

Metodologías activas para la enseñanza de programación a estudiantes de ingeniería civil informática

Eduardo A. Jones^{1*}, Claudia A. Jimenez¹, Pablo I. Ormeño¹ y Natalia A. Poblete²

(1) Universidad Viña del Mar, Ingeniería Civil Informática, Escuela Ingeniería y Negocios, Viña del Mar, Chile (correo-e: ejones@uvm.cl, cjimenez@uvm.cl, pablo.ormeno@uvm.cl)

(2) Instituto Profesional AIEP, Viña del Mar, Chile (correo-e: natalia.pobleteba@correaiep.cl)

* Autor a quien debe ser dirigida la correspondencia.

Recibido Nov. 24, 2021; Aceptado Ene. 18, 2022; Versión final Mar. 17, 2022, Publicado Jun. 2022

Resumen

El objetivo de este estudio es monitorear el impacto de la aplicación de metodologías activas de enseñanza para programación en la carrera de Ingeniería Civil Informática de la Universidad Viña del Mar (Chile). Los cursos de programación se encuentran actualmente en casi todos los primeros niveles de las carreras de ingeniería, especialmente los relacionados con informática. Se aplica un cuestionario para diagnosticar la relación de los estudiantes con los contenidos, al inicio y al término del semestre, realizando preguntas relacionadas con los primeros niveles de la taxonomía de Bloom. Los resultados muestran que la aplicación de metodologías activas mejora la tasa de aprobación de la asignatura y mejora las habilidades de recordar y comprender de los estudiantes. Sin embargo, no hay evidencia que demuestre si en la habilidad de aplicación hay algún impacto. Se concluye que las metodologías activas funcionan cuando son aplicadas en cursos de programación de computadoras.

Palabras clave: programación; metodologías activas; taxonomía; Bloom; enseñanza universitaria

Active methodologies for teaching programming to computer science students

Abstract

The main objective of this study is to assess the impact on students of applying active teaching methodologies for programming in the Ingeniería Civil Informática degree program at the Universidad Viña del Mar (Chile). First year programming courses are found in nearly all engineering degree programs, especially in computer science. A questionnaire is applied to assess the relationship between students and course contents, at the beginning and at the end of the semester, by asking questions related to the first levels of Bloom's taxonomy. The results show that the application of active methodologies improves both course pass rates and the students' abilities of remembering and understanding. However, there is no evidence to suggest that the ability to apply concepts is affected by active methodologies. It is concluded that active methodologies are effective when applied for teaching programming courses.

Keywords: programming; active methodologies; Bloom's taxonomy; university teaching

INTRODUCCIÓN

Dentro del trabajo de programación hay conceptos relevantes y más aún para esta investigación. Como, por ejemplo, se tiene el de algoritmo, que es un grupo finito de operaciones organizadas de manera ordenada y lógica que permiten solucionar un problema determinado (RAE, 2020). Es decir, se puede entender como una serie de instrucciones o reglas preestablecidas que permiten llegar a un resultado o a una solución, por ejemplo, al Algoritmo de Euclides, que sirve para calcular el máximo común divisor (Coutinho, 2009). El Pensamiento Algorítmico permite que un problema de la vida real sea transformado en Algoritmo o Programa (algoritmo escrito en un Lenguaje de Programación), es decir, una secuencia finita y determinista de pasos (Joyanes, 2003). Esto implica la capacidad de definir y enunciar con claridad un problema, pasando por su identificación y su modelado para que, de esta manera, este se pueda dividir en subproblemas más pequeños y manejables. Finalmente, se puede describir la solución a este problema. Todo esto, representa un gran uso de habilidades y también un gran problema para los estudiantes en general.

Lo anterior se ve reflejado al implementar nuevos planes de estudio, en los cuales se han introducido cursos de Programación en el primer año universitario (Prokofyeva et al., 2016). Como parte de las nuevas asignaturas, se exigen competencias que estén relacionadas con el aprendizaje de algún lenguaje de programación. Sin embargo, aprender a programar no es un ejercicio fácil, más aún si se tiene en cuenta que se involucran niveles de organización y sistematización de procesos cognitivos ligados a la organización y atención, así como también un alto grado de creatividad a la hora de resolver un problema. Es por ello que los estudiantes que se inician en programación suelen encontrar dificultades para aprender los conceptos básicos y el uso del lenguaje propio de la disciplina. Pero más aún, se visualiza mayor problemática para desarrollar el razonamiento algorítmico. A todo lo anterior se le suma la dificultad de resolver un problema usando un lenguaje de programación.

Estas dificultades pueden suponer un grave inconveniente en estudios de Ingeniería Informática, ya que mientras en otras especialidades el aprendizaje de un lenguaje de programación es útil, pero complementario, en los estudios de informática es fundamental, ya que, si los estudiantes no alcanzan las competencias exigidas en la asignatura inicial, su rendimiento a lo largo de los cursos puede verse seriamente afectado. La dificultad de aprendizaje de programación no es un problema exclusivo de los estudiantes de Chile, sino un problema a nivel mundial (Cheah, 2020). Además, si a la dificultad inherente de la programación se agrega que los estudiantes llegan con carencias formativas y con poca motivación, el problema crece. Las tasas de reprobación a la hora de analizar las asignaturas de programación terminan por disminuir la posibilidad de éxito de los estudiantes en la finalización de sus estudios y produce altas tasas de abandono (Juárez et al., 2016). Así también, otro factor relevante es que los nuevos planes de estudio traen nuevos requerimientos para la didáctica, tales como, el trabajo autónomo de los estudiantes, la evaluación mediante competencias y las Metodologías Activas de Aprendizaje (Silva y Maturana, 2017), además de agregar un componente lúdico que permita motivar a los estudiantes. Con esto se busca aumentar la motivación e involucramiento de los estudiantes con el objetivo de mejorar su rendimiento académico.

Es por todo lo anterior que en el siguiente artículo se presenta un estudio de la aplicación de Metodologías Activas en la asignatura de Programación de Computadores, de la carrera de Ingeniería Civil Informática de la Universidad Viña del Mar. Este trabajo es uno de los resultados asociados al proyecto “*Estudio de Metodologías y Herramientas para la enseñanza de Programación en alumnos de Ingeniería Civil Informática*”, financiado por Fondos de Investigación 2018 de la Universidad Viña del Mar. Esta investigación se realizó mediante un cuestionario, que da a conocer la mirada de los estudiantes respecto a temas técnicos relacionados con la disciplina. Luego, con los resultados iniciales obtenidos, se aplicaron metodologías activas de enseñanza en la asignatura (Trazar Código, Programación por Pares, Aprendizaje Basado en Problemas y Aprendizaje Basado en Proyectos (Ausín et al., 2016)). Para luego repetir las mismas preguntas al final del curso, aplicando el cuestionario y se compararon los resultados. Esto permitió evaluar la efectividad de las metodologías activas utilizadas. Se midió además el nivel de aprendizaje en el que están los estudiantes, mediante la Taxonomía de Bloom, para medir el grado de avance que obtienen dentro de la asignatura. La evaluación de la efectividad de los métodos se contrastó con el rendimiento académico y las tasas de reprobación y un análisis de la calidad de las respuestas.

OTROS ANTECEDENTES

En esta sección se presentan algunos conceptos teóricos que sustentan este estudio, como la Taxonomía de Bloom y las Metodologías Activas que son utilizadas en el estudio.

Taxonomía de Bloom

La taxonomía de Bloom es una herramienta que ha sido ampliamente utilizada para definir objetivos de aprendizaje en cursos de ciencias de la computación (Sobral, 2021). Esta taxonomía representa el proceso

de aprendizaje. Fue desarrollada en 1956 por Benjamín Bloom. La taxonomía enfatiza lo que un alumno puede hacer y cada una de sus etapas se conforma por un verbo. La lógica básica de la taxonomía radica en el hecho de que todo concepto para ser comprendido debe ser recordado, para poder ser aplicado debe ser comprendido, para poder analizarlo debe ser aplicado, para poder ser evaluado debe ser analizado y para poder crearlo debe ser recordado, comprendido, aplicado, analizado y evaluado. En la Figura 1 se pueden apreciar las etapas de la taxonomía.



Fig. 1: Niveles de la Taxonomía de Bloom

El dominio cognitivo dentro de esta taxonomía está diseñado para verificar el nivel cognitivo de un estudiante durante un examen. El desafío está en analizar si estas preguntas cumplen con los requisitos de la taxonomía de Bloom en los diferentes niveles cognitivos (Omar et al., 2012).

Metodologías Activas

Dentro de la literatura existen muchas Metodologías Activas que pueden aplicarse a los estudiantes de programación (Namratha et al., 2018) y que son usadas en las universidades para la enseñanza de la disciplina. Aprender a programar es una tarea de gran dificultad y es clave para lograr el éxito en los estudios de informática. Programar conlleva lograr dos objetivos. Primero es necesario que los estudiantes recuerden y comprendan los diferentes conceptos y estructuras básicas relacionadas con los algoritmos. El segundo objetivo es más complejo, ya que consiste en la aplicación y resolución de problemas de programación. Si bien ambos objetivos están ligados, uno no implica el otro. Además, se pueden aprender las estructuras y el uso de ellas, pero esto no implica que se pueda resolver problemas usándolas. También es posible que los estudiantes puedan determinar los procedimientos para resolver el problema, pero que no sean capaces de expresarlo como solución. Como Metodologías Activas se entiende a la enseñanza centrada en el estudiante, es su capacitación en competencias propias del saber de su disciplina. Estas estrategias conciben el aprendizaje como un proceso constructivo y no receptivo (Silva y Maturana, 2017). Específicamente, dentro de la disciplina de Programación se encuentran las siguientes metodologías Activas:

Programación por Pares

Consiste en asignar la resolución de un problema a dos estudiantes. La actividad se inicia con ambas partes acordando una solución, mientras un integrante de la pareja codifica la solución, el otro va revisando el código creado. Cuando pasa cierto tiempo decidido por el profesor, los roles se intercambian. El proceso se repite varias veces, hasta que se termine. O hasta que se haya encontrado la solución. Esta actividad permite aumentar la confianza y la capacidad de resolver problemas para los estudiantes menos experimentados, así como mejorar la retención. (Hagedorn, 1999)

Ejercicios de Trazas de Código a Mano

Habitualmente en los primeros cursos de programación se hace mucho énfasis en el desarrollo de programas a través de la lectura y traza de códigos. Esta es una de las partes más importantes y que consumen bastante tiempo para desarrollar un programa, a través de su depuración y la detección de fallas. Además, las personas que trabajen profesionalmente desarrollando programas, deberán tener la competencia con lectura y comprensión de código desarrollado por otros. Esta metodología se utiliza para poder evaluar de forma más adecuada las competencias obtenidas en lectura y trazas de código. Se combina con la resolución de problemas usando papel y lápiz, codificando con estas herramientas y determinando la solución del problema. Se puede utilizar con todos los tipos de componentes algorítmicos. (Öqvist y Nouri, 2018)

Aprendizaje Basado en Proyectos (ABP)

Es una metodología activa que se centra en el estudiante como generador de su aprendizaje, mediante el desarrollo de un proyecto que genera un producto que soluciona un problema real, donde el profesor asume un rol de facilitador de aprendizaje, respondiendo las preguntas y asesorando a los estudiantes (Torres et al., 2016). Existen múltiples definiciones de ABP. Educación 2020 lo define como “un método pedagógico que involucra a los estudiantes de una manera activa en su aprendizaje al pedirles que investiguen la respuesta a alguna pregunta o problema del mundo real, para luego crear una solución concreta. En todo este proceso, la comunidad escolar reflexiona sobre qué, cómo y por qué están aprendiendo. Finalmente, los estudiantes presentan sus proyectos a una audiencia mayor” (Educ2020, 2021), es decir, el objetivo final es la creación de la solución al problema planteado por el profesor.

Por último, el Ministerio de Educación, plantea que “el Aprendizaje Basado en Proyectos promueve que los alumnos se organicen, durante un periodo extendido de tiempo, en torno a un objetivo basado en una pregunta compleja, problema, desafío o necesidad, normalmente surgidas desde sus propias inquietudes, que pueden abordar desde diferentes perspectivas y áreas del conocimiento, fomentando la interdisciplinariedad. El proyecto culmina con la elaboración de un producto o con la presentación pública de los resultados”, (MinEduc, 2019), es decir, resume las definiciones anteriores, indicando que el proyecto termina con la generación de un producto, luego de un trabajo desarrollado en forma organizada por el equipo de proyecto, en un periodo extendido de tiempo. (Guo et al., 2020)

Aprendizaje Basado en Problemas (ABPro)

Es un método de enseñanza – aprendizaje centrado en el estudiante en el que estos adquieren conocimiento, habilidades y actitudes a través de situaciones de la vida real (Rodríguez y Fernández-Batanero, 2017). En esta metodología se pretende que el estudiante construya su conocimiento sobre la base de problemas y situaciones de la vida real y que, además, lo haga con el mismo proceso de razonamiento que utilizará cuando sea profesional (Ortiz, 2019), es decir, valorando e integrando el saber que los conducirá a la adquisición de competencias profesionales. La característica más innovadora del ABPro es el uso de los problemas como punto de partida para la adquisición de conocimientos nuevos y la concepción del estudiante como protagonista de la gestión de su aprendizaje. (Yew y Goh, 2016)

METODOLOGÍA

Idealmente, se busca motivar a los estudiantes con estas metodologías. La idea es realizar diferentes actividades de aprendizaje asociadas a las metodologías activas y con esto facilitar la difícil tarea de aprender a programar. Se busca alternar dichas actividades a lo largo del semestre y englobar todo en un proceso continuo de enseñanza – aprendizaje, donde los estudiantes van adquiriendo los conocimientos. La experiencia se llevó a cabo en la asignatura de Programación de Computadores de la carrera de Ingeniería Civil Informática de la Universidad Viña del Mar de Chile, el primer semestre (otoño) del 2019. En los semestres anteriores, los profesores notaron que los estudiantes, aun aprobando la asignatura, tenían ciertas deficiencias en la comprensión y aplicación de los contenidos. Por otro lado, las tasas de reprobación de la asignatura eran demasiado altas durante dichos años. En estos años, existieron esfuerzos puntuales en la aplicación de metodologías activas, sin planificación ni seguimiento de su uso. Además, la selección de las metodologías presentaba el principal desafío para el docente.

Con el objetivo de mejorar esta situación, se optó por la aplicación de un modelo compuesto de programación por pares, seguimiento de trazas de código de manera manual y aprendizajes basados en problemas y en proyectos, metodologías utilizadas a lo largo de todo el semestre. Como parte de la experiencia, se desarrollaron diversas actividades y materiales preparados por los docentes, que sirvieron de apoyo a estas metodologías activas introducidas en el aula, así como la creación de instrumentos de evaluación necesarios para medir el avance de los estudiantes. Concretamente se desarrollaron las siguientes actividades y materiales:

Para trabajo autónomo: Se preparó una serie de ejercicios resueltos paso a paso, con problemas de programación, para cada uno de los contenidos de la asignatura. Se propusieron problemas sin resolver con el objetivo de que los estudiantes se vieran enfrentados a este tipo de situaciones.

Para las clases teóricas: Conjunto de actividades entre pares con instrucciones dadas por el profesor a los estudiantes, junto a sugerencias sobre la dinámica a establecer.

Para las clases prácticas: Conjunto de ejercicios para realizar con programación en parejas. Junto a los ejercicios se desarrolla un manual para los profesores, con instrucciones acerca de cómo organizar y gestionar los grupos y la evaluación.

Para evaluación de la percepción: Se aplicaron dos cuestionarios, uno de diagnóstico inicial y el otro a la salida de la asignatura. Catorce (14) estudiantes de primer año de Ingeniería Civil Informática respondieron los dos cuestionarios. La encuesta fue diseñada bajo el enfoque Diseño Descriptivo Transversal (Díaz de Rada, 2007) y el detalle de las preguntas se muestra en la Tabla 1.

Este cuestionario tuvo como objetivo medir conocimientos generales de Programación. Las preguntas fueron divididas en 3 tipos siguiendo la taxonomía de Bloom (Gil et al., 2018). En las preguntas de la 1 a la 6 se refuerza el recordar conceptos, sin necesidad de un entendimiento muy profundo, identificando conceptos básicos. Las preguntas de la 7 a las 10 se enfocan en la comprensión de los conceptos mediante la ejemplificación. Finalmente, las preguntas 11 y 12 buscan aplicar contenido a un contexto definido. Esta división se muestra en la Tabla 2.

Tabla 1: Preguntas de conocimientos generales

1	¿Qué entiendes por programación de computadores?
2	¿Sabes lo que es un algoritmo?
3	¿Qué es una variable?
4	Defina arreglo
5	Describa una Estructura de datos
6	¿Qué es un Bubble Sort?
7	Mencione 3 tipos de datos primitivos en algún lenguaje
8	Mencione un tipo de dato que permita trabajar decimales
9	Menciona 2 estructuras de selección
10	Menciona 2 estructuras de repetición (bucles)
11	¿En qué contexto es recomendable utilizar for y while?
12	Si necesitamos detener una secuencia de una estructura ¿qué palabra utilizaremos

Tabla 2: Clasificación de preguntas del cuestionario

Rango de Preguntas	Taxonomía de Bloom	Descripción
1 a 6	Recordar	Identificación de conceptos básicos
7 a 10	Comprender	Ejemplificación de conceptos
11 a 12	Aplicar	Aplicación en un contexto definido

RESULTADOS Y DISCUSIÓN

La asignatura es calificada por medio de distintas evaluaciones teóricas y prácticas. Adicional a ello, se aplicó el cuestionario en dos oportunidades, al inicio y al término del semestre, para lograr determinar la adquisición de conceptos básicos de programación por parte de los estudiantes. Las respuestas fueron evaluadas como “correctas” o “incorrectas” por los docentes. A continuación, se muestra un conjunto de tablas con el análisis de los resultados de la aplicación de los cuestionarios. A partir del Test de Hipótesis se puede afirmar, con significancia del 95%, que los promedios del porcentaje de mejora de las diferencias entre las respuestas iniciales y finales son aceptables.

Preguntas Nivel Recordar

En la Tabla 3 se puede observar el desempeño de los estudiantes en el primer nivel, es decir, cuanto recuerdan de lo que les fue enseñado. Es posible apreciar que los conceptos básicos se identifican, ya que más de un 70% de los estudiantes respondió correctamente a la pregunta realizada al final de la asignatura, con un promedio de 66,7% de mejora. De acuerdo con la Taxonomía de Bloom, el estudiante es capaz de recordar conceptos de programación que son específicos y que son conocidos por todos. De esta forma el estudiante puede repetir conceptos teóricos en sus formas más básicas.

Tabla 3: Respuesta conocimiento general Nivel Recordar

	Inicio		Término		Diferencia	
	Incorrectos	Correctos	Incorrectos	Correctos	Correctos (Término - Inicio)	Término (Correctos - Incorrectos)
1. ¿Qué entiendes por programación de computadores?	64%	36%	29%	71%	35%	42%
2. ¿Sabes lo que es un algoritmo?	86%	14%	14%	86%	72%	72%
3. ¿Qué es una variable?	100%	0%	29%	71%	71%	42%
4. Defina Arreglo	100%	0%	21%	79%	79%	58%
5. Describa una estructura de datos	100%	0%	36%	64%	64%	28%
6. ¿Qué es un Bubblesort?	100%	0%	21%	79%	79%	58%
Promedio	91,7%	8,3%	25,0%	75,0%	66,7%	91,7%
Desviación Estándar	14,7%	14,7%	7,8%	7,8%	16,5%	14,7%

Preguntas Nivel Comprender

En la Tabla 4 se pueden ver las respuestas dadas por los estudiantes cuando se les pedía relacionar conceptos y demostrar comprensión. En este sentido se necesita que el estudiante interprete, traduzca o extrapole lo que se le está comunicando o que diga lo que sucede con relación a un hecho con el conocimiento previamente adquirido. Se puede apreciar que las respuestas fueron muy satisfactorias, pues casi un 80% de los estudiantes comprende los conceptos de programación relacionados con la pregunta, con un promedio de 87,8% de mejora.

Tabla 4: Respuesta conocimiento general Nivel Comprender

	Inicio		Término		Diferencia	
	Incorrectos	Correctos	Incorrectos	Correctos	Correctos (Término - Inicio)	Término (Correctos - Incorrectos)
7. Mencione 3 tipos de datos primitivos	100%	0%	14%	86%	86%	72%
8. Mencione un tipo de dato que permita trabajar con decimales	93%	7%	14%	86%	79%	72%
9. Menciona dos estructuras de selección	100%	0%	7%	93%	93%	86%
10. Menciona 2 estructuras de repetición (bucle)	93%	7%	0%	100%	93%	100%
Promedio	96,5%	3,5%	8,8%	91,3%	87,8%	96,5%
Desviación Estándar	4,0%	4,0%	6,7%	6,7%	6,7%	4,0%

Preguntas Nivel Aplicar

En la Tabla 5 se pueden observar las respuestas relacionadas con la aplicación de los conocimientos. En relación con la taxonomía de Bloom, en este nivel se requiere que el estudiante haga uso de la información que conoce y comprende (conceptos, métodos, teoría) para dar soluciones a los problemas. En este sentido, los resultados no fueron muy satisfactorios, obteniéndose un promedio de 54% de mejora. El análisis de las respuestas no sólo se realizó por medio de los resultados numéricos, sino que también se realizó un análisis de nubes de palabras en las respuestas. Este análisis contempla que las palabras más repetidas tienen mayor tamaño en la nube. En la Figura 2 se pueden ver las nubes de palabras sobre las respuestas respecto a algunos de los conceptos recordados. Aquí se observa que los estudiantes recuerdan los conceptos básicos, ya que las palabras más repetidas son adecuadas.

Tabla 5: Respuesta conocimiento general Nivel Aplicación

	Inicio		Término		Diferencia	
	Incorrectos	Correctos	Incorrectos	Correctos	Correctos (Término - Inicio)	Término (Correctos - Incorrectos)
11. ¿En qué contexto es recomendable utilizar for y while?	93%	7%	21%	79%	72%	58%
12. Si necesitamos detener una secuencia de una estructura, ¿qué palabra utilizamos?	100%	0%	64%	36%	36%	-28%
Promedio	96,5%	3,5%	42,5%	57,5%	54,0%	96,5%
Desviación Estándar	4,9%	4,9%	30,4%	30,4%	25,5%	4,9%

En la Figura 3 se pueden ver las nubes de palabras sobre las respuestas respecto a algunos de los conceptos comprendidos. Aquí se observa que los estudiantes comprenden los conceptos, ya que las palabras más repetidas son correctas. En la Figura 4 se pueden ver las nubes de palabras sobre las respuestas respecto a algunos de los conceptos aplicados. Aquí se observa que los estudiantes tienen problemas de aplicación, pues entre las palabras más repetidas se incluyen conceptos no adecuados. Finalmente, en la Tabla 6 se puede ver el porcentaje de Aprobación – Reprobación de la Asignatura el Año en que se aplicó este estudio en comparación a los años anteriores.

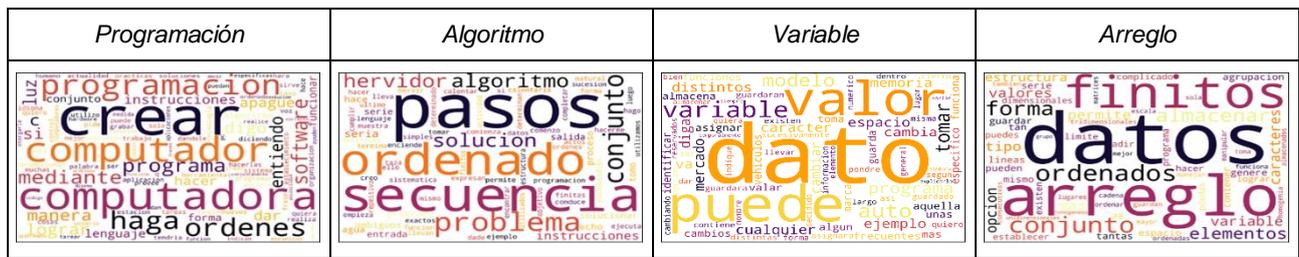


Fig. 2: Nube de Palabras para conceptos Nivel Recordar

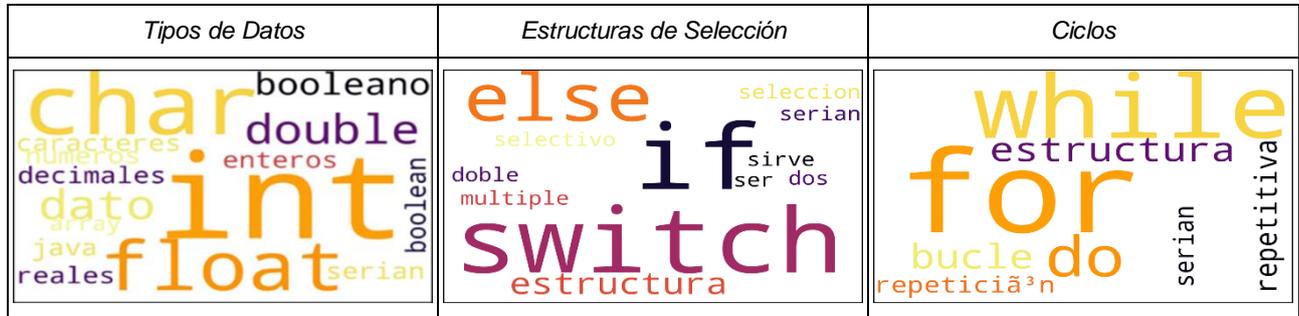


Fig. 3: Nube de Palabras para conceptos Nivel Comprensión



Fig. 4: Nube de Palabras para conceptos Nivel Aplicación

Al igual como lo indican Berssanette y de Francisco (2021), el uso de metodologías activas contribuye significativamente en el logro de los resultados de aprendizaje de programación, demostrando un aumento en la tasa de aprobación de los estudiantes en la asignatura estudiada, desde un 53,7% en el año 2018 a 65,2% el año 2019, que es el año en que se aplicaron estas metodologías. Durante los años 2020 y 2021 se implementó el Rediseño Curricular de Ingeniería Civil Informática y, en particular, la asignatura de Programación fue dividida en 2 (Algoritmos y Programación; y Programación Estructurada), con el fin de lograr la aplicación de los contenidos por parte de los estudiantes.

Tabla 6: Porcentaje de aprobación / reprobación desde 2015 a 2019

Año	%Aprobación	%Reprobación
2015	29,2%	70,8%
2016	42,4%	57,5%
2017	52,8%	47,3%
2018	53,7%	43,3%
2019	65,2%	34,7%

CONCLUSIONES

La aplicación de las metodologías activas en la asignatura Programación de Computadores de la carrera de Ingeniería Civil Informática de la Universidad Viña del Mar, históricamente considerada como una asignatura crítica, ha mejorado el porcentaje de aprobación, siendo el resultado mejor que los 4 años anteriores. Sin embargo, aquellos resultados de aprendizaje asociados a la aplicación de conocimientos no tuvieron un logro

significativo por parte de los estudiantes, a diferencia de los resultados de aprendizaje asociados a recordar y comprender. Las habilidades de recordar y comprender se refuerzan principalmente en el trabajo que se realiza resolviendo problemas, tanto de manera personal como trabajando con los compañeros, es decir, aplicando la programación por pares, las trazas de código a mano y la resolución de problemas, por lo que es posible concluir que estas metodologías activas funcionan cuando son aplicadas en un curso de Programación como éste.

REFERENCIAS

- Ausín, V., Abella, V., Delgado, V., y Hortigüela, D., Project-Based Learning through ICT: An Experience of Teaching Innovation from University Classrooms, <https://doi.org/10.4067/S0718-50062016000300005>, *Formación Universitaria*, 9(3), 31–38 (2016)
- Berssanette, J. H., y de Francisco, A. C., Active Learning in the Context of the Teaching/Learning of Computer Programming: A Systematic Review, <https://doi.org/10.28945/4767>, *Journal of Information Technology Education: Research*, 20, 201–220 (2021)
- Cheah, C. S., Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review, <https://doi.org/10.30935/cedtech/8247>, *Contemporary Educational Technology*, 12(2), ep272 (2020)
- Coutinho, S. C., *Numeros Inteiros E Criptografia RSA*, ISBN: 978-85-244-0124-4 (2009)
- Díaz de Rada, V., Tipos de encuestas considerando la dimensión temporal, <https://doi.org/10.5565/rev/papers/v86n0.814>, *Revista de Sociología*, 86, 131–145 (2007)
- Educ2020, Educación 2020, Aprendizaje basado en proyectos, <https://educacion2020.cl> (2021)
- Gil, V., Valencia, B., Hernández, S., y Yarce, S., Taxonomía de Bloom en la formulación de objetivos para un proyecto de lectoescritura en inglés, Simposio de Investigación USTAMED, Medellín, Colombia (2018)
- Guo, P., Saab, N., Post, L. S., y Admiraal, W., A review of project-based learning in higher education: Student outcomes and measures, <https://doi.org/10.1016/j.ijer.2020.101586>, *International Journal of Educational Research*, 102, 101586 (2020)
- Hagedorn, L. S., Factors Related to the Retention of Female Graduate Students over 30, <https://doi.org/10.2190/LKWC-EJUB-V5KM-BRNQ>, *Journal of College Student Retention: Research, Theory and Practice*, 1(2), 99–114 (1999)
- Joyanes, L., *Fundamentos de Programación: Algoritmos y Estructuras de Datos*, 3a edición, ISBN: 978-84-481-3664-2, McGraw-Hill Interamericana (2003)
- Juárez, J. L., López, M. C., y Villareal, Y., Estrategias para Reducir el Índice de Reprobación en Fundamentos de Programación de Sistemas Computacionales del I.T. Mexicali, *Revista de Gestión Empresarial y Sustentabilidad*, ISSN 2448-6043, 2(1), 25-41 (2016)
- MinEduc, Ministerio de Educación - Chile, Metodología de aprendizaje basado en proyectos, 1a edición, Unidad de Currículum y Evaluación del Ministerio de Educación, ISBN: 978-956-292-808-3, Santiago, Chile (2019)
- Namratha, M., Rekha, G S, y otros 3 autores, Active learning approach for python programming, *Journal of Engineering Education Transformations*, 32, 15–19 (2018)
- Omar, N., Haris, S. S., y otros 5 autores, Automated Analysis of Exam Questions According to Bloom's Taxonomy, <https://doi.org/10.1016/j.sbspro.2012.09.278>, *Procedia Computer Science*, 59, 297–303 (2012)
- Ortiz, E., El aprendizaje basado en problemas como experiencia de innovación y mejora docente universitaria, de Alfonso Javier García González (coordinador), <https://doi.org/10.22201/iisue.24486167e.2019.164.59223>, *Perfiles Educativos*, 41(164), 208–213 (2019)
- Prokofyeva, N., Uhanova, M., y otros 3 autores, Introductory Programming Training of First Year Students at Riga Technical University, <https://doi.org/10.1016/j.procs.2017.01.137>, *Procedia Computer Science*, 104, 286-293 (2016)
- RAE, Algoritmo, Real Academia Española de la Lengua, <https://dle.rae.es> (2020)
- Rodríguez, C., y Fernández-Batanero, J., Evaluación del Aprendizaje Basado en Problemas en Estudiantes Universitarios de Construcciones Agrarias, <https://dx.doi.org/10.4067/S0718-50062017000100007>, *Formación Universitaria*, 10(1), 61-70 (2017)
- Silva, J., y Maturana, D., Una propuesta de modelo para introducir metodologías activas en educación superior, *Innovación educativa (México, DF)*, 17(73), 117–131 (2017)
- Sobral, S. R., Bloom's Taxonomy to Improve Teaching-Learning in Introduction to Programming, <https://doi.org/10.18178/ijiet.2021.11.3.1504>, *International Journal of Information and Educational Technology*, 11(3), 148-153 (2021)
- Torres, J., Acevedo, D., y Montero, P., Proyectos de Aula Semestrales como Estrategia Pedagógica para la Formación en Ingeniería, <https://dx.doi.org/10.4067/S0718-50062016000300004>, *Formación Universitaria*, 9(3), 23-30 (2016)
- Yew, E. H. J., y Goh, K., Problem-Based Learning: An Overview of its Process and Impact on Learning, <https://doi.org/10.1016/j.hpe.2016.01.004>, *Health Professions Education*, 2(2), 75–79 (2016)