

TRABAJO FIN DE MÁSTER:

**CodeRacing:
Actividad basada en la competición
para incrementar la motivación hacia la
programación**

*Facultad de ciencias humanas y sociales
Master en formación del profesorado de educación secundaria.
Especialidad: sistemas electrónicos*

Leyre Sobrino Alzueta

ÍNDICE

Abstract	1
Introducción	1
Programación.....	1
¿Por qué es importante?.....	1
¿Por qué es difícil?	1
Métodos para enseñarla	2
Motivación, intrínseca / extrínseca.....	2
Motivación vs Competición	3
Programación en PHP.....	3
¿Por qué hemos hecho la investigación?	3
¿Qué hemos hecho?	4
Trabajos relacionados.....	4
Competición ACM	4
Competición vs competición y colaboración.....	5
Hardware, Software	5
Materia que se imparte	6
CodeRacing	6
Descripción del sistema	6
Casos de uso	6
Navegación	7
Evaluación de los problemas	9
Misiones	10
Implementación	11
Evaluación.....	11
Participantes	11
Recursos	11
Procedimiento.....	12
Medidas	13
Resultados	13
Pre-Encuesta, la asignatura.....	13
Post-Encuesta, la actividad.....	14
Variación Post y Pre	14
T-paired Test.....	15
Misiones superadas / Tiempo por misión / Intentos por misión.....	15
Correlación entre puntuaciones y misiones que superaron	17
Correlación entre pregunta 7 y visitas al ranking	17
Discusión.....	17
Explicación de resultados	17
Tipos de fallos que cometían	19
Fuentes de información o estrategias seguidas para resolver el problema	19
Comportamientos, Impresiones generales.....	19
Limitaciones	20
Trabajo futuro	20
Conclusiones	20
Referencias	20

Abstract

En el presente estudio se ha evaluado una nueva metodología para practicar problemas de programación. Para ello se ha creado una aplicación denominada CodeRacing con la que los alumnos compiten para superar misiones programando en PHP.

Durante el proceso se ha tenido en cuenta la importancia de la programación, tanto en el mundo laboral, como para los estudiantes de informática. Además, se analizan numerosos estudios que confirman que con este tipo de actividades competitivas se puede ayudar a incrementar la motivación de los alumnos para que aprendan eficazmente a programar. En nuestro experimento (n = 10) no se observó un incremento significativo de la motivación pero la impresión general de los alumnos fue favorable de cara a realizar más veces este tipo de actividad.

Introducción

Programación

Una definición popular de programación sería; el proceso de escribir, probar, depurar/solucionar problemas y mantener el código fuente de aplicaciones [1].

¿Por qué es importante?

Soloway (1993) respondía a la pregunta de por qué es importante aprender a programar argumentando que se aprenden estrategias eficaces y poderosas para resolver problemas, diseñar y pensar [3].

Estas características se producen porque la primera necesidad de la programación es dar solución a un problema, y entonces el estudiante necesita reflexionar sobre cómo comunicar la solución a una máquina, usando la sintaxis, de manera precisa (Papert, 1980; Szlávi and Zsakó, 2006) [3].

Por esto último se considera que también contribuye a desarrollar habilidades lingüísticas ya que deben aprender a transmitir al ordenador lo que se quiere desarrollar. (Hromokovič, 2006) [3].

Si un problema de programación es grande, el alumno aprende a dividirlo en subproblemas 78 M. Saeli *et al.* y a crear una solución central generalizada. Igualmente, el estudiante alcanza la capacidad de hallar soluciones legibles, usables y atractivas.

Por último, también cabe nombrar que se adquiere cierto sentido de dominio sobre la tecnología y establecen contacto con otras disciplinas como ciencias, matemáticas y construcción de modelos intelectuales (Papert, 1980) [3].

¿Por qué es difícil?

Desarrollar habilidades de programación es una de las competencias fundamentales que deben desarrollar los estudiantes de formación profesional de las ramas de informática y telecomunicaciones. Sin embargo, tanto los estudiantes como los profesores están de acuerdo que aprender a programar es una tarea difícil en los primeros años de estudio. Los que comienzan en este ámbito tienen que aprender a encontrar soluciones estructuradas a los problemas, expresarlas en una sintaxis rigurosa,

memorizarla, mecanizarla y aprender a entender cómo funciona la ejecución del programa. La combinación de estas tareas puede ser frustrante para los que empiezan a aprender a programar [2]. Por esto, los estudiantes noveles necesitan adecuar la enseñanza para motivarles con el objetivo de aprender a programar de una manera satisfactoria [4].

Métodos para enseñarla

Tradicionalmente la programación se ha enseñado mediante prácticas fundamentalmente individuales; sin embargo, en los últimos años, los profesores de esta materia han realizado diferentes métodos, más colaborativos mediante trabajo en parejas o en equipos para realizar proyectos. A través de estas actividades colaborativas los alumnos aumentan la confianza en sí mismos, desarrollan mejores programas y mejoran sus capacidades de programación [4].

El uso del aprendizaje competitivo para motivar a los estudiantes y mejorar su proceso de aprendizaje ha demostrado mediante estos métodos que los estudiantes participan más despertando sus instintos competitivos.

Sin embargo, existe diferente motivación entre los ganadores y los perdedores. Para evitarlo, el instructor tiene la difícil tarea de centrar la evaluación en el aprendizaje y divertimento, más que en ganar la competición [4].

Motivación, intrínseca / extrínseca

La motivación puede variar en cantidad y en tipo. El tipo de motivación depende de los objetivos o razones que dan lugar a la acción.

Los comportamientos intrínsecamente motivados, se definen como actuaciones desinteresadas que satisfacen las necesidades psicológicas innatas para la competencia y autonomía de la autodeterminación del individuo. La motivación intrínseca les permite aprender con mayor eficacia, profundidad y no requiere de incentivos.

Los comportamientos extrínsecamente motivados, son aquellos que se ejecutan porque van a obtener ciertos intereses. Es decir, lo que atrae al individuo no es la acción que se realiza en sí, sino lo que se recibe a cambio de la actividad realizada. Estos últimos varían en la influencia sobre la autodeterminación del individuo, para aumentarla sería necesaria la internalización e integración de estos procesos.

Revisando algunos estudios, resaltan las necesidades de determinar ciertas condiciones en el aula que propicien un contexto social para soportar los sentimientos de relación, competencia y autonomía. Estas necesidades psicológicas básicas servirán para mantener la motivación intrínseca y convertir las tareas extrínsecamente motivadas en influyentes para la realización del individuo [5].

En [6] hablan sobre el diseño de una actividad que mejora la motivación de los alumnos hacia el aprendizaje de ciertos temas de la arquitectura de los ordenadores.

El método mejora la motivación al permitir a los estudiantes verificar por sí mismos, en plataformas reales, la aplicación de sus conocimientos de arquitectura para mejorar el rendimiento del software, mostrándoles los progresos que consiguen.

Estos estudios afirman rotundamente que un estudiante motivado tiene mayor probabilidad de ser un alumno brillante.

Como resultado de mejorar la motivación, los alumnos pueden hacer más trabajo en menos tiempo y afrontar exámenes más difíciles [6].

Motivación vs Competición

Los estudios superiores proporcionan habilidades para facilitar un ambiente de cooperación y competición empresarial mediante actividades realizadas por equipos. Los instructores pueden utilizar estos ejercicios para maximizar el desarrollo profesional de los estudiantes preparándolos para la competencia y la cooperación a través de estructurar actividades de aprendizaje que requieran estas características. Esto hace que los profesores intensifiquen el aprendizaje enfocado al aumento de oportunidades laborales para el desarrollo de una futura carrera profesional de los estudiantes.

Lam, Yim, Law, & Cheung (2004) concluyeron que la competición tiene un impacto positivo para aumentar la motivación del aprendizaje en el aula.

Dettmer (2004) postuló que “aprender a perder” es un proceso valioso para los estudiantes que se preparan para ejercer profesiones en las cuales, normalmente, se está sometido a presión.

Por tanto se puede concluir que exponer a nuestros alumnos a pequeñas competiciones favorece los intereses de los alumnos y de las empresas de modo que estos desarrollan competencias profesionales [7].

Programación en PHP

CodeRacing evalúa los conocimientos del alumno en PHP.

PHP es un lenguaje de código abierto, interpretado en el lado servidor, multiplataforma, permite incluir HTML y es idóneo para el desarrollo Web [8].

El uso de PHP como código interpretado en el lado del servidor, facilita su portabilidad a cualquier servidor. También es un lenguaje que garantiza la compatibilidad con los estándares de Sistemas Operativos. Por lo tanto, con PHP tienes la libertad de elegir un sistema operativo y un servidor web.

Las posibilidades de PHP incluyen la salida de imágenes, ficheros en formato PDF e incluso generar películas Flash sobre la marcha.

Una de las fortalezas más importantes de este lenguaje es que soportan una amplia gama de bases de datos y otros protocolos de servicios como LDAP, IMAP, HTTP...

Por todos estos motivos consideramos importante adquirir una base en este lenguaje de programación durante los estudios superiores de los alumnos [9].

¿Por qué hemos hecho la investigación?

Esta investigación la hemos llevado a cabo debido a que consideramos fundamental realizar prácticas innovadoras que ayuden a los alumnos a aprender a programar.

Como hemos comentado anteriormente, los comienzos en este tipo de materias resultan frustrantes mediante el uso de los métodos tradicionales de aprendizaje.

Con este pequeño reto web los alumnos mejorarían su motivación a la hora de realizar programas cortos indicados en forma de enunciados. Además fomentarían el interés participando activamente en una competición y afianzando su conocimiento.

Otro de los objetivos fundamentales trataba de fomentar el autoaprendizaje pensando en su futura utilidad, preparando a los alumnos a ser independientes en su futura carrera profesional.

¿Qué hemos hecho?

Hemos desarrollado una aplicación web denominada CodeRacing, que proporciona a los alumnos diferentes misiones con dificultad incremental. Dicha aplicación valida automáticamente las respuestas de los alumnos y propone nuevos problemas. Asimismo permite jugar a varios usuarios simultáneamente y ver la misión en la que se encuentran. La intervención realizada sirvió para comprobar como influye en la motivación de los estudiantes y cuál es su respuesta al utilizar CodeRacing.

Pusimos a prueba esta aplicación durante aproximadamente 45 minutos en un grupo de 10 alumnos del módulo de Sistemas Operativos y Lenguajes de Programación. Este módulo forma parte del currículo del Grado Superior de Sistemas de Telecomunicaciones e Informáticos que se imparte en el instituto Donapea.

Trabajos relacionados

Competición ACM

ACM (Association for Computing Machinery) es un grupo de profesionales informáticos que organizan desde 1977 la competición ICPC (International Collegiate Programming Contest). Los equipos están formados por tres estudiantes que comparten un ordenador, trabajan juntos para resolver tantos problemas como sea posible de un conjunto de 9 retos inéditos. La participación ha ido incrementando a lo largo de los años, en las finales de 2006 contaron con la presencia de 83 equipos. En total, participaron alrededor de 5606 equipos de 1733 universidades de 84 países.

El equipo que resuelva más problemas en 5 horas gana.

Una vez que los jueces reciben el código lo compilan, lo prueban y envían un veredicto a los concursantes.

Los componentes del equipo deberán decidir la estrategia a seguir, por ejemplo, decidiendo resolver los ejercicios del más fácil al más difícil o al revés [10].

En [11] narra la experiencia, en 1999, de la universidad de Bergen, en Noruega realizando la competición entre sus estudiantes para participar en la ACM. La motivación principal de esta universidad al participar en ACM-ICPC fue crear un ambiente académico positivo para sus estudiantes. Como normalmente ocurre cuando estudias una carrera, los estudiantes no se consideraban profesionales informáticos hasta pasado el segundo año de sus estudios. Es por esto que querían realizar un evento que pudiese ayudarlos a sentirse más identificados con su futura profesión. Desde el comienzo de la competición pudieron comprobar que podría usarse como método de aprendizaje, afianzando sus conocimientos de informática y el trabajo en equipo. Además la competición les dio la posibilidad de analizar en qué medida la educación en la universidad prepara a los estudiantes para competir.

La opinión de los organizadores refleja que fue un evento realmente positivo y exitoso. Los alumnos opinaban de la misma manera. Aunque la mayoría acababan cansadísimos después de la competición, comentaban que se habían divertido y que repetirían.

Además, participar en el evento hizo que los estudiantes estudiaran sin tener exámenes a la vista, algo que de otra forma, no lo habrían hecho. Los organizadores comentaban que el tiempo y el dinero empleados en preparar la competición fue recompensado con creces.

Competición vs competición y colaboración

En [12] narra un proyecto de la Universidad de Iowa en 2004 para una asignatura de estructuras de datos utilizando juegos de competición entre los alumnos. Esta programación competitiva se basaba en que los estudiantes evaluaran su código y mejoraran sus programas a través de tareas, comparándolo con el código de su profesor y el código de sus compañeros.

En [13] y en [4] narra una metodología educativa basada, al igual que la anterior, en la competición pero incorpora también el aprendizaje colaborativo. Fue evaluada en la Universidad de Valladolid basándose en dos herramientas Moodle para la parte colaborativa. En ella los alumnos incluían nuevos retos con sus soluciones. Y la herramienta de la competición QUESTOURnament, creada para fomentar las habilidades de investigación, análisis y documentación en la asignatura de Redes de Telecomunicación.

En [14] de la Universidad de Roma, también se basan en un aprendizaje colaborativo y competitivo. Forman equipos y compiten para desarrollar técnicas para la protección de imágenes mediante marcas de agua.

En todos los casos destacan el incremento de motivación de sus alumnos como principal objetivo de estos juegos de programación. En [13] afirman que los alumnos que participaron obtuvieron mejores calificaciones, por lo que esta estrategia resulta realmente efectiva.

En [14] destaca que numerosos estudios demuestran que un aprendizaje colaborativo es un sistema efectivo para difundir el conocimiento, organizar a los estudiantes en equipos proporciona mejores resultados e incrementa las habilidades sociales de los alumnos.

Hardware, Software

En los artículos anteriores predomina la utilización de software para realizar competiciones en enseñanzas innovadoras.

En [15] destaca la importancia de la utilización software mediante las nuevas Tecnologías de la Información y la Comunicación ya que no requieren demasiada infraestructura.

En [14] recalca que la utilización de ordenadores de bajo coste y conexión inalámbrica a internet, en una de las clases de ordenadores, permite el diseño de la nueva metodología de aprendizaje comentada anteriormente.

Sin embargo existen otras materias, como la electrónica, que requieren de una infraestructura hardware menos común.

Por ejemplo, en [16] relata una pequeña competición que pusieron en marcha los profesores del departamento de Ingeniería Informática y Electrónica de la Universidad de California. Los estudiantes debían construir un vehículo autónomo de juguete con una cámara CCD de miniatura y un transmisor de 900Hz. También requerían de un PC que recibiese las imágenes y generara una señal de vídeo.

En [17] narran la experiencia de la universidad de Mercy participando en Intelligent Ground Vehicle Competition (IGVC). En algunos casos los componentes para construir el vehículo provenían de donaciones ya que los materiales resultaban ser muy caros.

En [18] los estudiantes de la academia Naval de Estados Unidos, fabrican microrobots con habilidades para jugar al fútbol. Estos participaron en la Robocup Nanogram League en el curso 2006-2007.

Materia que se imparte

En las competiciones enfocadas a la enseñanza que hemos comentado anteriormente se pueden englobar en ámbitos científicos o de ingeniería. Sobre todo se enfocan en dos grandes ámbitos; la programación y la electrónica.

En las competiciones de la ACM los concursantes refuerzan sus conocimientos en lenguajes de programación C, C++, C#, JAVA.

En [12] desarrollan en C++, el lenguaje de programación que imparten en la universidad de Iowa. Estos desarrollos los llevan a cabo utilizando el compilador gcc en máquinas con sistema operativo Linux. También una de las contribuciones de este proyecto fue una arquitectura que combinaba programas en C++ con las interfaces de usuario JAVA para desarrollar juegos.

En el QUESTOURnament de la Universidad de Valladolid los alumnos preparaban y corregían preguntas sobre la asignatura de Redes de Telecomunicación incluyendo temas sobre pilas de protocolos, estandarización, el modelo OSI, TCP/IP, medios de transmisión, codificación y modulación, multiplexación, el flujo de control de datos, etc. Es decir, todo el temario de la asignatura de Redes.

En las competiciones que requerían hardware de elementos electrónicos las materias de aprendizaje, en las que los estudiantes reforzaban sus conocimientos, eran de ámbitos electrónicos. En la competición [16] estimulan el interés de los estudiantes en ámbitos eléctricos e informáticos ya que también utilizan código para programar sus vehículos. En las competiciones IGVC y en la Robocop Nanogram League de [17] y [18] buscan fomentar el interés de los estudiantes en ingeniería de sistemas de control robótico.

CodeRacing

Descripción del sistema

CodeRacing es una aplicación Web desarrollada en PHP que permite a varios usuarios enviar soluciones a distintos problemas de programación de forma competitiva.

Para mantenerse en una buena posición en la clasificación, los competidores de CodeRacing deberán superar tantas misiones como puedan. Estas misiones se basan en realizar funciones en lenguaje PHP.

Este juego es accesible vía web mediante la dirección:

<http://www.infinityk.com/temp/a/coderacing/>

Casos de uso

Existen dos tipos de usuarios que pueden utilizar la aplicación CodeRacing: El usuario administrador y el resto de participantes de la competición.

El usuario administración tiene una opción más en el menú superior que le permite dar de alta a nuevos usuarios que deseen participar en una carrera de código. Asimismo, si se fija un tiempo finito para realizar una competición o se desea parar el juego cuando uno de los competidores gana, el usuario administrador podrá darle a “End Current Race” para finalizar la competición. Esta acción llevará a los participantes a la pantalla de inicio donde se loguean para entrar.

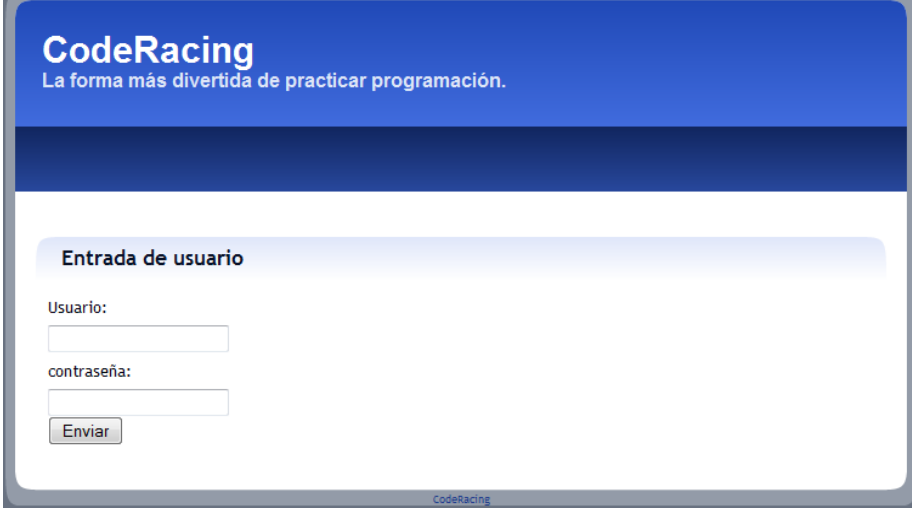
Las otras opciones del menú, están disponibles para ambos tipos de usuario.

La primera opción es “misiones” en la que se puede observar el enunciado de las funciones e introducir en una caja de texto el código PHP para superar el reto. En la

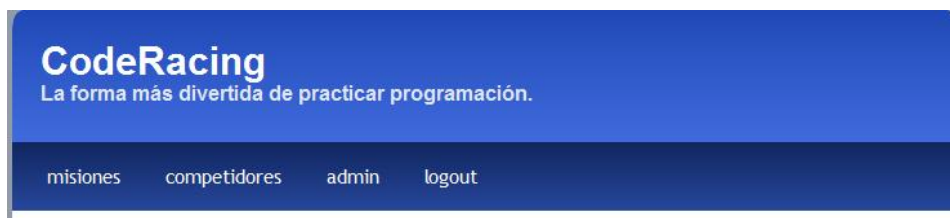
opción “competiciones” los usuarios pueden ver la clasificación de los participantes. Por último, está la opción de “logout” dónde se sale de la aplicación posicionándose en la pantalla inicial de login.

Navegación

Al acceder a la página nos pedirá loguearnos con un usuario y contraseña.

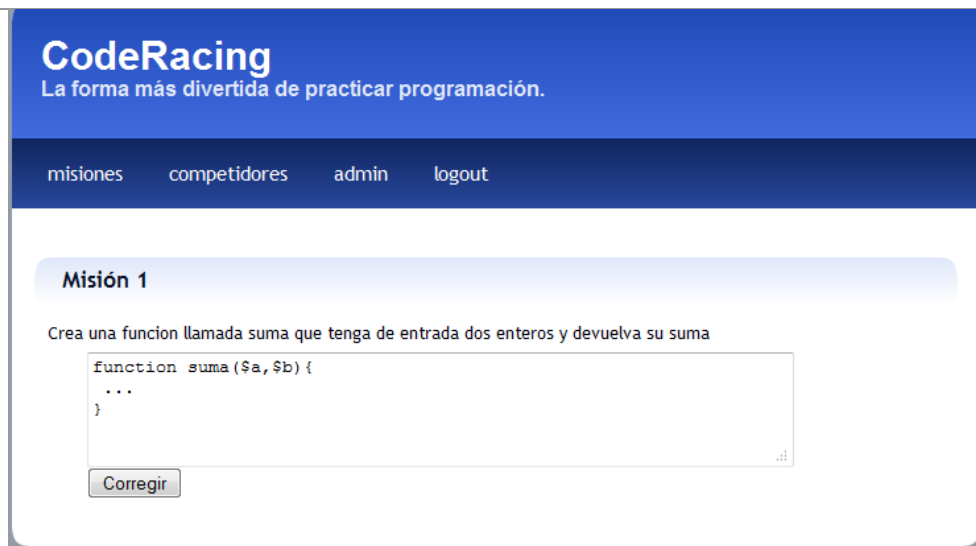


Las opciones más importantes del menú son misiones y competidores. La pantalla que mostramos a continuación incluye la opción admin, ya que en este caso nos logueamos como usuario administrador.



Como comentaba en el apartado anterior, al acceder a misiones podrán ver el enunciado de las mismas.

El juego tiene 8 misiones que, como hemos explicado, consisten en realizar funciones en lenguaje PHP. Al realizar la misión comprobarán si han tenido algún error. Si no tienen ninguno, la habrán superado e irán a la siguiente. De lo contrario, tendrán que volver a realizarla hasta que el código sea correcto y se obtenga el resultado esperado. Las misiones irán aumentando en dificultad.



Hay ocasiones en que los programadores dan al botón corregir y su programa no tiene errores de sintaxis, pero no da el resultado esperado. En este caso, CodeRacing les proporciona una pista, mediante un ejemplo, para que les ayude a descubrir en qué han fallado.

En la imagen que vemos a continuación, la misión consistía en realizar la suma de dos valores de entrada. Para los valores de entrada 1831 y 1369 esperaba que el resultado de la función devolviese el valor 3200 es decir, 1831 más 1369; sin embargo el valor que devuelve esta función es 1369, por lo tanto la misión ha sido fallida.



En la sección de competidores los alumnos podrán ver por qué misión van sus compañeros. Esta sección se pensó para aumentar la motivación de los alumnos y fomentar su espíritu competitivo.



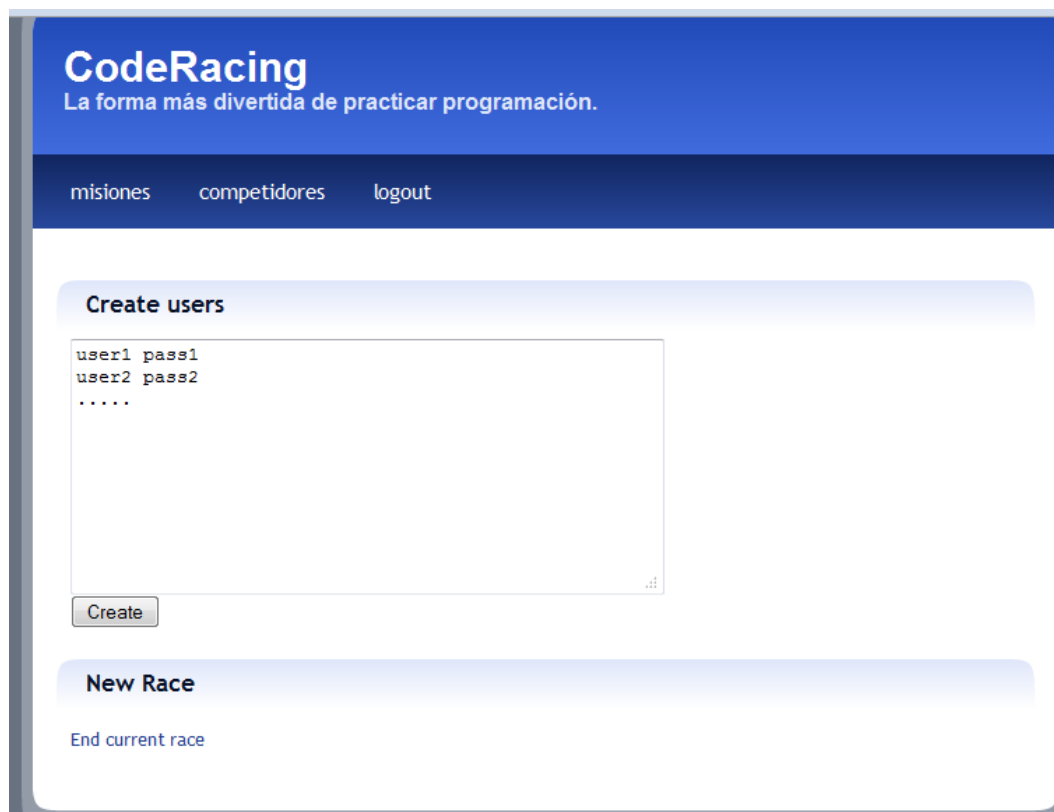
The screenshot shows the CodeRacing application interface. At the top, there is a blue header with the text "CodeRacing" and "La forma más divertida de practicar programación." Below the header is a dark blue navigation bar with the links "misiones", "competidores", and "logout". The main content area is white and features a section titled "Posiciones" (Positions). This section contains a table with two columns: "Usuario" (User) and "Posicion" (Position).

Usuario	Posicion
juan	3
lucia	1

La siguiente pantalla muestra la sección que ve sólo el administrador. En ella se crean usuarios y tiene la opción de parar la competición:

En la parte superior hay un cuadro de texto para crear nuevos participantes. Cada línea representará un nuevo usuario. La primera palabra de la línea será el nombre de usuario y la segunda palabra la contraseña.

En la parte inferior vemos el botón "End Current Race" que se utilizará para finalizar la carrera.



The screenshot shows the CodeRacing application interface from an administrator's perspective. It features the same blue header and navigation bar as the previous screenshot. The main content area is white and contains a section titled "Create users". Below this title is a text input field containing the following text:

```
user1 pass1
user2 pass2
.....
```

Below the text input field is a "Create" button. At the bottom of the main content area, there is a section titled "New Race" with a button labeled "End current race".

Evaluación de los problemas

Para describir las pruebas de corrección de las distintas misiones de la aplicación definiremos los conceptos de caja blanca y caja negra.

Se denomina caja negra a aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En otras palabras, de una caja negra nos interesará su forma de interactuar con el medio que le rodea [20]. Por tanto, en este caso deben estar muy bien definidas sus entradas y salidas, de su interfaz; en cambio, no se precisa definir ni conocer los detalles internos de su funcionamiento.

Por otro lado, en programación, se denomina cajas blancas a un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo. Así como las pruebas de caja negra ejercitan los requisitos funcionales desde el exterior del módulo, las de caja blanca están dirigidas a las funciones internas, evaluando la eficiencia del código. Entre las técnicas usadas se encuentran; la cobertura de caminos (pruebas que hagan que se recorran todos los posibles caminos de ejecución), pruebas sobre las expresiones lógico-aritméticas, pruebas de camino de datos (definición-uso de variables), comprobación de bucles [19].

Dicho de manera más sencilla, en las comprobaciones de caja negra se comprueba que la función realizada funcione. Sin embargo en las de caja blanca se comprobaría que el código implementado en la función sea eficaz, los bucles elegidos, las variables, que no tarde mucho tiempo en dar el resultado.

En CodeRacing se utilizan pruebas de caja negra para validar la solución enviada por los alumnos en cada una de las misiones. Para testear cada función programada, se generan numerosos valores de entrada y se comprueba que, efectivamente, con el código realizado por el participante se obtiene el mismo resultado que con una función válida previamente conocida por el sistema. De esta forma, en caso de que la función sea incorrecta, se le puede mostrar información sobre los datos de entrada, el valor esperado, y el obtenido con su código.

Misiones

A continuación describo en qué consisten las ocho misiones, de dificultad incremental, que deben realizar y superar los participantes de CodeRacing. Para llevarlas a cabo se les facilita a los participantes el enunciado y nombre de la función, indicando las variables de entrada que almacenarán números, matrices o palabras.

Misión 1

En la misión número 1 el competidor deberá realizar una función llamada *suma* que recibiendo dos valores enteros, como variables de entrada, devuelva la suma de estos.

Misión 2

Cuando superen esta misión se mostrará el enunciado de la misión 2. En ella deberán escribir una función llamada *maximo* que tenga de entrada dos números enteros y devuelva el mayor de los dos.

Misión 3

Cuando realicen correctamente este reto pasarán a la siguiente prueba y deberán realizar una función llamada *suma*, que tenga de entrada un array, es decir, una matriz o vector capaz de almacenar varios valores. Esta función deberá devolver la suma de todos estos valores contenidos en el array.

Misión 4

La cuarta misión también tendrá como variable de entrada una matriz y un número entero, la función se llamará *cuenta*. Para superar esta misión el resultado tiene que mostrar el número de veces que aparece el entero en la matriz.

Misión 5

En la quinta misión los participantes leerán el siguiente enunciado: Crea una función llamada *buscar* que tenga de entrada un array de enteros y un número y devuelva la posición en donde se encuentra la primera aparición del valor, la primera posición es el índice 0. Si no existe ninguna aparición se devolverá -1.

Misión 6

En la sexta misión llamada *primo* la función recibirá un número y tendrá que mostrar en pantalla si se trata de un número primo o no.

Misión 7

Cuando superen el desafío anterior deberán crear una función llamada *palin* que tenga de entrada una cadena, es decir, una palabra y devuelve verdadero si la palabra es palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda).

Misión 8

En la última misión, la ocho, deberán crear una función llamada *ordena* que tenga de entrada una matriz de números y devuelva esa matriz con los números ordenados de forma ascendente.

Implementación

La aplicación CodeRacing fue realizada en el lenguaje PHP utilizando MySQL como base de datos. Estaba alojada en un servidor para que los alumnos pudieran acceder simultáneamente desde cada uno de sus puestos de prácticas.

Evaluación**Participantes**

Para realizar la evaluación con CodeRacing participaron 10 alumnos de la asignatura Sistemas Operativos y Lenguajes de Programación del Grado Superior de Telecomunicaciones e Informática del instituto Donapea. Todos ellos eran de sexo masculino, el promedio de edad era de 22,1 años. Siendo el menor de 19 años y el mayor de 37. La desviación típica de la edad de los participantes es de 5,36.

Recursos

Para realizar este estudio utilizamos dos encuestas, la Pre-Encuesta y la Post-Encuesta, y la aplicación CodeRacing. En la Pre-Encuesta evaluamos el interés y dificultad de los alumnos hacia la asignatura de programación. Tenía 6 afirmaciones las cuales, al igual que en la Post-Encuesta, debían valorarlas indicando si estaban totalmente de acuerdo o en total desacuerdo. Estas son las siete opciones con las que podían responder:

Totalmente en desacuerdo (1)	Parcialmente en desacuerdo (2)	En desacuerdo (3)	Ni de acuerdo ni en desacuerdo (4)	De acuerdo (5)	Parcialmente de acuerdo (6)	Totalmente de acuerdo (7)
---------------------------------	-----------------------------------	----------------------	---------------------------------------	-------------------	--------------------------------	------------------------------

Las afirmaciones que incluía la Pre-Encuesta son:

1. La asignatura de Sistemas Operativos y Lenguajes de Programación me resulta muy interesante.

2. Creo que esta asignatura es muy útil.
3. Comparando con las demás asignaturas esta es la que menos aburrida me resulta.
4. Creo que se podrían utilizar otros sistemas para aprender a programar.
5. Tengo facilidad para resolver solo los ejercicios de programación.
6. Comparando con mis compañeros, esta asignatura me resulta fácil.

Al concluir la competición con CodeRacing realizaron la Post-Encuesta. En ella se incluían 7 afirmaciones sobre la actividad realizada con la aplicación, 6 afirmaciones que servirían para compararlas con las de la Pre-Encuesta y una última para evaluar la competitividad de los alumnos. Estas son las afirmaciones de la Post- Encuesta:

1. Esta actividad fue muy entretenida e interesante.
2. Creo que esta actividad podría tener alguna utilidad para mí.
3. Mientras estaba haciendo esta actividad, me he divertido bastante.
4. Creo que actividades de este tipo podrían motivarme hacia el estudio de la asignatura.
5. Esta actividad me ha ayudado a comprender que puedo aprender por mí mismo buscando información en Internet.
6. Hice muy bien esta actividad, en comparación con los otros compañeros.

Afirmación extra para medir la competitividad:

- Me gustó ver la situación de mis compañeros, ayudó a fomentar mi espíritu competitivo.

Procedimiento

Realicé esta actividad el último día de las prácticas. El viernes los alumnos tienen tres horas de la asignatura de Sistemas Operativos y Lenguajes de Programación. Dos antes del recreo y otra después. Para realizar la actividad seguí los siguientes pasos:

Antes de nada, con el usuario administrador, creé los usuarios que participarían en la actividad. En este caso diez alumnos que se encontraban en clase ese día.

Les sometería a la prueba durante unos 45 minutos en la última sesión, que tendría lugar después del recreo.

Antes de realizar la actividad les pasé la Pre-Encuesta sobre la asignatura, como comentamos, en ella evaluamos el interés, la capacidad y la facilidad del alumno en el temario del módulo.

A continuación, los estudiantes se marcharon al recreo y aprovechamos para preparar la actividad en los ordenadores. Los encendimos, abrimos el navegador e introdujimos la dirección del juego para que cuando volviesen estuviera listo para que comenzaran la carrera de código.

Me pareció importante insistirles que disponían de red para buscar por internet las soluciones que desconociesen.

Al sonar el timbre los alumnos fueron entrando y se repartieron por los ordenadores del aula. Cuando todos estuvieron sentados, introdujeron su usuario y contraseña y comenzaron la actividad.

Pasados los 45 minutos, mediante el usuario administrador, finalicé la carrera y les pasé la Post-Encuesta sobre la actividad.

Al finalizarla, automáticamente terminan la sesión y todos los usuarios se sitúan en la página de login.

Por último realizaron la Post-Encuesta opinando sobre la actividad que acababan de realizar.

Medidas

Durante la competición, el alumno que ganó consiguió llegar a la quinta misión, aunque no tuvo tiempo a superarla. Otros seis alumnos finalizaron intentando superar la cuarta misión, consiguiendo realizar con éxito los tres primeros retos. Los tres alumnos que faltan, superaron los dos primeros desafíos, la carrera finalizó mientras intentaban superar la tercera misión.

Entre todos, el número de intentos para realizar la misión uno fue de 43, para la misión dos 34 intentos. La misión tres, fue en la que más intentos realizaron, 83. En la cuarta fueron realizados 23 intentos por los siete alumnos que llegaron y para la quinta misión, el alumno que llegó, solamente le dio tiempo a intentarlo una vez.

En cuanto al tiempo empleado, hablaremos de la suma de tiempo de todos los alumnos en cada una de las misiones: Para la misión tres, la que más tiempo emplearon, de media fueron 1986 segundos. En la misión dos, la que menor tiempo emplearon, 576 segundos. En la primera misión emplearon 818 segundos, en la cuarta 919 y por último el único intento realizado para la misión cinco fue de 9 minutos. Para hablar de los alumnos a partir de ahora los enumeraremos del 1 al 10.

El alumno 4 miró 14 veces a la clasificación, fue el alumno que más la observó. El alumno 3 fue el siguiente con un total de siete vistazos. Los alumnos 6 y 9 miraron 5 veces al ranking durante la competición, el alumno 8 tres veces, los alumnos 1, 7 y 10 miraron dos veces y por último el alumno 2, no miró ninguna vez.

Resultados

Para realizar el estudio de los resultados relacionamos las afirmaciones de la Pre-Encuesta y la Post-Encuesta (la 1 con la 1, la 2 con la 2, y así sucesivamente). De esta manera evaluamos la asignatura tal y como la llevan a cabo en estos momentos y como sería si el aprendizaje se basase en actividades competitivas como CodeRacing.

Evaluaremos el interés, la motivación y la facilidad en cada alumno, tanto en la asignatura como en la actividad realizada.

Para realizar el estudio realizamos una serie de medidas estadísticas de tal manera que se valorarán las respuestas del 1 al 7 para obtener valores estadísticos, considerando 1 cuando se está totalmente en desacuerdo y 7 en total acuerdo.

Pre-Encuesta, la asignatura

En la primera afirmación se medía el interés por la asignatura. De las respuestas de los alumnos, obtenemos una media de 6,20 y una desviación típica de 0,79.

En la segunda, se afirmaba que la asignatura es de gran utilidad, se obtiene un 6,10 de media y un 0,88 de desviación.

En la tercera afirmación, mide lo divertida que les parece la asignatura, se obtiene un 5,00 de media. La desviación es mayor comparada con las anteriores, 1,63.

La siguiente afirmación se obtiene una media de 4,70 y una desviación de 1,16. En ella se afirmaba que se podría aprender la asignatura utilizando otros medios.

En la quinta decía que resolvían los ejercicios individualmente, la media es de 3,80 con una desviación de 1,40.

La sexta afirmación comparaba al alumno con el resto de sus compañeros afirmando tener más facilidad por la asignatura en comparación con los demás. Es la de menor media con un 3,40 pero mayor desviación, un 2,27.

La media total de todas las respuestas es de 4,87 sobre 7, con una desviación de 1,35. El Alfa de Cronbach es 0,61 por lo que la coherencia de la encuesta es ligeramente insuficiente. Seguramente debido a la sexta pregunta.

Post-Encuesta, la actividad

En la Post-Encuesta obtenemos las siguientes medias y desviaciones:

La primera afirmación, midiendo lo interesante que les había parecido la actividad se obtienen una media de 5,10 y una desviación de 1,52.

En la siguiente, afirmando lo útil que era la actividad, el promedio de las respuestas alcanza un 4,90 con una desviación de 1,66.

Afirmando lo divertida que les pareció la actividad la media es de un 5,00 y la desviación de 1,89.

En la cuarta afirmación se decía que la actividad podría motivarles en el estudio de la asignatura; se obtiene un promedio de 5,30 con una desviación de 2,11.

En la siguiente se obtiene un 5,00 de media y una desviación de 1,76. Esta media si habían comprendido que podían buscar información individualmente.

La última es la de menor media, 3,60 con una desviación de 1,84. Afirmando que la actividad le había resultado más fácil que al resto de compañeros.

La media total de todas las respuestas es de 4,82 y la desviación de 1,80.

El Alfa de Cronbach es 0,93 por lo que el cuestionario resulta coherente.

Variación Post y Pre

En la Pre-Encuesta el alumno 4 fue el alumno quien obtiene la menor media, un 4,00. Las afirmaciones que menor puntuación obtuvieron por su parte fueron las que hablaban sobre lo fácil que le resultaba la asignatura, es decir, a este alumno le resultaba complicada.

El alumno que más interés, facilidad y capacidad mostraba por la asignatura es el alumno 2, su media en las respuestas era de un 6,33.

La afirmación en la que los alumnos se mostraban más de acuerdo, fue en la que decía que la asignatura les parecía muy interesante. La media es de 6,08 sobre 7 y la desviación de 0,79, la menor desviación de todas.

Las afirmaciones con medias inferiores son las que decían que los alumnos eran capaces de resolver los ejercicios individualmente y la que afirmaba que la asignatura les resultaba más fácil que al resto de sus compañeros.

En esta última vemos que la desviación es mayor, es decir, las puntuaciones son menos homogéneas.

El alumno número 1 y el 3 son los alumnos que menor media de puntuación poseen siendo de 3,33 y 1,33 respectivamente, no disfrutaron realizando la actividad.

Los alumnos que mejor puntuaron la actividad fueron los alumnos 5, 6 y 2 afirmando estar de acuerdo o totalmente de acuerdo con la mayoría de afirmaciones. La competición de CodeRacing les resultó divertida, motivadora e interesante.

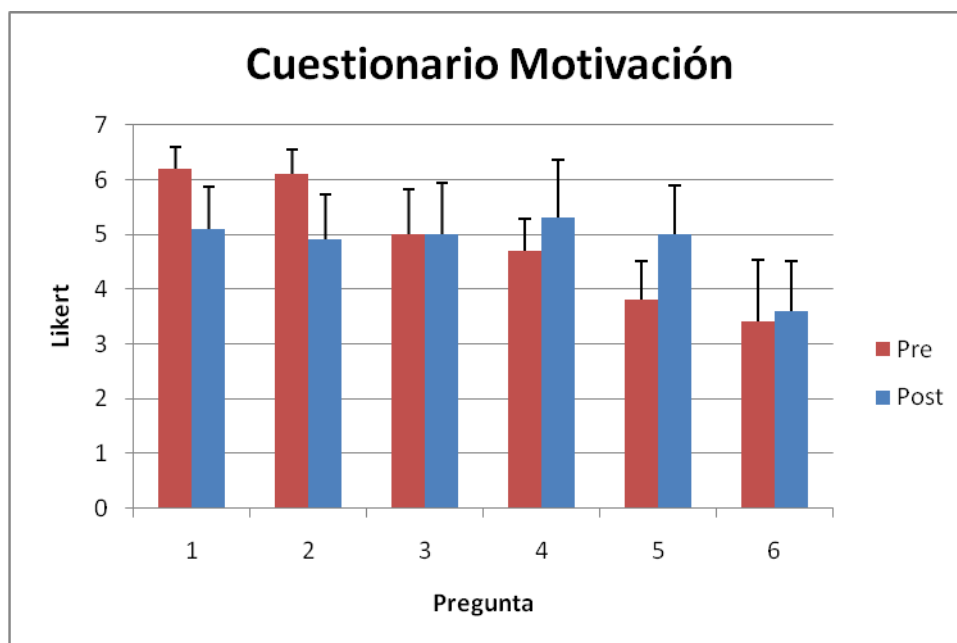
En cuanto a la media de cada una de las respuestas observamos que la mayor puntuación la obtiene la afirmación que decía que la actividad les podría motivar para el estudio de la asignatura aunque hay que destacar que es la que más desviación tiene. Esto lo provoca la puntuación de dos alumnos a los que no les resultó nada motivadora puntuándola con un valor de 3 o 1, estando totalmente en desacuerdo. La media de menor puntuación es para la afirmación que dice que la actividad les ha resultado más

fácil que al resto de sus compañeros. Esto se puede deber a que no hay excesiva diferencia en los resultados.

T-paired Test

Aunque en la media de la Post-Encuesta la puntuación es menor la diferencia es de 0,05 por lo que esta no es significativa. En cuanto a las desviaciones se 1,80 en la Pre frente 1,35 en Post, tampoco hay una diferencia importante.

En la gráfica que vemos a continuación se observa las diferencias en las medias de cada una de las afirmaciones.



Prueba t para medias de dos muestras emparejadas

	Variable 1	Variable 2
Media	4,81666667	4,86666667
Varianza	2,45339506	0,70864198
Observaciones	10	10
Coeficiente de correlación de Pearson	0,39605743	
Diferencia hipotética de las medias	0	
Grados de libertad	9	
Estadístico t	0,10865477	
P(T<=t) una cola	0,45793012	
Valor crítico de t (una cola)	1,83311292	
P(T<=t) dos colas	0,91586023	
Valor crítico de t (dos colas)	2,26215716	

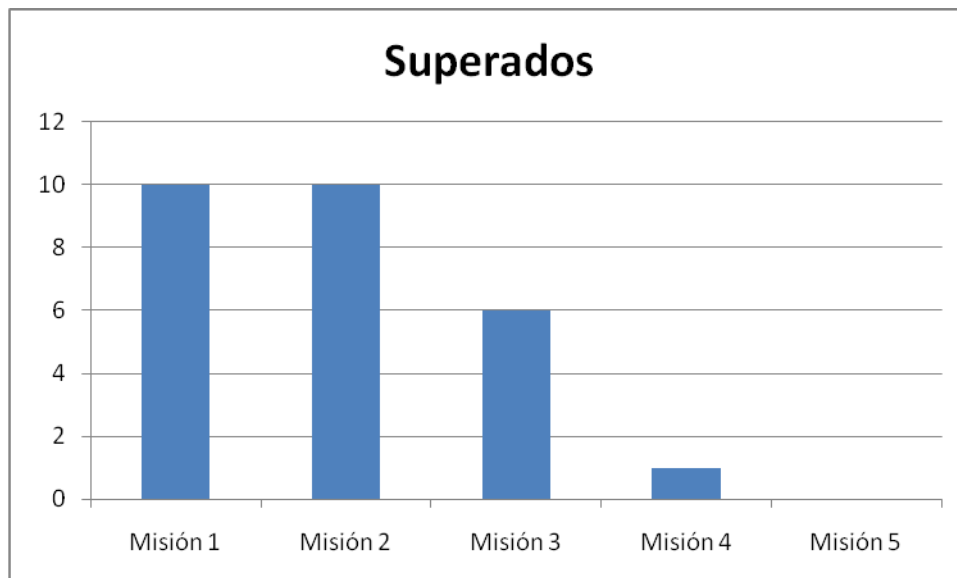
$t(9) = -0,108$ $p > 0.05$ No hay diferencia significativa

Misiones superadas / Tiempo por misión / Intentos por misión

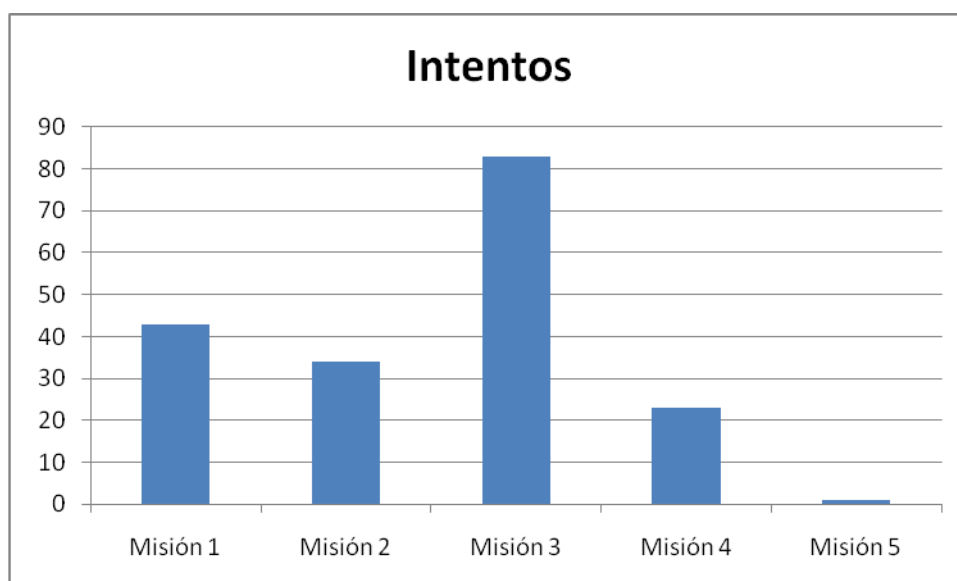
En la gráfica que se muestra a continuación observamos el número de alumnos que superó cada misión:

La primera y la segunda misión fueron superadas por todos los alumnos.

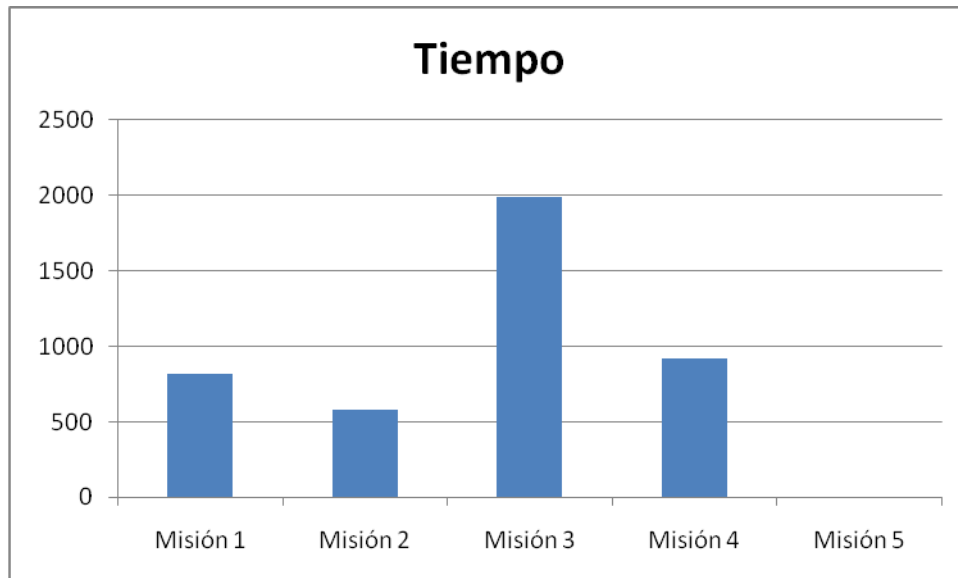
La tercera misión fue superada por 6 alumnos y la cuarta por un alumno.



En la siguiente gráfica se muestran el cómputo total de los intentos que realizaron por cada misión.



En la gráfica siguiente se representa el tiempo total que los alumnos emplearon por cada misión.



Correlación entre puntuaciones y misiones que superaron

El coeficiente de correlación entre las medias de las puntuaciones de la Post-Encuesta y las misiones que superaron es de 0,69. Siendo 1 la correlación perfecta entre estos dos valores y 0 que no existe ninguna relación podemos afirmar que la correlación es moderada tirando a alta.

Correlación entre pregunta 7 y visitas al ranking

En este caso relacionamos la competitividad de los alumnos, con la afirmación extra de la Post-Encuesta, y el número de vistazos al ranking. La fórmula estadística proporciona un resultado de 0,052 por lo que no existe correlación, más adelante aclaramos los motivos.

Discusión

Explicación de resultados

A continuación explicaré los resultados y la coherencia de los alumnos en las puntuaciones que dieron a las distintas afirmaciones que aparecían en las encuestas. Tuve la suerte de conocer a los estudiantes durante las prácticas del Master de Formación al Profesorado. Por este motivo me ha resultado más fácil relacionar las respuestas que dieron, con las actitudes que mostraban en el ámbito de la programación. En la Pre-Encuesta evaluamos el interés, capacidad y dificultad de los alumnos en la asignatura de Sistemas Operativos y Lenguajes de Programación.

Las respuestas de los alumnos en cuanto a la asignatura tienen bastante relación con lo bien que se les daba. Los alumnos que obtienen menor media son aquellos que repetían la asignatura o los que solían ser fieles a los exámenes de recuperación, es decir, los que encontraban mayor dificultad. El alumno que mayor media tiene en las afirmaciones de la Pre-Encuesta era uno de los que mejores notas solía obtener. La afirmación que mayor media obtiene es en la que se evalúa lo interesante que les parecía la asignatura. Además, es la de menor desviación por lo que denota bastante homogeneidad en las respuestas. Todos estaban de acuerdo con esa afirmación.

Los alumnos creen que la asignatura es útil ya que esta afirmación es la segunda de mayor puntuación y también tiene una desviación pequeña.

Por último, destaca la última afirmación por ser la que mayor desviación presenta. En ella, dos alumnos afirmaban que se les daba mejor la asignatura que al resto de sus compañeros, puntuándola con el máximo valor, un 7. Sin embargo, el resto de compañeros evaluaban la afirmación dándole un valor de 4 o menor, por este motivo hay más desviación.

En la Post-Encuesta, llama la atención la media individual de dos alumnos que puntuaron la actividad con valores muy bajos. En mi opinión, al ser alumnos bastante competitivos, es lógico que se mostraran más reacios a la actividad debido a que fueron los que quedaron últimos, solo superaron dos misiones. Los alumnos que demostraron divertirse más durante la actividad fueron, lógicamente, los que mejor puntuaron la Post-Encuesta. El alumno que más veces miró la clasificación es uno de los que mayor media obtiene.

Comparando las afirmaciones de la Pre-Encuesta con la Post-Encuesta, en las dos primeras se compara lo interesante y útil que les parece la asignatura en comparación con la actividad realizada. Los resultados muestran mayor satisfacción con la asignatura que con la actividad, creo que puede ser debido a que la competición fue de corta duración y en algunos casos perdieron tiempo en entender el funcionamiento de CodeRacing. En las siguientes afirmaciones, las terceras de cada encuesta, vemos que hay bastante paridad en cuanto a lo divertida que les resulta la asignatura y lo entretenida que les resultó la actividad.

De la cuarta se puede concluir que los estudiantes creen que realizando este tipo de actividades podría aumentar su motivación para el estudio de la asignatura. De la quinta afirmación se desprende que aunque en la asignatura les resulta complicado resolver los ejercicios solos, este tipo de actividades les ayudaría a encontrar las respuestas por sí mismos.

La correlación entre los alumnos que habían quedado mejor clasificados y la puntuación que daban en la Post-encuesta es moderadamente alta, algunos de ellos aunque no fueron los que mejor clasificación obtuvieron, se notó que se habían divertido y que el tiempo se les pasó rápido.

En cuanto al número de intentos y el tiempo empleado, llama la atención la misión tres con un valor medio de 1989 segundos empleados por los alumnos. Esto se debe a que en esta misión se pedía realizar un bucle, un concepto que suele resultar difícil de entender en los comienzos del aprendizaje de programación. También, aunque consideramos que la dificultad de las misiones va en aumento, observamos que en la misión dos emplearon menos tiempo e intentos que en la misión uno. Esto puede deberse a que, como nunca habían realizado actividades de este tipo, los primeros momentos fueron un tanto confusos.

En cuanto a la correlación que existe entre la afirmación de la competitividad y las veces que miraron a la clasificación es tan baja debido a que el alumno que más veces miró se encargaba de radiar a toda la clase cómo iban y quien era el ganador. Como el

alumno que ganó se mantuvo en cabeza durante toda la actividad no le hizo falta mirar ni una sola vez el ranking de clasificación, enfocando su concentración en realizar el mayor número de misiones posibles.

Tipos de fallos que cometían

Observando el código de los alumnos podemos observar algunos fallos de sintaxis típicos en los comienzos de programación en PHP.

Por ejemplo, en PHP para declarar una variable es necesario ponerle el símbolo \$, sin embargo en varios intentos vemos que los estudiantes se olvidaban de este símbolo.

En otro caso, omitían los paréntesis que engloban la condición de una sentencia IF.

Otras veces, como suele pasar tanto a novatos como a expertos, se olvidaban de poner el ; al final de las líneas.

También podemos encontrar errores funcionales o de comprensión.

Algunos de ellos no comprendían el significado de función y ponían un ejemplo dentro del código de su misión. Otros confundían devolver un valor en una función (que sería con la sentencia return) con escribir la función en la pantalla (con la palabra reservada echo).

Fuentes de información o estrategias seguidas para resolver el problema

Al comenzar la actividad me pareció importante insistirles en que podían buscar cómo realizar las misiones en internet. Considero que es esencial que aprendan a buscar, ellos mismos, las soluciones y que tengan en mente la cantidad de posibilidades que ofrece este medio, sobre todo en cuanto a la programación. Existen multitud de foros, tutoriales, etc.

Muchos de ellos ponían en el buscador de Google las funciones que tenían que realizar. En la tercera misión el alumno 2 estuvo atascado porque no encontraba el error, consiguió superar la misión buscando en el manual web de PHP.

Comportamientos, Impresiones generales

El tiempo en que estuvieron realizando la actividad se pasó muy rápido, en cuanto me quise dar cuenta ya sólo quedaban 5 minutos para que sonase la campana y finalizase la clase.

Al principio, a algunos de los alumnos les costó unos minutos situarse. Fui por los puestos intentando ayudarles a buscar la información para que fueran realizando las misiones.

El alumno dos, parecía tener mayor facilidad para superarlas. Al superar alguna levantaba los brazos en signo de victoria. El alumno 4, por su parte, miraba bastantes veces la sección de competidores recalando que el alumno 2 les iba ganando.

Antes de pulsar el botón “End Current Race” avisé a los alumnos de que la carrera había finalizado, sin embargo, alguno de ellos se quejaba y se preguntaba al mismo tiempo:

- ¡Eeeh! ¿Pero qué ha pasado?

Otro comentaba:

- Ahora que le había pillado el tranquilo...

Normalmente en clase no les permiten tener acceso a internet, por lo que están acostumbrados a llamar al profesor cada vez que encuentran problemas para solucionar

los ejercicios. Creo que para realizar esta actividad, y para su futuro profesional está bien que aprendan a buscar las soluciones solos, así lo hicieron.

Al alumno número 8 le costó situarse, debió estar despistado cuando les comenté que debían programar en PHP. Él pensaba que la competición era en lenguaje C. Una vez aclarado, las siguientes dos misiones le resultaron muy fáciles y consiguió llegar hasta la misión 4.

En el número de intentos del alumno 7 vemos claramente que las misiones iban aumentando de complejidad, ya que el número de intentos va en incremento. En la misión uno realiza solamente un intento, para la misión dos 9, para la tercera 12, la cuarta misión no le dio tiempo más que a realizar dos intentos.

Al alumno 9 le resultaron muy fáciles las dos primeras misiones. La tercera, al tratarse de un bucle le resultó más difícil, por tanto, no le dio tiempo a intentar la siguiente misión. Al final consiguió realizar esta.

Limitaciones

Existen algunas limitaciones significativas en el desarrollo de esta investigación.

Sólo diez alumnos participan en la prueba.

Para que se acostumbren a realizar este tipo de ejercicios habría que realizarlos más de una vez.

Y por último comentar que hubiera sido mejor emplear más tiempo para la competición, de manera que pudiesen llegar a realizar más ejercicios. Las últimas misiones, son más complejas por lo que hubieran requerido mayor perseverancia y mayor análisis por parte del alumno.

Trabajo futuro

Evaluando la actividad y conociendo otros trabajos relacionados, se puede proponer mejorar la aplicación mediante una metodología colaborativa además de competitiva. Se podría plantear a los alumnos que realizasen misiones con sus respectivos resultados de manera que ellos mismos fueran enriqueciendo la aplicación.

Trabajar en equipo también podría ayudarles a ser más eficaces. Suele ocurrir que si luchas individualmente, te rindes más fácilmente que si compites con varias personas en un mismo equipo.

Conclusiones

En este trabajo hemos observado que los alumnos responden bien ante la nueva forma de trabajar problemas prácticos cortos de programación.

Aunque el incremento de la motivación no es significativo, la impresión general y la observación cualitativa demuestran que es una técnica interesante para realizar en las clases prácticas de programación.

Referencias

[1] - 2012. Wikipedia. Disponible en: <http://es.wikipedia.org/wiki/Programación>.

[2] - Caitlin Kelleher, Dennis Cosgrove, David Culyba, Clifton Forlines, Jason Pratt, and Randy Pausch. Alice2: Programming without Syntax Errors. 2002. User Interface Software and Technology.

- [3] – Mara Saeli, Jacob Perrenet, Wim M.G. Jochems, Bert Zwaneveld. Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective .2011. Informatics in Education, Vol. 10, No. 1, 73–88.
- [4] – Elena Verdú, Luisa M. Regueras, María J. Verdú, José P. Leal, Juan P. de Castro, Ricardo Queirós. A distributed system for learning programming on-line. 2012. Computers & Education 58, 1–10.
- [5] – Richard M. Ryan y Edward L. Deci. Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. 2000. Contemporary Educational Psychology 25, 54–67.
- [6] – Mancia Anguita y F. Javier Fernández-Baldomero. Software Optimization for Improving Student Motivation in a Computer Architecture Course. 2007. IEEE TRANSACTIONS ON EDUCATION, VOL. 50, NO. 4.
- [7] – Simon Attle y Bob Baker. Cooperative Learning in a Competitive Environment: Classroom Applications. 2007. International Journal of Teaching and Learning in Higher Education, Volume 19, Number 1, 77-83.
- [8] – Disponible en: <http://www.eweek.com/c/a/IT-Management/10-Programming-Languages-You-Should-Learn-Right-Now/>.
- [9] – Disponible en: <http://janox.sourceforge.net/janox.php?page=p4&lang=eng>.
- [10] – Andrew Trotman y Chris Handley. Programming contest strategy. 2008. Computers & Education 50 821–837.
- [11] – Fredik Manne. Competing in Computing. 2000. In Proceedings of the 2000 Norsk Informatikkonferanse .
- [12] – Ramon Lawrence. Teaching Data Structures Using Competitive Games. 2004. IEEE TRANSACTIONS ON EDUCATION, VOL. 47, NO. 4.
- [13] – Luisa M. Regueras, Elena Verdú, María Jesús Verdú, y Juan P. de Castro. Design of a Competitive and Collaborative Learning Strategy in a Communication Networks Course.2011. IEEE TRANSACTIONS ON EDUCATION, VOL. 54, NO. 2.
- [14] – Federica Battisti, Giulia Boato, Marco Carli, y Alessandro Neri.2011. IEEE TRANSACTIONS ON EDUCATION, VOL. 54, NO. 3.
- [15] – Miguel A. Revilla, Shahriar Manzoor y Rujia Liu. Competitive Learning in Informatics: The Uva Online Judge Experience. 2008. Olympiads in Informatics, Vol. 2, 131–148.
- [16] – Ning Chen. A Vision-Guided Autonomous Vehicle: An Alternative Micromouse Competition. 1997. IEEE TRANSACTIONS ON EDUCATION, VOL. 40, NO. 4.
- [17] – Mark J. Paulik y Mohan Krishnan. A Competition-Motivated Capstone Design Course: The Result of a Fifteen-Year Evolution. 2001. IEEE TRANSACTIONS ON EDUCATION, VOL. 44, NO. 1.
- [18] – Samara L. Firebaugh, Member, IEEE, and Jenelle A. Piepmeier. The RoboCup Nanogram League: An Opportunity for Problem-Based Undergraduate Education in Microsystems. 2008. IEEE TRANSACTIONS ON EDUCATION, VOL. 51, NO. 3.
- [19] – 2012. Wikipedia. Disponible en: [http://es.wikipedia.org/wiki/Caja_blanca_\(sistemas\)](http://es.wikipedia.org/wiki/Caja_blanca_(sistemas))
- [20] – 2012. Wikipedia. Disponible en: [http://es.wikipedia.org/wiki/Caja_negra_\(sistemas\)](http://es.wikipedia.org/wiki/Caja_negra_(sistemas))