

Las Maratones de Programación, un paso más al campo de la Investigación

Sandra Marcela Guerrero Calvache, Esteban Danilo Díaz Riascos, Franco Esteban Córdoba Pérez, Gonzalo José Hernández.

Ingeniería de Sistemas, Facultad de Ingeniería, Universidad de Nariño, Calle 18 Cr 50
Ciudadela Universitaria Torobajo, Pasto, Nariño, Colombia

waynaudenar@gmail.com

Resumen. Las maratones de programación han tomado gran importancia pues estas se realizan a nivel mundial. Algunas de ellas como las maratones de la ACM/ICPC (Association for Computing Machinery/International Collegiate Programming Contest) que patrocina IBM con el fin de fomentar la creatividad, el trabajo en equipo y la innovación en la creación de nuevos programas de software, y permite a los participantes poner a prueba su capacidad para actuar bajo presión, estas van dirigidas a cualquier universidad del mundo que desee vincularse. A través de este artículo se ampliará sobre cómo es la metodología de desarrollo, con qué finalidad se hacen, como se realizan y que se debe tener en cuenta a la hora de participar. Al finalizar el documento, se presenta la investigación iniciada por la Universidad de Nariño con la cual se busca fortalecer estas habilidades en los estudiantes del programa de ingeniería de sistemas.

Palabras Clave: Algoritmos, Búsqueda de soluciones, Competitividad, Dificultades, Juez Online, Lenguaje de Programación, Lógica, Maratón, Métodos, Problemas, Tácticas, Veredicto.

1 Introducción

Desde el auge de la informática las grandes empresas de computación se han interesado en vincular buenos programadores a sus grupos de desarrollo, debido a esto, se han estado realizando hace varios años maratones de programación, las cuales se llevan a cabo anualmente, en estas participan varias universidades de todo el mundo donde son presentados problemas de lógica, matemáticos, trigonométricos, entre otros, que deben resolverse usando algoritmos de programación en el menor tiempo posible.

Estos eventos pretenden entrenar a estudiantes apasionados por la programación para que puedan competir a nivel nacional, suramericano e internacional. Estas también han servido para que estudiantes destacados sean llamados por grandes empresas.

Este documento está dividido en dos capítulos además del Capítulo 1 que corresponde a esta introducción. El Capítulo 2 presenta el estado del arte relacionado con la programación competitiva llevado a cabo por medio de las maratones de programación y el Capítulo 3 expone a la comunidad científica el proyecto de investigación iniciado por el programa de Ingeniería de Sistemas de la Universidad de Nariño para mejorar el desempeño de sus estudiantes en estas actividades.

2 Sobre las maratones de programación

Para muchas personas es desconocido aun el concepto de una maratón de programación, puesto que a pesar de que está relacionado con las ciencias exactas y de computación, hoy en día se ha vuelto más importante de lo que se cree. En este tipo de concursos se busca que personas con talento y capacidades exploren nuevas formas de plasmar sus conocimientos mediante la resolución de problemas de forma idónea, ayudándose de un lenguaje de programación para poner en marcha la solución deseada.

Existen además, páginas web donde entidades u organizaciones encargadas de promover este tipo de eventos tales como Google, ACIS, ACM-ICPC (ACM International Collegiate Programming Contest), REDIS (Red de Decanos y Directores de Ingeniería de Sistemas) entre otras, invitan a que estudiantes de establecimientos de educación superior participen y resuelvan los diversos tipos de problemas que estos alojan allí, algunos más complejos que otros y con un límite máximo de ejecución del programa para que sean evaluados a través de un juez-online.

Existe gran diversidad de situaciones problemáticas las cuales mediante técnicas, herramientas y análisis interpretativo se busca lograr los resultados pedidos en los enunciados, tan rápido como sea posible, poniendo a prueba entonces habilidades y destrezas.

Hoy en día, la Competición Internacional Universitaria ACM es una competición anual de programación y algoritmia entre universidades de todo el mundo patrocinada por IBM. Con esto, muchos equipos de diferentes lugares, participan en las maratones y cada año el interés por estas es cada vez mayor. Existen diversas fases para clasificar, comenzando por competiciones locales, regionales y luego la final mundial.

Por lo tanto, en Colombia la Organización de Maratones de Programación (OMPC), apoya y realiza dichas competencias. Las principales son dos actividades en las cuales se relacionan con la competición de ACM para realizar la clasificatoria respectiva estas son la Nacional realizada por lo general el mes de Septiembre y la Regional Latinoamericana en el mes de Noviembre, esta última simultánea con sedes en Argentina, Bolivia, Brasil, Colombia, Chile, Cuba, México, Perú y Venezuela.

A su vez existen aproximadamente doce actividades de entrenamiento en el año, las cuales los equipos se inscriben a las mismas para ir mejorando sus aptitudes, buscar la manera de resolver la mayor cantidad de ejercicios posibles y seguirse preparando y aprendiendo más cosas día con día, estas pueden ser de forma presencial en sedes estipuladas o vía virtual.

2.1 Las reglas del juego

En cuanto a las reglas de la competición oficial, el equipo debe estar conformado por un máximo de 3 integrantes acompañados por un coach, y bajo un tiempo límite de desarrollo para estas maratones de 5 horas para solucionar entre 8 y 10 problemas.

En el momento de la competición, el entrenador no puede intervenir en esta, a excepción del entrenamiento que se realiza momentos antes de la maratón oficial, la

cual consta de la resolución de dos ejercicios con el fin de que los concursantes se familiaricen con los equipos de trabajo, tengan claro cómo se maneja la plataforma y conozcan los entornos de programación que están disponibles.

Los lenguajes a emplear son C, C++ o Java según sea la preferencia de trabajo que se desee. Lógicamente el equipo que más cantidad de problemas resuelva es el ganador.

Entre los tipos de problemas más destacados en estos concursos son los siguientes:

Ad Hoc: Es un tipo de problema donde cada descripción y su correspondiente solución son únicas. Estos pueden clasificarse en directos los cuales requieren solo la traducción del problema para luego ser codificado, o de simulación donde hay unas reglas que deben ser simuladas para obtener una respuesta. De un punto de referencia de 10 problemas pueden aparecer 1 o 2 de este tipo.

Complete Search: Es un tipo de problema en donde se deben implementar búsquedas ya sean recursivas o iterativas. Un concursante debe desarrollar una solución completa de búsqueda cuando es evidente que no hay algoritmo inteligente disponible.

Divide & Conquer: Consiste en dividir el problema en dos o más sub-problemas, más sencillos, de manera que la solución global sea la unión de las sub-soluciones.

Algunos pasos para implementar este tipo de problemas son:

- Dividir el problema original en sub-problemas.
- Encontrar sub-soluciones para cada uno de los sub-problemas.
- Si es necesario, combinar las sub-soluciones para generar una solución completa al problema principal.

Greedy: Son los tipos de problemas relacionados con optimización. Consiste en elegir la opción más eficiente en cada paso local con la esperanza de llegar a la solución general más óptima. Pueden ser clásicos u originales.

Dynamic Programming: Es un procedimiento matemático diseñado principalmente para mejorar la eficiencia de cálculo de problemas de programación matemática seleccionados, descomponiéndolos en sub-problemas de menores tamaños y por consiguiente más fáciles de calcular. También clásicos u originales.

Graph: Utilizados para representar mapas de rutas, organización de procesos, espacios de búsqueda para juegos, circuitos lógicos y demás. Un grafo es simplemente una conexión de vértices y enlaces que almacenan información conectada entre ellos, haciendo que su búsqueda sea más rápida.

Mathematics: Es la aplicación de técnicas matemáticas para la resolución de problemas algorítmicos. Puede existir una combinación entre algunos de los tipos de problemas ya mencionados.

String Processing: Es la solución de problemas mediante la utilización de estructura de datos, y algoritmos eficientes para cadenas.

Computational Geometry: Son un conjunto de problemas con mucha más complejidad puesto que es más difícil encontrar el algoritmo correcto para dichas situaciones. Pueden estar relacionados con geometría básica o geometría computacional.

Some Harder Problems: En este caso puede haber una combinación de varios tipos de problemas, o que estos no entren en ninguna de las categorías mencionadas con anterioridad.

Como se puede observar existe una gran variedad de problemas, los cuales se relacionan mucho con el diseño de algoritmos que involucren a las ciencias exactas, tales como matemáticas, física, la lógica computacional y varios trucos de programación con el fin de que al implementar la posible solución esta sea la más óptima, con los recursos suficientes (tiempo), y generando la respuesta correcta.

Sin embargo cabe destacar que no siempre los resultados son los esperados para todos los equipos, puesto que en el transcurso de la resolución de los problemas se pueden presentar dificultades que pueden llegar a ser un obstáculo, pero que si se saben manejar con inteligencia, analizando mejor el problema, con los casos de prueba necesarios y calculando su tiempo, complejidad y con la memoria justa, se obtendrá soluciones que al final serán todo un éxito.

Es importante mencionar los veredictos que estos jueces vía online exponen cuando el código es enviado para ser evaluado, obteniendo como respuesta que este sea correcto o en caso contrario de que no funcione después de habersele aplicado los casos de prueba correspondientes, entonces así los maratonistas puedan depurar su código y hacer las correcciones respectivas para que este sea apto:

Accepted (Aceptado) (AC): El programa es correcto y corre normalmente en el tiempo previsto.

Presentation Error (Error de presentación) (PE): El programa da las salidas correctas pero éstas no están ajustadas al formato.

Wrong Answer (Respuesta Incorrecta) (WA): La respuesta que da el programa es incorrecta frente a los casos de prueba del juez de la plataforma.

Compile Error (Error de Compilación) (CE): El Compilador no comprende cómo funciona el programa. Puede existir mensajes que interfieren en dicho proceso.

Runtime Error (RE): El programa falla durante la ejecución debido a defectos de segmentación, comprobación de referencias inválidas (división entre cero).

Time Limit Exceeded (Tiempo Limite Excedido) (TL): El programa toma mucho tiempo en los casos de prueba, se requiere más eficiencia.

Memory Limit Exceeded (ML): El programa intenta usar mucho más memoria que la permitida.

Output Limit Exceeded (OL): El programa intenta imprimir demasiadas salidas, es decir está alojado en un ciclo infinito.

Restricted Function (Funciones Restringidas) (RF): El programa intenta usar funciones ilegales del sistema tales como `fork ()` o `fopen()`.

Submission Error (Error de Sumisión) (SE): Errores en el campo de información del problema cómo código de información o el número del problema.

Frente a los errores que pueden presentarse al momento de resolver un problema cabe mencionar que se debe tratar de buscar las mejores alternativas para mejorar esta situación, analizando las falencias que el programa puede presentar tanto en el rendimiento como en la ejecución del algoritmo.

Al momento de obtener las entradas y generar las salidas necesarias, se debe tener en cuenta los siguientes aspectos:

Entradas:

- La entrada debe leerse en forma estándar.

- La entrada contiene varios casos de prueba. Cada caso es descripto usando el número de líneas que dependan de cada problema.
- Cuando una línea contiene varios valores, estos son separados por únicos espacios. No debe haber líneas vacías.
- No hay datos extra después de los casos de prueba mencionados en la entrada.

Salida:

- La salida debe ser escrita en forma estándar.
- El resultado de cada caso debe aparecer en la salida usando el número de líneas requeridas dependiendo del problema.
- Cuando una línea de resultados contiene varios casos de prueba, estos deben ser separados por espacios en blanco. Ningún otro espacio debe aparecer en la salida. No debe haber líneas vacías.
- Después de la salida de todos los casos de prueba. Ningún dato extra debe ser escrito en la salida.

El archivo a enviar debe tener la siguiente estructura `<codename>.c`, `<codename>.cpp` o `<codename>.java`, indicando el título del problema respectivamente.

Al finalizar la jornada y en el momento de determinar los ganadores si llega a existir un caso de empate el equipo que tenga menos minutos de penalidad será el ganador, esta penalidad será la cantidad de minutos desde el inicio de la competencia y más 20 minutos extra por cada intento fallido, estos se suman solo en caso de que se resuelva el problema.

2.2 El entrenamiento es indispensable

Es importante incentivar además a la comunidad estudiantil universitaria a explorar de estos nuevos conceptos, poner a prueba sus conocimientos, investigar formas y metodologías de desarrollo, pero sobre practicar cada vez más, para que desde luego los resultados esperados sean los más satisfactorios para el grupo de trabajo.

El que regularmente se incentive a estar en constante práctica genera que la persona se estimule a resolver paradigmas de programación con mayor frecuencia, y para llevar a cabo dicho entrenamiento se presentan a continuación diversos sitios vía online.

Comenzando por la Universidad de Valladolid Online Judge (UVa) el cual contiene un gran conjunto de problemas de anteriores competiciones y de otras fuentes, todos estos completando una excelente colección para que el usuario esté dispuesto a fijarse nuevos retos.

Seguido por ACM IPC Live Archive el cual alberga los problemas de las maratones finales mundiales desde el año 2000.

Otro relevante sitio es Usaco Training Personalized Curriculum, un lugar para aprender acerca de los concursos relacionados con el área de programación, dirigido a cargo de las Olimpiadas de Computación USA.

Google Code Jam como un concurso virtual de arduos problemas algorítmicos que participan año tras año muchas personas de todo el mundo y es organizado por la compañía Google.

Finalmente Caribbean Online Judge (COJ) y TopCoder dando la oportunidad de seguir ofreciendo buenos ejercicios de diversos tipos de dificultad para que los concursantes se sientan motivados y capaces de encontrar la respuesta correcta.

2.3 Explorando el camino de la programación

Más allá de una idea de lo que es programación, es necesario al menos dominar un lenguaje de esta categoría, que permite demostrar lo que se quiere plasmar a través de la lógica, el análisis y la interpretación.

Existen varios componentes que hace que un concursante logre lo que se propone en una competición, por ejemplo es indispensable destacar que al momento de comenzar a resolver un problema este debe tomar en cuenta la categoría a la que pertenece, para tener un concepto previo de qué camino tomar.

Antes de empezar a codificar, es importante indagar si el algoritmo que se está creando antes y después de leer el enunciado es el más apropiado para la situación problemática y que sea eficiente en recursos de tiempo y complejidad.

A parte de los casos de prueba que se muestran, es necesario formular otros más que puedan ser pieza clave en la resolución del problema y que produzcan las salidas correctas dentro del límite de ejecución.

Entre el contenido que se debe manejar para implementar un buen algoritmo está el manejo de estructuras de datos lineales y no lineales, de búsqueda, grafos, la programación dinámica, y el uso correcto de librerías estándares, entre otras perspectivas que dependerán del interés del concursante en particular.

Aunque los enunciados están en inglés, esto no debe ser un impedimento para motivarse y seguir aprendiendo, puesto que si no se maneja este idioma, se pueden utilizar herramientas didácticas tales como diccionarios que faciliten la comprensión de los problemas sin pretexto alguno. Además para la ayuda de los concursantes, estos pueden llevar al momento de la competición material impreso con información que les sea útil como implementación de algoritmos, problemas matemáticos y demás ejercicios resueltos con anterioridad.

Con esto se demuestra aún con mayor claridad que más que un concurso es un compromiso que requiere dedicación, tiempo, pero sobre todo investigación, lectura en libros, artículos y demás documentos de apoyo en la rama de computación que sirvan para mejorar las prácticas de programación y que demuestre a que año tras año más instituciones se vinculen a participar, a estimular a sus estudiantes a crear nuevos desafíos, y las maratones son un buen comienzo para lograrlo.

2.4 ¿Porque participar?

En el momento de participar existen varias razones por las cuales un equipo de trabajo se siente motivado a seguir aprendiendo, entrenándose para fortalecer sus habilidades, adquirir nuevos conocimientos y fijarse nuevas expectativas, todo esto con el fin de que cada competencia sea un reto para avanzar, clasificar y estar al lado de los otros mejores equipos.

La importancia de participar radica en que esta es una buena oportunidad para empezar a fomentar el trabajo en equipo, respetar los criterios de cada uno de los

integrantes, y plantear así una solución más óptima a los problemas formulados con los aportes que cada uno puede llegar hacer.

Por otro tanto permite profundizar en tópicos avanzados que generalmente los cursos universitarios no incluyen, dando paso al estudio de los temas de ramas informáticas y no informáticas, mejorar notablemente la lógica y las técnicas de programación.

Entre otra de las ventajas que trae participar en las maratones es que estas competencias generen posibilidades a eventos de índole nacional en la parte computacional, y que entidades como Google, IBM y demás, abran sus puertas a estudiantes con talento para que puedan realizar sus pasantías y posteriormente tengan mayores oportunidades de empleo en una de estas prestigiosas compañías. Además el entrenamiento continuo involucra más dedicación por parte de los miembros del equipo, de la mano con la investigación, así como también saber perder y saber ganar, pues aunque el rendimiento no sea siempre el mejor, se debe seguir fortaleciendo el estudio, la disciplina, y no rendirse a la primera vez, sino por el contrario buscar mayores motivaciones manteniendo siempre el equilibrio entre una sana competencia y no adoptando técnicas erróneas por intentar de codificar rápido y buscar solo triunfos al participar.

3 El papel de las Universidades en las maratones de programación

Más de 15 universidades colombianas aproximadamente, están vinculadas a participar en las maratones de programación. Sería maravilloso que más establecimientos de educación superior no solo en nuestro país, sino alrededor del mundo, fomenten a los estudiantes a involucrarse más de lleno a este campo, a que exista mayor preparación, para darse a conocer, pues esto también genera grandes reconocimientos no solo para ellos sino también como Institución buscando una formación integral para que a medida de que existan equipos sobresalientes ayuden a las siguientes generaciones a prepararse cada vez más.

3.1 La labor de la Universidad de Nariño

La Universidad de Nariño es una de las instituciones que participa en las maratones de programación desde el año 2012.

Aunque esta institución ha tratado de ser sede de entrenamiento en algunas ocasiones, el reducido porcentaje de participación por parte de los estudiantes y los no tan favorables resultados que se obtienen por los pocos que participan, ha sido un punto clave para que esto no se lleve a cabo. Una de las posibles causas a esta situación, es que no existe un semillero de investigación para maratonistas en donde se trate de orientarlos al conocimiento y resolución de problemas, así también como un apoyo para realizar un análisis minucioso del planteamiento y ejecutar su respectiva implementación en un lenguaje de programación de la manera más óptima, junto con un seguimiento de las posibles dificultades que puedan surgir en el camino e impedir que este tipo de obstáculos obstruyan las ideas del programador.

3.2 Proyectos de Investigación

A raíz de lo mencionado, se ha planteado el desarrollo de un proyecto llamado “Análisis de competencias requeridas en los estudiantes del programa de Ingeniería de Sistemas de la Universidad de Nariño para participar apropiadamente en las maratones de programación”, en el cual se pretende establecer cuáles son las falencias que existen en el momento de involucrarse de lleno en las maratones de programación y las dudas que puedan surgir a consecuencia de la exigencia de los problemas, pues se observa que estos no solo requieren un conocimiento en un lenguaje de Programación, también es indispensable el manejo de las ciencias exactas y de habilidades lógicas que ayuden a interpretar el enunciado, además de entender cómo un juez electrónico evalúa la ejecución de una posible solución.

3.3 Objetivos del Proyecto de Investigación

Elevar el posicionamiento del departamento de Sistemas de la Universidad de Nariño a nivel nacional: Con esto se pretende ante todo, pertenecer al grupo de universidades que colaboran con la organización y difusión de las maratones de programación.

Establecer una maratón de programación interna en la Universidad de Nariño con la ayuda de un juez electrónico: Con esto se quiere dar a conocer este tipo de eventos a los estudiantes y a su vez lograr la creación de propios ejercicios para un entrenamiento adecuado y realizar una maratón oficial que permita hacer una preselección de los grupos que deseen representar a la universidad en otras ciudades en las maratones nacionales e internacionales.

Fomentar la motivación y participación del estudiantado en las maratones de programación de manera relevante: Aquí se busca incrementar el conocimiento, perfeccionar las habilidades y el desarrollo tecnológico.

Incrementar la confianza de los estudiantes del departamento de Sistemas para que se presenten con mayor interés a las maratones de programación: Que el número de equipos participantes ascienda considerablemente en los eventos programados para las maratones de programación.

Mejorar los resultados obtenidos hasta la fecha en las maratones de programación organizadas por ACIS: Ya que las tres participaciones anteriores por parte de la universidad no han tenido los mejores resultados se pretende que con el desarrollo del proyecto se permita que los estudiantes que participen obtengan un puntaje superior a los alcanzados.

3.4 Fases del Proyecto de Investigación

Además de los objetivos anteriormente planteados se postulan ciertas fases del desarrollo del proyecto las cuales se mencionan a continuación junto con el propósito de cada una de estas.

Fase inicial reconocimiento del problema: En esta fase se pretende establecer claramente las actividades que podrían ser la base fundamental de la investigación con el fin de cumplir con los objetivos planteados de la mejor manera posible. Ante ello se busca estructurar las ideas que ayuden la profundización del tema en las maratones de programación, su organización e importancia hoy en día.

Comenzando por la recolección de ejercicios de maratones anteriores y el análisis de la anatomía de los mismos, con los cuales mediante su formulación, se identificará el grado de dificultad, y con ello comprender la manera más adecuada de asimilarlos, interpretarlos, entenderlos y luego resolverlos tomando en cuenta los criterios de solución establecidos según la plataforma en la que se esté trabajando.

Posteriormente con el análisis del trabajo de los jueces electrónicos se busca establecer claramente su metodología de evaluación, las condiciones y limitaciones que estos imponen a la hora de participar, y la estructura interna de estos en la recepción de los problemas enunciados.

Para que esto se lleve a cabo es necesaria la búsqueda de ejercicios en las plataformas en línea de los diversos concursos realizados frecuentemente, la lectura de documentos acerca de la forma evaluativa de los jueces y sus diversos criterios e instalar en la Universidad de Nariño una plataforma de evaluación de juez online para comprender mejor su funcionamiento.

Fase intermedia de la investigación: Con la catalogación de los problemas se llevara a cabo un registro de cada uno de ellos, justamente con las características principales identificadas al profundizar los ejercicios y con el fin de concretar algunas pautas de resolución de una manera más adecuada, correcta y que a su vez sea informativa como un apoyo para los estudiantes que participan en estos concursos. Por otro lado en el entrenamiento en el circuito de maratones se busca implementar más el trabajo arduo en equipo, la búsqueda de conocimiento no solo en el área de programación, sino también en la lectura de algoritmos o documentos claves que faciliten la comprensión y desarrollo de los ejercicios, pero ante todo estimular a que más estudiantes se involucren en estas actividades que realmente son interesantes y que se trate de explorar ese talento que podría ser muy útil no solo como parte de la desenvoltura personal sino también como un camino para lograr que muchas otras personas se den a conocer en otros lugares de nuestro departamento y se reconozca aún más el nombre de la Universidad.

Es por eso que en esta fase se resalta el entrenamiento y participación activa en maratones ya sea de manera virtual o presencial que conlleven a trabajar de una manera más eficiente y de gran colaboración grupal e incentivar a más estudiantes a internarse de lleno en el área de la programación, a que exploren sus capacidades en ello e impulsarlos a que estas actividades se conviertan también en su hábito de estudio y de aprendizaje donde se busca irse preparando y perfeccionando considerablemente; junto con la ayuda de otros docentes o personas capacitadas en esta rama, sea de nuestra universidad o de otra diferente. Se busca que mediante entrevistas, reportes, encuestas,

entre otros, se obtenga información de vital importancia que nos lleven a deducir las causas de que existan falencias en algunos estudiantes en la resolución de dichos ejercicios, y las sugerencias que se puede seguir apropiándolas a las necesidades de este proyecto. Finalmente la implementación de una maratón interna teniendo en cuenta los factores relacionados a ello y buscando que el estudiante llegue más preparado y seguro de su conocimiento.

Fase final y conclusión de los procesos a investigar: Al establecer realmente cada uno de los procesos anteriormente mencionados y su funcionalidad, se concluye tomar los datos recolectados como aportes que permita la cualificación y cuantificación de la información propuesta en los problemas, y sean de gran utilidad para que cualquier persona interesada en este tema pueda sentirse apropiado con la temática antes, durante, y después de participar en las maratones o concursos de Programación. La documentación resguardará y apoyará esas ideas y se presentará un informe final que establezca los resultados de la investigación.

3.5 Identificación y caracterización de la innovación propuesta

Mediante los procesos investigativos acerca de las maratones de programación, más que la idea de generar mayor participación estudiantil se busca es la preparación idónea y asertiva en cada una de estas, asumiendo ante todo los conocimientos aprendidos en las asignaturas de Programación, Lógica, Matemáticas e Inglés para la continua y eficaz resolución de los problemas a tratar y por consiguiente aumentar el nivel de resultados de tal manera que sean los más óptimos y se descubra allí las dificultades que pueden existir y tratar de reducirlas justamente con las destrezas, aptitudes y agilidades de cada uno de los miembros del equipo; para esto se desea implantar una red de Programación Interna en la Universidad de Nariño.

Es una propuesta realmente interesante e innovadora puesto que en la Universidad no se ha puesto en ejecución proyectos tan significativos como es el explorar las capacidades de los estudiantes en esta área, estimularlos a su correcto acondicionamiento intelectual que pueda traer realmente efectos positivos y que con el acompañamiento de organizaciones dedicadas a ello hagan cumplir satisfactoriamente con este objetivo.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el Departamento de Sistemas de la Facultad de Ingeniería y de la Vicerrectoría de Investigaciones de la Universidad de Nariño y forma parte del proyecto de Investigación “Análisis de competencias requeridas en los estudiantes del programa de Ingeniería de Sistemas de la Universidad de Nariño para participar apropiadamente en las maratones de programación”.

Los autores desean expresar su agradecimiento al docente Gonzalo Hernández por su asesoramiento en el proyecto de investigación y al Centro de Investigaciones Udenar por su apoyo logístico.

Referencias

1. Steven Halim, Felix Halim: Competitive Programming 3 The New Lower Bound of Programming Contests. Handbook for ACM ICPC and IOI Contestants (2013).
2. Steven S. Skiena, Miguel A. Revilla: Programming Challenges. The Programming Contest Training Manual. Editors David Gries, Fred B. Schneider. Springer-Verlag New York, Inc (2003).
3. ACIS.(2014). ACM International Collegiate Programming Contest. Bogota DC, Colombia. <http://www.acis.org.co/index.php?id=556>
4. ACIS.(2013). Bogota DC,Colombia. <http://www.acis.org.co/index.php?id=48>
5. GUI (Grupo Universitario de Informática de Valladolid). (2014). Guía de uso UVa Online Judge. Valladolid, España. <http://foro.gui.uva.es/viewtopic.php?f=4&t=34>
6. Caribbean Online Judge (COJ). (2014). Acerca de. La Habana, Cuba. <http://coj.uci.cu/general/about.xhtml?lang=es>