

De las competencias de programación a una pedagogía educativa en La Paz (Bolivia)

From programming competitions to an educational pedagogy in La Paz (Bolivia)

Jorge Humberto Teran-Pomier

Instituto de Investigaciones en Informática, Universidad Mayor de San Andrés, Bolivia (teranj@acm.org)

Recibido el 6 de mayo de 2016; revisado el 27 de octubre de 2016; aceptado el 21 de noviembre de 2016; publicado el 2 de diciembre de 2016

RESUMEN:

La metodología para el aprendizaje de la programación de computadoras denominada programación competitiva por su origen en las competencias de programación, proporciona diversas ventajas en las clases de los primeros cursos de programación en la educación universitaria, suministrando retroalimentación instantánea a los estudiantes. Se presenta la forma en la que se deben elaborar los cursos de programación bajo esta metodología y los beneficios que aporta a los estudiantes y empresas.

El artículo se origina en una encuesta realizada a 20 docentes de programación de 7 universidades privadas y una pública. Muestra la situación en la enseñanza de la programación en la ciudad de La Paz, Estado Plurinacional de Bolivia, mostrando la factibilidad de implementar esta metodología. Los resultados evidencian que la retroalimentación de los resultados de las prácticas están directamente relacionados a la tasa de aprobación. Situación que hace muy deseable aplicar la programación competitiva como una herramienta pedagógica en la enseñanza de la programación.

PALABRAS CLAVE: LENGUAJE DE PROGRAMACIÓN, PROCESO DE APRENDIZAJE, MÉTODO DE ENSEÑANZA.

ABSTRACT:

The methodology for learning computer programming called competitive programming has

its origin in programming competitions, provides several advantages in the classes of the first programming courses in university education, by providing instant feedback to students. How to develop programming courses under this methodology and the benefits to students and businesses is presented.

This article starts with survey taken to 20 programming teaches in 7 private universities and one public. It shows the status of teaching programming in the city of La Paz Plurinational State of Bolivia, showing the feasibility of implementing this methodology. The results show that the feedback on programming assignments is directly related to student approval. This situation makes it very desirable to apply competitive programming as a pedagogical tool in teaching programming.

KEYWORDS: PROGRAMMING LANGUAGES, LEARNING PROCESS, TEACHING METHODS.

1 INTRODUCCIÓN

La programación cada día es más importante en las diferentes carreras universitarias. Por esto es necesario mejorar su enseñanza. Se presenta el estado de la programación competitiva en la Ciudad de La Paz y sus posibilidades de implementación como herramienta pedagógica.

2 PROGRAMACIÓN COMPETITIVA

Las competencias de programación tienen sus orígenes en las competencias de programación del año 1977 que organizó la universidad de Baylor (ICPC, 2016; Manzoor, 2008) en los Estados Unidos. Esta competencia, hoy en día, se ha convertido en una organización de nivel mundial que se desarrolla en todos los continentes y se realiza en varias etapas: primero las universidades seleccionan sus equipos; segundo, estos equipos participan en competencias nacionales. Los equipos clasificados participan en una competencia regional que clasifica a solo 100 equipos de tres personas a una competencia mundial. Los beneficios que aporta a los participantes son:

- Implementaciones más limpias y rápidas. Los concursos de programación hacen hincapié en la velocidad. Se vuelve importante para aprender a traducir los pensamientos en código rápidamente. Además, la escritura de código limpio ayuda a evitar el desperdicio de tiempo de depuración.
- Mejora de la capacidad de resolver problemas. Dado que los concursos de programación tratan sobre resolver problemas, participar en las competencias ayudan a desarrollar las habilidades de resolución de los mismos.
- Optimización de código: Generalmente se requiere optimizar una solución en el mayor grado posible. Esto permite que los participantes adquieran habilidades de optimización de código y a escribirlo eficientemente.

Desde su origen a la fecha se ha transformado de una forma de competir a ser una herramienta para el aprendizaje y en un factor importante para contratar personal experimentado en el campo de la Informática (RedGreenCode, 2016; Wikipedia, 2016). La programación competitiva es una forma de aprender y dominar la programación de computadoras. El desarrollar aplicaciones no requiere necesariamente ser un programador experto. Sin embargo, provee muchas habilidades y destrezas a los desarrolladores.

Existen varias definiciones de programación competitiva, en Wikipedia (2006) se define como un deporte de la mente que trata de programar, de acuerdo a las especificaciones dadas.

La definición dada por QUORA (2016) indica que la programación competitiva consiste en resolver problemas bien definidos escribiendo programas de computadora dentro de límites

especificados. Basado en lo anterior se presentan tres aspectos:

- Problemas bien definidos. Los problemas bien definidos contienen variables y deben hallar una respuesta en base a las posibles combinaciones de las mismas. Se especifican las restricciones de todas las variables, así como los supuestos necesarios.
- Programas de computadora. Se escriben programas que resuelven los problemas, que no tienen una interfaz gráfica, ya que solo se ingresan datos por la entrada estándar y los resultados se escriben en la salida estándar.
- Límites especificados. El resultado debe producirse dentro de límites de tiempo, memoria y lenguaje especificado.

La evolución hacia una herramienta de aprendizaje fue algo natural. Se han desarrollado diversas iniciativas. Duncan Smith en su página Red Green Code describe las ventajas de la programación competitiva para el aprendizaje presentando un método que ha denominado Learning Driven Development (ltd).

Una de las primeras experiencias de llevar los principios desarrollados al aula se realizó en la Universidad de Pekin desarrollando un sistema para la enseñanza de la computación que utiliza los principios de las competencias de programación. (Luo, et al, 2008).

Asociar el aprendizaje por juegos a la programación por juegos fue presentado por Burguillo (2010) para mostrar cómo motivar y estimular a los estudiantes para lograr mejor rendimiento académico.

La programación competitiva también proporciona una forma para que la universidad se relacione con la empresa y la sociedad (Nikov *et al.* 2014), proporcionando beneficios tanto para una como para las otras. Actualmente, es una forma que los empleadores utilizan para seleccionar talentos. Así tenemos diversas competiciones, mencionando a las más conocidas: Google Code Jam, Facebook Hackers Cup y, en Bolivia, el CodeRoad Coding Contest. Tener experiencia en este tipo de competiciones abre las puertas a mejores oportunidades una vez terminada su formación.

La evolución de los sistemas educativos hacia la educación virtual, ha aprovechado el desarrollo de los jueces virtuales para incorporarlos en los LMS (*Learning Management System*) (Llana, *et. al.* 2012). En este sentido la Universidad de Pekín ha desarrollado POP: Peking University Online Programming (Wang, 2014), donde han construido

una plataforma para el aprendizaje de la programación que se utiliza en cursos formales de la universidad y en actividades de auto aprendizaje.

Bierre *et al.* (2006) han desarrollado un sistema para la enseñanza de programación orientada a objetos basado en un juego con el que tiene que interactuar el programa de un estudiante compitiendo con otro. Cada uno es como un jugador en contra de otro.

Para el éxito de la programación competitiva es necesario verificar que los programas presentados no sean plagio (Hoon, 2008). Hoon ha desarrollado un modelo que presenta una métrica de similitud. Actualmente los sistemas de aprendizaje basados en programación competitiva tienen incorporados algún sistema para detectar el plagio en los programas desarrollados.

En la actualidad existen una gran cantidad de ejercicios publicados en diversos sitios que promueven la programación competitiva. Para que los mismos sean accesibles a los docentes y utilizables en clases es necesario establecer un sistema de clasificación. Esta puede ser por temática, por nivel de dificultad, tipo de algoritmo, tiempo requerido para resolver y otras medidas que se definan. Skupas (2009) desarrolla estas ideas que pueden ser utilizadas para asignar ejercicios a los participantes de un curso.

Una alternativa para desarrollar cursos basados en programación competitiva es elaborar una lista de problemas y asociarlo a un curso (Petit *et al.*, 2012) incorporando seguimiento y evaluación. Este método ha sido implementado en el sitio <http://jutge.org/>, que ya cuenta con cursos de algoritmos, aprender a programar, problemas matemáticos, Haskell, Python entre otros.

Los problemas deben estar bien definidos y considerar todas las posibilidades en los datos de prueba. Este no es un problema fácil y se han propuesto varios metodologías (Buzdalov, 2011) para generar los datos prueba.

La programación competitiva en la ciudad de La Paz, aún no ha sido analizada. En esta investigación tratamos de ver si la aplicación de esta metodología de enseñanza podría potencialmente aplicarse en las aulas.

3 METODOLOGÍA

Con la finalidad de realizar este estudio se han encuestado a profesores que dictan materias de programación en universidades públicas y privadas de la ciudad de La Paz. En total se encuestaron a 20

profesores de 8 universidades para establecer si la utilización pedagógica de la programación competitiva puede proporcionar ventajas en el aprendizaje y aprovechamiento de los estudiantes.

La encuesta realizada (ver anexo) contempla el lenguaje de programación, número de alumnos, aprobados, reprobados, asignación de ejercicios, tiempo de retroalimentación de la corrección de prácticas.

3.1 Resultados

En relación al lenguaje de programación utilizado se ve que los lenguajes de programación utilizados son el Java, c/c++ y Python (tabla 1).

Tabla 1. Lenguajes de programación utilizados en los primeros cursos de pregrado

| | Cursos | Opción 2 |
|-------------|--------|----------|
| Java | 14 | 70% |
| C/c++ | 5 | 25% |
| No responde | 1 | 5% |
| Total | 20 | 100% |

Dos docentes manifestaron que también utilizan Python en sus clases de c/c++.

Todos los lenguajes están disponibles en los jueces virtuales, con excepción del Python que no está disponible en todos los jueces; sin embargo, la mayoría de los jueces importantes lo incorporan.

Con relación al porcentaje de aprobación (tabla 2) se ve que en promedio solo aprueban el 55,2% de los alumnos. Para entender mejor este número se ha trabajado esta tasa en función de la cantidad de alumnos que participan en un curso, y el resultado nos muestra que a menor número de alumnos hay un mayor número de aprobados. En los cursos con menos de 20 alumnos aprueban hasta el 72% de los participantes.

Tabla 2. Porcentaje de aprobación en función al número de alumnos por cursos

| Número de alumnos | Alumnos | Aprobados | Porcentaje de aprobación |
|-------------------|---------|-----------|--------------------------|
| Hasta 20 | 111 | 80 | 72,1 % |
| De 21 a 40 | 265 | 142 | 53,6% |
| 41 y más | 198 | 95 | 48,0% |
| Total | 574 | 317 | 55,2% |

La programación competitiva provee una retroalimentación sobre las soluciones instantáneas, por esto es necesario asociar el porcentaje de aprobación según el tiempo de retroalimentación de prácticas presentadas. En este punto se hizo una división del tiempo de retroalimentación que tienen

los estudiantes para conocer si sus soluciones son correctas. Esto es el tiempo en que reciben sus tareas revisadas. Tomando en cuenta las respuestas dadas por los docentes, se vio por conveniencia dividir en dos grupos, los que demoran 5 o más días y los que demoran 4 o menos (tabla 3).

Tabla 3. Porcentaje de aprobación según el tiempo de retro- alimentación de prácticas presentadas

| Días | Alumnos | Aprobados | Porcentaje de aprobación |
|-----------|---------|-----------|--------------------------|
| 4 o menos | 253 | 174 | 68,4% |
| 5 o más | 321 | 143 | 44,5% |
| Total | 574 | 317 | 55,2% |

Los cursos con pocos alumnos (20 o menos estudiantes) generalmente revisan estas tareas en clases cuando han sido presentadas. Los otros cursos llevan estas prácticas para ser devueltas en una fecha posterior. El resultado nos muestra que al demorar menos en proporcionar una evaluación sobre los trabajos prácticos impacta en el aprovechamiento de los estudiantes.

Los jueces virtuales utilizan para probar que los programas enviados son correctos casos de prueba. Esto es un conjunto de datos que el estudiante no conoce y, al hacer correr el programa, verifica que los resultados generados son exactamente iguales que los que tiene almacenados. En los enunciados existen datos de ejemplo con los cuales el estudiante puede hacer pruebas y comprobar sus respuestas. Los datos de prueba son elaborados cuidadosamente y permiten probar si los resultados se obtienen con las restricciones dadas, esto es utilización de memoria y tiempo de proceso. Si vio que el 80% de los docentes utiliza casos de prueba en sus prácticas (tabla 4).

Tabla 4. Utilizan casos de prueba en las prácticas

| | SI | NO | Total Docentes | % |
|----------|----|----|----------------|-----|
| Utilizan | 14 | 4 | 20 | 80% |

Cuando preguntamos si conocen jueces virtuales (tabla 5), 8 de 20 docentes es decir el 40% conocen de jueces virtuales. Estos docentes son los que han participado en competencias de programación ya sea como competidores cuando fueron estudiantes o como entrenadores de equipos en su calidad de docente.

Tabla 5. Conocen la existencia de jueces virtuales

| | SI | NO | Total Docentes | % |
|---------|----|----|----------------|-----|
| Conocen | 8 | 12 | 20 | 40% |

Finalmente se quiso averiguar si el hecho de conocer los docentes utilizan jueces virtuales en sus clases y solo 10% (tabla 6) hacen uso de esta herramienta.

Tabla 6. Utilizan jueces virtuales en sus clases

| | SI | NO | Total Docentes | % |
|----------|----|----|----------------|-----|
| Utilizan | 2 | 18 | 20 | 10% |

4 DISCUSIÓN

Claudia Ott (2016) muestra 7 beneficios que la buena retroalimentación proporciona:

1. Ayuda a clarificar que es un buen rendimiento
2. Facilita el desarrollo del proceso de reflexión, auto aprendizaje
3. Proporciona información de buena calidad a los estudiantes sobre su aprendizaje.
4. Alienta a los profesores a una discusión con sus colegas sobre el aprendizaje.
5. Mejora la autoestima y motiva positivamente a los estudiantes.
6. Provee oportunidades de cerrar la brecha entre la eficiencia deseada y la actual.
7. Provee información a los profesores que puede ser utilizada para mejorar la enseñanza.

El proceso de programación tiene varias etapas que requieren una retroalimentación para el estudiante. Inicialmente entender el planteamiento del problema, establecer una estrategia de solución, desarrollar la solución y realizar muchas pruebas hasta obtener una respuesta correcta. El uso de jueces virtuales proporcionan todas estas características. Dan una retroalimentación inmediata y permite que el estudiante pueda realizar múltiples intentos hasta obtener una solución al problema planteado. Esto posibilita que pueda obtener una fluidez en el uso de la detección y corrección de errores.

La programación competitiva se basa en el uso de jueces virtuales para la evaluación de los ejercicios de programación presentados. Esto permite obtener los beneficios de la retroalimentación. En relación al estudiante se obtienen los beneficios 1, 2, 3, y 5.

A los docentes proporciona información permanente que ayuda a mejorar la enseñanza, conociendo las dificultades que van presentando los estudiantes en el desarrollo del curso. En la encuesta realizada se vio la importancia del tiempo de retroalimentación sobre los problemas y el aprovechamiento de los estudiantes.

Las respuestas que debemos dar para orientar las prácticas de retroalimentación efectiva en los cursos iniciales de programación (Ott, 2016) son:

- **Comunicación.** La retroalimentación se recibe en forma oportuna tras el reenvío de múltiples tareas. Se asegura que el estudiante recibe la retroalimentación. Se promueve el diálogo entre docentes y estudiantes.
- **Objetivos.** Los objetivos son claramente entendidos por los estudiantes.
- **Progreso actual.** La retroalimentación se proporciona de forma regular, con detalle y de acuerdo a los objetivos de los ejercicios. Se toman evaluaciones frecuentes para descubrir las dificultades de los estudiantes.
- **Actividades para mejorar.** Las estrategias proveen consejos de cómo mejorar para alcanzar los objetivos. Se da información que permita corregir y mejorar el aprendizaje en lugar de detallar fortalezas y debilidades.

La programación competitiva proporciona una solución adecuada a estas interrogantes. Primero con relación a la comunicación el estudiante recibe una retroalimentación instantánea, pudiendo a través del diálogo con el profesor resolver sus dificultades. Esto se da porque cuando el estudiante envía su solución a un ejercicio, que se traduce en un programa de computadora, el sistema proporciona una respuesta instantánea sobre la solución enviada. En este momento el estudiante sabe si el problema es correcto.

Segundo, al tener en cada sesión de práctica una lista de tareas que resolver quedan muy claros los objetivos que los estudiantes deben alcanzar. En la programación competitiva se presenta como una lista de problemas a resolver en un determinado tiempo.

Para medir el progreso el profesor tiene en cada instante un detalle de los problemas que están siendo solucionados y puede intervenir ayudando a los estudiantes en las dificultades encontradas. Los jueces virtuales proveen una lista que muestra primero los ejercicios resueltos y, segundo, las

dificultades y la cantidad de veces que un estudiante intenta resolver un ejercicio.

Finalmente, este detalle del progreso de los estudiantes sirve para retroalimentar al profesor para definir los ejercicios de las próximas sesiones de práctica reforzando el aprendizaje de los estudiantes.

En base a los resultados vemos que los primeros cursos de programación están adecuados para implementar con éxito la programación competitiva como una metodología para el aprendizaje. Podemos resumir los hallazgos en:

- El 80% de docentes preparan ejercicios con casos de prueba. Requisito fundamental para incorporar problemas a los jueces (tabla 4).
- Al reducir el tiempo de retroalimentación se mejora el aprovechamiento de los estudiantes (tabla 3).
- En cursos numerosos es posible dar retroalimentación rápida que mejorará el aprovechamiento.
- El 40% de los docentes conocen de jueces virtuales.
- Los jueces virtuales soportan todos los lenguajes de programación utilizados en los cursos de programación.

4.1 Consideraciones para crear un curso

Para establecer un curso basado en programación competitiva, debemos inicialmente enfocarnos en el nivel. Por ejemplo, un curso para el aprendizaje de la programación, algoritmos, etc. Existen diversos libros que se han escrito relacionando los problemas a jueces virtuales donde el estudiante puede probar los problemas propuestos (Halim, 2013; Skenia 2006). La mayoría de estos textos están son pensados para cursos preparatorios para competencias de programación o cursos de algoritmos avanzados.

La construcción de un curso basado en programación competitiva requiere inicialmente escoger el *software* de juez virtual que uno va a utilizar, por ejemplo un *software* utilizado en competencias o uno ya existente. Personalizar uno propio tiene algunas ventajas a la hora de organizar cursos, por ejemplo personalizar la forma de evaluación o establecer una forma más simple para incorporar problemas. En los jueces virtuales existen muchos problemas bien probados que se pueden utilizar; sin embargo, a la hora de aportar problemas se debe pasar por diversos filtros que aseguran la calidad de los problemas y la compatibilidad con la plataforma.

Para definir el contenido de un curso es necesario seguir los siguientes pasos:

- Establecer detalladamente el contenido clase por clase.
- Desarrollar los problemas que hacen al contenido y los casos de prueba que permitan verificar los resultados esperados.
- Estructurar los problemas para que el contenido asimilado sea gradual.

En la ejecución de los cursos hay que tomar en cuenta algunos aspectos:

- Ayudar a los estudiantes que se queden estancadas con algún ejercicio.
- Retroalimentar la planificación y los ejercicios en base de los resultados experimentados, con la finalidad de hacer las siguientes clases más productivas.

Esta metodología se puede utilizar naturalmente en cursos de programación y algoritmos. Sin embargo, también puede ser un complemento para asignaturas tales como matemática discreta, geometría, teoría de números y otras donde los ejercicios pueden ser planteados para resolverse enviando un programa a un juez virtual y que este sea evaluado en base de casos de prueba.

Las dificultades más evidentes están en la creación de los datos de prueba, y aquí hace falta una puntualización de las dificultades que se encuentran al momento de probar el problema. Para ejemplificar esto supongamos que queremos hallar todos los divisores de un número, para esto podemos comenzar con una solución trivial dividiendo por 1, 2, 3 y así sucesivamente. Claro que si los números son grandes el tiempo de proceso será mayor. Una optimización del problema es no calcular los divisores dos veces. Cuando hallamos un divisor en la realidad tenemos dos, por ejemplo si A es divisor de B, al dividir B por A tenemos un segundo divisor. Esta estrategia permite reducir considerablemente el tiempo de proceso. Por lo tanto, para obligar a los estudiantes a escribir soluciones eficientes, es necesario escoger cuidadosamente los datos de prueba y el tiempo de proceso con la finalidad de evitar que cualquier solución poco eficiente sea aceptada.

Una segunda dificultad es hacer que los datos escogidos abarquen todos los posibles casos descritos en el enunciado; por ejemplo, el tamaño máximo de la instancia a procesar. Esto hará que en muchos casos debamos adicionalmente escribir un programa que genere los datos prueba.

Finalmente tenemos que escribir el programa que resuelve el enunciado y probarlo en un el juez virtual para verificar que se procesa sin problemas. Claro está que, después de poner esto en consideración de los estudiantes, pueden aparecer preguntas y soluciones por la cual se verá en la necesidad de mejorar el enunciado, casos de prueba y/o la solución.

5 CONCLUSIONES

En este artículo hemos revisado las posibilidades y la funcionalidad de programación competitiva. Se recomienda organizar un curso basado en este paradigma. Para lograr el éxito se recomienda elaborar una serie de ejercicios elaborados desde un vista pedagógico. Se recomienda elegir un curso e implementar la metodología.

6 REFERENCIAS

- Bierre, K., Ventura, P., Phelps, A., and Egert C., (2006). *Motivating OOP by blowing things up: an exercise in cooperation and competition in an introductory Java programming course*. Proceedings of the 37th SIGCSE technical symposium on Computer science education. ACM, New York, NY, USA, 354-358. DOI=<http://dx.doi.org/10.1145/1121341.1121452>
- Burguillo, J., (2010). Using Game Theory and Competition-based Learning to Stimulate Student Motivation and Performance. *Computers & Education*, 55(2).
- Buzdalov, M., (2011). Generation of tests for programming challenge tasks using evolution algorithms. *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*. Natalio Krasnogor (Ed.). ACM, New York, NY, USA, 763-766. DOI=<http://dx.doi.org/10.1145/2001858.2002086>
- Halim, S., Halim, F. (2013). *Competitive Programing 3*, Universidad de Singapur.
- ICPC Fact Sheet (2016), Sitio web: <https://icpc.baylor.edu/worldfinals/pdf/Factsheet.pdf>
- Jeong-Hoon J., Gyun, W., and Hwan-Gue, C., (2008). An experience of detecting plagiarized source codes in competitive programming contests. *SIGCSE Bull.* 40(3), 369-369. DOI=<http://dx.doi.org/10.1145/1597849.138441>
- Llana, L., Martin-Martin, E., and Pareja-Flores C., (2012). FLOP, a free laboratory of programming. *Proceedings of the 12th Koli Calling International Conference on Computing Education Research*. ACM, New York, NY, USA, 93-99. DOI=<http://dx.doi.org/10.1145/2401796.2401807>
- Luo, Y., Wang, X., and Zhang, Z (2008). Programming grid: a computer-aided education system for programming courses based on online judge. *Proceedings of the 1st ACM Summit on Computing Education in China First ACM Summit on Computing Education in China*. ACM, New York, NY, USA. DOI=10.1145/1517632.1517643 <http://doi.acm.org/10.1145/1517632.1517643>

- Manzoor, S. (2008). Common mistakes in online and real-time contests. *Crossroads* 14(4), 10-16. DOI=<http://dx.doi.org/10.1145/1375972.1375976>
- Nikov, A., Kulev, I., Cavkovski P., Misev, A., Trajkovik, V., and Trajanov, D., (2014). CodeFu: coding competition as a tool for industry university collaboration. *Proceedings of the 2014 international workshop on Long-term industrial collaboration on software engineering ACM*, New York, NY, USA, 63-68. DOI=<http://dx.doi.org/10.1145/2647648.2647657>
- Ott, C., Robins, A., and Shephard, K., (2016). Translating Principles of Effective Feedback for Students into the CS1 Context. *Transactions. Computer. Education.* 16(1) DOI=<http://dx.doi.org/10.1145/2737596>
- Petit, J., Giménez, O., and Roura, S., (2012). Judge.org: an educational programming judge. *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. ACM, New York, NY, USA, 445-450.
- Quora (2016). *What is competitive programming*. Sitio web: <https://www.quora.com/What-is-competitive-programming-2>
- RedGreenCode (2016). *12-reasons-to-study-competitive-programming*, Sitio web: <http://www.redgreencode.com/12-reasons-to-study-competitive-programming/>
- Skenia, S., Revilla Ramos, M., (2006). *Concursos internacionales de informática y programación. Manual de entrenamiento por Internet*. Springer, New York, USA.
- Skupas, B., Dagiene, V., and Revilla, V., (2009). Developing classification criteria for programming tasks. *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education*. ACM, New York, NY, USA, 373-373. DOI=<http://dx.doi.org/10.1145/1562877.1563015>
- Smith, D., Red Green Code, Sitio web <http://www.redgreencode.com/>
- Wikipedia. (2016). *Competitive programming*, Sitio web: https://en.wikipedia.org/wiki/Competitive_programming
- Wang, Q., Li, W., and Xie, T., (2014). Educational programming systems for learning at scale. *Proceedings of the first ACM conference on Learning @ scale conference*. ACM, New York, NY, USA, 177-178. DOI=<http://dx.doi.org/10.1145/2556325.2567868>
5. Lenguaje de programación utilizado en su materia:
6. Horas de teoría por todo el curso (semestre/año/módulo):
7. Horas de práctica en laboratorio por todo el curso
8. Cuántos alumnos cursan clases en un semestre típico
9. De este número cuántos alumnos:
Aprueban Reprueban Abandonan
10. ¿Cuál es el texto que considera más importante para su clase (título, autor,...)?
11. Describa cómo se realizan las sesiones prácticas (laboratorio)
12. Describa cómo se realiza la evaluación de los ejercicios
13. Para los ejercicios que prepara para el curso ¿elabora casos de prueba para verificar que las soluciones sean correctas? Sí/No
14. ¿Cuánto tiempo pasa hasta que los estudiantes obtengan una retroalimentación sobre su tarea días?
15. ¿Conoce herramientas de validación automática, tales como pruebas unitarias o jueces virtuales? ¿Cuáles?
16. ¿Ha utilizado jueces virtuales en sus clases? Sí/No
17. Para participar en un taller de programación competitiva como herramienta pedagógica qué día y hora de la semana le viene bien.

Día Hora

Anexo: encuesta sobre la enseñanza de la programación en los cursos iniciales

OBJETIVO. El objetivo de la encuesta es realizar un diagnóstico de cómo se desarrollan los cursos iniciales de programación, principalmente como se llevan las prácticas.

RESPONSABLE: M.Sc. Jorge Teran Pomier

1. Universidad:
2. Nombre de la Materia de Programación:
3. Nombre del Docente
4. Email: