

López, C.; Ferreiro, V.; Brito, J. & Garambullo, A. (2011). *La generación del milenio, su diversidad y sus implicaciones en el desarrollo académico*. Recuperado de: https://docs.fajardo.inter.edu/Acad/revisionPEG/Articulacion%20Educacion%20General/La%20generacion%20del%20milenio,%20su%20diversidad%20y%20sus%20implicaciones%20en%20el%20desarrollo%20academico_2011.pdf

OECD (2013). Learning environments and innovative practice, en *Innovative Learning Environments*. OECD Publishing, <http://dx.doi.org/10.1787/9789264203488-3-en>

Przybylski, A. (2014). *Electronic Gaming and Psychosocial Adjustment*. Recuperado de: <http://pediatrics.aappublications.org/content/early/2014/07/29/peds.2013-4021>

Abstract: Design teaching is a complex activity where several actors characterized by a dynamic approach participate. In this context, ambient learning must be flexible in order to embrace the necessities of the millennium generation. We propose a gamification strategy in order to bring to the classroom projects linked with the Industry.

Keywords: Workshop, design, teaching, learning, gamification

Resumo: O ensino do design é uma atividade sofisticada onde os atores são agentes dinâmicos. Por isso, é necessário que os ambientes de aprendizagem sejam flexíveis para responder às necessidades da geração milênio. Se propõe o gamification como estratégia para a oficina de design de projetos reais vinculados à Indústria.

Palavras chave: Oficina - design - ensino - aprendizagem - gamification.

(*) **Carlos Fabián Bautista Saucedo.** Diseñador de la Comunicación Gráfica y Maestría en Ciencias y Artes para el Diseño (Universidad Autónoma Metropolitana Unidad Xochimilco). Especializado en investigación aplicada, gestión del diseño, análisis de las emociones del usuario y branding. Ha participado en agencias de publicidad, estudios de diseño interactivo y consultor privado desde 1998. Es académico con trayectoria ininterrumpida desde 2001. Ha participado en diseño curricular, desarrollo de proyectos de investigación aplicada en vinculación con la industria y administración docente y publicado artículos en foros académicos a nivel internacional. Actualmente es profesor en CETYS Universidad, siendo Director de la Escuela de Ingeniería, Campus Tijuana.

Diagnóstico de los estudiantes de ingeniería en diseño de entretenimiento digital para facilitar el aprendizaje de los fundamentos de programación

Actas de Diseño (2021, diciembre),
Vol. 35, pp. 50-54. ISSN 1850-2032.
Fecha de recepción: julio 2016
Fecha de aceptación: agosto 2017
Versión final: diciembre 2021

Andrés Bedoya Tobón (*)

Resumen: Existe un problema de bajo rendimiento, deserción estudiantil y desmotivación durante el estudio de la programación de software, que se presenta en diferentes materias del pensum de ingeniería y particularmente en los cursos de fundamentos de programación de la carrera de Ingeniería en Diseño de Entretenimiento Digital. Para entender las necesidades de aprendizaje del estudiante se realiza un diagnóstico por medio de encuestas a estudiantes de Ingeniería de Sistemas e Informática, Ingeniería electrónica, Ingeniería en Diseño de Entretenimiento Digital y a diseñadores gráficos en la Universidad Pontificia Bolivariana. El fin de esto es poder identificar factores claves en el aprendizaje, desde el punto de vista de ingenieros y diseñadores. Adicionalmente, se hizo una prueba piloto de un material creado para facilitar la solución de exámenes escritos y se presentan los resultados como un avance sobre lo que se ha encontrado hasta el momento.

Palabras clave: Programación - enseñanza - ingeniería - encuesta - diseño gráfico - software.

[Resúmenes en inglés y portugués y currículum en p. 54]

Introducción

Existe una dificultad para el aprendizaje de los fundamentos de programación de estudiantes de Ingeniería en Diseño de Entretenimiento Digital. Las causas identificadas para este problema en el aula de clase son:

- Mal uso del tiempo: El tiempo invertido para resolver ejercicios de manera autónoma, estudiar para los exámenes y los talleres de estudio no es suficiente. Esto se ha identificado porque los estudiantes pierden los exámenes escritos con preguntas que están presentes en los talle-

res de estudio, que se brindan para estudiar de manera voluntaria, previamente a la presentación del examen.

• **Desmotivación:** Algunos estudiantes de Ingeniería de Entrenimiento Digital no ven en su perfil profesional la programación como una materia atractiva para seguir profundizando, ya que esperan especializarse en animación o en diseño de experiencias interactivas, donde no van a necesitar ese saber.

• **Las capacidades de saberes previos son insuficientes:** Los requisitos previos del estudiante no son adecuados, pues presenta problemas con el dominio de un inglés básico, no reconoce o aplica los conceptos del álgebra de Boole y principios básicos de lógica, no reconocen los tipos de razonamientos y se les dificulta entender conceptos abstractos.

• **Metodología de estudio incorrecta:** El estudiante al parecer no es consciente del grado de exigencia de la universidad y no prepara las tareas y demás obligaciones con la debida organización y responsabilidad.

• **Mala pedagogía:** Una metodología en el aula con un enfoque que no es apropiado o no enfrenta el problema respondiendo a las necesidades del estudiante.

• **Material de apoyo docente:** No existe material oficial propio de la Facultad de Ingeniería en tecnologías de la información y la comunicación (TIC) que permita al estudiante guiarse mientras estudia el curso. Además, el material que se encuentra en la biblioteca de la UPB normalmente es avanzado, está en inglés, es abundante en contenido (muchos temas importantes para estudiar) o en ocasiones el estudiante no lo adquiere porque es costoso.

El problema finalmente tiene los siguientes efectos sobre el estudiante:

• **Cambio de carrera:** Cambio para carreras como Diseño Gráfico o profesiones afines, debido a la dificultad en comprender los fundamentos de programación o las ciencias básicas.

• **Deserción escolar:** Cancelación temprana de la materia por bajo rendimiento.

• **Perder el curso:** Muchos estudiantes pierden el curso, por bajas notas y desinterés.

• **Mala salud (estrés):** Algunos estudiantes han demostrado ansiedad en los exámenes, manifestándose por medio de enfados, desespero, lágrimas, entrega del examen sin contestar puntos, entre otras cosas.

• **Frustración:** Algunos estudiantes se frustran frente a la solución a un problema y se dejan vencer fácilmente por el desafío.

Lo anterior se contrastó con algunas ideas de Jorge Villalobos (2007), expuestas en el congreso colombiano de computación:

- Alta deserción
- Baja demanda de los cursos
- Alto nivel de plagio
- No se cumplen las expectativas de los estudiantes
- Problemas en los cursos avanzados
- Problemas de coordinación entre secciones
- Problemas de formación de profesores

- Poco interés de los profesores
- Manejo de la motivación y la frustración

Este era el panorama en 2004, en el contexto analizado por Villalobos (2007), y todavía en 2015 se viven muchos de estos problemas en el contexto de Ingeniería de Diseño de Entrenimiento Digital, con un nuevo factor que entra a crear un contexto diferente: la inclusión de estudiantes con un perfil de diseñador.

Además, se revisó la investigación realizada en la enseñanza de programación centrada en la evaluación con hojas de trucos o “*cheat-sheets*” (de Raadt, 2012), de tal forma que sirvió como punto de partida para proponer el folleto de apoyo para la clase.

La diferencia con la propuesta de “*cheat-sheet*” difiere en que el estudiante es el que realiza la hoja de trucos que va a sacar en el examen y luego se mide el impacto en resultados del examen en estudiantes que usan la hoja de trucos y en los que no. Se toman por tanto las siguientes ideas para dirigir esta parte de la investigación:

• “Exámenes a libro abierto pueden tener beneficios, como reducir la ansiedad, desenfatisa la memorización y reduce la trampa” (de Raadt, 2012).

• “Los estudiantes adaptan la información a sus propias necesidades y escriben la información que ellos todavía necesitan aprender. El acto de escribir y organizar la información para la hoja de trucos permite a más estudiantes llenar los vacíos en su conocimiento” (de Raadt, 2012).

• “El proceso de crear la hoja de trucos, puede también ser educativo... No hay estudios que reporten el uso de hojas de trucos en exámenes de ingeniería de computación” (de Raadt, 2012).

• “En circunstancias del mundo real, los programadores confían en recursos con información específica, tales como recursos de sintaxis y ejemplos de solución de problemas... ellos no esperan memorizar información específica” (de Raadt, 2012).

Lo que se busca con el proyecto, en esta etapa, es poder lograr un diagnóstico que sirva como punto de partida para el contexto actual de Ingeniería en Diseño de Entrenimiento Digital y obtener unos resultados experimentales para identificar factores claves en el proceso de aprendizaje del estudiante, de tal manera que permita diseñar material didáctico que responda a las necesidades de enseñanza del estudiante (usuario) por medio de la prueba de un mínimo producto viable, realizado para la primera fase de validación.

Metodología

En este diagnóstico, el lenguaje de programación seleccionado para enseñar a programar es irrelevante, porque la enseñanza de los fundamentos de programación tiene la ventaja de ser una materia cuya temática se mantiene conceptualmente de la misma forma desde noviembre de 1996, año en el cual fue definido el estándar ECMA

que actualmente se conoce como el estándar ECMA-262 (International, 2015), definido en la norma ISO/IEC 16262:2011 (ISO, 2011).

La norma define técnicamente la sintaxis del lenguaje de programación y algunas características importantes, como la declaración de variables, el contexto de las variables, etc., y en la actualidad muchos de los lenguajes que se usan profesionalmente y en la academia siguen este estándar. El uso de ECMAScript se ha trasladado más allá del simple *scripting* y ahora es usado por el espectro completo de tareas de programación en muchas escalas y ambientes diferentes. El uso de ECMAScript se ha expandido, gracias a las características y facilidades que provee. ECMAScript es ahora un muy completo lenguaje de programación de propósito general. Algunas de las facilidades de ECMAScript son similares a las usadas en otros lenguajes de programación, en particular C y Java (International, 2015).

Es un aliciente aprender a programar, independiente del lenguaje, porque no se enseña a usar una herramienta, sino que se brinda la capacidad de adaptarse y crear soluciones de *software* independiente del lenguaje, siendo más importante la solución de un problema por medio del pseudocódigo y el diagrama de flujo. Sin embargo, es importante anotar, que la selección del lenguaje de programación es clave para la enseñanza en el aula. Se revisó un artículo *CS1 programming language option* (Giangrande, 2007) al respecto, y se encontró que el lenguaje es relevante en la enseñanza de elementos más abstractos, como es la programación orientada a objetos, que es un tema que se sale del contexto del problema actual. Pero es importante resaltar esta opción para poder entender por qué se limita el problema a los conceptos de programación básica.

Puristas pueden argumentar que los estudiantes necesitan estar familiarizados con lo que actualmente está pasando 'adentro' de la máquina antes que sean introducidos en la abstracción propuesta por lenguajes de programación de alto nivel. En la otra mano, el argumento puede ser que es más beneficioso enseñar lenguajes de alto nivel primero, y después 'pelar la cebolla' para exponer la arquitectura subyacente de la máquina y todos los asuntos asociados con ese nivel (Giangrande, 2007). El enfoque seleccionado para la investigación actual es partir de los fundamentos de la máquina y entender las estructuras básicas de control y manejo de memoria, teniendo en cuenta mantener un nivel de profundidad intermedio en la herramienta de programación, ya que trabajar a más bajo nivel, sin la metodología apropiada, puede ser causa de desmotivación.

Los lenguajes de programación a este nivel eran relativamente simples, con una instrucción haciendo una cosa, y la operación tenía una relación uno a uno con los cambios en la arquitectura subyacente... Aprender a programar a este nivel es muy tedioso, pero asegura que los estudiantes aprendan qué está pasando realmente adentro de la máquina (Giangrande, 2007). Para mantener la motivación alta, se propone partir de lo simple y, por medio de logros tempranos, lograr enganchar al estudiante y motivarlo.

Se encontró, además, la recomendación de herramientas de aprendizaje, que aunque se enseñan temas como pro-

gramación orientada a objetos, que ya de entrada estamos descartando como una opción, se resalta la metodología usada para motivar al estudiante a aprender.

BlueJ fue especialmente diseñado para enseñar programación orientada a objetos usando *Java*, programas completos pueden ser desarrollados y ejecutados con el ambiente de *BlueJ* (of kent). Lo que hace esta herramienta de aprendizaje es agregar herramientas gráficas para realizar un *debug* visual a la aplicación y, de esta forma, el estudiante poder realizar un seguimiento del diagrama de flujo de la aplicación que está diseñando, mediante su ejecución paso a paso, algo similar como la operación paso a paso de *visual studio*.

Alice es otra herramienta de desarrollo para introducir fácilmente a un estudiante a la programación orientada a objetos (Mellon, s.f.). Se resalta de Alice la gamificación, la lúdica de aprendizaje y sus herramientas didácticas, las cuales serán claves como punto de partida para diseñar el material de apoyo del proyecto de maestría.

Adicionalmente, realizando una revisión en sitios de tutoriales en línea, se resalta *code.org* (Ali & Partovi, s.f.), que enseña a programar usando personajes populares como *Star Wars* o *Frozen*. Sin duda es un muy buen referente, teniendo en cuenta el apoyo que tiene esta iniciativa de empresas como *Google*, *Microsoft*, *Infosys Foundation USA*, *Omidyar Network*, *Starup: Education*, *Quadvirium Foundation*, *Sales Force*, *Verizon*, *Juniper Networks*, *AmazonSmile*, *ACM*, entre muchos más, siendo además muy importante como referente para conocer las empresas que actualmente están impulsando iniciativas que fomenten la enseñanza de la programación.

Teniendo en cuenta el problema, se realizó la creación de un folleto que resume el contenido del curso de fundamentos de programación, el cual funcionó como material oficial de apoyo durante 2 cursos de fundamentos de programación en el presente año (2015), con el fin de ser material de consulta para el estudiante en evaluaciones escritas, donde solamente se puede usar lápiz, papel, borrador, el enunciado de evaluación y el folleto como "cheat-sheet" (de Raadt, 2012), que es una hoja de trucos que resume los principales conceptos vistos en clase, con el fin de concentrar la evaluación en la solución de ejercicios teórico-prácticos, y disminuir la ansiedad por no recordar estos conceptos en la evaluación.

Se evaluó el material con estudiantes, para mirar su pertinencia en el proceso de aprendizaje por medio de encuestas con interrogantes relacionados, que ayudan a determinar los factores que intervienen en el aprendizaje del estudiante (usuario) actual de los cursos de programación.

Resultados

La primera parte del proyecto, tuvo como alcance realizar un folleto de 32 páginas, que resumiría el contenido de la materia, de tal forma que los estudiantes del curso de fundamentos de programación lo pudieran usar como referente en todo el semestre, además de ser el único material autorizado para usar durante exámenes escritos de la materia.

El estudiante se apropió del material, incluso lo portaban en la billetera y en sus útiles de uso diario. El folleto se desarrolló para ser usado solo en los exámenes escritos, pero luego en el aula de clase se evidenciaron otros usos, como los que se mostraron anteriormente.

El material fue registrado y cuenta con código ISBN y código de barras. Adicionalmente es posible imprimir el archivo de manera gratuita.

El material es regalado a los estudiantes durante el curso, pero por fines de registro se asigna un valor de \$5000 (COP). Este material fue una autoedición, que se inspiró en la distribución de los fanzines y cómics. Se ha entregado gratuitamente a más de cien personas, entre alumnos, docentes, coleccionistas, dibujantes, diseñadores, ingenieros, entre otros. Luego de haber sido usado en dos cursos de fundamentos de programación, se hizo una encuesta con los siguientes resultados:

- 71% de los estudiantes son de sexo masculino
- 29% de los estudiantes son de sexo femenino
- El 71% de los estudiantes tiene entre 18 y 22 años
- El 19% de los estudiantes tiene entre 23 y 25 años
- El 7% de los estudiantes tienen entre 26 y 29 años
- El 3% son mayores a 29 años

97% de los estudiantes les gusta la programación o le interesa aprender a programar, con lo cual podemos decir que la programación por sí misma es un motivante para aprenderla.

Todos los estudiantes, estuvieron de acuerdo en que la programación es importante. Cabe resaltar que es importante principalmente para solucionar detalles técnicos o problemas generales, ya que se preguntó específicamente si era solo en la carrera y solo el 28% asocia la programación a su relevancia para la carrera: la mayoría cree que es importante principalmente para ser autónomo resolviendo problemas de programación.

Todos han tenido relación con programación, para ninguno era nuevo el tema.

Los exámenes escritos no son muy populares entre los estudiantes: solamente un 3% considera que es una buena estrategia metodológica. De esta pregunta se resalta la preferencia por presentar un examen en el computador, pero previamente haber realizado un taller con preguntas similares para un 49%. Claro que aparece un indicador muy importante para el uso del folleto como material de apoyo, 19% prefiere estudiar un taller y luego presentar un examen escrito. Esto se podría interpretar como que el estudiante le gusta ser guiado por medio del taller y ponerse a prueba en el examen, y el 16% piensa de la misma forma pero amparado en el folleto como apoyo durante el examen.

Los estudiantes consideran que un examen escrito no es una estrategia para aprender como tal, pero sí al examen le combinamos un taller de estudio o un folleto, la preferencia aumenta en 35%, mientras que el 55% considera que estudiar un taller y luego hacer un examen en el computador es la mejor opción, aunque el computador gana como metodología. Para el estudiante es importante tener un material de apoyo, en este caso el taller de estudio, ya que solo el 10% considera que el examen en el computador sin taller sea buena metodología.

Esta respuesta es realmente satisfactoria, ya que los estudiantes prefieren usar un folleto antes que un libro: el 55% prefirió el folleto.

Discusión

Se puede entender la creación del folleto como una alcahuetería por parte del docente con el estudiante y, más para este caso, la idea de crear el folleto respondió al desinterés de algunos estudiantes de tener cuaderno y de realizar siempre las mismas preguntas que ya se habían respondido innumerables veces en la clase.

Sin embargo, en el desarrollo de esta fase de investigación, se pudo constatar que el problema se presenta en varios contextos con causas muy similares y la creación del folleto se valida en la creación de una hoja de trucos estándar para su uso en el aula. El estudiante requiere, además, de una guía para preparar un examen con ejercicios y teoría, que sea propuesta oficialmente por el docente, de tal forma que el material simplifique el contenido que será evaluado.

La información que traen los libros es extensa y presentan enfoques diferentes, que pueden distraer al estudiante de lo básico que debe aprender en el curso. Por tanto, se debe hacer un énfasis contante en el contenido y repetir las veces que sean necesarias, para garantizar que la comunicación entre el docente y el estudiante sea adecuada por medio de un proceso de retroalimentación constante y permita una validación de la apropiación de los conocimientos impartidos.

Aunque en las encuestas se evidenció que el estudiante quiere ser evaluado en el computador, no es la última palabra, porque se debe diseñar una prueba que permita realmente evaluar las capacidades que se desarrollan en el curso de fundamentos de programación para desarrollar soluciones a problemas.

Las certificaciones técnicas de lenguajes de programación, se presentan en un computador, pero hay un estricto control del tiempo, del acceso a internet o a cualquier tipo de material durante la prueba. De pronto, en esto no pensó el estudiante en el momento de incluir al computador en el proceso de evaluación. Finalmente, la creación del mínimo producto viable que es el folleto, fue muy efectiva, se cumplió con el producto y se pudo hacer la investigación. Esta es una ventaja que cuenta el proyecto y fue algo que no se evidenció en los proyectos consultados.

Conclusiones

Se recolectaron las siguientes sugerencias de los estudiantes, las cuales van a ser tenidas en cuenta para las siguientes fases del proyecto:

- Documentación en internet, ejercicios resueltos, ejemplos de clase, material práctico (No teoría).
- Presentaciones interactivas.
- Usaría una página que aplique y enseñe los conceptos básicos.
- Video tutorial.
- PDF. El folleto se pierde.

Estas son claves principalmente para el diseño del producto, y se hicieron recomendaciones para mejorar el uso del producto por parte de los estudiantes.

- El mínimo producto viable desarrollado se registró con ISBN y código de barras, para realizar alianzas de distribución con librerías y para tener datos de distribución en próximas fases del proyecto porque, además de ser una herramienta que sirva en la clase, debe ser un producto rentable.
- Se realizará difusión de esta investigación para ser punto de partida para proyectos similares y poder enriquecer otros proyectos sobre enseñanza de programación que están vigentes en el mundo actualmente.
- Se obtuvieron los datos para identificar factores importantes sobre la forma como le gusta aprender a esta generación de estudiantes, resaltándose la participación de las TIC en este proceso, específicamente en el uso de talleres, un folleto impreso y el computador, como herramientas de evaluación a considerar en una siguiente fase.
- Se considera para la siguiente fase, tener presente la evaluación de los conceptos básicos, teniendo en cuenta un enfoque de bajo nivel (Meza, Garrido, & Astudillo, 2003) pero en un curso previo al curso de fundamentos de programación, donde se le enseñará al estudiante los conceptos de lógica desde un ambiente electrónico básico. Esto se sale del contexto del proyecto, pero será clave revisar su pertinencia, ya que se identificó que una de las causas del problema son los saberes previos.

Referencias bibliográficas

- Partovi, Ali, & Partovi, H. (s.f.). *Code.org launched in 2013 as a bootstrapped project of co-founders Ali and Hadi Partovi*. Recuperado de: <https://code.org/>
- de Raadt, M. (2012). Student Created Cheat-Sheets in Examinations : Impact on Student Outcomes. *14th Australasian Computing Education Conference*, pp.71-76.
- Giangrande, J. E. (2007). Cs1 programming language options. *Language*, pp.153-160.
- International, E. (2015). *ECMAScript® 2015 Language Specification*. Recuperado de: <http://www.ecma-international.org/ecma-262/6.0/index.html>
- ISO. (06 de 2011). ISO/IEC® 16262:2011 Information technology -- Programming languages, their environments and system software interfaces -- *ECMAScript language specification*. Recuperado de: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=55755
- Mellon, C. (s.f.). *An Educational Software that teaches stundts computer programming in a 3d environment*. Recuperado de: <http://www.alice.org/index.php>
- Meza, F.; Garrido, R. y Astudillo, C. A. (2003). Un lenguaje de Bajo Nivel Como Apoyo al Aprendizaje en el Primer Curso de Programación de las Carreras de Ingeniería. *Revista de la Sociedad Chilena de Ciencia de la Computación*, 4.
- of Kent, U. (s.f.). *Blue J*. Recuperado de <http://www.bluej.org>
- Villalobos, J. (2007). *Tutorial: Cómo enseñar a programar: un enfoque efectivo*.
- Abstract:** There is a problem of low performance, student desertion and demotivation during the study of software programming, which is presented in different subjects of the engineering curriculum and particularly in the courses of programming fundamentals of the Engineering Degree in Entertainment Design Digital. In order to understand the student's learning needs, a diagnosis is made through surveys to students of Systems Engineering and Computer Science, Electronic Engineering, Design Engineering in Digital Entertainment and Graphic Designers at the Universidad Pontificia Bolivariana. The purpose of this is to be able to identify key factors in learning, from the point of view of engineers and designers. Additionally, a pilot test was made of a material created to facilitate the solution of written exams and the results are presented as an advance on what has been found so far.
- Keywords:** Programming - teaching - engineering - survey - graphic design - software.
- Resumo:** Existe um problema de baixo rendimento, deserção estudantil e falta de motivação durante o estudo da programação de software que se apresenta nas matérias do currículo de engenharia e em particular nos cursos de fundamentos de programação da carreira de Engenharia em Design de Entretenimento Digital. Para entender as necessidades de aprendizagem do estudante se faz um diagnóstico através de questionários a estudantes de Engenharia em Sistemas e Informática, Engenharia Eletrônica, em Design de Entretenimento Digital e a designers gráficos na Universidade Pontificia Bolivariana, com o propósito de identificar fatores chave na aprendizagem, desde o ponto de vista de engenheiros e designers. Adicionalmente, se fez uma prova piloto de um material criado para facilitar a solução de provas escritas e se apresentam os resultados como um avanço ao momento.
- Palavras chave:** programação - ensino - engenharia - questionário - design gráfico - software.
- (* **Andres Bedoya Tobón.** Ingeniero informático de la Universidad Pontificia Bolivariana (UPB), estudiante de maestría en TIC con énfasis en desarrollo de software y estudiante de diseño gráfico. Trabajó como desarrollador de proyectos de software para Pragma S.A y como docente. Actualmente es profesor interno en la Facultad de Ingeniería Informática de la UPB.