



UNIVERSIDAD
DE LA FRONTERA
VICERRECTORÍA DE PREGRADO

PROYECTOS DE DESARROLLO E INNOVACIÓN DOCENTE 2019



FONDO CONCURSABLE



Ozobot en el Aula: Didácticas para dispositivos programables como soporte a las asignaturas de programación de computadores.

Nombre Docente(s) Responsable	Docente Responsable: Jaime Díaz
Nombre docentes colaboradores:	Docentes Colaboradores: M. Dieguez, S. Sepúlveda, A. Cravero.
Departamento	Cs. de la Computación e Informática - DCI

I. RESUMEN

Los escenarios de la educación superior, buscan avanzar hacia procesos formativos pertinentes a las nuevas demandas. Como académicos, y especialistas en las Ciencias de la Computación, buscamos centrar nuestra docencia en el aprendizaje, utilizando lo mejor de las herramientas ya disponibles.

Uno de los problemas, es lo que sucede el curso de “*Programación de Computadores*”. Asignatura dictada para todas las carreras de Ingeniería Civil (400 estudiantes por semestre), en los primeros ciclos formativos. El curso en cuestión cuenta con altas tasas de deserción y un bajo nivel de motivación (Dieguez, 2016).

Nuestro proyecto desarrolló material específico para el uso de dispositivos programables (OZOBOTs), que mezclan intervenciones físicas con las digitales. Se aplicaron en tres módulos distintos, mezcladas con iniciativas de Pensamiento Computacional.

Dentro de los resultados y evaluaciones destacan: (i) Promedios de notas de los módulos, (ii) Porcentajes de aprobación y (iii) Tasas de abandono de los estudiantes.

II. INTRODUCCIÓN

La programación de computadores se considera actualmente como una habilidad fundamental para los nativos digitales, por lo que se considera como un área de estudio transversal (McLaughlin et al. 2014).

Diversas propuestas buscan introducir la programación de computadores durante los primeros ciclos formativos (Ricardo Gacitúa, Diéguez, and Vidal 2017) de una forma sistemática y basada en metodologías de enseñanza, no solo para proveer de conocimiento técnico a los estudiantes, sino también para apoyar el desarrollo de competencias cognitivas, tales como el desarrollo de habilidades de resolución de tareas, desarrollo del pensamiento lógico y trabajo colaborativo (Fábrega Lacoa, Fábrega Lacoa, and Blair 2016).

En este escenario varias propuestas metodológicas asociadas a la enseñanza de la programación han sido impulsadas (Medeiros, Ramalho, and Falcão 2019; Borges et al. 2018; Ricardo Gacitúa, Diéguez, and Vidal 2017). Todas ellas buscan potenciar el desarrollo tanto de las habilidades técnicas de programación como de las habilidades cognitivas para la resolución de problemas (Pensamiento computacional). Sin embargo, Scherer et al. (Scherer 2016), plantea que no existe evidencia empírica que demuestre el grado de desarrollo de estas habilidades en los estudiantes. Se debe agregar, además, que muchas de las habilidades de Pensamiento computacional actualmente reconocidas están asociadas con la resolución de problemas y el uso de dispositivos electrónicos (Grover and Pea 2013).

Esta iniciativa, presenta los resultados enmarcados en un marco de trabajo para la enseñanza de programación para las diferentes carreras de ingeniería de la Facultad de Ingeniería y Ciencias de la Universidad de La Frontera, Chile. Dicho marco de trabajo es flexible, pues se adapta a cada programa formativo e incluye el uso de dispositivos electrónicos programables, y sistemático, puesto que está basado en una metodología de enseñanza de programación (R\G Methodology (Ricardo Gacitúa, Diéguez, and Vidal 2017)) la cual ya ha sido probada con éxito en la formación de estudiantes de informática.

Dentro de los síntomas identificados preliminarmente, se encuentra: (i) Una importante tasa de reprobación (Dieguez, 2016). Se ha identificado un universo de 400 estudiantes por semestre académico, donde sólo aprueba un 55,91% de ellos. (ii) Una alta tasa de deserción del curso (Dieguez, 2016). De todo el universo de estudiantes en nivel para tomar el curso en cuestión, el 25% de ellos decide dejarlo. Y (iii) Poca motivación. Es complejo dar a conocer los beneficios de la programación a estudiantes que vienen con un prejuicio anterior. Es importante recordar que este curso es transversal para las carreras de Ingeniería Civil.

III. OBJETIVO

Objetivo General

- Generar didácticas para productos electrónicos programables que entregarán soporte a las asignatura - *Programación de Computadores* - para el desarrollo del pensamiento computacional.
 - Aprender a programar, trae beneficios en todo nivel, pero requiere de un esfuerzo adicional: Asimilar lenguajes y herramientas no siempre alineadas con aspectos “cotidianos”, donde además los avances son individuales y no instan los aspectos colaborativos ni de inclusión (Kong, Chiu, and Lai 2018; Bray and Tangney 2017; van Roy and Zaman 2018).

Objetivos Específicos

- Diseñar los aspectos metodológicos a ser aplicados en los materiales complementarios.
 - Diseño e implementación de las estrategias y/o metodologías para la mejora del aprendizaje a incorporar en los materiales complementarios.
 - Definición de las estrategias metodológicas para el diseño y seguimiento de las actividades de trabajo autónomo, utilizando la nueva estructura del campus virtual y sus potencialidades (como por ejemplo, la gamificación) (Martín-Gutiérrez et al. 2015).
- Desarrollar los recursos de apoyo.
 - Elaboración del material educativo, potenciadores del aprendizaje. En formato digital y físico, y que contarán con directa relación con los aprendizajes esperados en una actividad curricular.
- Aplicar la propuesta de solución.
 - El equipo investigador, realizará una aplicación piloto en uno de los módulos del curso de Programación de Computadores. La creación y evaluación de la misma propuesta, buscará dar sustento a la iniciativa.
- Evaluar y difundir la propuesta de solución.
 - Coordinación de una instancia de difusión interna de resultados en el DCI, para luego enviar los resultados finales de la investigación a un congreso nacional.

IV. MÉTODOS.

Para enfrentar la problemática descrita, se presenta un marco de trabajo para la enseñanza de programación para las diferentes carreras de ingeniería de la Facultad de Ingeniería y Ciencias de la Universidad de La Frontera. Dicho marco de trabajo es flexible, pues se adapta a cada programa formativo e incluye el uso de dispositivos electrónicos programables como parte de la enseñanza de programación, y es sistemático, puesto que está basado en una metodología de enseñanza de programación (R&G Methodology (Ricardo Gacitúa, Diéguez, and Vidal 2017)) la cual ya ha sido probada con éxito en la formación de estudiantes de informática.

La Fig. 1 ilustra el marco de trabajo, el cual considera dos áreas de trabajo: A) Común y B) Especializada.



Fig. 1. Marco de trabajo para la enseñanza de programación para las diferentes carreras de ingeniería, Universidad de La Frontera, Chile

Área común. La componente común se enfoca en el desarrollo de las habilidades cognitivas asociadas a la resolución de problemas, las cuales son transversales y aplicables a cualquier problema de ingeniería.

Para este componente se utiliza como base la metodología Rápida y Global (R&G) (Ricardo Gacitúa, Diéguez, and Vidal 2017), la cual fomenta la programación sistemática, centrada en seguir un proceso disciplinado para el diseño de soluciones.

Área especializada. Tomando en consideración la diversidad de perfiles y especialidades de los estudiantes de ingeniería, el marco de trabajo plantea la utilización de un conjunto de mecanismos y tecnologías diferenciadas, que permita adecuar la asignatura a los diferentes perfiles de los alumnos. Desde el punto de vista práctico, se contextualiza el desarrollo de las soluciones a problemas relacionados a la especialidad a través de la utilización del uso de dispositivos electrónicos programables, tales como **Ozobots**, Lego, Arduino, Robots y dispositivos móviles con fines didácticos y prácticos.

Por ejemplo, los Ozobots son pequeños robots inteligentes que permiten introducir a principiantes o a quienes recién se inician en la programación, así como el aprendizaje del lenguaje STEM.

Se trata de un dispositivo que los ayuda a programar y a conocer las consecuencias de la programación. Dentro de sus características, destacan sus capacidades auto-programables y el reconocimiento de patrones e instrucciones en papel.

Otros ejemplos son robots del tipo Lego, Arduinos entre otros, los cuales ya se están utilizando en cursos de programación.

La flexibilidad del marco de trabajo está dada por la combinación de adecuación y uso de estos componentes tecnológicos acorde al perfil de cada área de la ingeniería.

Por ejemplo, para los alumnos de ingeniería ambiental se pueden definir problemas de uso de sensores medioambientales usando tarjetas Arduinos. Para ingeniería en construcción se pueden realizar programas basados en lego, entre otros.

Dada la diversidad de elementos para complementar las clases teóricas, tales como juegos y dispositivos físicos programables, se permite ofrecer variedad de problemas y soluciones (que combinan mecanismos físicos y digitales) y que, además, poseen elementos complementarios para el aprendizaje de la programación, tales como desafíos incrementales, retroalimentación inmediata personalizada y visualización de las consecuencias de sus errores.

La introducción de este tipo de dispositivos, enlazada con los diseños obtenidos sistemáticamente mediante el uso de una metodología, ayudan a satisfacer la necesidad de los distintos perfiles profesionales de ingeniería, de contextualizar los problemas y las soluciones a la especialidad del estudiante, en un ambiente práctico, y contextualizado a diversos dominios de ingeniería. Además, introduce al estudiante en el conocimiento y manejo de nuevas tecnologías (componentes electrónicos programables), dándole una nueva perspectiva respecto del tipo de problemas que se pueden abordar con la programación.

El marco de trabajo, adicionalmente, provee de un conjunto de ejemplos y problemas adecuado a cada perfil del área de ingeniería. Uno de los aspectos relevantes a considerar es que todas las actividades propician el trabajo colaborativo, por sobre el trabajo individual.

Respuesta a Objetivo General

- *Generar didácticas para productos electrónicos programables que entregarán soporte a las asignatura - Programación de Computadores - para el desarrollo del pensamiento computacional.*

Por cada sesión de trabajo con el equipo interno, se generan minutas de avance y acuerdo. Por cada una de ellas, se discuten los alcances y resultados de aprendizaje que pretenden ser reforzados.

Finalmente, se generaron tres intervenciones con los dispositivos

- **Actividades lúdicas de reconocimiento.** (i) Reconocimiento de dispositivos programables, (ii) capacidades y funciones base, (iii) historias lúdicas y aplicación
- **Estructuras de Control.** (i) Introducción a condiciones IF - ELSE - FOR - WHILE. (ii) Resolución de problemas tipo.
- **Cierre de módulo - Evaluación Integral.** (i) Evaluación y aplicación de un test evaluativo.

Entregables generados

A continuación, escenarios y documentación generada a modo de ejemplo (ver Fig 2, Fig. 3).

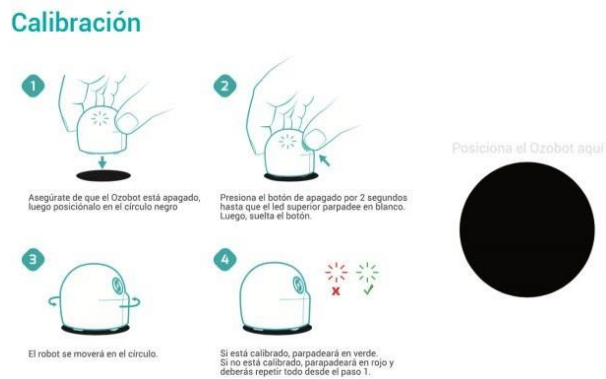


Fig. 2. Documentación tipo instruccional - Utilización OZOBOT

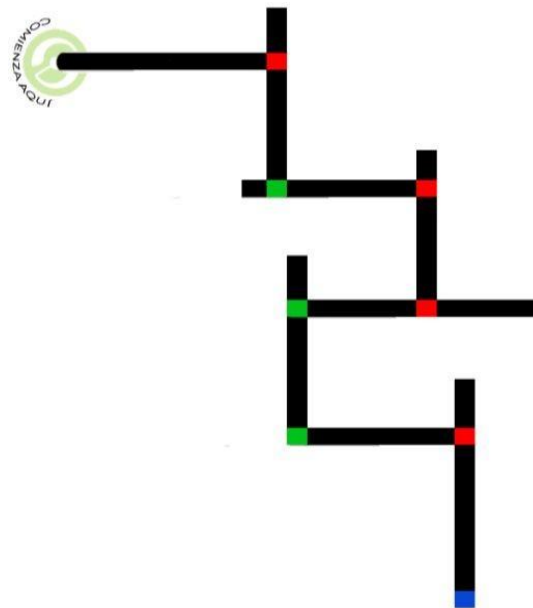


Fig. 3. Problema original de resolución de problemas mediante estructuras de Control.

V. RESULTADOS

A fin de evaluar la propuesta se implementó una versión piloto del marco de trabajo en tres módulos del curso de programación, lo cual significó un grupo de 69 estudiantes considerados. Se utilizó un grupo de control compuesto por dos módulos (46 estudiantes).

Este grupo trabajó de manera tradicional (centrado en el producto - código) con los mismos profesores que imparten los módulos intervenidos. Por consideraciones administrativas, se decidió realizar una implementación parcial del marco de trabajo, es decir, se implementó la componente común, con el uso de la metodología R&G, y parte de la componente especializada, interviniendo solo algunas carreras y algunos temas de la asignatura, dado que actualmente las carreras de ingeniería están en un proceso de rediseño instruccional.

De manera de medir el impacto de la intervención, se compararon los resultados académicos de ambos grupos y también las respuestas de los estudiantes a una encuesta de percepción y evaluación.

Los indicadores que se estudiaron a partir de los resultados académicos son: (i) Promedios de notas de los módulos, (ii) Porcentajes de aprobación y (iii) Tasas de abandono de los estudiantes.

Además de lo anterior, se analizaron los resultados de las encuestas, a partir de las opiniones positivas y las opiniones negativas de los estudiantes de ambos módulos en relación al desarrollo del curso.

Resultados Académicos

- *Promedios de notas de los módulos.* En este ítem se observa que no existe una gran diferencia en el promedio final de ambos grupos, (4.2 - con metodología vs 4.1 sin metodología) muestra que el promedio del grupo intervenido es superior al grupo de control por solo una décima. Se observa además que, en ambos casos, los promedios están por sobre la nota mínima de aprobación del curso (i.e. 4.0). En este caso, la diferencia en los promedios es mínima.
- *Tasas de aprobación.* De los resultados presentados en la Fig. 4 se aprecia un impacto positivo en las tasas de aprobación. La figura muestra que el 58% de los estudiantes que utilizaron el marco de trabajo finalizó el curso con un promedio superior al mínimo de aprobación, lo cual está por sobre el promedio histórico de la asignatura. Por su parte, el grupo de control que trabajó con el método tradicional presentó un porcentaje de aprobación de un 43%, lo cual está muy por debajo del promedio, que es de un 55%.
En este caso existe una marcada diferencia en los porcentajes de aprobación entre ambos grupos. Esto se puede explicar debido a las diferencias que existen entre las notas obtenidas por los estudiantes en el grupo de control.

En efecto, varios estudiantes obtuvieron notas sobresalientes, muy superiores al resto de los estudiantes. Sin duda, esta situación eleva el promedio final del grupo. Por su parte, el grupo intervenido presenta notas más homogéneas entre los estudiantes.

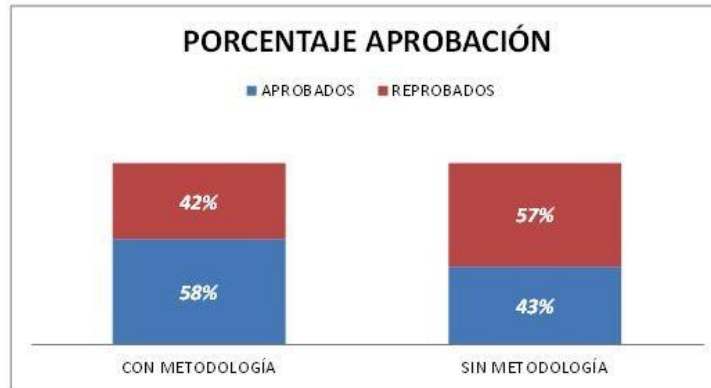


Fig. 4 Tasas de aprobación

- Tasas de abandono de los estudiantes.* En lo que se refiere a la deserción de los estudiantes en ambos grupos, el grupo que utilizó el marco de trabajo presentó una tasa de abandono de un 17%, valor bastante menor que las tasas históricas del curso, que es de un 47%, tal como lo muestra la Fig. 5. En el caso del grupo de control, este índice fue de un 43%, valor que se encuentra dentro de los rangos históricos. Esto evidencia que la nueva forma de trabajo ha aumentado la motivación y disminuido la frustración de los estudiantes, lo que les permitió mantenerse en el curso hasta el final del semestre.



Fig. 5 Tasas de deserción

VI. CONCLUSIONES

Esta propuesta presenta un marco de trabajo para la enseñanza de programación para las diferentes carreras de ingeniería de la Facultad de Ingeniería y Ciencias de la Universidad de La Frontera. Dicho marco de trabajo es flexible, pues se adapta a cada programa formativo e incluye el uso de dispositivos electrónicos programables como parte de la enseñanza de programación, y es sistemático, puesto que está basado en la metodología R&G, la cual ya ha sido probada con éxito en la formación de estudiantes de informática.

De los resultados de la evaluación, cuatro aspectos son relevantes de destacar:

- *Los promedios de los grupos observados presentan una mínima diferencia, siendo superior la del grupo intervenido por una décima de nota.*
- *La tasa de aprobación fue más alta en los estudiantes que fueron enseñados utilizando el marco de trabajo.*
- *La tasa de deserción de los estudiantes se redujo con el uso marco de trabajo, en contraste con aquellos estudiantes que no fueron enseñados con el marco de trabajo.*
- *Las opiniones de los estudiantes difieren ampliamente en ambos grupos, pues se aprecia que el centro atención y de análisis es diferente. Las opiniones de aquellos estudiantes que usaron el marco de trabajo se centraron principalmente en aspectos relacionados con las metodologías, los problemas y sus soluciones, tales como por ejemplo aspectos de notación, facilidad de uso, documentación, entre otros. En cambio, aquellos estudiantes que no fueron enseñados con el marco de trabajo, centraron sus opiniones en las características personales del profesor, tales como simpatía, puntualidad, entre otros aspectos, lo que claramente indica que los aspectos de impacto para los estudiantes son la personalidad y competencias del profesor. Este resultado implica que el marco de trabajo logra abstraerse del docente, dejando de ser éste el centro del curso. Por otra parte, indica además que el marco de trabajo logra estandarizar una forma de enseñanza independiente del docente.*

Los desafíos futuros que se han de enfrentar dicen relación con:

- *Utilizar el marco de trabajo para la enseñanza de programación común para todos los alumnos de ingeniería, independiente de su área de especialidad.*
- *Regirse por un programa estandarizado que puede flexibilizarse en el ámbito de dominio de los problemas y/o las tecnologías prácticas que se utilizan. De este modo, todos los profesores de programación utilizarán los mismos enfoques metodológicos y herramientas, variando sólo el nivel de ejemplos y dominio de aplicación.*
- *Utilizar una plataforma común de soporte al enfoque metodológico, lo cual facilitará la introducción y uso del marco de trabajo.*

VII. REFERENCIAS

- Borges, R. P., P. R. F. Oliveira, R. G. da R. Lima, and R. W. de Lima. 2018. "A Systematic Review of Literature on Methodologies, Practices, and Tools for Programming Teaching." *IEEE Latin America Transactions* 16 (5): 1468–75. <https://doi.org/10.1109/TLA.2018.8408443>.
- Bray, Aibhín, and Brendan Tangney. 2017. "Technology Usage in Mathematics Education Research
A Systematic Review of Recent Trends." *Computers & Education*. <https://doi.org/10.1016/j.compedu.2017.07.004>.
- Dieguez, "Resultado diagnóstico asignatura ICC202," Temuco, 2016.
- Fábrega Lacoa, Rodrigo, Jorge Fábrega Lacoa, and Alissa Blair. 2016. "La Enseñanza de Lenguajes de Programación En La Escuela: ¿ Por Qué Hay Que Prestarle Atención?" <http://repositorio.minedu.gob.pe/handle/MINEDU/6019>.
- Grover, Shuchi, and Roy Pea. 2013. "Computational Thinking in K–12: A Review of the State of the Field." *Educational Researcher* 42 (1): 38–43. <https://doi.org/10.3102/0013189X12463051>.
- Kong, Siu-Cheung, Ming Ming Chiu, and Ming Lai. 2018. "A Study of Primary School Students' Interest, Collaboration Attitude, and Programming Empowerment in Computational Thinking Education." *Computers & Education* 127 (December): 178–89. <https://doi.org/10.1016/j.compedu.2018.08.026>.
- Martín-Gutiérrez, Jorge, Peña Fabiani, Wanda Benesova, María Dolores Meneses, and Carlos E. Mora. 2015. "Augmented Reality to Promote Collaborative and Autonomous Learning in Higher Education." *Computers in Human Behavior* 51 (October): 752–61. <https://doi.org/10.1016/j.chb.2014.11.093>.
- McLaughlin, Stephen, Martin Sherry, Eileen Doherty, Marian Carcary, Clare Thornley, Yi Wang, Marianne Kolding, et al. 2014. "E-Skills: The International Dimension and the Impact of Globalisation - Final Report 2014." <http://eprints.maynoothuniversity.ie/5559/>.
- Medeiros, R. P., G. L. Ramalho, and T. P. Falcão. 2019. "A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education." *IEEE Transactions on Education* 62 (2): 77–90. <https://doi.org/10.1109/TE.2018.2864133>.
- Ricardo Gacitúa, B., M. Diéguez, and E. Vidal. 2017. "Forming Software Architects in Early Stages: From Craft to Engineering." In *2017 36th International Conference of the Chilean Computer Science Society (SCCC)*, 1–8. <https://doi.org/10.1109/SCCC.2017.8405130>.
- Roy, Rob van, and Bieke Zaman. 2018. "Need-Supporting Gamification in Education: An Assessment of Motivational Effects over Time." *Computers & Education* 127 (December): 283–97. <https://doi.org/10.1016/j.compedu.2018.08.018>.
- Scherer, Ronny. 2016. "Learning from the Past—The Need for Empirical Evidence on the Transfer Effects of Computer Programming Skills." *Frontiers in Psychology* 7 (September): 48. <https://doi.org/10.3389/fpsyg.2016.01390>.