

## Enseñando Pensamiento Computacional en el Ciclo Básico de las carreras de Ingeniería

Pamela Viale<sup>1,2</sup>, Claudia Deco<sup>1,2</sup>, Cristina Bender<sup>1,2</sup>

<sup>1</sup> Facultad de Química e Ingeniería del Rosario,  
Universidad Católica Argentina, Campus Rosario, Argentina.  
{pamelaviale, cdeco, cbender@uca.edu.ar}@uca.edu.ar

<sup>2</sup> Facultad de Ciencias Exactas, Ingeniería y Agrimensura,  
Universidad Nacional de Rosario, Rosario, Argentina.  
{pamela, deco, bender}@fceia.unr.edu.ar

**Resumen.** Los futuros ingenieros de la sociedad actual necesitan formarse en los fundamentos de las ciencias de la computación para poder aprovechar la tecnología para asistirse en la resolución de problemas de manera automática y/o semi-automática a través de aplicaciones. El Pensamiento Computacional hace referencia a un conjunto de competencias necesarias para la formulación y resolución de problemas. La enseñanza del Pensamiento Computacional permite que los ingenieros puedan hacer uso de las nuevas tecnologías para resolver diferentes problemas que se les presentan en su actividad profesional de manera más rápida y eficiente. El estudio del Pensamiento Computacional ha tenido influencia en las investigaciones relacionadas con los procesos de aprendizaje y enseñanza, sensibilizando a las instituciones educativas a abordar el problema desde el punto de vista de la formación. Teniendo este objetivo en mente se propuso un plan para agregar contenidos sobre Pensamiento Computacional en materias de ciclo básico de las carreras de Ingeniería de la Facultad de Química e Ingeniería del Rosario (UCA). Se desarrollaron algunas primeras experiencias con este nuevo enfoque en los cursos iniciales de dichas carreras y los resultados son promisorios.

**Palabras clave:** Pensamiento Computacional, Programación, Carreras de Ingeniería.

### 1 Introducción

Desde hace ya varios años, estudiosos de todo el mundo afirman que la enseñanza del Pensamiento Computacional (PC) resulta beneficiosa para cualquier profesional, independientemente de la especialidad en la que se desarrolle. Se usa el término Pensamiento Computacional para hacer referencia a un conjunto de competencias necesarias para la formulación y resolución de problemas de manera que puedan ser resueltos por un agente de procesamiento de información [1, 2, 3]. El PC promueve la utilización de cuatro estrategias principales para la solución de problemas:

1. Descomposición de un problema en subproblemas: involucra el análisis de problemas complejos y su descomposición en problemas más pequeños, más fáciles de analizar.
2. Reconocimiento de patrones: cada uno de estos problemas más pequeños puede ser analizado en profundidad para identificar problemas similares ya resueltos.
3. Abstracción: para buscar las soluciones a los problemas encontrados es necesario focalizarse en los detalles importantes e ignorar información no relevante.
4. Pensamiento algorítmico: pueden proponerse una serie de pasos o reglas a seguir para crear una solución para cada uno de los subproblemas encontrados.

Una solución elaborada utilizando las estrategias del PC, puede ser fácilmente implementada en un sistema computacional, construyendo así una solución eficiente a un problema inicial complejo.

El estudio del Pensamiento Computacional ha tenido influencia en las investigaciones relacionadas al entendimiento y desarrollo de los procesos de enseñanza y aprendizaje, provocando que se aborde el tema desde el punto de vista de la formación en los distintos niveles educativos [4, 5, 6].

Se ha decidido entonces focalizar este trabajo en la formación de estudiantes de las carreras de Ingeniería de la Facultad de Química e Ingeniería del Rosario (UCA) en las competencias relacionadas al Pensamiento Computacional. Para esto, se está trabajando en la elaboración y adaptación de los materiales didácticos para las materias “Informática” e “Informática General” del ciclo básico de las carreras de Ingeniería de nuestra facultad.

## 2 Planteo del Problema

Según la ISTE (International Society for Technology in Education) y CSTA (Computer Science Teacher Association) el Pensamiento Computacional es un proceso para la resolución de problemas. Este proceso permite, entre otras cosas, aprender a formular problemas de manera que una computadora u otras herramientas puedan asistir en su resolución, permite analizar y organizar datos según cierta lógica, modelar y simular datos, automatizar algoritmos, identificar, analizar e implementar diferentes alternativas de solución para lograr un objetivo, y generalizar y transferir el proceso de solución de un problema a otros similares. Una forma de lograr la aprehensión de estos conceptos y prácticas de resolución de problemas es mediante la implementación de dichas soluciones en lenguajes que puedan ser interpretados y ejecutados por un ordenador, o sea, en algún lenguaje de programación.

Existen numerosos grupos de investigación que hacen del pensamiento computacional su principal objeto de estudio. Son de interés particularmente para este trabajo, dos equipos: el grupo de investigación de Karen Brennan de la Universidad de Harvard, perteneciente a la Graduate School of Education, Cambridge, Estados Unidos, y el equipo de Marcos Román-González de la Universidad Nacional de Educación a Distancia (UNED), de Madrid, España.

Brennan y su equipo [2] han desarrollado un marco de referencia para el estudio del pensamiento computacional a través del uso online de su herramienta Scratch. El estudio de esta herramienta y, a su vez, la participación en workshops, les ha permitido proponer una definición propia de pensamiento computacional que involucra tres dimensiones diferentes. La primera de estas dimensiones involucra los conceptos computacionales, conceptos que no son exclusivos de la herramienta en cuestión, sino que son comunes a la mayoría de los lenguajes de programación, como por ejemplo, secuencias, bucles, paralelismo, eventos, condicionales, operadores lógicos y matemáticos, datos, etc. La segunda dimensión consiste en el uso de estrategias o prácticas computacionales, entre ellas podemos citar la experimentación e iteración, el testeo y debugging, la reutilización de proyectos anteriores, la abstracción y modularización, entre otras. La última de las dimensiones trata las perspectivas computacionales, esto involucra el cambio de mirada que produce el uso de la herramienta en sus usuarios: cambia su forma de expresarse con respecto al área de las ciencias de la computación, aprecian la importancia de conectarse con otros y producir colaborativamente, y se cuestionan sobre el mundo, les permite darse cuenta que podrían ayudar a resolver problemas cotidianos mediante el desarrollo de herramientas propias.

Con respecto a Marcos Román-González y su equipo, nuestro principal interés se centra en un test que permite evaluar los conocimientos que posee un individuo sobre el pensamiento computacional [7].

**Tabla 1.** Vinculación entre Dimensiones de Brennan y Test de Román-González (Tabla original en [7], Pág 3)

Dimensiones de K. Brennan	Descripción	Componentes	¿Evaluado por Test de Román-González?
Conceptos Computacionales	Conceptos usados por durante la programación	Secuencias	Sí
		Bucles	Sí
		Eventos	No
		Paralelismo	No
		Condicionales	Sí
		Operadores	Sí
		Datos	No
Prácticas Computacionales	Prácticas de resolución de problemas durante el proceso de programación	Experimentación e Iteración	No
		Testeo y Debugging	Parcialmente
		Reutilización de proyectos	Parcialmente
		Abstracción y Modularización	Parcialmente
Perspectivas Computacionales	Cambios que produce aprender a programar	Forma de expresarse	No
		Conexión con otros	No
		Cuestionamientos	No

En la Tabla 1 vemos cómo se relacionan las dimensiones propuestas por el equipo

de Brennan y el test de PC propuesto por el equipo de Román-González. Se muestra de manera bien clara cuáles son los conceptos computacionales que se evalúan en el test. También se hace evidente que dicho test sólo permite evaluar parcialmente las prácticas computacionales y que resulta imposible el uso del mismo para evaluar las perspectivas computacionales. El test propuesto por Román-González, permite evaluar los conocimientos sobre PC teniendo en cuenta las capacidades de abstracción, descomposición, reconocimiento de patrones y la habilidad para comprender/corregir algoritmos. Para ello consta de 28 preguntas. Cada pregunta evalúa diferentes capacidades que se resumen en la Tabla 2.

**Tabla 2.** Conocimientos evaluados por cada pregunta del test (Tabla original en [8], Págs 118 y 119).

Pregunta	Abstracción	Descomposición	Reconocimiento de Patrones	Algoritmo
1	x			x
2	x			x
3	x			x
4		x	x	x
5		x	x	x
6		x	x	x
7	x	x		
8			x	x
9			x	x
10		x	x	x
11	x	x	x	x
12	x	x	x	x
13	x	x		x
14	x		x	x
15	x	x	x	x
16				x
17			x	x
18			x	x
19				x
20			x	x
21	x	x		x
22	x	x	x	x
23	x	x	x	x
24				x
25	x	x	x	x
26	x	x	x	x
27	x	x	x	x
28	x	x	x	x

Se aplicó el test a los alumnos durante el primer día de clases, tratando así de obtener una idea de los conocimientos sobre el pensamiento computacional con los cuales llegan a nuestra universidad. A partir de esto se pudo armar una primera fase de mejora del proceso de enseñanza - aprendizaje para fortalecer en los estudiantes los aspectos más débiles. En este sentido, durante el dictado, se hizo fundamental hincapié en aquellos relacionados al pensamiento computacional. Los principales temas abordados fueron los siguientes: introducción a la algoritmia, abstracción y generalización, funciones y colecciones. Al finalizar el cuatrimestre se solicitó a los mismos alumnos que respondan nuevamente a las mismas preguntas. Analizando las respuestas de estos alumnos que realizaron el test antes de tomar el curso y luego del mismo, se observó que se obtuvo un mejor desempeño por parte de los estudiantes [9]. Por esto se ha decidido realizar la experiencia presentada en este trabajo.

El objetivo es mejorar los procesos de enseñanza y aprendizaje de las competencias involucradas en el Pensamiento Computacional, para que los futuros ingenieros estén más preparados para asumir los retos que demanda la sociedad contemporánea. La viabilidad del proyecto está sustentada en los avances obtenidos mediante trabajos previos de los integrantes en el área y su interacción con otros grupos de investigadores latinoamericanos [10]. Se propone desarrollar una didáctica para la enseñanza del pensamiento computacional y la programación para estudiantes de las carreras de ingeniería y a su vez proponer metodologías activas y lúdicas para la utilización y entrega de los materiales en experiencias educativas presenciales y en línea.

### **3 Desarrollo de la Experiencia**

Se comenzó con las primeras experiencias en la materia Informática que se dicta en el primer año de las carreras de Ingeniería. Se desarrolló material para estas asignaturas centrados en aquellos conocimientos principales del Pensamiento Computacional, como ser, la abstracción, la descomposición de problemas en subproblemas, el reconocimiento de patrones y algoritmos. En las prácticas se utiliza la estrategia de aprendizaje basado en proyectos. Se busca motivar a los estudiantes de las diferentes carreras mediante el desarrollo de prototipos que les sean de utilidad en su profesión. La idea es que los estudiantes comprendan, a través del desarrollo de estos proyectos, la importancia que estas habilidades tienen para su vida profesional y personal futura. Se busca incentivarlos mediante el desarrollo de proyectos de su elección. Los temas de los mismos son libres, pero se les pide a los alumnos que reflexionen sobre prototipos y/o herramientas que les puedan ser de utilidad en su labor futura como profesionales, y con eso en mente propongan un proyecto a desarrollar. La dificultad del mismo es evaluada por el docente, quien, teniendo en cuenta la propuesta de los estudiantes, brinda el acuerdo o adecúa la dificultad teniendo en mente los alcances de la materia. Con este plan de innovación se busca que nuestros egresados puedan hacer uso de las tecnologías para resolver los problemas que se les presenten en su actividad profesional, para dar soluciones a una sociedad cada vez más exigente, que demanda respuestas cada vez más rápidas y eficientes.

A modo de ejemplo, podemos citar algunos de los proyectos desarrollados por los alumnos del primer año de la carrera de Química, durante el segundo semestre del año 2020. Nuestros estudiantes han desarrollado prototipos para: informar sobre las peligrosidades de reactivos en laboratorios, aceptar o rechazar medios de cultivos coloreados en función de su absorbancia, calcular pesos moleculares, realizar cálculos estequiométricos y calcular energía en celdas electroquímicas, con el objetivo de optimizar la generación de energía en baterías. A continuación se amplían tres de los proyectos grupales presentados por nuestros alumnos.

### **PROYECTO 1: Peligrosidades de Reactivos en el laboratorio**

*Descripción de la aplicación:* Esta aplicación tiene como finalidad saber cuáles son las peligrosidades de los reactivos en el laboratorio, cuál es su correcta manipulación y a su vez agilizar el tiempo de clasificación de ellos. Los químicos trabajan constantemente con reactivos los cuales algunos suelen ser muy peligrosos y causan daño a la salud, también al medio ambiente o pueden causar riesgos físicos. Pero a su vez esta aplicación la podría usar cualquier persona inexperta que quiera usar reactivos y no sepa cuáles son los recaudos que debe tomar, cuáles son los peligros, cuál es su correcta manipulación y así poder facilitar su uso. Esta aplicación también servirá para saber, por ejemplo, que los reactivos inflamables los debemos manejar lejos de los mecheros o los reactivos que son tóxicos para el medio acuático debemos desecharlos en los lugares correspondientes. Es decir, su modo de emplearlo. Al abrir la aplicación el usuario deberá elegir una de las 5 categorías de peligrosidad de los reactivos, la cual al elegir una se le mostrará 3 opciones más, las cuales serán la definición de dicha peligrosidad, Como emplear dicho reactivo de forma segura y ejemplos de reactivos más frecuentes. Cada una contará con su descripción y al terminar de leerlo se le preguntará al usuario “¿desea volver al menú anterior?” si este contesta que no se quedará allí, pero si contesta que sí volverá a menú anterior, donde tendrá la posibilidad de elegir otra opción o contestar si desea o no volver al menú anterior, si este decide volver al menú de categorías volverá a tener la opción de elegir otra peligrosidad o “volver a inicio” donde se le preguntará si desea salir de la programa, si éste contesta que sí, se cerrará el programa, si contesta que no se lo enviará nuevamente al menú de las distintas peligrosidades.

*Menú de opciones:* El usuario puede elegir una de las siguientes opciones: 1.Explosivo, 2.Inflamable, 3.Comburente, 4.Corrosivo, 5.Toxicos, 6.Irritación cutánea, 7.Peligro por aspiración, 8.Peligroso para el medio ambiente, 9.Salir.

*Ejemplos de datos involucrados:* Sustancia Corrosivas: reactivos que pueden destruir una superficie metálica o un tejido ya sea inerte o vivo, manipular con la vestimenta adecuada como guantes, batas de laboratorio y gafas de seguridad. Por ejemplo: oxidantes fuertes, ácidos y bases fuertes.

*Ejemplo de casos de uso:* Si el usuario no elige una opción entre el 1 y 9 se le mostrará el mensaje “Error en la opción ingresada”. Si el usuario ingresa, por ejemplo, la opción 8. Peligroso para el medio ambiente, la respuesta será: Sustancias Peligrosas para el Medio Ambiente (N): contienen reactivos cuyo uso inapropiado pueden causar daños graves al medioambiente, ya sea inmediato o futuro para uno o más componentes del medio ambiente mediante su incorrecta eliminación como

residuos, por ejemplo a la capa de ozono (clorofluorocarbonos) o al medio ambiente acuático (Cinc). Su clasificación es difícil, dependerá de la concentración, de la forma física, lugar de vertido, etc. Su forma recomendable de uso es que los desechos sean los mínimos y que cumplan los requisitos correspondientes. Por ejemplo: Cianuro de Potasio, Benceno, Asbestos, Cinc.

### **PROYECTO 2: Calculadora de peso molecular**

*Descripción de la aplicación:* La idea principal es la de ayudar a estudiantes o profesionales al momento de realizar cálculos o resolver problemas con pesos moleculares de compuestos o elementos químicos. A modo de agilizar el proceso de una manera interactiva, eficaz y confiable con el fin de reducir los márgenes de errores y alivianar el trabajo. El usuario tendría que ingresar la fórmula del compuesto o el elemento y el programa arrojará su peso molecular/atómico automáticamente.

*Menú de opciones:* Frente a la pregunta ¿Qué peso desea saber?, el usuario puede elegir una de las siguientes opciones: 1.Peso molecular, 2.Peso atómico, 3.Salir.

*Ejemplos de datos involucrados y cómo serán los cálculos a realizar:* Los cálculos que se realizan serán solamente suma y multiplicación. Si el usuario ingresa el símbolo de un elemento se arrojará el resultado sin hacerle ninguna operación aritmética, por ejemplo si el usuario ingresa: Fe sabemos que ese símbolo pertenece al hierro y pesa 55,847 u. Si ingresa una molécula diatómica por ejemplo yodo: I2 tenemos que multiplicar el peso molecular del yodo por 2:  $126,9045 \times 2$  y arrojamos el resultado de la siguiente manera: “El peso molecular es 253.809 u”. Si se ingresa una molécula con más de un elemento por ejemplo ácido sulfúrico H<sub>2</sub>SO<sub>4</sub>, se tendrá que multiplicar el peso del hidrógeno por 2 sumar el peso del azufre y sumar el resultado de multiplicar el oxígeno por 4. Y mostrar en pantalla: “El peso molecular es:”

*Ejemplo de casos de uso:* Al ingresar a la aplicación aparece el siguiente mensaje: ¿QUÉ PESO DESEA SABER?, si el usuario elige la opción 2, la aplicación le solicita que INGRESE EL SÍMBOLO QUÍMICO, y el sistema le devuelve El PESO ATÓMICO ES: “...”.

### **PROYECTO 3: Cálculos estequiométricos**

*Descripción de la aplicación:* El objetivo principal del programa es, a través de la concentración que el usuario ingrese, ya sea en molaridad (moles de soluto por litros de solución) o en normalidad (número de equivalentes por litro de solución), determinar la masa (en el caso de un sólido) o el volumen (en el caso de un líquido) de reactivo que se debe tomar para obtener la solución que se desea. De este modo, el programa, colabora con los cálculos del usuario, minimizando los posibles errores y optimizando el tiempo de trabajo.

*Menú de opciones:* Para elegir la concentración que desea utilizar, el usuario puede elegir una de las siguientes opciones: 1. Molaridad, 2. Normalidad, 3. Salir.

*Ejemplos de datos involucrados y cómo serán los cálculos a realizar:* Existirán 2 unidades de concentración para trabajar en el programa, la primera es molaridad, en este caso el usuario deberá ingresar datos como “masa molar de la sustancia”, la “concentración de la solución a preparar” y “el volumen de la solución”. Desde aquí

existirán dos caminos posibles, si la droga es líquida se necesitará que el usuario indique la densidad de la misma, si la droga es sólida no se necesitan datos adicionales. El programa desarrollará una serie de multiplicaciones y divisiones en donde el resultado será “la cantidad de droga a pesar o a pipetear”, según si es sólida o líquida. Si se selecciona normalidad el programa actuará de la misma forma, solamente que pedirá como dato adicional “la cantidad de equivalente químicos de la sustancia”.

*Ejemplo de caso de uso:*

MENSAJE EN PANTALLA	DATOS QUE INGRESA EL USUARIO
Concentración a utilizar: Molaridad/Normalidad	Molaridad
Droga: Líquida/Sólida	Sólida
Ingrese la masa molar del soluto en gramos por mol:	40
Ingrese el volumen de la solución que desea preparar en mililitros:	250
Ingrese la concentración de la solución en molaridad:	0.1

La respuesta final de la aplicación es: Se debe tomar 1 gramo de la droga.

#### 4 Resultados Preliminares

Durante estas primeras experiencias se ha notado una mayor participación de los estudiantes en el proceso de aprendizaje así como también una mayor motivación por la programación. Creemos que esto se debe fundamentalmente a la propuesta de trabajo basado en proyectos de su interés profesional.

Como cierre de la experiencia se solicitó a los alumnos que comenten lo realizado mediante las respuestas a tres preguntas: ¿Qué dificultades encontraron al realizar el trabajo práctico?, ¿Creen que los contenidos vistos sobre el pensamiento computacional / algoritmia podrán serles de utilidad en un futuro?, y ¿Qué se podría mejorar en la materia para otros años?

Podemos compartir algunos de los comentarios hechos por los alumnos al finalizar el cursado:

- “Nos encontramos, en primera instancia, frente al desafío de pensar en algún tipo de programa que nos ayude a llevar a cabo con menor dificultad alguna tarea relacionada con nuestra carrera. Una vez que decidimos cuál sería la tarea que el programa debería realizar, nos encontramos frente a otra dificultad: pensar en la estructura del mismo y armarlo. Creemos que estos contenidos nos servirán para tener una idea general, sobre qué es el pensamiento computacional o la algoritmia. Esto, nos será de mucha utilidad en un futuro, ya que cada vez, se van introduciendo más y más estos conceptos en el mundo del trabajo. Nos pareció que la materia fue dictada de forma muy ágil, didáctica y clara. Se llegaron a



entender muy bien los conceptos y ejemplos planteados.”

- “Creemos completamente que los contenidos dados sobre pensamiento computacional y algoritmia nos van a ser útil al finalizar la carrera ya que nos plantea nuevas formas de ver los problemas y encontrarle soluciones más sencillas apoyándonos en la tecnología”
- “Creemos y esperamos que nos ayude ya que es una nueva manera de pensar y ver las cosas en este mundo tan tecnológico, que va a seguir creciendo en cuanto a la tecnología y por lo tanto cada vez más vamos a tener que apoyarnos en este tipo de contenidos.”

Los trabajos realizados por los estudiantes han sido de calidad y mostraron la adquisición de los conceptos de pensamiento computacional. Consideramos que estas primeras experiencias son muy positivas y seguiremos planteando nuevas estrategias para guiar y motivar a nuestros alumnos.

## 5 Conclusiones

La inclusión de la enseñanza del Pensamiento Computacional en diferentes carreras permite que los nuevos profesionales puedan hacer uso de las nuevas tecnologías para resolver de manera más eficiente diferentes situaciones problemáticas que se les presentan en su actividad laboral. Propusimos entonces, no sólo enseñar sino también evaluar el grado de aprendizaje de conocimientos básicos en ciencias de la computación. Para esto se trabajó, en particular, en las materias del ciclo básico de las carreras de Ingeniería de nuestra Facultad. En este sentido, se planteó evaluar la didáctica aplicada y el grado de entendimiento por parte de nuestros alumnos, para proponer cambios donde se detectaron problemas de aprehensión de conceptos.

En este trabajo se presentaron las experiencias realizadas en la carrera de Química donde se propuso a los estudiantes solucionar problemas propuestos por ellos y vinculados a su carrera mediante la utilización de estrategias de Pensamiento Computacional. En este sentido, los trabajos realizados por los estudiantes han sido de calidad y mostraron la adquisición de los conceptos de Pensamiento Computacional. Consideramos que estas primeras experiencias son muy positivas y seguiremos planteando nuevas estrategias para guiar y motivar a nuestros alumnos.

## Referencias

1. Wing J. Computational Thinking Benefits Society. Social Issues in Computing. 2014.
2. Brennan K., Resnick M. New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 Annual Meeting of the American Educational Research Association (Vancouver, Canada). 2012.

3. Román-González M., Perez-González J. C., Jiménez-Fernández C. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72: 678-691. 2017.
4. García M., Deco C., Bender C., Collazos C. Invited paper: Robotics Based Strategies to Support Computational Thinking: The Case of the Pascual Bravo Industrial Technical Institute. *Journal of Computer Science and Technology (JCS&T)* Vol. 17 No. 1. April 2017. pp 59-67. 2017.
5. García M., Deco C., Bender C., Collazos C. . Herramientas de Diseño para el Desarrollo de Competencias en Educación Básica, Media y Tecnológica: Experiencia en el Instituto Técnico Industrial Pascual Bravo de Colombia. *Revista TE&ET (Argentina)*. Mayo 2017.
6. Casali A., Deco C., Viale P., Bender C., Zamarini D., Monjelat N. Enseñanza y aprendizaje del pensamiento computacional y la programación en los distintos niveles educativos. Mayo 2020 - XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020, El Calafate, Santa Cruz). pp 595-599. 2020.
7. Román-González M., Perez-González J. C., Jiménez-Fernández C.. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72: 678-691, 2017
8. Brackmann C. P.. Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica. Universidade Federal do Rio Grande Do Sul, Centro interdisciplinar de novas tecnologias na educação, Programa de Pós-Graduação em informática na educação, 2017.
9. Viale P., Deco C., Bender C. Introduciendo conocimientos sobre el Pensamiento Computacional en los primeros años de las carreras de Ingeniería. En *Proceedings del XIII Congreso Argentino Ingeniería Industrial (COINI)*. Organizado por Asociación Argentina de Carreras de Ingeniería Industrial y Afines (AACINI). Virtual. 19 al 24 Octubre 2020.
10. Rodríguez del Rey Y., Cawanga Cambinda I., Deco C., Bender C., Avello-Martínez R., Villalba-Condori K. Developing Computational Thinking with a Module of Solved Problems. In *Computer Applications in Engineering Education*. March 2020.