

# Design Considerations for Introducing PBL in Computer Engineering

Mayela Coto  
Escuela de Informática  
Universidad Nacional  
Heredia, Costa Rica  
mcoto@una.ac.cr

Sonia Mora  
Escuela de Informática  
Universidad Nacional  
Heredia, Costa Rica  
smora@una.ac.cr

Marianne Lykke  
Escuela de Informática  
Universidad de Aalborg  
Aalborg, Dinamarca  
mlykke@hum.aau.dk

*Abstract*— Nowadays, universities are driven to adopt pedagogical approaches that enable them to better meet the needs of the new generation of students and industry. This article presents an ongoing experience in innovating the computer engineering curriculum through a problem-based and project-oriented learning approach to develop skills such as problem solving, effective communication, teamwork and lifelong learning, which are increasingly demanded by employers and society. The investigation determined an initial set of design principles that can be used to guide an implementation strategy of the approach. The methodology allows continuous evaluation and refinement of the design. The findings so far point to a number of obstacles, from student, teachers and management, which must be overcome to achieve an effective result.

*Keywords:* curriculum innovation, pedagogical approaches, computer engineering, PBL.

## I. INTRODUCCIÓN

Estudios recientes [1, 2, 3] han indicado que los profesionales actuales requieren, además de un sólido conocimiento de su disciplina, del desarrollo de otras habilidades. Las ciencias de la computación, la informática, la ingeniería de sistemas y las áreas afines<sup>1</sup> no son la excepción. La ACM [1] identifica entre las características que deben tener los futuros graduados de esta disciplina: habilidades para la solución de problemas, para comunicarse efectivamente, para el trabajo efectivo en grupos, la responsabilidad profesional y la capacidad de aprendizaje a lo largo de la vida. En la misma línea, Mills y Treagust [4] señalan que el profesional en ingeniería informática debe ser capaz de hacer frente a continuos cambios tecnológicos, trabajar en situaciones inciertas, responder a necesidades diversas y algunas veces contradictorias, y comprender los aspectos sociales, legales, éticos y culturales involucrados en el ejercicio de su práctica profesional.

Este panorama obliga a las universidades a demostrar que sus programas de estudio forman graduados que, además

<sup>1</sup> A las cuales por simplicidad llamaremos en adelante ingeniería informática

de tener un sólido conocimiento disciplinario, cuentan con habilidades y destrezas como la comunicación, la capacidad de analizar el entorno social y económico, la resolución de problemas reales, la innovación, el trabajo colaborativo y la investigación. Sin embargo, los programas de estudio generalmente se estructuran alrededor del contenido técnico y prestan poco énfasis al desarrollo de dichas competencias.

A pesar de los múltiples retos que hoy enfrentan las universidades, la enseñanza en la disciplina informática ha sido tradicionalmente deductiva [5], es decir, el docente presenta un tema por medio de una charla donde expone los principios generales, muestra ejemplos ilustrativos, ofrece a los estudiantes prácticas y trabajos sobre aplicaciones similares y, finalmente, pone a prueba la capacidad del alumnado de repetir el mismo tipo de razonamiento mediante trabajos o exámenes. Sin embargo, un principio bien establecido de la psicología educativa es que la gente está más motivada para aprender aquellas cosas para las que percibe su utilidad con claridad. Para estimular a un estudiante no es suficiente prometerle que algún día va a necesitar esos conocimientos o habilidades. Una alternativa preferible es la enseñanza y el aprendizaje inductivo [5]. En lugar de iniciar con los principios generales y, finalmente, llegar a las aplicaciones, la enseñanza comienza con los detalles, preguntas para ser contestadas, un estudio de caso para analizar o un problema complejo del mundo real que es necesario resolver. A medida que los estudiantes tratan de analizar los datos o resolver el problema, generan una necesidad por conocer principios, hechos, reglas y procedimientos. Es en este momento cuando se les presenta la información necesaria o se les motiva a descubrirla por sí mismos.

Partiendo de la premisa que memorizar, almacenar y reproducir información no es la mejor estrategia para estudiar ingeniería informática; que la capacidad para resolver problemas es un elemento vital dentro de esta disciplina; y que la mayoría de las actividades del profesional en ingeniería informática tiene como marco el desarrollo de proyectos, se plantea la implementación de un enfoque pedagógico basado en problemas y orientado a proyectos en los cursos de

programación de la carrera de Ingeniería de Sistemas de la Universidad Nacional de Costa Rica (UNA), mediante una intervención educativa guiada por las siguientes preguntas de investigación:

1. ¿Cómo puede implementarse un enfoque pedagógico basado en problemas y orientado a proyectos en la disciplina de la ingeniería informática, con el propósito de desarrollar las habilidades requeridas actualmente por los graduados, en la resolución de problemas de la vida real?
2. ¿Cómo puede el uso de este enfoque pedagógico contribuir a reducir las tasas de deserción en los cursos de programación?
3. ¿Qué principios de diseño pueden ser utilizados para guiar una estrategia de implementación de un enfoque pedagógico basado en problemas y orientado a proyectos en cursos de programación?

En la primera parte de este artículo se presentan los conceptos de aprendizaje basado en problemas y aprendizaje basado en proyectos. En la segunda sección se exponen algunos resultados obtenidos en la introducción y el desarrollo de dichos enfoques en ambientes universitarios. La tercera parte introduce una experiencia de un enfoque híbrido entre el aprendizaje basado en problemas y en proyectos. En la cuarta sección se explica la metodología seleccionada y en la quinta y sexta partes se presentan respectivamente los resultados obtenidos del análisis del contexto y un diseño inicial de solución. Finalmente se realiza una reflexión final sobre el trabajo realizado.

## II. LOS ENFOQUES INDUCTIVOS DE ENSEÑANZA

De acuerdo con Prince y Felder [6, 5], entre los métodos inductivos de enseñanza y aprendizaje se encuentran el aprendizaje basado en investigación, el aprendizaje basado en casos, el aprendizaje basado en problemas y el aprendizaje basado en proyectos. Todos ellos comparten las características de estar centrados en el estudiante, ser métodos constructivistas, involucrar un aprendizaje activo y fomentar el aprendizaje colaborativo.

El aprendizaje basado en problemas se promueve cuando los estudiantes son enfrentados a un problema auténtico como punto de partida para la adquisición e integración de nuevos conocimientos. Prieto [7] señala que este enfoque puede mejorar la calidad del aprendizaje en las universidades de formas diversas, pues fomenta competencias en la resolución de problemas, la toma de decisiones, el trabajo en equipo y las habilidades comunicativas. Además favorece el desarrollo de la capacidad de búsqueda, el manejo de información y de investigación, pues el estudiante se ve motivado a comprender e investigar acerca de un problema para así llegar a proponer una solución adecuada.

Por su parte, el enfoque basado en proyectos se fundamenta en la idea de que los estudiantes desarrollen proyectos con aplicación en el mundo real como punto de

partida de su proceso de aprendizaje. Esto implica dejar de lado la enseñanza tradicional y apearse a un enfoque en el cual las actividades interdisciplinarias se promueven y estimulan el aprendizaje colaborativo y centrado en el estudiante. El uso de proyectos como parte del proceso de enseñanza-aprendizaje en un programa de estudios es bien conocido en el área de la ingeniería informática y usualmente estos se enfocan en la aplicación de conocimiento y, posiblemente, en la integración de conocimientos previamente adquiridos [4].

Ambos enfoques tienen muchas características en común: los dos son centrados en el estudiante y requieren una actitud y participación activa de su parte, son multidisciplinarios y se basan en la colaboración, logran una mayor motivación del estudiante y una mayor responsabilidad de estos en el proceso de aprendizaje. Por otro lado, Perrenet, Bouhuijs y Smits [8] identifican algunas diferencias entre los dos enfoques: el trabajo en proyectos es más cercano a la realidad profesional de los ingenieros; usualmente los proyectos requieren de un período mayor de tiempo, a diferencia del enfoque basado en problemas que puede tener más corta duración. El aprendizaje basado en proyectos está más orientado a la aplicación de los conocimientos, mientras que el enfoque basado en problemas está más dirigido a su adquisición. El uso del aprendizaje basado en proyectos suele estar complementado con cursos temáticos (por ejemplo matemáticas y física en el área de ingeniería), además la gestión del tiempo y de recursos por parte de los estudiantes, y el manejo de roles juegan un papel muy importante, mientras que en el aprendizaje basado en problemas no necesariamente

## III. INVESTIGACIÓN PREVIA SOBRE EL USO DE PBL

La efectividad del enfoque PBL (problem-based learning) ha sido analizada en diferentes estudios en el contexto universitario. Dochy, Segres, Van den Bossche y Gijbels [9] revisaron 43 estudios empíricos de los efectos del PBL en la adquisición de conocimiento y en el desarrollo de habilidades para resolver problemas. Con respecto al primer elemento, los resultados mostraron que el estudiantado adquiere más conocimiento en el corto tiempo, si se utiliza un enfoque de enseñanza tradicional; sin embargo, los estudiantes que experimentaron el PBL fueron capaces de retener el conocimiento adquirido por un período mayor. Con respecto a la adquisición de habilidades, el efecto del enfoque PBL fue positivo en forma contundente.

En la misma perspectiva, Prince [10] concluye que aquellos docentes que adopten el PBL es posible que no perciban una mejora significativa en el rendimiento académico de los estudiantes, pero sí un efecto positivo en la actitud hacia el aprendizaje, hábitos de estudio, la retención del conocimiento y la capacidad para aplicarlo, así como el desarrollo del pensamiento crítico y habilidades para resolver

problemas, particularmente si la experiencia se acompaña de formación explícita sobre estos procesos.

Nuutila, Törmä y Malmi [11], en un estudio realizado sobre la aplicación de PBL en cursos iniciales de programación, observaron un descenso significativo en la tasa de abandono. Los autores indican que, además de programación, los estudiantes adquirieron habilidades relacionadas con el trabajo colaborativo, el estudio independiente y la comunicación. El estudio les dio seguimiento a los estudiantes en cursos avanzados de programación, y los datos demuestran que las puntuaciones medias de los alumnos procedentes del curso de PBL han sido ligeramente mejores que las de los otros estudiantes. Similarmente, Hamalainen [12] reporta los resultados obtenidos en un curso de introducción a los conceptos teóricos de ciencias de la computación y concluye que los estudiantes generan un mayor compromiso con un curso PBL que con uno de corte tradicional.

De esta manera, la investigación en enfoques pedagógicos identifica al PBL como un aliado para resolver parcialmente algunos de los problemas que enfrenta la formación de estudiantes en ingeniería informática; sin embargo, para Mills y Treagust [4] otros enfoques inductivos como el aprendizaje basado en proyectos pueden resultar más apropiados, porque reflejan mejor el comportamiento profesional de un ingeniero informático. Kolmos [13] menciona que este último enfoque promueve condiciones para el aprendizaje a partir de los procesos de discusión y facilita el desarrollo de las competencias de dirección de proyectos y de colaboración, indispensables para los ingenieros informáticos.

Hay que considerar que la enseñanza basada en proyectos presupone una estrategia educativa integral y no únicamente un complemento en el proceso de aprendizaje. Cuando un plan de estudios está organizado por proyectos, la mayoría de los cursos están ligados con la realización de un determinado proyecto. Existen pocos planes de estudios organizados por proyectos en su totalidad. El ejemplo más cercano a un enfoque puro es la Universidad de Aalborg en Dinamarca, donde los proyectos y los cursos relacionados a estos conforman un 75% del plan de estudios [4]. Otras universidades que utilizan el enfoque de organización por proyectos como el foco principal de sus programas de estudios de ingeniería son Roskilde en Dinamarca; Bremen, TU Berlin, Dortmund y Oldenburg en Alemania; Delft and Wageningen en Holanda; Monash University y Central Queensland University en Australia, y Olin College en los Estados Unidos [5].

Para capitalizar las ventajas de ambos enfoques, algunas veces se utilizan conjuntamente, es decir, el punto de partida del proceso de aprendizaje es un problema, pero la solución al mismo se presenta mediante la realización de un proyecto. Tal es el enfoque de la Universidad de Aalborg en Dinamarca, que se introduce en la siguiente sección.

#### IV. EXPERIENCIAS CON UN ENFOQUE HÍBRIDO: APRENDIZAJE BASADO EN PROBLEMAS ORGANIZADO POR PROYECTOS

La Universidad de Aalborg, en Dinamarca, desde su creación en 1974 se interesó en un enfoque en el cual los estudiantes tuvieran un rol más activo en la construcción del conocimiento, que los hiciera sentirse más comprometidos con su propio aprendizaje y que garantizara altos estándares de calidad. Desde esta perspectiva, el problema, el proyecto y el trabajo en grupo de estudiantes y supervisores son la manera de organizar el aprendizaje, la enseñanza y la investigación en esta universidad. Esta forma se conoce como el Modelo PBL de Aalborg [14] o Project-Oriented Problem Pedagogy (POPP), y sus principios base son: *orientado a la resolución de un problema*, como base del proceso de inicio del aprendizaje; *organizado por proyectos*, como la forma de abordar el problema; *integración de la teoría y la práctica*; el estudiante tiene *un nivel de decisión* en la selección del problema y en cómo organizar el proyecto, *trabajo colaborativo* en grupos de tres o más estudiantes, y la *realimentación* de compañeros y profesor [14, 13].

Entendiendo que: (1) la capacidad para resolver problemas es un elemento vital dentro de la disciplina informática, y (2) una gran parte de las actividades laborales de estos profesionales tiene como marco el desarrollo de proyectos, se propone utilizar un enfoque híbrido – inspirado en el modelo de Aalborg – como punto de partida para realizar una innovación curricular en el plan de estudios de Ingeniería de Sistemas, específicamente en el área de programación.

#### V. METODOLOGÍA PARA IMPLEMENTAR EL POPP EN EL ÁREA DE PROGRAMACIÓN

El objetivo de la intervención educativa es generar cambios en la forma de abordar los cursos de programación del plan de estudios y, a partir de aquí, obtener evidencia empírica sobre la forma en que un área disciplinaria como la ingeniería informática puede moverse de un modelo basado primordialmente en la entrega de información, hacia una pedagogía que favorece el aprendizaje como la construcción del conocimiento mediante la colaboración, la formulación de problemas y la realización de proyectos.

Con este objetivo en mente se seleccionó la *investigación basada en diseño* para abordar el problema, la cual surge como un paradigma para el estudio del aprendizaje en el contexto a través del análisis y diseño sistemático de herramientas y estrategias educativas [16]. De acuerdo con Wang y Hannafin [17] el fin de la metodología es mejorar las prácticas educativas en contextos reales. El proceso, flexible pero sistemático, es guiado por teorías y principios de diseño sensitivos al contexto y se basa en un proceso participativo de colaboración entre investigadores y profesionales involucrados en el ejercicio de la práctica educativa.

La investigación basada en diseño puede verse como un modelo de cinco fases [18, 19] donde los participantes son parte de un proceso conjunto e iterativo de análisis, diseño, desarrollo e implementación. La primera de ellas, "análisis del problema" requiere que los investigadores y académicos realicen un análisis del contexto, en este caso el de los cursos de programación del plan de estudios. La segunda fase "diseño de soluciones con un marco teórico" implica la creación de prototipos de soluciones basadas en los principios de POPP, los objetivos del plan de estudios de Ingeniería de Sistemas, la perspectiva estudiantil y docente, y la revisión de la bibliografía. La tercera fase "Evaluación y puesta en marcha de la solución propuesta en la práctica" se refiere a un proceso iterativo de prueba y mejora de los principios de diseño y de la solución prototipo inicial. La cuarta fase "documentación y reflexión sobre los principios de diseño" se refiere al proceso de reflexión retrospectiva sobre el diseño y sus resultados. Este proceso de reflexión con los académicos permite refinar la propuesta inicial para ser aplicada en el próximo semestre, así como documentar las fortalezas y debilidades. La última fase "difusión y adopción en contextos más amplios" se refiere a la adopción y sostenibilidad del proceso de innovación realizado, así como a la difusión del mismo a otros contextos de la Universidad Nacional.

El proceso, aquí descrito, es guiado por principios teóricos y por el conocimiento de la "cultura" propia de la unidad académica, se basa en un proceso participativo de colaboración entre investigadores de la Universidad de Aalborg y de la Escuela de Informática, así como académicos y estudiantes de esta última. La intervención se realiza en el área de programación, en particular en la secuencia de cursos: Fundamentos de Informática, Programación I, Programación II, Programación III y Programación IV. Una de las ventajas de la investigación basada en diseño es que permite monitorear los cambios introducidos en los cursos, analizar las posibles discrepancias entre el diseño realizado y las respuestas producidas por los participantes y, a partir de esto, proponer cambios que logren un abordaje más eficiente del enfoque pedagógico por parte de los docentes, estudiantes y la gestión del plan de estudios.

A continuación se presentan algunos resultados obtenidos en las primeras dos fases: análisis del problema y diseño de soluciones iniciales.

## VI. ANÁLISIS DEL CONTEXTO

Se considera fundamental para el proyecto el diseñar soluciones prototipo con probabilidad de éxito. Esto es de vital importancia, ya que, como Guskey [20] sugiere, un cambio significativo en las actitudes y prácticas de los docentes se produce con mayor énfasis después de que han obtenido pruebas de la mejora del aprendizaje de sus estudiantes. En consecuencia, este proyecto sustenta el cambio como un

proceso fundamentalmente basado en experiencias de aprendizaje positivas para los docentes.

Para lograr este diseño de soluciones "exitosas" se hace necesario analizar el modelo pedagógico de la UNA, que sirve como marco de referencia a todo este proceso; estudiar las condiciones de la Escuela de Informática y el estado de preparación que tienen académicos, estudiantes y el currículo para enfrentar un proceso de cambio como el propuesto, así como conocer las limitaciones y dificultades que, desde su perspectiva, podrían enfrentar.

### A. El modelo pedagógico de la UNA

La Universidad Nacional cuenta desde el año 2008 con un modelo pedagógico que entiende la enseñanza y el aprendizaje como un proceso social, histórico y cultural que va más allá de la simple transmisión de conocimientos. La enseñanza y el aprendizaje se basan en el análisis y el cuestionamiento de la realidad; la investigación y el trabajo práctico sobre el contexto en el que un estudiante y su carrera se sitúan; el desarrollo de habilidades para la innovación y la resolución de problemas; la negociación de conflictos; grupos de trabajo interdisciplinarios, y la toma de decisiones basadas en información confiable y oportuna [21].

El espíritu de este nuevo modelo pedagógico es muy amplio y general, y no apunta a ninguna estrategia didáctica específica. Los principios generales y flexibles implican algunas dificultades cotidianas, especialmente para los profesores sin formación pedagógica, pues la interpretación e implementación del modelo descansa, en gran medida, en la comprensión que los académicos alcancen sobre sus principios, las propias experiencias en educación, sus concepciones de la enseñanza y el aprendizaje, sus conocimientos pedagógicos, sus habilidades para aplicar diferentes estrategias didácticas y la influencia de su propia disciplina.

El enfoque POPP de la Universidad de Aalborg puede ser entendido como una posible concreción del modelo pedagógico de la UNA. Ambos modelos tienen como objetivo un esquema de enseñanza-aprendizaje centrado en el alumno, donde la principal función de los docentes es la facilitación del aprendizaje de los estudiantes. En ambos, la enseñanza se entiende como un proceso complejo y multidireccional, mediante el cual se construye y se comparte el conocimiento.

### B. Perspectiva docente

Para conocer la perspectiva de los académicos, sobre la utilización del enfoque POPP, se realizó un taller de sensibilización con los académicos de la Escuela de Informática. Participaron ocho docentes de diferentes áreas disciplinarias. El objetivo de esta actividad fue identificar los retos que implica trabajar con POPP como enfoque

pedagógico. La actividad permitió establecer las barreras actuales que evitarían que el proyecto fuese exitoso, las metas a las cuales se desea llegar y las acciones que deben ejecutarse para disminuir la brecha entre las barreras y las metas. Entre los resultados obtenidos se encuentran:

- **Barreras:** Una estructura y administración curricular rígida; escasa posibilidad de trabajo colaborativo entre profesores; escasa motivación de estudiantes; escasa motivación laboral, y la no conceptualización de la pedagogía como un elemento importante del trabajo en el aula.
- **Metas:** Mejorar el clima organizacional; formar profesionales con capacidad de formular y resolver problemas; contar con un currículo más acorde a las necesidades de la sociedad; promover un ambiente de trabajo más orientado a metas comunes, al trabajo colaborativo y a la búsqueda de la excelencia, y mejorar el vínculo universidad – sociedad.
- **Acciones:** Diseñar una estrategia gradual para el cambio; mejorar el plan de estudios; y diseñar una actualización pedagógica continua e innovadora que permita el desarrollo de destrezas y habilidades docentes pertinentes.

Para efectos de diseño esto sugiere que, desde la perspectiva docente, la intervención que se haga en el aula debe considerar una estructura y una administración curricular poco flexible y un cuerpo docente que no confía plenamente en la pedagogía como vehículo para mejorar su desempeño docente y el aprendizaje de los estudiantes. Esto conduce a promover un cambio paulatino y con experiencias exitosas, que motive a los docentes a hacer pequeñas modificaciones y a compartir su conocimiento y experiencia, de tal manera que se mejoren los procesos de comunicación y aprendizaje entre ellos. Es importante además dar seguimiento a los estudiantes para valorar su aprendizaje y de esta manera contar con información que conduzca a una mejora en el plan de estudios.

### C. Perspectiva estudiantil

Para conocer la perspectiva estudiantil se elaboró un cuestionario, compuesto de preguntas abiertas y cerradas, con el objetivo de valorar la preparación de los mismos hacia un currículo organizado desde un enfoque POPP. Se tomaron en cuenta los principios básicos del enfoque y se consideraron cinco grupos del curso EIF206 Programación III, que se ubica en el cuarto nivel de carrera. La población total fue de 118 estudiantes y se seleccionó una muestra de 68 estudiantes a los cuales se les aplicó el instrumento. Los principales resultados obtenidos se presentan a continuación.

1) *Con respecto a la solución de problemas:* Un 76% de los estudiantes afirmó que en el plan de estudios se utilizan problemas como estrategia de aprendizaje en los diferentes cursos de la carrera (no exclusivamente cursos de programación), además un 79% indicó que les gusta resolver

proyectos con problemas o situaciones conocidas, y únicamente un 57% indicó que les gusta el reto de explorar situaciones poco comunes. Con respecto al grado en que consideraban que los problemas a resolver en los cursos eran auténticos problemas de la vida real, solo 55 estudiantes opinaron y únicamente 7 de ellos piensan que es de esta forma, 46 estudiantes manifiestan que son problemas de la vida real en un ambiente ficticio, y el resto, 2 estudiantes piensan que son totalmente hipotéticos. Para efectos de diseño, esto sugiere que se les debe proporcionar a los estudiantes experiencias que promuevan en ellos una actitud positiva hacia situaciones no conocidas y que demanden una actividad cognitiva más compleja. Se debe además reforzar la utilización de situaciones de la vida real más próximas a su futuro laboral y profesional.

2) *Con respecto al nivel de decisión deseado y ejercido por los estudiantes:* de 52 estudiantes un 98% indicó que les gustaría poder influir en los problemas que resuelven, sin embargo únicamente un 58% afirmaron que tienen algún nivel de decisión en los problemas que el docente les plantea, mientras que un 42% opina que no tiene ningún poder al respecto. Adicionalmente el 75% se siente capacitado para decidir qué problemas puede resolver. Estos datos sugieren que es conveniente incrementar los niveles de autonomía de los estudiantes, brindándoles algún nivel de decisión en el problema a resolver, el tipo de proyecto, la forma de presentar los resultados, etc.

3) *Con respecto al contacto con proyectos de la vida real y la preparación obtenida para realizarlos:* En los cursos de programación los estudiantes realizan de dos a tres proyectos en un ciclo lectivo. Al preguntarles a los estudiantes si se sienten en capacidad para realizar proyectos reales en una empresa o institución, un 35% de los estudiantes dice sentirse bien preparado porque tienen los conocimientos necesarios para hacerlos. El 65% de los estudiantes expresa sentirse incapaz de enfrentar esta labor debido a que no tienen los conocimientos necesarios, les falta experiencia laboral y el tipo de proyectos hipotéticos realizados en el aula no los han preparado para enfrentarse a uno real. Considerando que el resolver situaciones auténticas no solo es un factor que motiva a los estudiantes, sino que también los entrena para su futuro laboral, se deriva necesario enfrentarlos a proyectos que presenten problemáticas ejemplares o representativas que les permitan ejercitar lo que tendrán que realizar frecuentemente en su vida profesional.

4) *Con respecto al trabajo en grupo:* Un 72% de los estudiantes consideran que el trabajar en equipo es una adecuada estrategia de aprendizaje debido a que esto les permite aprender con otros, además, es una habilidad importante en su desarrollo profesional. Un 28% no lo considera adecuado debido a los conflictos que se presentan tales como desigualdad de tareas e incompatibilidad de

horarios y personalidades. Adicionalmente únicamente un 19% expresó que nunca o casi nunca se generan conflictos cuando trabajan en grupo. Un 58% de los estudiantes dice resolver estos conflictos de forma independiente, un 36% requiere ayuda del profesor y un 4% expresa no resolverlos nunca. En el desarrollo del trabajo grupal un 63% dice recibir poca o ninguna supervisión del docente. Los datos sugieren que es importante mantener el trabajo en equipo como una buena estrategia para compartir el conocimiento y aprender entre todos; sin embargo, se debe preparar al estudiante para enfrentarse adecuadamente a los conflictos que pueden surgir, y a los docentes, para brindar la supervisión necesaria y suficiente.

5) *Con respecto a la responsabilidad sobre el proceso de aprendizaje:* Un 28% de los estudiantes piensa que el proceso de aprendizaje es únicamente responsabilidad del profesor; un 31% opina que es responsabilidad del profesor, pero que ellos deben complementarlo, y un 41% plantea que es responsabilidad de ambos. Estos datos sugieren que si bien los estudiantes en la mayoría asumen su responsabilidad en el proceso de aprendizaje, se les deben brindar oportunidades donde ejerzan mayor control y activen el aprendizaje autónomo.

En general, los resultados anteriores apuntan a la pertinencia de ir moviendo los procesos de enseñanza y aprendizaje hacia un esquema de POPP, pues este les puede brindar a los estudiantes mayores posibilidades de decidir sobre el problema que quieren resolver, a integrar conocimientos y áreas diversas de la disciplina, a tener un mayor contacto con problemas de la vida real y a responsabilizarse por su propio aprendizaje. Sin embargo, algunos aspectos culturales deben irse cambiando paulatinamente, como el tipo de problemas a resolver, la responsabilidad del aprendizaje y la forma de realizar el trabajo en grupo. Tanto docentes como estudiantes deben ir aprendiendo a ejercer nuevos roles consecuentes con un enfoque didáctico diferente.

#### D. Consideraciones de los cursos de programación

Los cinco cursos del área de programación son parte fundamental del plan de estudios, debido a las destrezas que desarrollan en los estudiantes y al interés que generan en ellos. En particular, el curso Fundamentos de Informática es el primer contacto que los estudiantes tienen con el área disciplinaria. A su vez, los cursos de programación, conjuntamente con los de matemáticas, son los que tienen tasas más elevadas de reprobación y deserción.

Según datos de la Escuela de Informática, en el período comprendido entre el 2006 y el 2011, la tasa promedio de reprobación (contempla la deserción, pues hasta la fecha no existen registros individuales de esta condición) de los cursos de programación, puede observarse en la Tabla 1.

TABLA 1. NIVELES DE REPROBACIÓN DE LOS CURSOS DE PROGRAMACIÓN

Curso de programación	% reprobación
Fundamentos de Informática	51,33%
Programación I	35,98%
Programación II	30,24%
Programación III	20,36%
Programación IV	16,72%

Como puede verse de estos resultados, el porcentaje de reprobación es mayor entre más inicial sea el curso, y lo mismo sucede con la observación que han realizado los docentes sobre la deserción. Esto apunta a un problema de múltiples causas, una de ellas es que algunos de los estudiantes que ingresan a la carrera descubren en los primeros cursos que no es la disciplina que les gustaría estudiar; otra de las causas se refiere al cambio de razonamiento que deben realizar los estudiantes para enfrentar la resolución algorítmica de problemas. De cualquier manera, estos indicadores plantean un reto adicional para el proyecto, a partir de la premisa de que un enfoque pedagógico centrado en el estudiante pueda contribuir a mejorar estos índices, especialmente en los primeros cursos.

## VII. DISEÑO DE UNA SOLUCIÓN INICIAL

En la fase inicial del proyecto, se propone una intervención educativa que contemple dos aspectos importantes: (1) un proceso de formación docente y (2) la introducción paulatina de algunos de los principios del POPP en los cursos de programación, a través de la definición y solución de problemas.

### A. Formación docente

De acuerdo con lo establecido en la etapa anterior, se evidencia la necesidad de acompañar el proceso de innovación curricular de una estrategia de formación docente que les permita a los académicos (1) fortalecer las competencias en el diseño de ambientes de aprendizaje que promuevan, en los estudiantes, la construcción del conocimiento de una forma integral y cercana al mundo real; y (2) desarrollar las competencias necesarias para la gestión de un currículo enfocado en la solución de problemas y organizado por proyectos.

Este proceso de capacitación docente pretende consolidar la cátedra de programación, estableciendo gradualmente las bases para la formación de una comunidad de práctica [22]. En este sentido, el aprendizaje de los docentes se conceptualiza como un proceso continuo, resultado de un esfuerzo colaborativo en el que reciben apoyo de sus colegas, de expertos externos, de investigadores y de la administración [23, 24]. Se busca un proceso de reflexión colectiva, de continuo rediseño y experimentación que permita resolver retos de la práctica docente y que posibilite la positiva transferencia de lo aprendido al aula [25].

Con este fin se programan espacios donde los docentes “aprendan compartiendo” con actividades tanto formales como informales para que compartan inquietudes y resultados acerca de la implementación del POPP en sus cursos (formulación de los problemas, diseño de las actividades de aprendizaje, evaluación, seguimiento, etc.). Se diseñó como primera actividad un curso en línea, de ocho semanas (de octubre a diciembre 2011), con apoyo docente desde la Universidad de Aalborg. El objetivo de este curso era exponer a los profesores al cambio que ha enfrentado la educación superior en la historia y cómo las recientes necesidades de los estudiantes remiten a nuevas estrategias en el aula para mejorar los procesos de aprendizaje. A través de las actividades desarrolladas, los profesores participantes lograron apropiarse de elementos tanto teóricos como prácticos de los temas abordados y dar los primeros pasos para lograr que los cursos a su cargo integren nuevas estrategias didácticas. En este curso se matricularon inicialmente 15 profesores, pero únicamente 8 de ellos cumplieron con los objetivos del curso.

Adicionalmente, se plantea la necesidad de procesos cortos de capacitación semestrales, cuya meta sea que gradualmente los académicos desarrollen competencias que les permitan ir diseñando ambientes de aprendizaje más acordes con los principios del POPP y que puedan ir experimentando y evaluando estas nuevas experiencias en el salón de clase. Esta oportunidad de ir “experimentando” en el aula con pequeñas transformaciones les brinda a los docentes la ocasión de percibir el aula como un ámbito de investigación. Consecuentemente, la introducción de los principios del POPP en los cursos de programación se plantea ajustada a las necesidades y posibilidades de los docentes, a la madurez de los estudiantes y a la naturaleza de cada uno de los cursos, como se expone en la próxima sección.

### B. Introduciendo los principios del POPP

El POPP incorpora una serie de principios didácticos como base para el diseño del ambiente de aprendizaje [26, 27, 13] a continuación algunos de ellos:

- La formulación e investigación de problemas ejemplares: A fin de comprender y resolver el problema, los estudiantes tienen que pasar por diferentes etapas sistemáticas: investigación preliminar, formulación, consideraciones teóricas y metodológicas, experimentación y reflexión, pues el problema se considera el punto de partida y el aprendizaje se organiza en torno a este.
- Control de los participantes: El proceso de aprendizaje es dirigido por los estudiantes en colaboración con el supervisor/facilitador, el estudiante tiene el espacio para definir y formular su propio problema. Se trata de vincular la problemática a sus experiencias e intereses, con el objetivo de generar una mayor motivación hacia el aprendizaje.

- Aprendizaje interdisciplinario: Los problemas pueden extenderse más allá del marco disciplinario conocido, lo cual debe ser considerado cuidadosamente por el docente cuando planifica el proceso de enseñanza.
- Proyectos conjuntos de aprendizaje y acción: El aprendizaje se concibe como un acto social que se lleva a cabo a través del diálogo, la comunicación y la colaboración en grupos mixtos. A menudo, los proyectos se realizan en colaboración con empresas e instituciones públicas.

La Tabla 2 muestra la incorporación de los principios del POPP en aspectos concretos, ubicados gradualmente, de tal manera que el estudiante vaya adquiriendo mayor nivel de decisión y a su vez el docente vaya asumiendo más su rol de facilitador.

Actualmente la mayoría de los cursos del área de programación se encuentran en el nivel 1, donde el docente y los contenidos juegan un papel central en el proceso de aprendizaje. De esta manera se genera una estrategia gradual de incorporación del enfoque que permita, en dos años, construir y evaluar experiencias exitosas [20], y a partir de aquí, ir moviendo los cursos del nivel I actual hacia el deseado nivel III propuesto.

TABLA 2. NIVELES DE INTRODUCCIÓN DE LOS PRINCIPIOS POPP

Principios POPP	Niveles de introducción		
	Nivel 1	Nivel 2	Nivel 3
<i>Resolución de problemas</i>	Problemas diseñados por el docente	Problemas ejemplares y poco estructurados Docente formula el problema	Problemas ejemplares poco estructurados Estudiante identifica el problema o necesidad
<i>Autonomía del estudiante</i>	Docente proporciona todas las directrices del proceso de aprendizaje	Docente identifica las necesidades y los objetivos de aprendizaje	Estudiante identifica y establece sus necesidades de aprendizaje basado en la problemática identificada
<i>Trabajo en grupo</i>	División de tareas entre los miembros	Aprendizaje colaborativo, manejo de conflictos	Aprendizaje colaborativo y liderazgo eficaz
<i>Integración teoría/práctica</i>	Conocimiento de teorías y métodos	Identificación y selección de teorías y métodos	Investigación sobre teorías y métodos
<i>Seguimiento y realimentación</i>	Al inicio y final del proyecto	En cada una de las etapas del proyecto	Continua e integrada en el aula
<i>Relación con la industria</i>	Problemas hipotéticos	Problemas ejemplares contextualizados en la industria	Coordinación industria y sus necesidades

Principios POPP	Niveles de introducción		
	Nivel 1	Nivel 2	Nivel 3
Reflexión	Sobre el producto final Mínima sobre el proceso de aprendizaje	Sobre el proceso de aprendizaje, integración de teoría y práctica y la solución hallada	Permanente y continua sobre el proceso de aprendizaje Sobre la solución, teorías y métodos utilizados, relación con la industria y el proceso de aprendizaje
Evaluación	Evaluación asignada por el docente	Auto-evaluación, co-evaluación Grado de flexibilidad de acuerdo a intereses de los estudiantes	Auto-evaluación, co-evaluación Interiorización de su nivel de logro

La Figura 1 esquematiza el proceso gradual de incorporación de los principios del POPP en los cursos de programación. Como puede apreciarse, se trata de un proceso continuo de refinamiento donde los resultados obtenidos a nivel de un ciclo tienen impacto en el ciclo siguiente, y los resultados obtenidos en un determinado curso tienen implicaciones en el diseño que se realiza en el próximo curso de la secuencia.

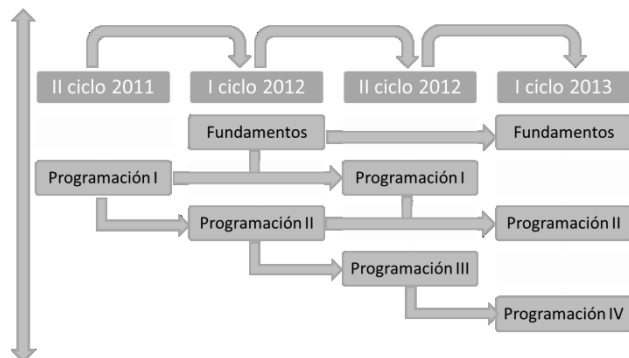


Figura 1. Proceso gradual de inclusión de los principios del POPP

### C. El problema: eje central del diseño

En un esquema didáctico inductivo el problema es el estímulo para el aprendizaje y proporciona el contexto y la oportunidad para aprender e investigar teorías y conceptos, y aplicarlos a situaciones auténticas de la vida real. En el caso de la disciplina informática, y particularmente en el área de programación, la solución de problemas se concretiza mediante la formulación, ejecución y presentación de un proyecto programado, donde el producto final es, en la mayoría de los casos, un programa computacional que resuelve, de alguna manera, la problemática identificada. De esta manera “un buen problema” puede articular todos los otros principios del enfoque POPP, y el proyecto se convierte

en el medio mediante el cual los estudiantes buscan soluciones viables para resolverlos.

En este contexto, el diseño de esta intervención educativa se centra en ir generando, en estudiantes y docentes, una nueva cultura hacia la resolución de problemas. Se plantea trabajar con la cátedra de programación en el desarrollo y aplicación de los siguientes aspectos: objetivos de aprendizaje, tipos de problemas, contenido y forma de resolución.

- **Objetivos de aprendizaje:** El problema es un vehículo a través del cual los estudiantes obtienen conocimiento y adquieren las habilidades deseadas en el curso [7], por eso debe tener una relación estrecha con los objetivos de aprendizaje de los cursos de programación, y conducir a los estudiantes a investigar y a aplicar los conocimientos deseados sobre paradigmas, lenguajes, métodos, buenas prácticas de programación, etc. De acuerdo con el principio de aprendizaje interdisciplinario, el problema debe incluir contenidos de otros cursos, de esta forma el problema contribuye a explorar vínculos entre las distintas áreas de la ingeniería informática.
- **Tipos de problemas:** Los problemas deben ser poco estructurados, abiertos y complejos, plantear un significado ambiguo, deben tener elementos desconocidos, varias soluciones e integrar enfoques de más de una disciplina, pues los estudiantes deben tener la necesidad de investigar. En este sentido, se debe trabajar en la cátedra de programación para moverse hacia proyectos que planteen problemas con estas características, y de esta manera, fomentar la construcción activa del conocimiento [27]. Esto conlleva a promover un cambio en la cultura, tanto de docentes como de estudiantes, porque en las condiciones actuales un problema poco estructurado puede ser percibido por los estudiantes como falta de conocimiento o de esfuerzo por parte de los docentes y repercutir en una evaluación poco favorable de su desempeño.
- **Contenido de los problemas:** Se deben formular problemas acerca de situaciones reales en los proyectos de programación, que abarquen los temas de los cursos y ejerciten lo que los estudiantes tendrán que hacer en su futuro profesional. Esto implica que los problemas deben ser ejemplares y a la vez ajustarse al nivel de conocimiento e intereses de los estudiantes.
- **Forma de resolución:** En los proyectos de programación es muy común el “divide y vencerás”, por lo que se busca diseñar problemas que promuevan el verdadero trabajo en grupo, donde cada uno de los estudiantes se haga responsable de su desempeño individual dentro del equipo, pero a la vez se estimule la interdependencia positiva, en el sentido de que los estudiantes deban depender los unos de los otros para lograr la meta común. Se hace necesario un proceso de formación que les permita a los estudiantes desarrollar



las habilidades necesarias para que el grupo funcione en forma efectiva, tales como el liderazgo, la reflexión crítica y la solución de conflictos.

Es fácil deducir que un factor clave de la intervención propuesta reside en la capacidad de elaborar o seleccionar problemas con las características antes mencionadas. Idealmente este proceso debe ser un trabajo en equipo, esto permite darle unidad a la secuencia de los cinco cursos de programación y asegurar un crecimiento incremental en los objetivos de aprendizaje. Se pretende que, como grupo, los docentes puedan elaborar un conjunto de proyectos para cada uno de los cursos y que los estudiantes puedan seleccionar, a partir de estos. Así se asegura, por un lado, que los proyectos respondan a los contenidos y objetivos de los cursos y; por otro, que los estudiantes se motiven al escoger el problema que les resulte más interesante. Se espera, además, que este proceso de construcción colaborativa de los proyectos entre los docentes contribuya a la cohesión de grupo y sea el punto de arranque para la creación de la comunidad de práctica.

#### D. Potenciales limitaciones

Cualquier proyecto de cambio siempre se enfrenta a obstáculos. El análisis del contexto realizado y las primeras intervenciones en los cursos de Fundamentos de Informática, Programación I y Programación II, ponen en evidencia algunas de las problemáticas a las que se enfrenta esta iniciativa:

- Inestabilidad del cuerpo docente en la cátedra de programación: los docentes varían de un semestre al otro, y los cursos muchas veces están en manos de docentes con jornadas docentes parciales.
- Falta de liderazgo positivo en el área que logre darles unidad y coherencia a los cursos tanto desde el punto de vista técnico como pedagógico.
- La complejidad, todavía presente hoy en día, de abordar la programación orientada a objetos de manera adecuada.
- Escasa comunicación positiva entre los docentes y dificultades para consolidar un trabajo grupal efectivo.
- No se tiene una directriz clara de la administración de la unidad académica con respecto a la incorporación de este tipo de enfoques. Esto implica que a pesar de que el proyecto cuenta con el apoyo de las autoridades de la Escuela, no tiene carácter obligatorio ni hay una directriz sobre la obligatoriedad ni conveniencia de seguir las pautas generadas por el proyecto. Esto significa que los docentes no están obligados a hacer cambios, ni a seguir los lineamientos dados por el proyecto.

## VIII. REFLEXIÓN FINAL

La intervención aquí descrita está en proceso. Se ha realizado un importante esfuerzo en entender el contexto y en un diseño inicial que permita responder adecuadamente al mismo. Los resultados obtenidos hasta el momento muestran que el proceso de cambio no es sencillo, la alineación de los planes de estudio con los enfoques pedagógicos deductivos es muy fuerte, y el proceso de adopción de enfoques inductivos requiere de cambios significativos tanto en el rol de los docentes como en el de los estudiantes.

Lograr una mayor autonomía de los estudiantes; obtener un seguimiento más continuo y oportuno de los docentes; valorar la reflexión sobre el proceso de aprendizaje; y entender los contenidos como un medio para desarrollar habilidades, son algunos de los retos que se plantean en cada una de las intervenciones.

Aunque la mayoría de los docentes indican estar abiertos a hacer cambios que mejoren el aprendizaje y motiven más a los estudiantes, estos procesos implican más trabajo y un cambio de cultura de los actores involucrados. La estrategia utilizada hasta ahora para lograr los cambios deseados en la práctica docente, está alineada con un proceso continuo de divulgación, de formación y de seguimiento. Como tal, no todos los profesores en un curso específico adoptan el enfoque de la misma manera y en la misma medida.

El consolidar a los docentes del área de programación como una verdadera comunidad de práctica -un grupo de trabajo y aprendizaje colaborativo efectivo- apunta como uno de los mayores retos. Además es claro que se requiere de una línea de gestión clara y de una estructura de gestión curricular flexible que permita modificar las estructuras de evaluación tradicionales.

La metodología utilizada permitirá ir evaluando resultando parciales e ir ajustando el diseño de tal manera que se adapte a las características particulares de cada curso, de docentes y de estudiantes, y a partir de esto lograr un abordaje más efectivo del enfoque POPP.

## REFERENCES

- [1] "Computer Science Curricula 2013", ACM, 2011.
- [2] T. W. Hissey, "Enhanced Skills for Engineers," de *Proceedings of the IEEE*, 2000.
- [3] OIT, "Una fuerza de trabajo provista de formación para un crecimiento sólido, sostenible y equilibrado," Ginebra, 2010.
- [4] J. Mills y D. Treagust, "Engineering Education - Is Problem-Based or Project-Based Learning the Answer?," *Australasian Journal of Engineering Education*, vol. 4, 2003.
- [5] M. Prince y R. Felder, "Inductive teaching and learning methods: Definitions, comparisons, and research bases," *Journal of Engineering Education*, vol. 95, pp. 123-138, 2006.

- [6] M. Prince y R. Felder, "The Many Faces of Inductive Teaching and Learning," *Journal of College Science Teaching*, vol. 36, n° 5, 2007.
- [7] L. Prieto, "Aprendizaje activo en el aula universitaria: el caso del aprendizaje basado en problemas," *Miscelánea Comillas: Revista de Ciencias Humanas y Sociales Vol.64. Núm.124. Págs. 173-196*, vol. 64, n° 124, pp. 173-196, 2006.
- [8] J. Perrenet, P. Bouhuijs y J. Smits, "The suitability of problem-based learning for engineering education: theory and practice," *Teaching in higher education*, vol. 5, n° 3, pp. 345-358, 2000.
- [9] F. Dochy, M. Segres, P. Van den Bossche y D. Gijbels, "Effects of Problem-Based Learning: A Meta-Analysis," *Learning and Instruction*, vol. 3, pp. 533-568, 2003.
- [10] M. Prince, "Does Active Learning Work? A Review of the Research," *Journal of Engineering Education*, vol. 93, n° 3, pp. 223-231, 2004.
- [11] E. Nuutila, S. Törmä y L. Malmi, "Pbl and computer programming - the seven steps method with adaptations," *Computer Science Education*, vol. 15, n° 2, pp. 123-142, 2005.
- [12] W. Hamalainen, "Problem-based learning of theoretical computer science," de *Proceedings of the 34th Annual Conference on Frontiers in Education*, 2004.
- [13] Kolmos, "Estrategias para desarrollar currículos basados en la formulación de problemas y organizados en base a proyectos," *Educar*, vol. 33, pp. 77-96, 2004.
- [14] S. Barge, "Principles of problem and project basic learning. The Aalborg PBL Model," Aalborg University., Aalborg, 2010.
- [15] L. Krogh, "Introduction to The Teacher Training Course for Assistant Professors at Aalborg University," Aalborg University, Aalborg, 2009.
- [16] Design-Based Research Collective, "Design-Based Research: An Emerging Paradigm for Educational Inquiry," *Educational Researcher*, vol. 32, n° 1, pp. 5-8, 2003.
- [17] F. Wang y M. J. Hannafin, "Design-based research and technology-enhanced learning environments," *Educational Technology Research and Development*, vol. 53, n° 4, pp. 5-23, 2005.
- [18] M. Coto, "Designing for Change in University Teaching Practices. The case of UNAgora: A community of Practice Approach to Facilitate University Teacher Professional Development in ICT and Project-Oriented Problem Pedagogy," Aalborg, 2010.
- [19] T. C. Reeves, "Design Research from a Technology Perspective," de *Educational Design Research*, London, Routledge, 2006, pp. 52-66.
- [20] T. Guskey, "Professional Development and Teacher Change," *Teachers and Teaching: theory and practice*, vol. 8, n° 3/4, pp. 381-391, 2002.
- [21] Universidad Nacional, "Modelo Pedagógico," Universidad Nacional, Heredia, 2007.
- [22] E. Wenger, *Communities of practice: learning, meaning, and identity*, Cambridge, U.K.: Cambridge University Press, 1998.
- [23] C. Rhodes y S. Beneicke, "Coaching, mentoring and peer-networking: challenges for the management of teacher professional development in schools," *Journal of In-Service Education*, vol. 28, n° 2, pp. 297-310, 2002.
- [24] M. S. Schlager y J. Fusco, "Teacher professional development, technology, and communities of practice: Are we putting the cart before the horse?," de *Designing for Virtual Communities in the Service of Learning*, Cambridge MA, Cambridge University Press, 2004.
- [25] K. Wing Lai, K. Pratt, M. Anderson y J. Stigter, "Literature Review and Synthesis: Online Communities of Practice," Dunedin, New Zealand, 2006.
- [26] L. Dirckinck-Holmfeld, "Designing Virtual Learning Environments Based on Problem Oriented Project Pedagogy," de *Learning in Virtual Environments*, Frederiksberg C, Samfundslitteratur Press, 2002, pp. 31-54.
- [27] E. Graaff y A. Kolmos, "Characteristic of problem-based learning," *International Journal of Engineering Education*, vol. 19, n° 5, 2003.
- [28] S. Butler, "Problem based learning at NCSU," 2002. [En línea]. Available: <http://www.ncsu.edu/pbl/>.
- [29] J. O'Kelly, "Designing a Hybrid PBL course: A Case study of First Year Computer Science at NUI, Maynooth," de *Handbook of Enquiry and Problem Based Learning*, T. Barrett, I. Mac Labhrainn y H. Fallon, Edits., Galway, CELT, 2005.