada actividad (se explicará mejor en apartados siguientes), y la plataforma se encarga de realizar los cálculos necesarios y actualizar esta barra de progreso.

A continuación aparecen los bloques de actividades del curso:

- **Medición [Pre-test].** Actividad de medición que debe realizarse con la inicialización del curso. Permite obtener una medida de cómo están las capacidades de resolución de problemas y/o pensamiento computacional de cada alumno y del grupo antes de comenzar a desarrollarlas. Son iguales para ambos grupos (experimental y control).
- **Actividades.** Bloque de actividades para el grupo determinado (experimental o control, ya que son diferentes para cada grupo) incluidas dentro del curso. Por defecto están ordenadas por dificultad, pero se ha implementado un sistema de Drag-and-Drop para que el profesor pueda cambiarlo a su gusto.
- **Medición [Post-test].** Actividad de medición que debe realizarse con la finalización del curso. Permite obtener una medida de cómo están las capacidades de resolución de problemas y/o pensamiento computacional de cada alumno y del grupo al finalizar el estudio. Son iguales para ambos grupos (experimental y control).

Cada actividad dentro de un curso tiene un ciclo de vida, que es marcado por el profesor y se encuentra comprendido por:

- **Fase 1 (En Espera).** La actividad está en el curso, pero no es accesible por el alumnado para su realización. En esta fase la actividad aún puede ser removida del curso por el profesor.
- **Fase 2 (Activada).** La actividad se encuentra activada para su realización. Se puede observar visualmente dicha condición, ya que se vuelve de un color azulado claro. Permite al alumnado entrar a la misma y registrar su realización.
- **Fase 3 (Finalizada).** La actividad se encuentra finalizada. El profesor deja su registro de la actividad, y establece la duración real que tuvo.

Deja de ser accesible para el alumnado. Permite al profesor calificar la actividad a aquellos alumnos que registraron su realización. Visualmente la actividad deja de mostrarse como accesible. Además, si es una actividad propia del profesor se le permitirá proponerla para la plataforma (esto será explicado en detalle en el apartado 5.4.9).

5.4.8. Organización al alumnado

En la plataforma existen dos grupos de alumnado, el del grupo experimental y el del grupo de control. A continuación, explicaremos la diferencia entre ambos.

- **Grupo Experimental.** Alumnado que realizará actividades relacionadas con el desarrollo del pensamiento computacional .
- **Grupo de Control.** Alumnado que realizará actividades que ni fomentan ni desarrollan el pensamiento computacional.

¿Por qué esta separación del alumnado? Permite realizar una comparación de resultados al final del curso, y es que las pruebas de medición (pre-test y post-test) son las mismas para ambos grupos, y lo teórico nos dice que deberíamos observar una mejora de resultados del grupo experimental frente al de control.

Para organizar al alumnado en estos dos grupos el profesor dispone de un botón denominado “*Gestionar alumnos*” dentro del curso, donde podrá ver un listado con todos los alumnos matriculados, cambiarles el grupo al que pertenecen, e incluso desmatricularles.



Abstractly ADMINISTRACIÓN - MIS CURSOS CUENTA -

Alumnos matriculados

Regresar al curso

Pedro Example | Grupo Experimental | Desmatricular
Juan Exposito | Grupo Experimental | Desmatricular
Carlos Fernandez | Grupo Control | Desmatricular
Alessandro Lopez | Grupo Control | Desmatricular
Juan | Grupo Experimental | Desmatricular
Laura Example | Grupo Experimental | Desmatricular

Computational Thinking Platform by Alexis Daniel Fuentes Pérez GitHub

Por último, cada grupo dispondrá de una vista diferente del curso y los profesores podrán alternar entre ambas con el botón “*Cambiar vista*”.

5.4.9. Añadir actividades al curso

Los profesores pueden añadir mas actividades a su curso si lo desean, ya sea porque han terminado el curso antes de lo estimado o porque decidieron quitar algunas actividades y añadir otras nuevas.

Para ello deben dirigirse a la barra de acciones del curso, y seleccionar “Añadir actividades”, y se redirigirá a la siguiente vista.

The screenshot shows the 'Actividades disponibles' page. At the top, there is a navigation bar with the 'Abstractly' logo and links for 'ADMINISTRACIÓN', 'MIS CURSOS', and 'CUENTA'. Below the navigation bar, the title 'Actividades disponibles' is centered. There are two buttons: 'Crear actividad' (green) and 'Regresar al curso' (blue). The page is divided into two main sections: 'Actividades de desarrollo' and 'Actividades de control'. Each section has a 'Filtro: Edad' dropdown menu. Below each filter is a table of activities. In the 'Actividades de desarrollo' section, there are four activities: 'Puzzle: Aprender Drag & Drop', 'Laberinto: Secuencia', 'Mapas Felices', and 'Muévelo, muévelo'. Each activity has an 'Edad' column showing '< 12' and an 'Añadir al curso' button. In the 'Actividades de control' section, there is one activity: 'Origami: Perrito', also with an 'Edad' of '< 12' and an 'Añadir al curso' button. At the bottom of the page, there is a footer with the text 'Computational Thinking Platform by Alexis Daniel Fuentes Pérez' and a 'GitHub' link.

Se le mostrarán las actividades de la plataforma que no pertenecen a su curso separadas en dos paneles, uno de actividades para el grupo experimental y otro para el de control. Dispone además de unos filtros para localizar actividades de interés más rápido.

Para cada actividad se mostrará la edad para la que está dirigida, y las acciones.

- **Añadir al curso.** Si se dispone de tiempo en grupo al que va dirigida la actividad.
- **No dispone de tiempo.** No le dejará añadirla hasta que consiga algo de tiempo en ese grupo.

Además, la plataforma permite que el profesor cree sus propias actividades y las añada a su curso. Para ello debe darle al botón verde que aparece en la parte superior de “Añadir actividades”. Este procedimiento será explicado con detalle en el apartado ‘Creación de actividades’.

5.4.10. Registrar una actividad

El seguimiento de la realización de actividades se realiza mediante los registros. En un registro se puede llegar a guardar el usuario, la actividad en sí, el curso, la duración de la actividad, la edad que tenía el usuario al realizarla, una calificación, una retroalimentación sobre la dificultad percibida, y una retroalimentación de satisfacción con la misma.

Estos registros sirven para determinar la realización de una actividad por parte de un usuario, ya sea estudiante o profesor. Sin embargo, el procedimiento para realizarlo puede llegar a cambiar un poco:

- **Estudiante.** Disponen de un botón al final de cada actividad para registrar su realización. Se les pedirá que puntué la satisfacción y la dificultad percibida.

Abstractly MIS CURSOS CUENTA ▾

Laberinto: Secuencia

Registro de realización

Curso: VIERA Y CLAVIJO 1A

Grado de satisfacción

Bajo Medio Alto

Grado de dificultad

Fácil Adecuado Difícil

Guardar

Computational Thinking Platform by Alexis Daniel Fuentes Pérez GitHub

- **Profesor.** Cuando el profesor cierra una actividad en el curso, se abre automáticamente el registro de la actividad y se le pide la duración real que ocupó y la satisfacción y dificultad observada. De esta manera, la actividad pasará a finalizada y se actualizará el progreso del curso.

Abstractly ADMINISTRACIÓN ▾ MIS CURSOS CUENTA ▾

Laberinto: Secuencia

Registro de realización

Curso: VIERA Y CLAVIJO 1A

Tiempo real invertido (en minutos)

Grado de satisfacción

Bajo Medio Alto

Grado de dificultad

Fácil Adecuado Difícil

Guardar

Computational Thinking Platform by Alexis Daniel Fuentes Pérez GitHub

Cuando un registro se almacena en la plataforma, se actualizan de forma automática los niveles de satisfacción y dificultad de esa actividad dentro de la plataforma. Estos datos son accesibles por los administradores, y los veremos más adelante.

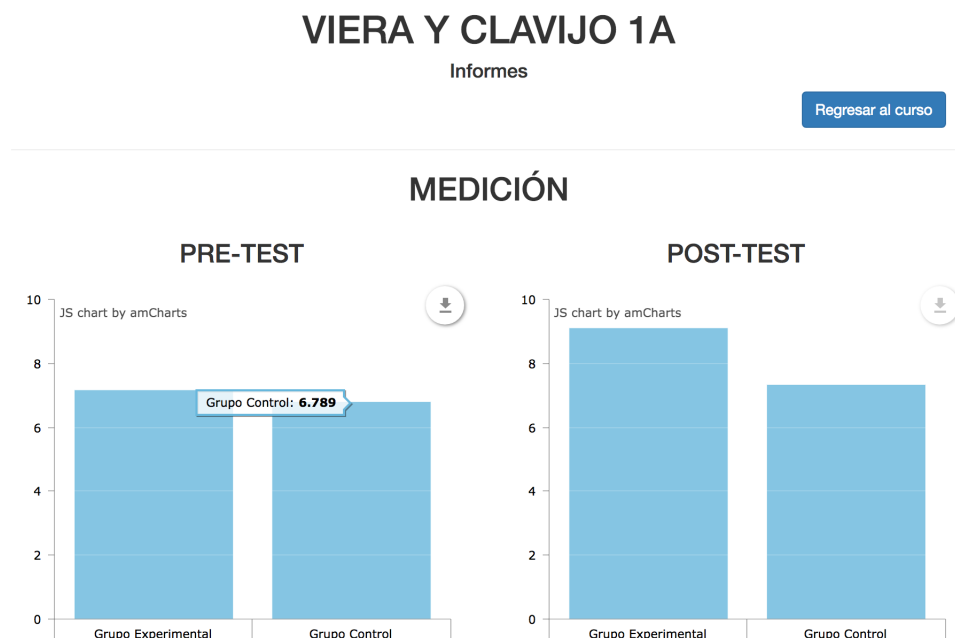
5.4.11. Generación de informes

Uno de los puntos fuertes de la plataforma es la generación automática de informes para los cursos. Se ha utilizado la biblioteca externa de *amCharts* [1] para facilitar la representación gráfica de los datos.

Estos informes permiten obtener de una manera fácil y sencilla datos reales sobre el progreso del curso, como calificaciones, satisfacciones, comparaciones entre los diferentes grupos, etc. Estos datos son representados mediante gráficos y pueden descargarse (*.jpg*, *.png*, *.svg*, *.pdf*), exportarse (*.csv*, *.xls*, *.json*) o imprimirse desde la propia plataforma.

A continuación veremos los diferentes tipos de informes que nos genera:

- **Informes de medición.** Se realizan diferentes tipos de informes en relación a las calificaciones obtenidas en las actividades de medición (pre-test y post-test). En cada uno de los gráficos, siempre se enfrentan el resultado obtenido por el grupo experimental y el resultado obtenido por el grupo de control, y a su vez, cada gráfico se enfrenta para el pre-test y el post-test.
 - **Curso.** Se muestra la media de las calificaciones obtenidas por cada grupo.

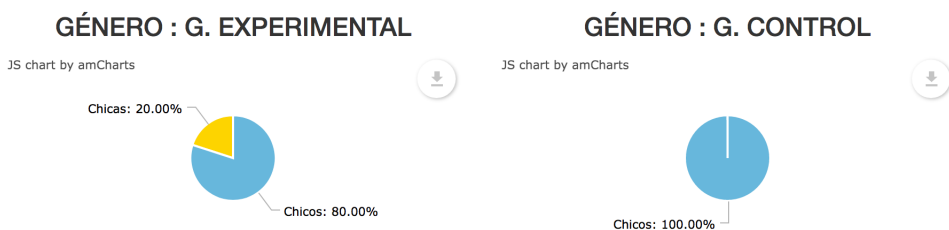


- **Género.** Se muestra la media de las calificaciones obtenidas por género para cada grupo.
- **Edad.** Se muestra la media de las calificaciones obtenidas por edad para cada grupo.

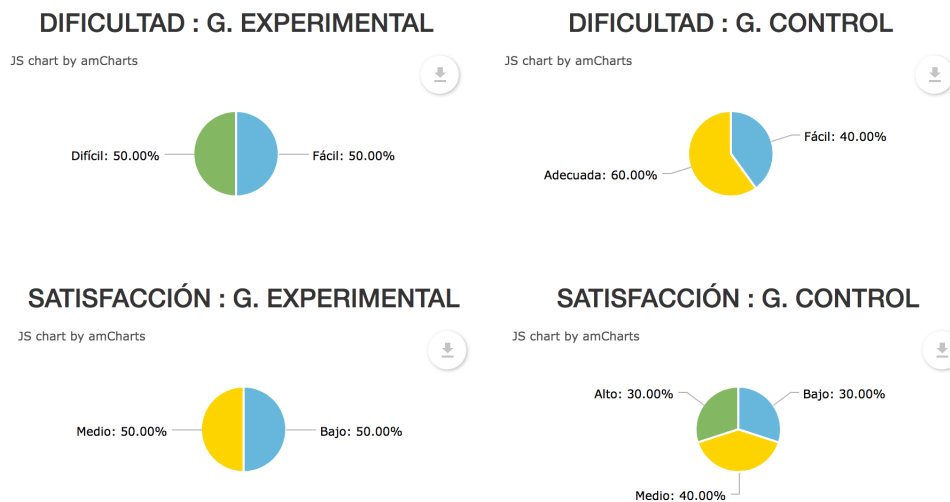


■ **Informes del curso.** Estos informes muestran información significativa del curso.

- **Género.** Muestra el porcentaje de alumnos de cada género por grupo.



- **Dificultad.** Muestra los porcentajes de dificultad medios para cada actividad realizada en el curso. Cada actividad puede obtener los valores de dificultad: fácil, apropiada, y difícil.
- **Satisfacción.** Muestra los porcentajes de satisfacción medios para cada actividad realizada en el curso. Cada actividad puede obtener los valores de satisfacción: bajo, medio, y alto.



5.4.12. Creación de actividades

Abstractly permite la creación de actividades desde la propia plataforma. Existen dos maneras de crearlas:

- Cuenta de **profesor**. Desde la vista de “Añadir actividades” los profesores tienen también la opción de “Crear actividad”. Podrán crear actividades para su grupo experimental o su grupo de control. Estas actividades solo aparecerán en su curso, pero Abstractly les ofrece la posibilidad de contribuir con el crecimiento de la plataforma y una vez esta actividad haya sido finalizada, le saldrá una opción llamada “Proponer”. Además el profesor podrá modificarla tantas veces como quiera mientras la actividad se encuentra en ‘Fase I’ (En Espera). Los administradores podrán ver estas actividades propuestas, e incluirlas o no a la plataforma.
- Cuenta con permisos de **administrador**. En la barra de herramientas de “Administrador” se puede acceder a la administración de actividades, y desde ahí crear nuevas.

Al crear una actividad se le asigna un título, una duración estimada de realización, un tipo (experimental, control, o medición), y además es necesario especificar si es unplugged o no (que pueda ser realizada sin necesidad de un ordenador), y a qué edad pertenece. Por último, toca escribir el contenido de la actividad, que bien puede ser en texto plano o HTML.

La única diferencia que existe entre el creador de actividades del profesor y el de un administrador, es que el administrador puede crear actividades de medición y el profesor no, y que las actividades creadas por un administrador son directamente insertadas en la plataforma.

Crear nueva actividad

Título

Duración (en minutos)

Tipo

Desarrollo Medición Control

Unplugged

True False

Edad

< 12 12 - 14 14 - 16 16 - 18

Contenido

Guardar

5.4.13. Actividades de medición


Las actividades de medición, también conocidas como pre-test y post-test, nos permiten evaluar las capacidades de pensamiento computacional del alumnado.

En un principio se planteó la realización de nuestros propios test para la evaluación, pero estos no son triviales y requieren de la evaluación por parte de un jurado para ser validados, y debido a la escasez de tiempo decidimos descartar la idea. En su lugar, se decidió incorporar dentro de la plataforma los test de Bebras (UK) [2], ya que están validados y presentan una gran variedad en cuanto a dificultades.

Estos test tienen duración límite de 40 minutos y están formados por un total de 18 preguntas, que se dividen en 6 de dificultad A (fácil), 6 de dificultad B (media), y 6 de dificultad C (difícil). La forma de puntuar las respuestas es la siguiente:

Dificultad	Correcta	Incorrecta	Sin responder
A	+6 puntos	0 puntos	0 puntos
B	+9 puntos	-2 puntos	0 puntos
C	+12 puntos	-4 puntos	0 puntos

Toda esta información se presenta al alumnado en una pantalla previa al comienzo de la prueba.

 Abstractly
ADMINISTRACIÓN - MIS CURSOS - CUENTA -

Bebras Intermediate (C15)

Objetivo
Resuelve todos los problemas que puedas en **40 minutos**
No se espera que los resuelvas todos

Reglas
Existen tres niveles de dificultad: A, B, y C
Cada nivel de dificultad está formado por **6 problemas**
Las reglas de puntuación son las siguientes

Dificultad	Correcta	Incorrecta	Sin responder
A	+6 puntos	0 puntos	0 puntos
B	+9 puntos	-2 puntos	0 puntos
C	+12 puntos	-4 puntos	0 puntos

Comenzar

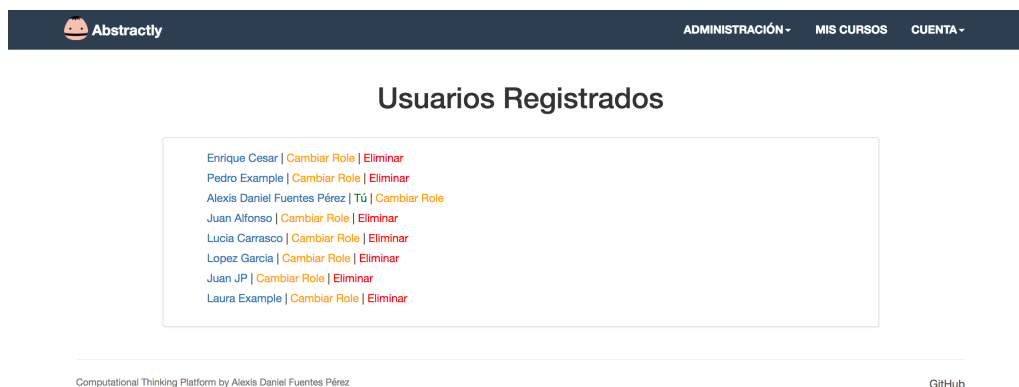
Una vez calculada la puntuación total se normaliza sobre 10 y es la calificación final que se guarda en el registro de la actividad de medición. Si el alumno abandona el test durante su realización o sale del mismo, el test se enviará automáticamente con las respuestas que tenga marcadas en ese momento. Por último destacar que estas actividades no pueden ser realizadas ni accedidas por el profesorado para evitar que los alumnos puedan ser informados del contenido de los mismos o tengan una idea previa del estilo de preguntas.

5.4.14. Administración de usuarios

Los usuarios con permisos de administrador pueden acceder a la administración de usuarios desde el menú superior en “Administración”.

En esta sección podrán ver un listado con todos los usuarios registrados en Abstractly, y se pueden realizar diferentes acciones:

- **Acceso a perfiles.** Se puede acceder al perfil de cualquier usuario, sea estudiante, profesor o administrador.
- **Modificación de rol.** Puedes cambiar el rol de cualquier usuario, permitiendo así la adjudicación controlada del rol de profesor bajo petición. Recordar que los únicos roles que existen en la plataforma son los de ‘usuario básico’, ‘alumno experimental’, ‘alumno control’, y ‘profesor’, y que la condición de administrador no cuenta como rol y se otorga únicamente desde el servidor por motivos de seguridad. Esto permite además a un administrador ser profesor, o alumno al mismo tiempo que administrador para así poder probar nuevas funcionalidades o bugs.
- **Eliminación de cuenta.** Puedes eliminar una cuenta de usuario (acción peligrosa, presenta verificación) siempre que esta no posea permisos de administrador.



Usuarios Registrados

Enrique Cesar	Cambiar Role	Eliminar
Pedro Example	Cambiar Role	Eliminar
Alexis Daniel Fuentes Pérez	TU	Cambiar Role
Juan Alfonso	Cambiar Role	Eliminar
Lucia Carrasco	Cambiar Role	Eliminar
Lopez Garcia	Cambiar Role	Eliminar
Juan JP	Cambiar Role	Eliminar
Laura Example	Cambiar Role	Eliminar

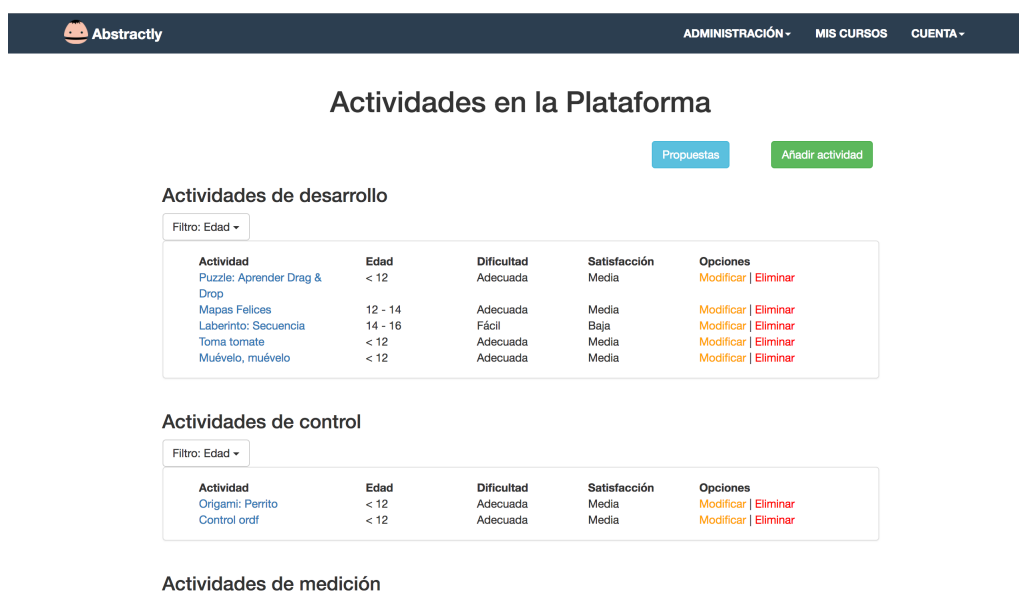
Computational Thinking Platform by Alexis Daniel Fuentes Pérez GitHub

5.4.15. Administración de actividades

Los usuarios con permisos de administrador pueden acceder a la administración de actividades desde el menú superior en “Administración”. Desde aquí podrán ver el listado de actividades incluidas dentro de la plataforma, separadas por tres grandes paneles: actividades para alumnado experimental, actividades para alumnado de control, y actividades de medición. Cada panel posee un filtro para poder localizar actividades de forma más rápida o centrarse en algún grupo en concreto.

Junto a cada actividad, se muestra la edad la que está dirigida, la retroalimentación proporcionada por el alumnado en cuanto a la dificultad percibida y a la satisfacción, y las opciones que se puede realizar:

- **Modificar.** Permite modificar la actividad, ya sea por algún error o por datos provenientes de la retroalimentación de los usuarios.
- **Eliminar.** Borra la actividad de la plataforma (acción peligrosa, pide confirmación).



Actividades en la Plataforma

Propuestas [Añadir actividad](#)

Actividades de desarrollo

Filtro: Edad ▾

Actividad	Edad	Dificultad	Satisfacción	Opciones
Puzzle: Aprender Drag & Drop	< 12	Adecuada	Media	Modificar Eliminar
Mapas Felices	12 - 14	Adecuada	Media	Modificar Eliminar
Laberinto: Secuencia	14 - 16	Fácil	Baja	Modificar Eliminar
Toma tomate	< 12	Adecuada	Media	Modificar Eliminar
Muévelo, muévelo	< 12	Adecuada	Media	Modificar Eliminar

Actividades de control

Filtro: Edad ▾

Actividad	Edad	Dificultad	Satisfacción	Opciones
Origami: Perrito	< 12	Adecuada	Media	Modificar Eliminar
Control ordf	< 12	Adecuada	Media	Modificar Eliminar

Actividades de medición

5.4.16. Alojamiento de archivos

Abstractly permite a los administradores alojar archivos dentro de la plataforma, con el fin de poder tener a nuestra disposición cualquier material necesario para las actividades, así como recursos para nuevas funcionalidades, y que no tenga que depender de terceros y evitar así problema de pérdida de archivos.

Aquí se podrá ver un listado de todos los archivos que han sido subido por administradores desde la propia plataforma, y junto a cada archivo el identificador de administrador que lo subió. Para obtener el enlace al recurso, basta con clickarlo o abrirlo.



Capítulo 6

Conclusiones y líneas futuras

Abstractly es una plataforma que presenta todos los conceptos necesarios para desarrollar y medir el pensamiento computacional y que ha sido desarrollada y pensada exclusivamente para ello, por lo que ofrece un servicio, un producto y una dedicación mucho mayor que la que podría llegar a servir un Moodle cualquiera.

Una de las cosas más importantes a destacar es que esta plataforma está preparada para que cualquier persona, docente o no, pueda llegar a impartir un curso sin necesidad de poseer conocimientos algunos en la materia, solo ganas de enseñar y fomentar este tipo de educación. Otros de sus fuertes, su diseño intuitivo y simple, permite complementar este deseo de impulsión del pensamiento computacional ya que es fácil de usar por cualquier niño, además de llamativo.

Si Abstractly se diera a conocer en los colegios y se fomentara su utilización en pocos meses podrá crecer por sí sola, debido a la retroalimentación que recibe de los alumnos que la usan y a la posibilidad que la plataforma ofrece al profesorado para deje su granito de arena y proponga nuevas actividades.

Por último, se ha intentado que esta plataforma fusione el potencial de las actividades que ofrecen otras plataformas como Code.org, con la posibilidad de medir el pensamiento computacional, de crear un formato de curso amigable y bien estructurado, y de obtener informes estadísticos en cualquier momento en todos los aspectos destacables a estudiar y, además, que estos puedan personalizarse y descargarse.

En cuanto a las líneas futuras, existen algunas tareas que quedan pendiente para mejorar en Abstractly y que no han podido terminarse o realizarse debido al tiempo disponible.

- Realización de testing. Desarrollar tests para comprobar el correcto funcionamiento de todas las características de la plataforma, y que estos se lancen de forma automática con cada nueva versión.
- Pila de actividades. Aumentar la cantidad de actividades existentes dentro de la plataforma, tanto para el grupo experimental como para el grupo de control, para mejorar la experiencia en la plataforma y presentar cursos no repetitivos de buena calidad.
- Puesta en marcha en aulas. Para la difusión de la plataforma en las aulas es necesario lograr los siguientes items.
 - Obtener un dominio, preferiblemente *http://abstractly.org*, por lo que habría que dar la plataforma a conocer como una organización sin ánimo de lucro.
 - Obtener dynos comerciales de Heroku, para proporcionar un servicio más profesional.
 - Dar a conocer a plataforma, sus objetivos y alcances en diferentes congresos, institutos y colegios.

Capítulo 7

Summary and Conclusions

Abstractly is a platform that presents all the concept necessary for develop and measure computational thinking and it has been developed and designed exclusively for it, so it offers a service, a product, and a dedication much greater than a Moodle could serve.

One of the most important things to note is that this platform is prepared so that anyone, teacher or not, can teach a course without having some knowledge in this subject, he just have to want to teach and encourage this type of education. Other strengths, its intuitive and simple design allows to complement this desire to boost computational thinking because it is easy to use by any child, as well as flashy.

If Abstractly is distributed in schools and encouraged its application, in a few months it would grow on its own, due to the feedback it receives from the students who use it and the possibility that the platform offers the teachers to propose new activities.

Finally, I have been tried that this platform fuses the potential of the activities offered by other platforms such as Code.org, with the possibility to measure computational thinking, to create a friendly and well structured course format, and to obtain statistical reports in any time and, in addition, that these reports can be customized and downloaded.

To the future, there are some tasks that are pending to improve in Abstractly and that could not be finished or realized due to the time available.

- Testing. Develop tests to verify the correct functioning of all the characteristics of the platform, and that these test are launched automatically with each new version.
- Activities stack. Increase the number of existing activities in the platform, both for the experimental group and the control group, to improve the experience in the platform and to present non-repetitive courses of good quality.
- Start-up in classrooms. For the diffusion of the platform in the classrooms it is necessary to achieve the following items:
 - Obtain a domain, preferably *http://abstractly.org*, so we would have to give the platform to know as a non-profit organization.
 - Get commercial dynos of Heroku, to provide a more professional service.
 - To make known the platform, its objectives and scope in different congresses, institutes and schools.

Capítulo 8

Presupuesto

8.1. Presupuesto

A continuación se muestra el presupuesto estimado para la realización de este proyecto y su explotación durante un año para corresponder con la duración de un curso escolar y poder aplicar la metodología.

Recurso	Cantidad	Precio unidad	Total (un año)
Horas de trabajo	300	20€	6000€
Heroku Standard Dyno	1	25€/mes	300€
Dominio 'abstractly.org'	1	9,99€/año	9,99€
Equipamiento informático	1	320,75€/ año	320,75€
Total	-	-	6630,74€

Bibliografía

- [1] amcharts. <https://www.amcharts.com/>.
- [2] Bebras. <http://challenge.bebas.uk/>.
- [3] Blockly. <https://developers.google.com/blockly/>.
- [4] Code.org. <https://code.org/>.
- [5] Csunplugged. <http://csunplugged.org/>.
- [6] Gambas. <http://gambas.sourceforge.net/en/main.html>.
- [7] Gamefroot. <http://gamefroot.com/>.
- [8] Heroku. <http://heroku.com>.
- [9] Hour of code. <https://hourofcode.com/>.
- [10] Mit app inventor. <http://appinventor.mit.edu/explore/>.
- [11] Pensamiento computacional y programación visual por bloques en el aula de Primaria. https://www.researchgate.net/publication/311881914_Pensamiento_computacional_y_programacion_visual_por_bloques_en_el_aula_de_Primaria.
- [12] Programación de computadores y desarrollo de habilidades de pensamiento en niños escolares: fase exploratoria. http://eduteka.icesi.edu.co/pdfdir/Icesi_Investigacion_Scratch_FaseI.pdf.
- [13] Programamos. <https://programamos.es/>.
- [14] Scratch. <https://scratch.mit.edu/>.
- [15] Snap! <http://snap.berkeley.edu/>.
- [16] Teaching london computing. <https://teachinglondoncomputing.org/>.

- [17] Luis Alfredo Sánchez Ruiz Dirigida por and Javier Alejandro Corredor Aristizábal. Comprendiendo el pensamiento computacional: experiencias de programación a través de scratch en colegios públicos de bogotá. <http://www.bdigital.unal.edu.co/53679/1/luisalfredosanchezruiz.2016.pdf>.
- [18] Mañana Andrea Carolina Chingaté Avila Abderrahim Florez Ortega Jaramillo Perez Gilberto. Incidencia de la aplicación de talleres de estimulación de procesos cognitivos en el desarrollo de habilidades basicas del pensamiento en los niños y niñas de cuarto de primaria del colegio Alfonso López Pumarejo (ied) jornada mañana. <http://repository.unilibre.edu.co/bitstream/handle/10901/8737/DOCUMENTOTESIS.pdf?sequence=1&isAllowed=y>.
- [19] Juan-Jose Boix-Tormos. Estudio de la influencia del aprendizaje del pensamiento computacional en las materias de ciencias en alumnos de secundaria. 2016. <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/52982/6/jjboixTFM0716mem%C3%B2ria.pdf>.
- [20] Karen Brennan and Mitchel Resnick. New frameworks for studying and assessing the development of computational thinking. 2012. http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf.
- [21] Alan Bundy. Computational thinking is pervasive. *Journal of Scientific and Practical Computing Noted Reviews*, 1(2):67–69, 2007. <https://core.ac.uk/download/pdf/28961399.pdf>.
- [22] Linda W. Erner, Jill Denner, Shannon Campe, and Damon C. Kawamoto. The Fairy Performance Assessment: Measuring Computational Thinking in Middle School. <https://pdfs.semanticscholar.org/5e9b/8a50566ca39b96774b1c3c76f62dacd563ce.pdf>.
- [23] Alexis Daniel Fuentes Perez. Abstractly. <http://abstractly.herokuapp.com/>.
- [24] Google. Computational thinking for educators. https://computationalthinkingcourse.withgoogle.com/course?use_last_location=true.
- [25] Marcos Román González. *Computational thinking test: design guidelines and content validation*. International Association of Technology, Education and Development (IATED), 2015. https://www.academia.edu/20221156/COMPUTATIONAL_THINKING_TEST_DESIGN_GUIDELINES_AND_CONTENT_VALIDATION.

- [26] Cristian L Vidal, Carlos Cabezas, José H Parra, and Leopoldo P López. Experiencias Prácticas con el Uso del Lenguaje de Programación Scratch para Desarrollar el Pensamiento Algorítmico de Estudiantes en Chile. *Formación universitaria*, 8(4):23–32, 2015. http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-50062015000400004&lng=en&nrm=iso&tlng=en.
- [27] Jeannette M. Wing. Computational Thinking. *COMMUNICATIONS OF THE ACM March*, 49(3), 2006. <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>.
- [28] Jeannette M. Wing. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881), 2008. <http://rsta.royalsocietypublishing.org/content/366/1881/3717.short>.
- [29] Jeannette M. Wing. Computational Thinking. 2012.