

**Desarrollo e implementación de un programa de clase para el semillero  
de programación**

**Autor:**

Ana Echavarría Uribe

**Tutor:**

Juan Francisco Cardona Mc'Cormick

Universidad Eafit

Escuela de Ingeniería

**Universidad Eafit**

**Ingeniería Matemática**

**Medellín**

**2013**

## Tabla de Contenidos

<b>1. Planteamiento del Problema</b>	<b>2</b>
<b>2. Objetivos generales y específicos</b>	<b>2</b>
2.1. Objetivo General . . . . .	2
2.2. Objetivos Específicos . . . . .	2
<b>3. Antecedentes</b>	<b>3</b>
<b>4. Justificación</b>	<b>4</b>
<b>5. Alcance</b>	<b>4</b>
<b>6. Metodología Propuesta</b>	<b>4</b>
<b>7. Cronograma de Actividades</b>	<b>5</b>
<b>8. Presupuesto</b>	<b>6</b>
8.1. Personal . . . . .	6
8.2. Materiales . . . . .	6
<b>9. Propiedad Intelectual</b>	<b>6</b>

## **1. Planteamiento del Problema**

El semillero de programación es un grupo de la universidad que tiene como objetivo preparar a los estudiantes para las maratones de programación realizadas por la ICPC (International Collegiate Programming Contest). En este semillero se enseñan los temas más útiles para estas competencias que son los algoritmos de grafos, strings y teoría de números, paradigmas de programación como la programación dinámica y la recursividad y estructuras de datos.

Durante los últimos años el semillero ha estado a cargo de estudiantes destacados en las maratones de programación bajo la supervisión de docentes del departamento de ingeniería de sistemas; sin embargo, durante este tiempo, no se desarrolló un plan de trabajo para el curso. Lo que se busca en esta práctica investigativa es desarrollar e implementar un programa de clases para el semillero de programación que le de a los estudiantes las bases teóricas y prácticas para enfrentarse por primera vez a las maratones de programación.

## **2. Objetivos generales y específicos**

### **2.1. Objetivo General**

Desarrollar e implementar un programa de clases para el semillero de programación que de a los estudiantes las bases necesarias para participar en las maratones de programación de la ICPC.

### **2.2. Objetivos Específicos**

- Enseñar los conceptos básicos de la STL (Standard Template Library) de C++.
- Introducir a los estudiantes a los diferentes jueces en línea.
- Buscar y sugerir problemas en los diferentes jueces que permitan a los estudiantes aplicar los conceptos aprendidos en las clases.
- Discutir las soluciones a los problemas propuestos luego de que los estudiantes hayan intentado resolverlos de manera independiente.
- Crear diapositivas con los temas enseñados y enviarlas a los estudiantes para que les sirvan como material de estudio en casa.

- Enseñar los algoritmos de recorrido de grafos en profundidad (DFS) y en anchura (BFS) y sus aplicaciones.
- Enseñar los algoritmos para hallar los caminos más cortos en grafos: Dijkstra, Bellman-Ford y Floyd-Warshall.
- Enseñar en qué consiste el paradigma de programación dinámica y los problemas más comunes que se resuelven usando este paradigma.
- Mostrar cómo se pueden hallar las ocurrencias de una palabra en un texto de manera rápida usando el algoritmo de KMP.
- Enseñar los algoritmos de teoría de números más útiles para las competencias de programación.
- Enseñar el algoritmo de Kruskal para hallar el árbol de mínima expansión.
- Enseñar los algoritmos de geometría más usados en las competencias de programación.

### 3. Antecedentes

El semillero de programación fue implementado hace varios años en la Universidad EAFIT y ha permitido que los estudiantes con interés en la programación, las matemáticas y los algoritmos tengan un espacio para aprender, interactuar y socializar problemas. En los años anteriores, el semillero era un espacio muy productivo para los estudiantes avanzados ya que durante las clases se socializaban y solucionaban problemas de dificultad media y alta lo que le permitía a quienes iban mejorar su habilidad de solución de problemas. Sin embargo, para los principiantes, el nivel del semillero era avanzado y se trataban problemas que ellos no tenían la fundamentación teórica para resolver.

El semestre 2012-1 se dividió el semillero de programación en dos grupos: básico y avanzado. En el semillero avanzado se siguieron resolviendo y discutiendo problemas como se hacía anteriormente y el semillero básico se enfocó en desarrollar las habilidades de programación de los estudiantes de primero y segundo semestre. Los resultados de la creación de un grupo básico fueron positivos.

## **4. Justificación**

Actualmente, los estudiantes que pertenecen al semillero de programación son en su mayoría de tercer semestre, lo que quiere decir que tienen conocimientos acerca de cómo programar mas no conocen las técnicas más utilizadas en la solución de problemas de maratones de programación como los son los algoritmos de grafos, strings y geometría, la programación dinámica y los conceptos y algoritmos básicos de teoría de números. Este semestre se busca crear un programa de clases que se ajuste al nivel de los estudiantes y les de la fundamentación teórica necesaria para resolver los problemas más comunes presentados en las maratones de programación.

Se busca que la documentación y diapositivas utilizadas este semestre sirvan como material de clase para futuras generaciones del semillero y para que aquellos que continúen a cargo del semillero de programación el semestre entrante sepan cuáles son los conceptos que los estudiantes actuales conocen y puedan continuar enseñándoles aquellos temas que los estudiantes todavía no conocen.

Se espera que los temas enseñados a los estudiantes durante este semestre y el próximo sirvan para que ellos tengan un buen desempeño en la Maratón Nacional de Programación ACIS/REDIS que se realiza en octubre y tengan la posibilidad de participar en la Maratón Regional Suramericana ACM-ICPC y en la Maratón Mundial ACM-ICPC bajo el nombre la Universidad como ha ocurrido en ocasiones anteriores.

## **5. Alcance**

La implementación de este proyecto permitirá que los estudiantes participantes en el semillero de programación, en su mayoría pertenecientes a las carreras de ingeniería de sistemas e ingeniería matemática de la Universidad se motiven y estén preparados para participar en el Circuito de Maratones de Programación ACIS/REDIS y en la Maratón Nacional de Programación ACIS/REDIS. De obtener buenos resultados en esta última competencia, los estudiantes podrían tener la oportunidad participar bajo el nombre de la Universidad en la Maratón Regional Suramericana ACM-ICPC y en la Maratón Mundial ACM-ICPC.

## **6. Metodología Propuesta**

El semillero de programación consiste en una clase semanal que dura entre 1 hora y media y 2 horas. La estructura de cada clases se compone de tres etapas principales:

1. Discusión y solución de los problemas propuestos como tarea en la sesión anterior.
2. Exposición del nuevo tema a trabajar.
3. Explicación y discusión breve de los problemas propuestos para la siguiente sesión.

Para desarrollar cada clase es necesario investigar acerca del tema a enseñar, preparar documentación que le permita a los estudiantes repasar lo aprendido y buscar y solucionar problemas que se ajusten al nivel y los temas aprendidos en la clase.

## 7. Cronograma de Actividades

El semillero consistirá de 15 clases que se dictarán los todos los viernes desde el 1 de febrero hasta el 17 de mayo de 2013. Los temas a tratar en cada clase son:

Fecha	Tema
Febrero 1	Introducción a C++ y al juez de programación Codeforces ¿En qué consiste una maratón de programación? Solución a un problema básico de maratón de programación
Febrero 8	Arreglos en C++, Vectores de C++ y Grafos
Febrero 15	STL de C++
Febrero 22	Algoritmos de recorrido de grafos: BFS y DFS
Marzo 1	Aplicaciones de los algoritmos de BFS y DFS
Marzo 8	Algoritmos de Dijkstra y Bellman-Ford
Marzo 15	Solución de problemas de repaso
Marzo 22	¿Qué es la programación dinámica? Soluciones básicas con programación dinámica Algoritmo de Floyd-Warshall
Abril 5	Aplicaciones de la programación dinámica: Longest Common Substring (LCS) y 0-1 Knapsack
Abril 12	Algoritmo de Knuth-Morris-Pratt
Abril 19	Algoritmos básicos de teoría de números
Abril 26	Solución de problemas de repaso
Mayo 3	Algoritmo de Kruskal para hallar el árbol de mínima expansión (MST)
Mayo 10	Algoritmo de Edmonds-Karp para hallar el máximo flujo
Mayo 17	Algoritmos de geometría

## 8. Presupuesto

### 8.1. Personal

La dedicación de la estudiante Ana Echavarría Uribe al proyecto será de 10 horas a la semana entre el 28 de enero y el 25 de mayo de 2013. Estas horas serán remuneradas al valor que tenga establecido la Universidad EAFIT para las monitorías y el costo será asumido por el Departamento de Ingeniería de Sistemas.

### 8.2. Materiales

Para la realización del proyecto será necesario el uso de un aula de clases cada semana durante dos horas. Esta aula será proporcionada por la Universidad EAFIT.

## 9. Propiedad Intelectual

Los resultados del trabajo pertenecen en un 50 % a la estudiante Ana Echavarría Uribe y en un 50 % al tutor Juan Francisco Cardona Mc.Cormick. El contenido del trabajo podrá ser utilizado en su totalidad para el desarrollo del semillero de programación sin previo aviso a los autores.

## Bibliografía

- [1] CORMEN, T.H., LEISERSON, C.E., RIVEST, R.L., STEIN, C. *Introduction to Algorithms*. The MIT Press, 3ra ed. 2009.
- [2] MIRZAYANOV, M. *Codeforces*. [www.codeforces.com](http://www.codeforces.com)
- [3] SKIENA, S.S, REVILLA, M.M. *Programming Challenges: The Programming Contest Training Manuals*. Springer, 2003.
- [4] HALIM, S.HALIM, F. *Competitive Programming: Handbook for ACM ICPC and IOI Contestants*. Lulu, 2010.
- [5] HALIM, S.HALIM, F. *Competitive Programming 2: Handbook for ACM ICPC and IOI Contestants*. Lulu, 2011.