

Análisis cuantitativo y cualitativo del aprendizaje de Programación I en la Universidad Central del Ecuador

Jéfferson Beltrán^a, Héctor Sánchez^b, Mercedes Rico^b

^a Facultad de Ingeniería Ciencias Físicas y Matemática, Universidad Central del Ecuador, Ciudadela Universitaria Av. América, Quito, Ecuador
jtbeltran@uce.edu.ec

^b Centro Universitario de Mérida, Universidad de Extremadura, Av. Santa Teresa de Jornet 38 06800, Mérida, España
sasah@unex.es, mricogar@unex.es

Resumen. La presente investigación analizó, cuantitativa y cualitativamente los procesos de aprendizaje de Programación I en la Universidad Central del Ecuador, con el fin de identificar las principales problemáticas, y proponer métodos y herramientas que conduzcan a mejorar la motivación y el compromiso de los estudiantes. Para dar respuesta a nuestro objetivo, se realizó un estudio estadístico del porcentaje de estudiantes que aprobaron y reprobaron Programación I. Mediante encuestas electrónicas anónimas, a docentes y estudiantes de Programación I, se analizó cualitativamente la estrategia de aprendizaje. El análisis de resultados nos muestra que un alto porcentaje de estudiantes reprobó la materia, siendo el docente de la asignatura la variable que más influye en el índice de aprobados en la misma. Además, se observa una relación entre los estudiantes que aprobaron o reprobaron Programación II, con el profesor que cursaron Programación I. La metodología, herramientas y principalmente el nivel de exigencia usados por los docentes, no fue estándar y la motivación de los alumnos por el auto-aprendizaje fue muy baja. Como posible solución a los problemas identificados y herramienta de apoyo en el proceso de aprendizaje de Programación I, se propuso el uso de Gamificación de tareas autónomas en un ambiente e-Learning para incentivar y mejorar de esta manera la motivación del estudiante.

Palabras Clave: e-Learning, gamificación, motivación, programación.

1 Introducción

Uno de los principales problemas que enfrentan los docentes es cómo conseguir de sus estudiantes una motivación adecuada y el compromiso en el proceso de aprendizaje. El aprender a programar no está fuera de esta problemática.

La programación es una habilidad difícil de adquirir, que se aprende mejor con la práctica y que será más efectivo si la práctica es auto-dirigida [1]. El papel del profesor, por lo tanto, es clave para motivar a los estudiantes y que participen de manera apropiada. Con una adecuada motivación, sus actitudes podrían cambiar a medida que avanzan a través del curso. Algunos estudiantes pueden descubrir un interés genuino en la programación y desarrollar una motivación intrínseca, que es lo que realmente se debe buscar. Otros, en cambio, pueden llegar a valorar el aprendizaje de programación más por razones extrínsecas. En la actualidad, es una materia que se enseña en varias carreras de ingeniería, circunstancia que provoca un problema añadido, pues los

estudiantes de las diferentes carreras, pueden tener diferentes motivaciones para aprender a programar. Los profesores deben saber por tanto cómo afrontar la motivación de los estudiantes [1].

La motivación es un condicionante decisivo del aprendizaje. Es importante conocer las expectativas de los estudiantes, cuáles son sus motivaciones, teniendo en cuenta que la interacción entre motivaciones intrínsecas y extrínsecas, así como la desmotivación, se encuentran relacionadas con diversos factores que inciden en su rendimiento. Estas motivaciones se relacionan, al menos en parte, con las características del medio socio psicológico, socio económico y cultural de pertenencia y con la experiencia escolar [2].

La motivación en el aprendizaje de programación es baja, debido a la novedad de la asignatura, la dificultad de los alumnos en aprender conceptos abstractos y en su posterior uso en la resolución de problemas, y a la falta de capacidad de trabajo continuo. Los alumnos buscan analogías directas entre problemas resueltos en clase respecto a un problema nuevo que se les plantea. Normalmente no tienden a aplicar los conceptos teóricos, sino que aplican las mismas “recetas” en el mismo orden. Estos problemas, hacen que algunos alumnos abandonen la asignatura prematuramente [3].

Esta falta de motivación entre los estudiantes y en consecuencia las altas tasas de abandono y deserción implícitas, está asimismo relacionada en cierta medida con dificultades de adaptación a la universidad, bien por causas académicas (mala preparación previa, bajo rendimiento, etc.) o bien por causas sociales o de la institución [4]. A esto se une la falta de un estudio a fondo acerca de las habilidades que los estudiantes deben adquirir, lo que conlleva una reducción en el programa de los cursos, limitándose a un recorrido de estructuras sintácticas de un lenguaje de programación específico. El objetivo de estos cursos no es únicamente que el estudiante aprenda a escribir un programa de computador, sino que deben generar una variedad de competencias asociadas con cualquier profesional en ingeniería (aprender a entender un problema, a plantear soluciones efectivas, a manejar lenguajes para expresar una solución, a utilizar herramientas que entiendan esos lenguajes, a probar que la solución sea válida, a justificar las decisiones tomadas, etc.) [5]. El modelo pedagógico que se propone en el Proyecto CUP12, está fundamentado en cuatro estrategias exitosas aplicadas en ingeniería y otras áreas de la ciencia: aprendizaje activo, aprendizaje basado en problemas, aprendizaje incremental, aprendizaje basado en ejemplos [5].

El propósito de la presente investigación, es identificar las principales problemáticas del proceso de aprendizaje de Programación I en la Universidad Central del Ecuador, con el objeto de proponer mejoras al proceso, debido a como mencionan los autores, las condiciones socio psicológicas y socio económicas propias de las instituciones educativas, impactan en los logros académicos y emocionales de los estudiantes, es por esto, la importancia de realizar un estudio cualitativo y cuantitativo del aprendizaje de programación dentro de la propia universidad.

En el capítulo 2 se hace referencia a estudios previos realizados por otros investigadores, a más de los mencionados en la introducción, en el capítulo 3, se describe la metodología de investigación utilizada, para luego en el capítulo 4, mostrar los resultados que se obtuvieron en el estudio, al final, en el capítulo 5, se detallan las conclusiones, y se propone el trabajo futuro en función de las principales problemáticas encontradas.

2 Revisión de literatura

Las tasas de aprobación y reprobación indican que aproximadamente la mitad de los estudiantes reprobaban una o más asignaturas durante el primer año en carreras de ingeniería [11].

En la carrera de Ingeniería de Sistemas y Computación de la Universidad Nacional de Colombia, sede Bogotá, Periodo 2013-I, en la materia de Algoritmos, alrededor del 30% de estudiantes no aprueban la asignatura, en Programación Orientada a Objetos, alrededor del 41% de los alumnos reprobaban la materia [12].

Para resolver problemas motivacionales que surgen en los procesos de aprendizaje y enseñanza de programación, varios autores han realizado estudios que muestran que la Gamificación aplicada a la educación, mejora la motivación y el compromiso del estudiante [6], mediante la aplicación de la mecánica y dinámica de juego a las tareas y los procesos e-Learning [7], también mencionan, que la motivación aumenta mediante facetas de autonomía y competencia [8].

Los alumnos de hoy son “nativos digitales”, para motivarlos, se deben diseñar actividades centradas en los estudiantes para el desarrollo de competencias, las estrategias basadas en juego, pueden contribuir al desarrollo de las mismas, tanto específicas, como transversales, al mismo tiempo que aumentan la motivación del alumno por el aprendizaje [9].

Se realizó un modelo para el aprendizaje de programación utilizando técnicas de Gamificación, y un prototipo de una herramienta de software, basada en WordPress y BadgeOS, que implementa parcialmente el modelo para demostrar la factibilidad técnica de su implementación, un grupo de tres expertos, determinó la validez del modelo y recomendó su implementación en un salón de clases, como herramienta para apoyar un curso tradicional de programación [10].

3 Metodología

La investigación se realizó en la Facultad de Ingeniería Ciencias Físicas y Matemática de la Universidad Central del Ecuador, la universidad más antigua, y la segunda por el número de estudiantes del Ecuador, en las Carreras de Ingenierías: Informática, Computación Gráfica, Diseño Industrial y Matemática, donde se imparte Programación I, durante el semestre abril – septiembre de 2015.

Programación I se imparte en el primer semestre, donde el estudiante adquiere los fundamentos de la programación y los conocimientos iniciales para diseñar algoritmos, que son implementados en programas básicos y sencillos de computador en lenguaje Java, usando NetBeans o Eclipse como herramienta de desarrollo de software, utilizando las diferentes técnicas algorítmicas de la programación estructurada y de la programación orientada a objetos (clases, objetos, atributos y métodos). Programación II, es la continuación de Programación I, donde el alumno aprende la programación orientada a objetos usando Java, así como también, conoce UML para realizar el análisis y diseño de software.

Las materias se evalúan con exámenes (50%) y actividades complementarias (50%), como tareas, lecciones, participación en clase y prácticas. El semestre se divide en dos hemisemestres, cada uno evaluado sobre 20 puntos, el estudiante que reúna 27,5 superará la materia, mientras que con valores inferiores, puede optar a un examen de recuperación, excepto si cursa el ciclo con tercera matrícula. Para rendir el examen de recuperación, se suman las notas de los dos hemisemestres y se divide por dos, a este cociente se agrega la nota del examen de recuperación. Si el estudiante no aprueba la materia, o se retira teniendo alguna nota parcial, se considera reprobado.

Se analizó cuantitativamente, el porcentaje de estudiantes que aprobaron y que reprobaron Programación I. Además, con un enfoque cualitativo, se estudió la estrategia de aprendizaje de la materia.

Análisis cuantitativo. Para determinar el porcentaje de estudiantes que aprobaron y reprobaron Programación I, se realizó un análisis estadístico, entre los semestres 2009-2010 y 2014-2015 con los datos entregados por la Facultad, estudiantes inscritos y matriculados, y de ellos quienes aprobaron y reprobaron.

Las variables geográficas y demográficas que se analizaron fueron las siguientes:

- Género: Masculino y Femenino.
- Geografía: Si el estudiante es de Pichincha o de otra provincia.
- Tipo de colegio donde se graduó en la secundaria: Fiscal, Fisco misional, Municipal, Particular Laico, Particular Religioso, Otro.
- Número de matrícula: Primea, Segunda, Tercera.
- Carrera: Ingenierías Informática, Diseño Industrial, Matemática y Computación Gráfica.
- Semestre: 2009-2010, 2010-2010, 2010-2011, ..., 2014-2015.
- Profesor: Profesor que dictó la materia en el semestre.

Un similar análisis, pero de menor profundidad, se hizo con Programación II, entre los semestres 2010-2010 y 2014-2015, se analizó el número de estudiantes aprobados y reprobados y el docente que dictó Programación II, junto con el profesor con el que el alumno cursó Programación I.

Análisis cualitativo. Se envió un correo electrónico a los docentes y estudiantes que dictaron y que se matricularon, respectivamente en Programación I, en el semestre 2014-2015, invitándoles a llenar una encuesta electrónica anónima. La encuesta sirvió para analizar cualitativamente la estrategia usada en el aprendizaje de Programación I, estuvo abierta durante un mes, durante el semestre 2015-2015.

La encuesta que se realizó a los docentes, incluyó los siguientes temas:

- El uso de herramientas tecnológicas utilizadas en el aprendizaje de Programación I: plataforma virtual, audio, video, internet, software especializado, redes sociales, correo electrónico. Se buscó comprender qué herramientas tecnológicas se usan en Programación I.
- Aspectos metodológicos que más favorecen en el aprendizaje de Programación I (listados en el syllabus de la materia): exposición oral, ejercicios dentro de clase, conferencias, prácticas de laboratorio, trabajos de investigación, talleres, exposición audiovisual, ejercicios fuera del aula,

lecturas obligatorias, prácticas de campo, desarrollo de proyectos, consulta en internet. Se buscó conocer las metodologías que usan en Programación I.

- Los principales problemas a los que se enfrenta el estudiante en el aprendizaje de Programación I, a través de una pregunta abierta.
- La motivación como factor importante en el aprendizaje de Programación I.

La encuesta que se hizo a los estudiantes, abordó los siguientes temas:

- Preguntas geográficas y demográficas para identificar la situación del estudiante de Programación I.
- El uso de herramientas tecnológicas utilizadas en el aprendizaje de Programación I. Las mismas herramientas y objetivo de la encuesta realizada a los docentes.
- Recursos y tecnologías que posibilitan el aprendizaje de Programación I: motivan al estudiante, hacen más fácil el aprendizaje, permiten mantener el interés en clases, favorecen la práctica, optimizan el aprendizaje, no son los más adecuados para el aprendizaje.
- Aspectos metodológicos que más favorecen el aprendizaje de Programación I (listados en el syllabus de la materia). Las mismas metodologías y objetivo de la encuesta hecha a los docentes, además de saber cuáles son las que más favorecen al aprendizaje según los alumnos.
- La contribución de la metodología usada, en cuanto a analizar y diseñar algoritmos utilizando fundamentos y técnicas de programación estructurada, y a la construcción de programas sencillos usando Java.
- Formas de evaluar que favorecen el aprendizaje de Programación I: pruebas parciales, trabajos y tareas fuera del aula, participación en clase, informes de laboratorio, examen final, asistencia a prácticas, trabajo en equipo. Se buscó entender cuáles son las formas de evaluar que más contribuyen en el aprendizaje.
- La motivación como factor importante en el aprendizaje de Programación I. Se investigó qué tan motivado estuvo el estudiante para aprender a programar.

4 Resultados

4.1 Análisis cuantitativo

Entre los semestres 2009-2010 y 2014-2015, se inscribieron 2980 estudiantes, de los cuales 2858 se matricularon (96%), y 122 se retiraron sin tomar la materia (4%). De los estudiantes que se matricularon, 1519 (53%) aprobaron, y 1339 (47%) reprobaron. Se evidencia que existen problemas en el proceso de enseñanza – aprendizaje, debido al alto porcentaje de estudiantes que no lograron superar la materia.

Género. El 70% de los estudiantes matriculados son hombres, y el 30% son mujeres. De los hombres matriculados, el 54% aprobó, y el 46% reprobó. De las mujeres matriculadas, el 52% aprobó, y el 48% reprobó.

Geografía. La mayoría de estudiantes son de la Provincia de Pichincha, 81%, el 92% de ellos son de Quito, el 19% son de provincia. El 53% aprobaron, y 47% reprobaron, sin importar si los estudiantes son de Provincia o de Pichincha.

Estudios secundarios. El 76% de los estudiantes de Programación I, fueron de colegios fiscales o estatales; de colegios particulares o privados el 19%, el 5% restante se distribuye entre colegios municipales, fisco misional y otros. De los alumnos que venían de colegios fiscales, el 53% aprobaron y el 47% reprobaron; entre el 54% y 58% aprobaron, y entre el 42% y 46% reprobaron los estudiantes de colegios particulares.

Las variables: género, tipo de provincia, tipo de colegio, no influyeron en que el estudiante apruebe o no Programación I, debido a que son similares los porcentajes en estas variables, a los porcentajes generales de estudiantes aprobados (53%) y reprobados (47%).

Carrera. El 43% de los alumnos se matricularon en la carrera de Ingeniería Informática, el 22% en Ingeniería en Computación Gráfica, el 20% en Ingeniería en Diseño Industrial, y el 15% restante, en Ingeniería Matemática. En Informática, el 59% de los estudiantes aprobó, y el 41% reprobó. En Computación Gráfica aprobaron el 51%, y el 49% reprobaron; en Diseño Industrial encontramos un 47% de aprobados, y 53% de reprobados; mientras que en Ingeniería Matemática el 48% aprobó, y el 52% reprobó. Fue mayor el porcentaje de estudiantes que aprobaron Programación I, en Informática y Computación Gráfica, carreras donde la materia es de especialidad. En los semestres de estudio, los cursos tuvieron estudiantes de las distintas carreras, si bien el pensum es el mismo, es necesario adaptar los contenidos y ejercicios a la necesidad de cada carrera, y de esta manera tener mayor motivación de los alumnos, desde el semestre abril - septiembre 2105, los cursos tienen solamente estudiantes de una misma carrera.

Un alto porcentaje de estudiantes que pierden la materia no la vuelven a tomar. El 29% de los estudiantes, aunque aprobaron Programación I, no tomaron Programación II. Entre las posibles razones, destacamos el cambio de carrera, el abandono de la misma, o por postergar la matrícula y tomarla cuando el alumno haya aprobado el resto de materias del semestre.

Número de matrícula. El 75% de los estudiantes matriculados fueron de primera matrícula, el 23% de segunda, y el 2% de tercera matrícula. De los estudiantes que tomaron primera matrícula, el 50% aprobaron y el 50% reprobaron, de quienes tomaron segunda matrícula, aprobaron el 59% y el 41% reprobaron, y de quienes se matricularon en tercera matrícula, aprobaron el 78% y reprobaron el 22%.

Semestre. Hasta antes del semestre 2012-2012, con excepción del semestre 2011-2012, el porcentaje de estudiantes reprobados fue mayor que el de aprobados. De los semestres 2012-2013 en adelante, la relación fue lo contrario, como se muestra en la Fig. 1.

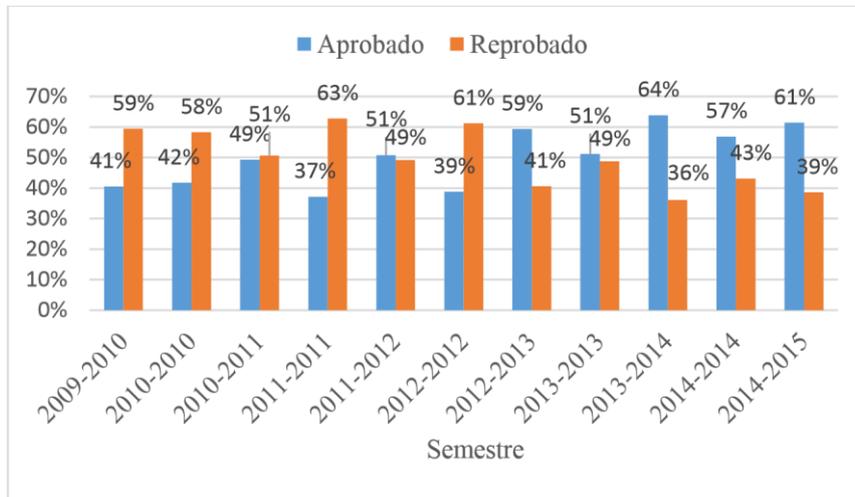


Fig. 1. Porcentaje de estudiantes que aprobaron y reprobaron Programación I, entre los semestres 2009-2010 y 2014-2015.

Profesorado. En el periodo 2012-2013 se integra el Profesor 2, en el semestre 2013-2013 ingresan los Profesores 5 y 14, en el semestre 2013-2014 se integra el Profesor 22, en el periodo 2014-2014 ingresan los Profesores 1, 11 y 6. El docente con mayor antigüedad y que se mantiene aún es el Profesor 4.

En la Tabla 1, se lista el porcentaje de estudiantes que aprobaron y reprobaron Programación I por cada docente. Con el Profesor 5, el porcentaje de aprobados fue del 87%; con el Profesor 2 del 71%, mientras que con el Profesor 4 el índice de aprobados se situó en 24%, y con el Profesor 14 en 26%. Aunque el syllabus y el programa docente es el mismo, se aprecia que el porcentaje de alumnos aprobados y reprobados entre profesores no fue equitativo, y en algunos casos difieren mucho del porcentaje general, 53% aprobados y 43% reprobados.

Tabla 1. Porcentaje de estudiantes que aprobaron y reprobaron Programación I, por cada docente.

Profesor	Estudiantes aprobados (%)	Estudiantes reprobados (%)
Profesor 5	87	13
Profesor 2	71	29
Profesor 1	69	31
Profesor 11	65	35
Profesor 8	64	36
Profesor 7	62	38
Profesor 21	58	42
Profesor 12	57	43

Tabla 1.

Profesor	Estudiantes aprobados (%)	Estudiantes reprobados (%)
Profesor 6	57	43
Sin nombre	52	48
Profesor 23	46	54
Profesor 22	43	57
Profesor 20	34	66
Profesor 14	26	74
Profesor 4	24	76

Adicionalmente, para encontrar correlaciones entre si un estudiante aprueba o reprobaba Programación I y las variables analizadas, se realizó una matriz de correlaciones y un análisis de componentes principales, no encontrándose correlaciones mayores. A continuación, se realizó un análisis de influenciadores clave, incluido en el complemento de minería de datos de Microsoft SQL Server, que usa el algoritmo de Bayes, y se halló que la variable que más impacto relativo tiene para que un estudiante apruebe o no, es el profesor con el que cursó la materia, como se ve en la Tabla. 2.

Tabla 2. Análisis de influenciadores clave para que un alumno apruebe o no Programación I.

Variable	Valor	Favorece Aprobado	Favorece Reprobado
Profesor	Profesor 4		100,0
Profesor	Profesor 5	55,6	
Profesor	Profesor 2	51,8	
Profesor	Profesor 14		38,8
Número matrícula	1		15,9
Número matrícula	3	11,2	
Profesor	Profesor 1	8,1	
Profesor	Profesor 20		8,1
Número matrícula	2	6,0	
Profesor	Profesor 11	3,2	
Profesor	Profesor 22		2,3

Entre los semestres 2009-2010 a 2014-2014 aprobaron Programación I 1188 estudiantes, pero solamente el 71% de los mismos se matriculó en Programación II, en los semestres 2010-2010 a 2014-2015. De los estudiantes que se matricularon en Programación II, el 72% aprobaron el curso, y el 28% reprobaron.

Con los profesores 15 y 9 aprobaron el 52%, y reprobaron el 48%, en cambio, con el Profesor 19, aprobaron el 94%, y reprobaron el 6%.

En la Fig. 2, se muestran los docentes de Programación I, profesores 5 y 2 (con quienes más aprobaron), Profesor 4 y Profesor 14 (con quienes menos aprobaron), en las columnas, están los docentes de Programación II que mayor cantidad de alumnos tuvieron, el Profesor 19 (con quienes más aprobaron), y Profesor 9 y Profesor 15 (con quienes menos aprobaron). Con el Profesor 19, el 94% de estudiantes aprobaron Programación II, sin importar con qué docente tomaron Programación I. Los estudiantes que más aprobaron Programación II, con los profesores 9 y 15, vienen de los cursos de Programación I, con los profesores Profesor 4 y Profesor 14 (con quienes menos aprobaron Programación I), en cambio, los alumnos que más reprobaron Programación II, con los profesores Profesor 9 y Profesor 15, tomaron Programación I, con los profesores Profesor 2 y Profesor 5 (con quienes más aprobaron Programación I). Existe entonces una relación entre el porcentaje de estudiantes que aprobaron y reprobaron Programación II, con el profesor que dictó Programación I.

El porcentaje de estudiantes aprobados y reprobados en Programación I y II, es muy diferente y depende del profesor que dictó la materia, en particular, de la metodología y herramientas usadas, y de la exigencia y dificultad de las pruebas y exámenes.

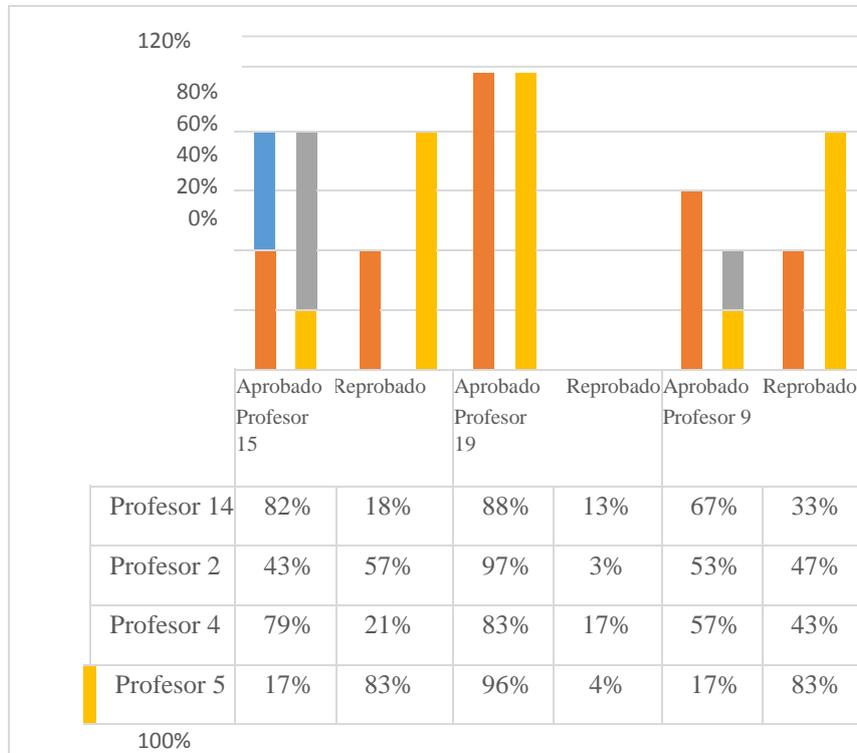


Fig. 2. Relación de alumnos que aprobaron Programación I y luego cursaron Programación II.

4.2 Análisis cualitativo

4.2.1 Encuesta a docentes. De los 10 docentes que impartieron Programación I, en el semestre 2014-2015, uno salió de la universidad. La encuesta fue respondida por 6 docentes.

Herramientas tecnológicas. De las herramientas tecnológicas usadas en el aprendizaje de Programación I, consideraron que únicamente el software especializado, como PSeInt, NetBeans o Eclipse, cumple con los requerimientos de aprendizaje. Internet, usado principalmente como recurso de búsqueda bibliográfica, cumple con el 50,0%, al igual que la plataforma virtual Sakai, además que su uso no está generalizado, ni estandarizado, el resto, tan sólo cumple con el 33,3%, como se muestra en la Tabla 3. Los recursos y herramientas tecnológicas usadas, no cumplen en la mayoría de los casos, con los requerimientos de aprendizaje, además que los docentes no las usaron en forma estandarizada. Existe entonces, una oportunidad de mejora que es necesario analizarla, pues los mismos estudiantes consideraron que su uso favorece en el aprendizaje de Programación I.

Tabla 3. Cumplimiento de requerimientos de aprendizaje de las herramientas usadas en Programación I.

Si cumple Herramientas	Cumple (%) los	parcialmente o mínimamente requerimientos de aprendizaje (%)
Software especializado	100,0	0,0
Plataforma virtual	50,0	50,0
Internet	50,0	50,0
Audio	33,3	66,7
Video	33,3	66,7
Redes sociales	33,3	66,7
Correo electrónico	33,3	66,7

Metodología. En la Tabla 4, se resume la respuesta de los docentes, sobre los aspectos metodológicos en la enseñanza de Programación I, aquellos donde el maestro tiene control dentro del aula, son los que cumplen mayormente con los requerimientos de aprendizaje, no así, en donde el estudiante realiza trabajo autónomo.

Tabla 4. Cumplimiento de requerimientos de aprendizaje de los aspectos metodológicos usados en Programación I.

Metodología	Si cumple (%)	Cumple parcialmente o mínimamente los requerimientos de aprendizaje (%)
Prácticas de laboratorio	100,0	0,0
Ejercicios dentro de clase	83,3	16,7
Prácticas de campo / Resolución de problemas	83,3	16,7
Talleres	66,7	33,3
Exposición audiovisual	66,7	33,3

Tabla 4.

Metodología	Si cumple (%)	Cumple parcialmente o mínimamente los requerimientos de aprendizaje (%)
Desarrollo de proyectos	66,7	33,3
Trabajos de investigación	66,7	33,3
Conferencias (profesores invitados)	66,7	33,3
Exposición oral (clase magistral)	66,7	33,3
Consulta en internet	50,0	50,0
Lecturas obligatorias	50,0	50,0
Ejercicios fuera del aula	33,3	66,7

El 83,4% de los docentes, consideraron que la metodología usada en la enseñanza de Programación I es muy buena o excelente y que la motivación es un factor importante en el aprendizaje de Programación I.

Los principales aspectos metodológicos, sin tener una priorización, que los docentes consideraron se deben mejorar, en el aprendizaje de programación son los siguientes:

- Examen unificado y exigencia similar, por parte de los profesores dentro de la misma carrera.
- Estandarización del contenido y alcance de la materia para todos los profesores dentro de la misma carrera.
- Reforzar los contenidos, guías, prácticas de laboratorio y los recursos para el proceso de enseñanza – aprendizaje.
- Mejorar el acceso a internet, laboratorios y el uso de la plataforma virtual.

También mencionaron en la encuesta, que los principales problemas que enfrenta el estudiante, sin tener una priorización, en el aprendizaje de Programación I son:

- Falta de motivación para el autoaprendizaje.
- Falta de conocimientos matemáticos básicos previos.
- Deficiencia de equipos en los laboratorios.
- Deficiencia en el acceso a internet.
- Muchos alumnos por curso.

La metodología, recursos y herramientas tecnológicas que usaron los profesores, no fue la misma, y tampoco la exigencia en la evaluación, esto produjo los distintos resultados entre docentes.

4.2.2 Encuesta a estudiantes. De los 539 alumnos matriculados en Programación I, en el semestre 2014-2015, se logró enviar un correo a 505 estudiantes, invitándoles a que contesten las preguntas en forma anónima, contestaron 85. La muestra es estadísticamente significativa, para un porcentaje de error del 10%, y un nivel de confianza del 95%.

Perfil de los alumnos que responden la encuesta. El 36,5% de los estudiantes que respondieron la encuesta son mujeres, y el 63,5% son hombres, su edad fluctúa entre 17 y 28 años. El 69,4% estudiaron en colegios fiscales o públicos, en colegios particulares o privados el 25,9%, y el 4,7% en colegios municipales.

Estudiaron Ingeniería Informática el 32,9%, Ingeniería en Diseño Industrial el 29,4%, Ingeniería en Computación Gráfica el 21,2%, e Ingeniería Matemática el 16,5%. El 54,1% fueron de primera matrícula, y el 45,9% de segunda. El 70,6% pasó el curso y el 29,4% no.

El 72,9% de los alumnos están en la universidad que querían estudiar, y el 27,1% no. El 56,5% no estudian la carrera que querían, el 43,5% sí. Esto es un factor de desmotivación, que explica en parte, el alto porcentaje de estudiantes que repiten la materia y/o abandonan la carrera en los primeros semestres.

El 30,6% de los alumnos, estudió alguna asignatura relacionada con programación en el colegio, el 69,4% no lo hizo.

Herramientas tecnológicas. Internet como herramienta de búsqueda bibliográfica, es la herramienta que más se usó como apoyo en el aprendizaje de Programación I, un 67,1% de estudiantes así lo consideraron, el software especializado, como PSeInt, NetBeans o Eclipse, un 62,3%, la plataforma virtual Sakai, sólo un 32,9%, la herramienta que menos se usó, fue las redes sociales, un 20,0%. El 44,7% de los estudiantes respondieron que fue muy bueno y excelente el uso de las herramientas tecnológicas, el 44,7% respondieron que fueron buenas las herramientas usadas, y el 10,6% afirmó que son regulares o malas.

Respondieron que el uso de los recursos y herramientas tecnológicas favorecen la práctica en un 84,7%, que motivan al estudiante, hacen más fácil el aprendizaje, permiten mantener el interés y optimizan el aprendizaje entre 70,6% y 76,4%, pero un 30,6% de estudiantes pensaron en forma indiferente, están en desacuerdo o totalmente en desacuerdo con la efectividad del uso de las herramientas usadas. Se evidencia nuevamente que el uso de las herramientas tecnológicas no está estandarizado, ni generalizado entre los docentes, y que puede ser optimizado su uso para una mayor efectividad en el aprendizaje de Programación I.

Metodología. Los estudiantes se sintieron más a gusto con las metodologías que se realizaron en el aula, como prácticas en el laboratorio, y ejercicios dentro de clase. Tienen menor agrado por los trabajos autónomos, como ejercicios fuera del aula, trabajos de investigación, y lecturas obligatorias. La exposición oral o clase magistral, es también una metodología no muy acogida por los alumnos, como se muestra en la

Tabla 5. Prefirieron la práctica a la teoría cuando aprendieron a programar. Las metodologías usadas por los docentes, al igual que las herramientas, no fueron las mismas entre los docentes. El semestre tiene 96 horas de clase presencial, de las 6 horas semanales, 4 se proponen sea en el laboratorio y 2 en el aula, además el trabajo autónomo es de al menos 144 horas. Es importante notar que no todos los docentes cumplen esta distribución de carga horaria. El 45,9% de los alumnos, afirmaron que en general la metodología usada en el aprendizaje de Programación I, fue excelente y muy buena, el 54,1% consideró que sólo fue buena, regular y mala.

Tabla 5. Metodologías y técnicas usadas en la enseñanza de Programación I.

Metodología	Totalmente de acuerdo y de acuerdo (%)
Prácticas de laboratorio	98,9
Ejercicios dentro de clase	96,4
Talleres	88,2
Prácticas de campo / Resolución de problemas	88,2
Consulta en internet	81,2
Exposición audiovisual	81,2
Desarrollo de proyectos	77,6
Ejercicios fuera del aula	71,7
Trabajos de investigación	69,4
Conferencias (profesores invitados)	68,2
Exposición oral (clase magistral)	55,3
Lecturas obligatorias	48,2

En general, los alumnos respondieron que no tienen problema con las formas de evaluar el curso, en forma individual, así como trabajo en equipo y prácticas.

El 94,2% de los estudiantes, consideraron que están totalmente de acuerdo y de acuerdo, que es importante aprender a programar para su futuro profesional, sin importar la carrera que estudian. El 55,3% de alumnos, respondieron que aprender a programar es difícil, y tan sólo el 17,7% consideraron que no lo es, el resto es

indiferente. El 74,1% considera que no es aburrido aprender a programar, tan sólo en 5,9% piensa que sí lo es, los demás respondieron indiferente.

Motivación. El 98,8% afirmaron que la motivación es un factor importante en el aprendizaje de Programación I. El grado de motivación entre muy alto y alto, fue del 48,2%, el 32,9% fue bueno, regular y malo, el 18,9%. La motivación disminuye, mientras la complejidad de los ejercicios aumenta, no se mantiene constante durante el curso. Se debe proponer una metodología, que al menos mantenga y mejor aún, haga crecer la motivación de los estudiantes durante todo el curso.

Valoración de conocimientos adquiridos. Solamente el 17,6% de los alumnos que respondieron la encuesta, están muy satisfechos con el curso de Programación I, el 50,6% está satisfecho, pero también hay un porcentaje alto, del 31,8%, que no lo está. Si se añade que históricamente, el 47,0% de los estudiantes reprobó la asignatura, se plantea nuevamente la necesidad de proponer una nueva metodología estandarizada, con un mismo grado de exigencia entre los docentes, que utilice herramientas tecnológicas para optimizar el proceso, y que motive a los estudiantes al aprendizaje autónomo.

5 Conclusiones y trabajo futuro

Del análisis cuantitativo, se concluye que existen problemas en el proceso de enseñanza – aprendizaje de Programación I, debido al alto porcentaje de estudiantes que reprobaron, y que muchos de ellos no vuelven a tomar la materia. De los alumnos que aprobaron, el 29%, no tomaron Programación II.

Las variables geográficas y demográficas: género, geografía, tipo de colegio, no influyeron en que el estudiante apruebe o no Programación I.

La variable que mayor impacto relativo tiene en que los alumnos aprueben o reprobren Programación I, es el docente con el que tomaron la materia, debido a que no se usó una metodología, y herramientas estándares, y principalmente al nivel de dificultad en las evaluaciones, especialmente en pruebas y exámenes, pues no se utilizó examen unificado. Además, existe relación entre el porcentaje de estudiantes que aprobaron y reprobaron Programación II, con el profesor que dictó Programación I.

Es necesario entonces evaluar, proponer y estandarizar la metodología de aprendizaje y se recomienda un nivel de exigencia similar en la evaluación, para todos los docentes que imparten una misma materia, además de adaptar los contenidos y ejercicios en cada carrera.

Las herramientas que usaron los docentes, en particular la plataforma virtual Sakai, no cumple totalmente con los requerimientos de aprendizaje de Programación I, además que su uso es limitado, no está estandarizado, ni generalizado. Debe evaluarse mejorar la herramienta o cambiarla por otra que brinde mejores funcionalidades y facilidades.

Los aspectos metodológicos donde el profesor tiene control dentro del aula son los que cumplen mayormente con los requerimientos de aprendizaje, además que son los de mayor agrado para los estudiantes, les gusta la práctica, mas no la teoría. Los trabajos

autónomos son los que menos les gusta a los alumnos, esto muestra que existe una falta de motivación, para el autoaprendizaje. Se debe plantear entonces, métodos para mantener e incrementar la motivación por los trabajos y tareas fuera del aula.

Casi la mitad de los estudiantes, un 56,5%, no estudian la carrera que querían. Esto es un factor de desmotivación, que explica en parte, el alto porcentaje de estudiantes que repiten la materia y/o abandonan la carrera en los primeros semestres.

La motivación es un factor importante en el aprendizaje de Programación I. Es muy importante para el futuro profesional del estudiante aprender a programar, no es aburrido, pero sí difícil aprender a programar.

Únicamente el 17,6% de los estudiantes están muy satisfechos con el curso, lo que plantea nuevamente la necesidad de proponer una nueva metodología estandarizada, con un mismo grado de exigencia entre los docentes, que utilice herramientas tecnológicas para optimizar el proceso, que motive a los estudiantes al aprendizaje autónomo, y que mantenga la motivación durante todo el curso.

Como trabajo futuro, sin pretender resolver todos los problemas encontrados, como apoyo en el aprendizaje de Programación I, se propone el uso de Gamificación en un curso de programación presencial en un ambiente e-Learning, usando Moodle y otras herramientas complementarias, y de esta manera incrementar la motivación y el compromiso del estudiante con las tareas autónomas.

Moodle tiene varios módulos, como asignación de puntos, insignias, tableros, despliegue de recursos mediante condiciones (niveles), retroalimentación en línea para ejercicios Java, etc., para Gamificar un curso. De esta manera, también se estaría mejorando la funcionalidad de la plataforma.

Finalmente, y como se indicó en la sección de resultados, el 83,4% de los docentes, están de acuerdo y totalmente de acuerdo, en que la Gamificación usando un ambiente e-Learning puede ayudar en el proceso de aprendizaje de Programación I.

Agradecimientos

Los autores desean expresar su agradecimiento a la Facultad de Ingeniería Ciencias Físicas y Matemática de la Universidad Central del Ecuador, por la entrega de los datos históricos para realizar el análisis cuantitativo, y a los docentes y estudiantes que gentilmente contestaron la encuesta, para realizar el análisis cualitativo.

Referencias

1. Jenkins, T.: The Motivation of Students of Programming. University of Leeds (2001)
2. Steinmann, A., Bosch, B., Aiassa, D.: Motivación y Expectativas de los Estudiantes por aprender Ciencias en la Universidad. Un estudio exploratorio. Revista Mexicana de Investigación Educativa, vol. 18, núm. 57, pp. 585 a 598 (2013)
3. Salamó, M., Camps, J., Vallespí, C., Vernet, D., Llorá, X., Berndó, E., Garrel, J., González, X.: Iniciativas para motivar a los alumnos de Programación. Departamento de Informática. Universidad Ramon Llull (2001)

4. García, A., Lías, A., Mahillo, Á., Pinero, R.: Abandono de primer año en la Ingeniería Informática. Universidad Politécnica de Madrid (2014)
5. Villalobos, J.: Proyecto CUP12 – Una solución integral al problema de enseñar y aprender a programar. Universidad de los Andes (2009)
6. Chevtchenko, A.: Gamified Education. Introducing Game Elements into the School Environment to Enhance Student Motivation and Performance. Erasmus University. Rotterdam (2013)
7. Muntean, C.: Raising engagement in e-learning through gamification. Babes-Bolyai University. Rumania (2011)
8. Larsen, E.: Game of Exams. Department of Informatics. University of Oslo (2013)
9. González, C., Mora, A.: Técnicas de gamificación aplicadas a la docencia de Ingeniería Informática. Departamento de Ingeniería Informática. Universidad de La Laguna (2014)
10. Arenas, R.: Modelo para la Motivación del Aprendizaje de la Programación utilizando Gamification. Facultad de Ingeniería. Pontificia Universidad Javeriana. Bogotá (2014)
11. Caballero, L., Castillo, Y., Bermúdez, C.: Comparación de las tasas de aprobación, reprobación, abandono y costo estudiante de dos cohortes en carreras de Licenciatura en Ingeniería en la Universidad Tecnológica de Panamá. UTP (2014)
12. Rodríguez, G.: Pérdida de Asignaturas Universidad Nacional de Colombia, Sede Bogotá Periodo 2013-I. Universidad Nacional de Colombia. Bogotá (2013)