

Enseñanza Basada en Proyectos y Herramientas de Código Abierto

Msc. Marco A. Alvarez, Marco.Alvarez@usu.edu
Computer Science Department
Utah State University

1. Introducción

La educación superior en computación presenta intrínsecamente diversos desafíos, afectados no solamente por la naturaleza dinámica y compleja de éste campo, sino también por las barreras que estudiantes universitarios pueden enfrentar en su proceso de aprendizaje. A pesar de ser un campo bastante atractivo hoy en día, la dificultad en aprender a programar aliada a la necesidad de dominar fundamentos matemáticos muchas veces conducen a estados de desmotivación y deserción en los estudiantes a lo largo de su vida universitaria. En este sentido, relato en el presente artículo experiencias vividas tanto en la Universidad Católica Dom Bosco (UCDB) situada en Campo Grande , Brasil, donde era profesor y director de la carrera de Ingeniería de Computación, así como en Utah State University (USU), situada en Logan, Utah, Estados Unidos donde actualmente estoy por concluir el doctorado en ciencias de la computación. Es necesario que apunte inicialmente, que experiencias de cursos basados con estrategias de enseñanza basadas en proyectos han contribuido a mantener a los alumnos motivados en la universidad así como les ha servido a seguir caminos exitosos luego de concluir el bachillerato. He podido observar que éste panorama positivo se presenta aún cuando la realidad de cada departamento es distinta, mientras en el primero no se ofrece postgrado, en el segundo, se ofrecen maestría y doctorado habiendo consecuentemente mayor infraestructura y know-how para actividades de investigación.

Para efectos ilustrativos tomare como ejemplo-base el curso de Visión Computacional, que actualmente representa una importante y creciente área de investigación que viene alimentando la productividad en la industria en general así como participando en mejorar la calidad de vida de la gente. De hecho, tecnologías de visión computacional ocupan también espacio en diversas aplicaciones web, y dan soporte a muchas otras disciplinas académicas tales como Biología, Física, Robótica, Geoprocesamiento, etc. Una tendencia general en las universidades es ofrecer cursos de visión computacional en los últimos años del pregrado de carreras de ciencia de la computación, e ingenierías afines, así como también en programas de maestría y doctorado con líneas de investigación apropiadas.

La enseñanza basada en proyectos es una excelente alternativa para que estudiantes puedan aprender tecnología de punta y así facilitar su transición de la universidad hacia una industria competitiva o su involucramiento futuro en programas de maestría o doctorado. En este artículo presento evidencia que da soporte a ello, bajo dos escenarios

distintos. Al mismo tiempo, describiré como es que la utilización de herramientas de código abierto como recursos pedagógicos puede contribuir directamente a motivar a los estudiantes a trabajar y resolver problemas del mundo real al mismo tiempo que retornar su contribución a la comunidad del software libre.

Los cursos que son basados en proyectos poseen ciertas ventajas bien definidas, tales como:

- Motivar a los estudiantes a pensar creativamente y de manera independiente.
- Ayudar a los estudiantes a tener experiencia práctica en aplicaciones del mundo real a través de la solución de problemas reales utilizando tecnologías libres;
- Enseñar a los estudiantes a asimilar lo último de la investigación científica, pensar en su propia solución, presentar resultados experimentales, y preparar publicaciones;
- Ayudar a los estudiantes a establecer auto-confianza así como pasar por experiencias reales de trabajo en grupo;
- Motivar a los estudiantes a seguir estudios de post-grado, o insertarse en la industria del sector.

Adicionalmente, el uso de software libre, es decir herramientas de código abierto, en actividades pedagógicas ameniza la curva de aprendizaje y facilita el desarrollo, experimentación y mantenimiento de los proyectos a cargo de los estudiantes, los cuales son inducidos a diseñar y desarrollar sus soluciones bajo las premisas del código abierto haciendo que compartan con la comunidad el código del software derivado de sus proyectos. Inclusive, en algunos casos es posible aprovechar software desarrollado por determinados grupos de estudiantes como herramientas didácticas, una vez que los componentes que desarrollan están asociados con proyectos reales y son excelentes ejemplos para ilustrar lo alcanzado por el trabajo colaborativo. Dos ejemplos desarrollados bajo estos lineamientos son un visualizador gráfico para filtros de partículas y un plugin para experimentar con transformadas Hough. El primero incluye ejemplos e instrucciones para usarlo en un sistema de rastreamiento de ratones en experimentos de campo abierto. El segundo está disponible de manera gratuita en <http://rsb.info.nih.gov/ij/plugins/hough-circles.html>.

2. Organización del Curso

Un curso de visión computacional puede ser ofrecido en un ciclo académico, generalmente de 18 semanas incluyendo una introducción a los fundamentos, teorías y algoritmos pertinentes. Es necesario indicar que durante el dictado del curso el profesor debe destacar la importancia de conceptos matemáticos provenientes de cursos básicos tales como Álgebra, Cálculo, Estadística, Matemática discreta, así como también se debe mantener un especial énfasis en las potenciales aplicaciones de los algoritmos de visión computacional.

Para poder acompañar los rápidos cambios tecnológicos subyacentes al área de computación, se pueden incorporar diversos componentes prácticos durante las clases. Esta integración ayuda a los estudiantes a comprender intuitivamente los conceptos fundamentales. Si es que el curso se encuentra en un departamento o grupo de investigación que trabaja con proyectos relacionados al tópico que es enseñado, o mejor aún en los cuales el profesor participa, sería muy recomendable tocar durante las clases detalles relacionados a dichos proyectos, lo cual contribuiría a un mejor entendimiento de las actividades de investigación realizadas en el entorno de los estudiantes, así como expone a los mismos a participar en dichos proyectos.

De manera general un curso de visión computacional podría cubrir los siguientes tópicos: (1) Introducción a la Visión Computacional; (2) Formación de imágenes y modelos de representación; (3) Filtros; (4) Operaciones Morfológicas; (5) Extracción y selección de características; (6) Segmentación; (7) Rastreamiento; (8) Tópicos complementares como visión estereoscópica, realidad aumentada, aprendizaje y reconocimiento de patrones, y métodos estadísticos para segmentación y reconocimiento de objetos.

Para cada tópico es necesario revisar conceptos matemáticos, de ciencia de la computación e investigación operativa. Por ejemplo, espacios vectoriales y transformaciones lineares son explorados en conjunto con espacios de colores y vectores de características. Autovalores y autovectores son necesarios en diversos algoritmos de selección de características. Cálculo diferencial, en particular gradientes, son cubiertos cuando la detección de bordes y algoritmos de flujo óptimo son presentados. Algunos conceptos de autómatos, lenguajes formales y teoría de grafos serán necesarios en algoritmos de reconocimiento sintáctico de patrones.

3. Herramientas de software libre

Herramientas de software libre deben ser utilizadas durante el curso para dar soporte a actividades didáctico-pedagógicas. El proceso de enseñanza se puede beneficiar de diversas maneras, entre las cuales destacan. (1) Utilización de ambientes, plataformas, APIs, o aplicaciones de propósito general que pueden servir como base para el desarrollo de proyectos de investigación puesto que los estudiante se concentran más en la solución del problema que en la implementación de algoritmos y técnicas ya establecidas en la literatura; (2) Utilización de applets o animaciones en la Web que permitan al estudiante visualizar el comportamiento de los algoritmos que estudia en el curso, así como percibir intuitivamente los efectos de la configuración de diferentes valores en sus parámetros, y (3) Utilización de lenguajes, compiladores, y librerías específicas que permitan una rápida implementación al mismo tiempo que el estudiante puede disponer de su código fuente para compartirlo con la comunidad, esto se hace de acuerdo a las prácticas comunes al software libre.

En el caso de visión computacional, destacan ambientes y plataformas muy interesantes como:

Nombre	Descripción	URL
The R Project	Ambiente para estadística	http://www.r-project.org/
Scilab	Plataforma de computación numérica	http://www.scilab.org/
ImageJ	Procesamiento de imágenes en java	http://rebweb.nih.gov/ij/
Weka	Minería de datos en Java	http://www.cs.waikato.ac.nz/ml/weka/

por otro lado una lista de paquetes y applets puede ser encontrada en la pagina del curso de visión computacional de la UCDB: <http://www.gpec.ucdb.br/pistori/disciplinas/visao>.

Es importante destacar que los estudiantes deben tomar clases en laboratorios con acceso a internet y al software apropiado para el curso. De esta manera tienen la posibilidad de explorar conceptos en la práctica. Con este propósito, una buena recomendación es brindar algunos minutos de la clase para que los estudiantes puedan buscar en la web información relacionada al tópico cubierto en el momento, así los alumnos pueden entender el concepto en perspectivas distintas durante las clases y solicitar la ayuda del profesor para responder a preguntas más ingeniosas ganando así mayor profundidad en su proceso de aprendizaje. Inclusive, los estudiantes pueden ser retribuidos haciendo que los recursos y enlaces más útiles encontrados por ellos puedan ser adicionados a la página del curso.

4. Enseñanza basada en proyectos

Al adoptar una metodología de enseñanza basada en proyectos, el punto crítico está en que los estudiantes concluyan un proyecto al finalizar el curso, donde en el transcurso del ciclo el profesor esté altamente comprometido a proporcionar rápido auxilio en cada fase. Los proyectos pueden ser distribuidos en grupos de 2 a 3 estudiantes (UCDB) promoviendo el trabajo en grupo aunque también es posible que los alumnos desarrollen sus proyectos individualmente, aumentando así su capacidad de investigar de manera independiente (USU). La decisión queda de acuerdo al criterio del profesor y en sintonía con las otras actividades de los estudiantes en la carrera e institución donde están vinculados.

Es una excelente práctica, fundamental en mi opinión, no dejar a los alumnos libres de asesoramiento y evaluación a lo largo del curso. Una alternativa con ese propósito es la distribución de la carga del proyecto en la presentación de reportes intermedios durante todo el semestre. Por ejemplo, en la UCDB a lo largo de 18 semanas los estudiantes deben presentar:

1. [cuarta semana] Artículo preliminar: estudiantes forman el grupo basado en sus intereses y de acuerdo a las sugerencias del profesor en la primera semana. Cada grupo comienza a leer papers relacionados al proyecto escogido, y el profesor ayuda a cada grupo a clarificar dudas y a explicar detalles técnicos de ser necesario. Cada grupo comienza a escribir un paper preliminar, que contiene una revisión extensiva de la literatura, una introducción que contenga motivación, objetivos, y metodología a ser empleada durante el proyecto. El grupo debe entregar 3 a 6 páginas utilizando un formato LATEX al finalizar la cuarta semana;
2. [Décima semana] Artículo intermediario: cada grupo discute con el profesor las técnicas propuestas en el artículo preliminar. Los estudiantes son también orientados sobre cómo escribir artículos científicos. De esta manera, en este periodo los estudiantes distribuyen su carga de trabajo entre el desarrollo del proyecto en sí y la escrita de un artículo. Al finalizar la décima semana, los estudiantes deben entregar una copia del artículo conteniendo el progreso obtenido hasta el momento;
3. [Décimo-octava semana] Artículo final: Cada grupo continúa trabajando en sus proyectos bajo la orientación del profesor. Algunas modificaciones desde la propuesta del artículo intermediario pueden ser permitidas luego de discusiones y la debida argumentación con el profesor. Al final de la décimo-octava semana los

estudiantes deben entregar todos los programas, datos y recursos utilizados en el proyecto, y la versión final del artículo.

Al final, de acuerdo al criterio del profesor, los estudiantes podrían presentar sus proyectos en la realización de un workshop interno, contribuyendo así a que también desarrollen sus capacidades de presentar y sustentar sus contribuciones hacia la comunidad académica.

5. Consideraciones finales

En diferentes cursos la metodología seguida por los profesores para acompañar efectivamente el desarrollo de los proyectos así como para evaluar el progreso de los estudiantes puede variar medianamente. No obstante, aún así es posible obtener buenos resultados. Por ejemplo, en el curso de visión computacional ofrecido en la USU se sigue también una metodología basada en proyectos pero con diferencias tanto en el acompañamiento como en la evaluación cuando comparado con el curso ofrecido en la UCDB. A pesar de ello, ambas instituciones han sido particularmente exitosas al emplear una metodología de enseñanza basada en proyectos, puesto que, en ambas instituciones los alumnos no solamente concluyeron sus proyectos, si no, publicaron sus papers en conferencias de destaque.

De la misma manera, la mayoría de los estudiantes optaron por: seguir su vida académica dando continuidad a sus proyectos de investigación en sus respectivas universidades; ingresar rápidamente en programas de maestría y doctorado; o buscar un espacio en la competitiva industria del sector. Los detalles de estos resultados estarán disponibles en un paper que aún está en proceso de publicación. De cualquier manera, estas historias de éxito evidencian que combinando el desarrollo de proyectos de investigación con actividades de enseñanza tienen un impacto positivo en la formación y el futuro de los estudiantes.

Por otro lado, el impacto de la utilización de herramientas libres no solamente ha repercutido de manera directa en el desempeño de los estudiantes, sino que también ayuda a cerrar la brecha que separa departamentos y universidades localizadas en diferentes realidades, tales como las de países desarrollados y con tradición en investigación, y países en proceso de desarrollo.

6 Agradecimientos

El presente artículo se basa en un trabajo desarrollado por el autor y los profesores Hemerson Pistori y Mauro Conti Pereira vinculados a la UCDB y la profesora Xiaojun Qi vinculada a la USU, los cuáles suministraron datos de los cursos que tienen a su cargo, así como participaron arduamente en la discusión de las ideas aquí expuestas.