

# Influencia del pensamiento computacional en las habilidades cognitivas propias de los métodos de resolución de problemas

HOP Volumen 23 #2 julio - diciembre

Influence of computational thinking on the cognitive skills of problem solving methods



Augusto Ramiro **Brugés Romero**  
Yadira del Pilar **Camperos Villamizar**



HOP Volumen 23 # 2 Junio-Diciembre

# hop 23

*Dorian*

Photo By/Foto:



De:

Planeta Formación y Universidades

**HORIZONTES PEDAGÓGICOS**

ISSN-I: 0123-8264 | e-ISSN: 2500-705X

Publicación Semestral

ID: [10.33881/0123-8264.hop.23206](https://doi.org/10.33881/0123-8264.hop.23206)

Title: Influence of computational thinking on the cognitive skills of problem solving methods

Título: Influencia del pensamiento computacional en las habilidades cognitivas propias de los métodos de resolución de problemas

Alt Title / Título alternativo:

[en]: Influence of computational thinking on the cognitive skills of problem solving methods

[es]: Influencia del pensamiento computacional en las habilidades cognitivas propias de los métodos de resolución de problemas

Author (s) / Autor (es):

Brugés Romero & Camperos Villamizar

Keywords / Palabras Clave:

[en]: Abstraction, Benchmarking; cognition; computational thinking; computer programming; problem solving; Skills development

[es]: Abstracción, Cognitivo; desarrollo de habilidades; evaluación; pensamiento computacional; programación informática; solución de problemas.

Submitted: 2021-12-23

Accepted: 2021-12-31

## Resumen

El desarrollo de software se ha convertido en un aspecto fundamental en la formación de ingenieros y se ha establecido como una competencia diferenciadora al momento de acceder al campo laboral; sin embargo, su aprendizaje no es una tarea fácil y muchos estudios apuntan a que el principal inconveniente se encuentra en la debilidad cognitiva de los estudiantes para resolver problemas, lo cual es un problema recurrente en otras áreas de formación, por lo que se hace necesario fortalecer la comprensión, el análisis, la síntesis y la evaluación a nivel general.

Desde esta perspectiva, se planteó una investigación con un enfoque cualitativo y diseño fenomenológico, cuyo objetivo principal estuvo en analizar la influencia del pensamiento computacional en el mejoramiento de las habilidades cognitivas propias de los métodos de resolución de problemas. Los informantes de la investigación fueron 10 estudiantes del curso de programación I y II en el periodo 2021-I, y utilizó como técnicas de recolección de información pruebas al iniciar y al finalizar el curso, la observación participante y la entrevista semiestructurada.

Los resultados de los instrumentos aplicados evidenciaron que los estudiantes presentan problemas de comprensión e interpretación, así como desconocimiento de conceptos básicos que obstaculizaron el correcto planteamiento de soluciones a los problemas presentados en las pruebas iniciales. En la prueba diagnóstica la mayoría de los estudiantes obtuvieron bajos resultados, lo que indica que existen deficiencias para solucionar problemas, ya que los elementos del pensamiento computacional son aplicados de manera mínima y los aciertos obedecen más a procesos mentales de orden inferior como memoria, observación y comparación. Los resultados de la prueba final y el análisis de los demás instrumentos aplicados muestran que la adquisición de conocimientos sobre programación mejoró las habilidades de abstracción y análisis; aumentaron la creatividad; incluyeron elementos de diseño, planificación y seguimiento; desarrollaron la metacognición, la síntesis y la evaluación.

La presente investigación arrojó como resultado que el desarrollo del pensamiento computacional le permite al estudiante mejorar la forma en que resuelve los problemas, adquiriendo habilidades cognitivas y metacognitivas para utilizar nuevos métodos y estrategias que generan alternativas de solución más creativas, ágiles y eficientes. El uso adecuado del conocimiento y los elementos adquiridos en torno al pensamiento computacional y el aprendizaje de la programación, así como la mejora de habilidades asociadas al análisis y la comprensión, permiten optimizar la forma en que solucionan problemas.

## Abstract

Software development has become a fundamental aspect in the training of engineers and has been established as a differentiating competence at the time of accessing the labor field; however, its learning is not an easy task and many studies suggest that the main drawback is in the cognitive weakness of students to solve problems, which is a recurring problem in other formation areas, it is necessary to strengthen understanding, analysis, synthesis and evaluation at a general level.

From this perspective, a research with a qualitative approach and phenomenological design was proposed, whose main objective was to analyze the influence of computational thinking, developed through the learning of programming, in the improvement of cognitive skills of problem solving methods. The informants of the research were 10 students of the programming I and II course in the period 2021-I, and used as information collection techniques tests at the beginning and end of the course, participant observation and semi-structured interview.

The results of the applied instruments showed that students present problems of understanding and interpretation, as well as lack of knowledge of basic concepts that hindered the correct approach of solutions to the problems presented in the initial tests. In the diagnostic test, the majority of students obtained low results, which indicates that there are deficiencies to solve problems, since the elements of computational thinking are applied in a minimal way and the correct answers are more due to mental processes of inferior order such as memory, observation and comparison.

The results of the final test and the analysis of the other instruments applied show that the acquisition of programming knowledge improved abstraction and analysis skills; increased creativity; included elements of design, planning and monitoring; developed metacognition, synthesis and evaluation.

The present research obtained as a result that the development of computational thinking allows students to improve the way they solve problems, acquiring cognitive and metacognitive skills to use new methods and strategies that generate more creative, agile and efficient solution alternatives. The proper use of the knowledge and elements acquired around computational thinking and programming learning, as well as the improvement of skills associated with analysis and understanding, allow optimizing the way they solve problems.

## Citar como:

Brugés Romero, A. R. & Camperos Villamizar, Y. d. (2021). Influencia del pensamiento computacional en las habilidades cognitivas propias de los métodos de resolución de problemas. *Horizontes Pedagógicos*, 23 (2), 57-70. Obtenido de: <https://horizontespedagogicos.iberro.edu.co/article/view/2342>

Ing Augusto Ramiro **Brugés Romero**, Mgtr Esp  
ORCID: [0000-0002-1047-1071](https://orcid.org/0000-0002-1047-1071)

Source | Filiación:

Universidad de Pamplona

BIO:

Ingeniero Electrónico egresado de la Universidad de Antioquia, Magister en Educación y Especialista en Gestión de Proyectos Informáticos de la Universidad de Pamplona, SEI-Certified PSP Developer, certificado por el Software Engineering institute – Carnegie Mellon University

City | Ciudad:

Bogotá DC [co]

e-mail:

[abruges@unipamplona.edu.co](mailto:abruges@unipamplona.edu.co)

Dra Yadira del Pilar **Camperos Villamizar**, Dra Mgtr Esp  
ORCID: [0000-0002-1926-2479](https://orcid.org/0000-0002-1926-2479)

Source | Filiación:

Universidad de Pamplona

BIO:

Doctora en educación Universidad Pedagógica Experimental libertador (Venezuela), Magister en Educación Universidad de Pamplona, Especialista en Pedagogía Universitaria y Psicóloga. Actualmente docente de la Universidad de Pamplona

City | Ciudad:

Bogotá DC [co]

e-mail:

[yadira.camperos@unipamplona.edu.co](mailto:yadira.camperos@unipamplona.edu.co)

# Influencia del pensamiento computacional en las habilidades cognitivas propias de los métodos de resolución de problemas

Influence of computational thinking on the cognitive skills of problem solving methods

Augusto Ramiro **Brugés Romero**  
Yadira del Pilar **Camperos Villamizar**

## Introducción

Aunque son múltiples los factores que se han identificado al interior de los cursos de programación, en cuanto a la dificultad que tienen los estudiantes para apropiarse los conceptos relacionados con los lenguajes de programación y su diagnóstico puede resultar aún más complejo, se puede mencionar que la deficiencia en la formación lógica y analítica del estudiante, la apatía por el aprendizaje de los contenidos, la poca comprensión lectora, la debilidad de los conocimientos básicos previos, la incapacidad de correlación y deducción y algunas veces, los errores metodológicos de enseñanza, son las principales causas de los bajos resultados en los cursos de programación, sin embargo, algunos estudios previos, permiten inferir que el problema principal puede radicar en la debilidad cognitiva de los estudiantes para resolver problemas, de allí el planteamiento que surge en ésta investigación de desarrollar el pensamiento computacional como herramienta para mejorar dichas habilidades.

*Soloway afirma que los docentes de los cursos introductorios de programación son conscientes que enseñar la semántica y la sintaxis de un lenguaje de programación no es suficiente para el aprendizaje de la programación, se considera de mayor importancia fortalecer habilidades para solucionar problemas, la idea es lograr que el aprendiz pueda transferir lo aprendido en programación a otras situaciones de resolución de problemas, no sólo como una habilidad vocacional, sino como un vehículo para aprender la resolución efectiva de problemas. (Santimateo, Nuñez, & González, 2018, pág. 14)*

Son varios los estudios que se han desarrollado al respecto, particularmente para esta investigación se realizaron diferentes consultas bibliográficas que sirvieron como suministro para el desarrollo de la misma y permitieron contrastar los resultados obtenidos con los ya existentes.

Por ejemplo, Zúñiga et al. (2014) en su artículo “El Desarrollo del Pensamiento Computacional para la Resolución de Problemas en la Enseñanza Inicial de la Programación” mencionan que los estudiantes deben desarrollar una gran variedad de habilidades, más allá de la memorización de sentencias que permitan codificar, pues antes de escribir un código se requiere entender un problema, es decir, abstraer, modelar y analizar, con el fin de plantear soluciones acordes a las necesidades. Desde esta perspectiva, el estudiante deberá desarrollar habilidades genéricas adicionales que contribuyan al mejoramiento de la metodología usada para la resolución de problemas (Zúñiga, Rosas, Fernández, & Guerrero, 2014).

Otro estudio plasmado en el artículo “Experiences in Learning Problem-Solving through Computational Thinking” de los autores Fernández et al. (2018), publicado en la revista Computer Science & Technology, describe dos alternativas diferentes con el fin de poder incluir el pensamiento computacional como estrategia para la resolución de problemas. Su objetivo estuvo en evidenciar que el pensamiento computacional se puede usar como medio para mejorar el proceso de resolución de problemas basado en habilidades cognitivas de descomposición, abstracción, reconocimiento de patrones y algoritmos (Fernández, Zúñiga, Rosas, & Guerrero, 2018). Los resultados de las experiencias planteadas en el estudio aportan elementos de valor a la actual investigación, ya que evidencian la importancia de incluir el pensamiento computacional en el aprendizaje de la resolución de problemas en diferentes áreas.

El siguiente referente se encuentra en el estudio “Evaluación de habilidades del pensamiento computacional para predecir el aprendizaje y retención de estudiantes en la asignatura de programación de computadoras en educación superior” de los autores Rojas y García (2020), publicado en la Revista de Educación a Distancia en 2020. El objetivo del estudio se centró en establecer el diagnóstico de las habilidades del pensamiento computacional de estudiantes que inician el curso de programación en la Universidad Tecnológica de Puebla – México y cuáles de dichas habilidades favorecen el aprendizaje de la programación. Los resultados de la investigación se relacionan entonces con la evaluación de los elementos del pensamiento computacional en los estudiantes y las competencias que estos adquieren y mejoran una vez finalizado el curso (Rojas & García, 2020).

Con el planteamiento del problema y la evidencia existente en estudios similares, la presente investigación buscó mostrar el pensamiento computacional como una alternativa para el desarrollo y mejoramiento de las habilidades cognitivas en la solución de problemas de índole tecnológico, que permitan fortalecer en los estudiantes las capacidades propias para plantear soluciones a través de algoritmos que sean llevados con éxito a un lenguaje de programación, disminuyendo con ello el índice de deserción y la pérdida del curso, lo que beneficiará a los estudiantes al generarles una formación integral que potencia las posibilidades laborales en este mundo mediado por las tecnologías de la información y sus sistemas.

## Marco teórico

### El Constructivismo

Aunque en la actualidad son muchas las corrientes, teorías y metodologías que se encuentran para interpretar la forma como el individuo adquiere y desarrolla el conocimiento, tal como el asociacionismo que se basa en los conocimientos ya adquiridos o como el racionalismo que se basa en la experiencia a partir del contacto con el entorno, en el caso del pensamiento computacional debemos iniciar nuestro estudio con el constructivismo. Terceros (2019) menciona en su artículo programación creativa: pensamiento computacional y constructivismo, que:

*Pensar el desarrollo del pensamiento computacional no es posible sino es a partir de un proceso constructivista enlazado con la tecnología, mediante el enfoque de problemáticas de los estudiantes ricas en contexto, donde el conocimiento del problema es tan necesario como las herramientas de solución, generando un entorno dinámico para la reflexión, la creatividad. (Terceros, 2019, pág. 122)*

Definido como una teoría epistemológica, el constructivismo fue desarrollado con el fin de analizar la forma en que adquirimos y transformamos el conocimiento, teniendo sus bases contemporáneas con el desarrollo de la teoría cognitiva-evolutiva desarrollada por Jean Piaget y que junto con otros planteamientos constructivistas formularon diversas teorías sobre el aprendizaje, de las cuales es pertinente mencionar:

- ➔ Teoría cognitiva de Piaget. Sostiene que el proceso de maduración biológica conlleva al desarrollo de estructuras cognitivas cada vez más complejas.
- ➔ El aprendizaje social de Vygotsky. Se fundamenta en el aprendizaje como resultado de la interacción del individuo con el entorno.
- ➔ El aprendizaje significativo de Ausubel. En esta teoría se asume que el individuo relaciona su conocimiento con uno adquirido previamente y que en dicho proceso el nuevo conocimiento es perdurable. Este proceso combina aspectos lógicos, cognitivos y afectivos. (Ortiz Granja, 2015, pág. 98).

La postura de desarrollo de habilidades cognoscitivas desarrolladas desde el enfoque constructivista plantea que lo más relevante en el proceso de aprendizaje es el desarrollo de tales habilidades y no los contenidos. La enseñanza debe centrarse en el desarrollo de capacidades para observar, clasificar, analizar, deducir y evaluar, prescindiendo de los contenidos, de modo que una vez alcanzadas estas capacidades pueden ser aplicadas a cualquier tópico (Araya, Alfaro, & Andonegui, 2007, pág. 90).

Desde el punto de vista constructivista, el aprendizaje obtenido por el individuo se va logrando a medida que este va alcanzado ciertos niveles de maduración, lo que en el actual estudio se logra a medida que este va adquiriendo y afianzando nuevos conocimientos. Estos conocimientos permiten el desarrollo de nuevas habilidades que se van perfeccionando paulatinamente a través de la experimentación de posibles soluciones a un problema dado y de las simulaciones realizadas para obtener los resultados.

# Pensamiento Computacional

Los orígenes del pensamiento computacional se remiten al término utilizado por la doctora Jeannette M. Wing, directora del instituto de ciencia de datos de la Universidad de Columbia y consultora informática de la Universidad de Carnegie Mellon, en un artículo publicado en la revista *Communications of the ACM* en marzo de 2006. Dicho artículo pretendía dar un acercamiento a las ventajas que tenía que las personas pensarán como informáticos, sin tener necesariamente que ser expertos en el área.

Aunque existen varios intentos por precisar con claridad el término desde diferentes enfoques, ubicamos inicialmente la definición dada por la doctora Wing (2011), a quien se le reconoce como la precursora del concepto, la cual estableció el pensamiento computacional como “el proceso del pensamiento que involucramos en la solución de problemas para que las soluciones planteadas puedan ser llevadas a cabo por un agente de procesamiento de información” (Wing, 2011, pág. 1). Otra definición es aportada por Aho (2012) en la que expresa que el pensamiento computacional “son los procesos de pensamiento involucrados en la formulación de problemas para que sus soluciones puedan representarse como pasos y algoritmos computacionales” (Aho, 2012, pág. 832).

Los elementos que componen el pensamiento computacional se resumen a continuación y se representan en la figura 1, ubicando la abstracción como uno de los elementos centrales de la computación y estableciéndolo como un componente crucial para el futuro del desarrollo científico y tecnológico. Y aunque la abstracción es una habilidad inherente a todos los seres humanos, cada individuo la aplica de forma única debido a que se desarrolla de manera diferente (Zapotecalt, 2018).

## Descomposición.

Hace referencia a la capacidad de dividir problemas grandes en problemas más pequeños, de manera que sea lo más independiente posi-

ble pero que aporte de manera significativa a la solución. En términos computacionales se puede relacionar con el término módulo.

## Abstracción.

Proceso en el cual se obtienen las propiedades y/o cualidades más relevantes de algún objeto, procedimiento o situación en particular. Cuando se abstrae se debe ser capaz de generalizar, pero también de tener la habilidad de quitar los detalles no relevantes.

## Patrón.

Permite tomar una muestra o modelo para ser replicado o utilizado según las necesidades. Debe ser genérico, sin embargo, debe adaptarse a las necesidades particulares de cada situación. En la figura se representa mediante la generalización.

## Análisis.

Proceso mediante el cual se definen objetivos, se identifican elementos simples, se relacionan, se buscan estrategias de desarrollo para plantear alternativas claras, ordenadas, coherentes y precisas. Requiere de técnicas de inferencia, deducción y pensamiento lógico.

## Evaluación.

Permite medir la eficacia del método propuesto y la solución obtenida a través de mecanismos que arrojan los resultados obtenidos y que tan acordes fueron a los objetivos planteados. En la evaluación se detectan aciertos y desaciertos y se usa como suministros para mejorar en situaciones similares.

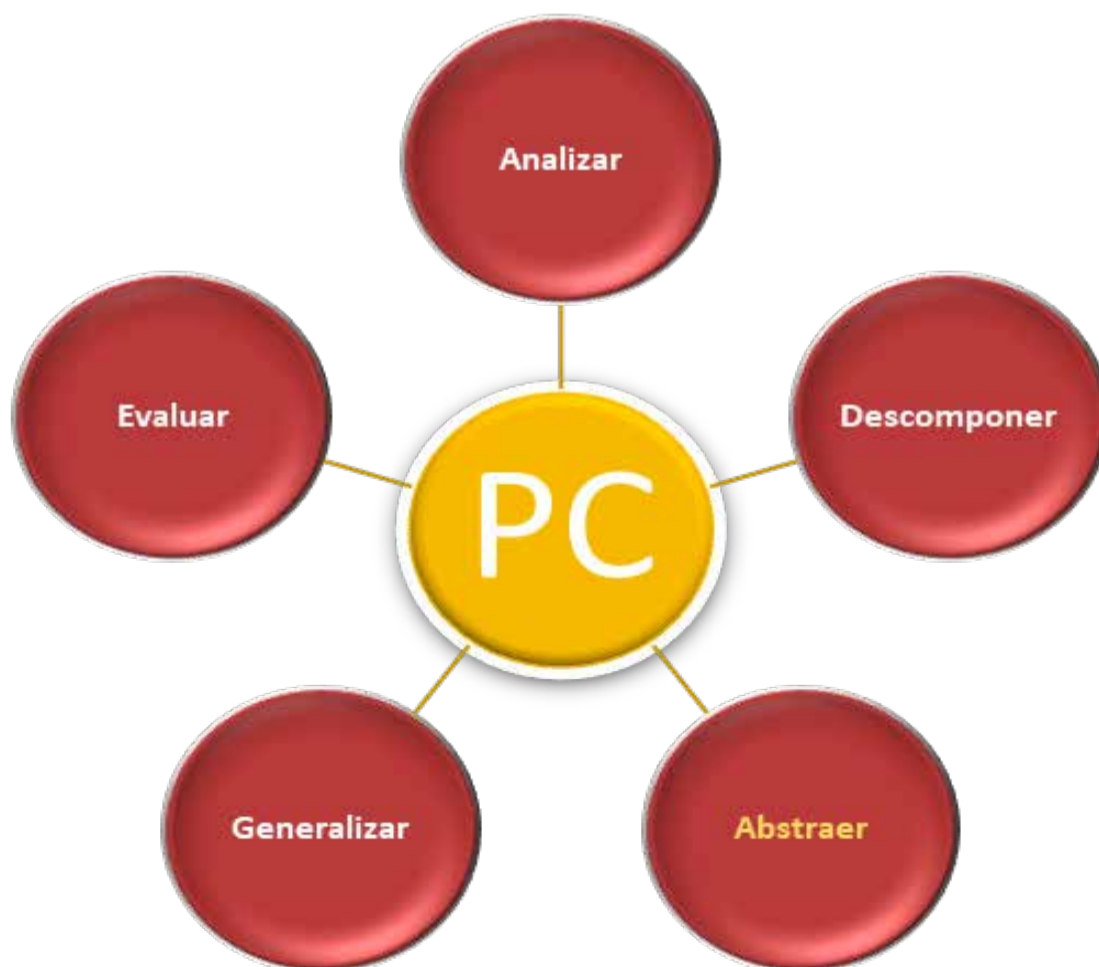


Figura 1 Componentes del pensamiento computacional

# Habilidades Cognitivas

En palabras de Rigney (1978), citada por Herrera (2001),

*Las habilidades cognitivas son entendidas como operaciones y procedimientos que puede usar el estudiante para adquirir, retener y recuperar diferentes tipos de conocimientos y ejecución ... suponen del estudiante capacidades de representación (lectura, imágenes, habla, escritura y dibujo), capacidades de selección (atención e intención) y capacidades de autodirección (autoprogramación y auto-control). (Herrera Clavero, 2001, pág. 1)*

Para Velásquez et al. (2013) “el pensamiento es una experiencia interna e intrasubjetiva, a través de la cual podemos abstraer, discriminar, inferir, inventar, encontrar respuestas, resolver problemas, analizar, reflexionar, lo cual se manifiesta mediante la elaboración de hipótesis, razonamiento y emisión de juicios” (Velásquez Burgos, Remolina de Cleves, & Calle Márquez, 2013, pág. 25). Es entonces por ello que debemos diferenciar los niveles de complejidad que podemos observar al momento de analizar las habilidades cognitivas de los individuos, encontrando procesos simples como la observación y la comparación, o procesos más complejos como el análisis, la síntesis y la abstracción.

Para Herrera (2001), las habilidades cognitivas las puede clasificar en estrategias cognitivas y meta cognitivas, que las resume de la siguiente manera:

- cognitivas: Son las facilitadoras del conocimiento, aquellas que operan directamente sobre la información: recogiendo, analizando, comprendiendo, procesando y guardando información en la memoria, para, posteriormente, poder recuperarla y utilizarla dónde, cuándo y cómo convenga.
- metacognitivas: Son las facilitadoras de la cantidad y calidad de conocimiento que se tiene, su control, su dirección y su aplicación a la resolución de problemas o tareas.

Para Bloom, el pensamiento de las personas se puede clasificar por niveles, de manera que el proceso de aprendizaje sea ordenado y se permitan definir claramente los objetivos del aprendizaje. Bloom plantea su taxonomía y divide los niveles cognitivos de las personas en tres grandes grupos (Conocimiento, Comprensión y Procesos Superiores) cada uno de ellos con características propias que determinan desarrollos de habilidades diferentes. En el planteamiento de problemas más complejos, como los que se utilizan en los cursos de programación, se espera que sus soluciones requieran el uso de elementos correspondientes al tercer nivel según Bloom, puesto que deberán investigar y experimentar con alternativas, extrayendo la información relevante y tomando decisiones de acuerdo a la necesidad de cada problema. La taxonomía sugerida por Benjamín Bloom se presenta de manera resumida en la tabla 1 (Gutiérrez Ávila, de la Puente Alarcón, Martínez González, & Piña Garza, 2012, pág. 52)

**Tabla 1. Niveles Cognitivos Según Bloom**

Nivele Cognitivo	Actividades de Aprendizaje
1. Conocimiento	Memorización de hechos, datos, conceptos, definiciones
2. Comprensión	Explicación y/o interpretación del significado de nueva información
3. Aplicación	Utilización de un concepto o principio para resolver tareas
4. Procesos Mentales Superiores	Separación de una hipótesis, ensayo o idea en sus partes para identificar interrelaciones y jerarquía de ideas
5. Análisis	Producción de juicios o conocimientos nuevos a partir de sus componentes.
6. Síntesis	Es el proceso de trabajar con fragmentos, partes, elementos, organizarlos, ordenarlos y combinarlos para formar un todo, un esquema o estructura que antes no estaba presente de manera clara.
7. Evaluación	Formulación de juicios sobre el valor de las ideas, soluciones y métodos, incluye la comparación de varios métodos de solución

Fuente: Aprendizaje Basado En Problemas, Gutiérrez et al (2012).

Para las técnicas de solución de problemas, el individuo debe ser capaz de desarrollar y perfeccionar aquellas habilidades de nivel superior que le permitan tener las herramientas necesarias para transformar su entorno. Dentro de estas habilidades, como se mencionó anteriormente, se encuentra el análisis, la síntesis, la argumentación y la abstracción (Velásquez Burgos, Remolina de Cleves, & Calle Márquez, 2013).

desarrollo de los pasos que conlleven a la solución del problema, estableciendo un conjunto de lineamientos que van desde la comprensión del problema hasta la evaluación de la solución. Los métodos de resolución de problemas se han venido desarrollando a lo largo de los siglos soportados por distintas corrientes psicológicas, cada una de las cuales ha realizado su interpretación dependiendo de su orientación; en los últimos años se ha hecho énfasis para la resolución de problemas desde el marco de las ciencias exactas.

## Métodos para la Solución de Problemas

Un problema es una relación entre un conjunto de situaciones (entradas) y un conjunto de soluciones (salidas) que permite establecer formalmente la relación deseada entre las unas y las otras. El problema se resuelve si se obtiene al menos una solución correspondiente para cada escenario (Zapotecalt, 2018).

La figura 2 muestra la descripción de pasos para la solución de problemas según Polya, sin embargo, existen muchos otros métodos que siguen el enfoque numérico o matemático planteado por él (Pozo, Pérez, Angel, & Postigo, 1994, pág. 25).

Cuando se hace referencia al método para la solución de problemas se hace alusión a la selección de una estrategia a seguir para el

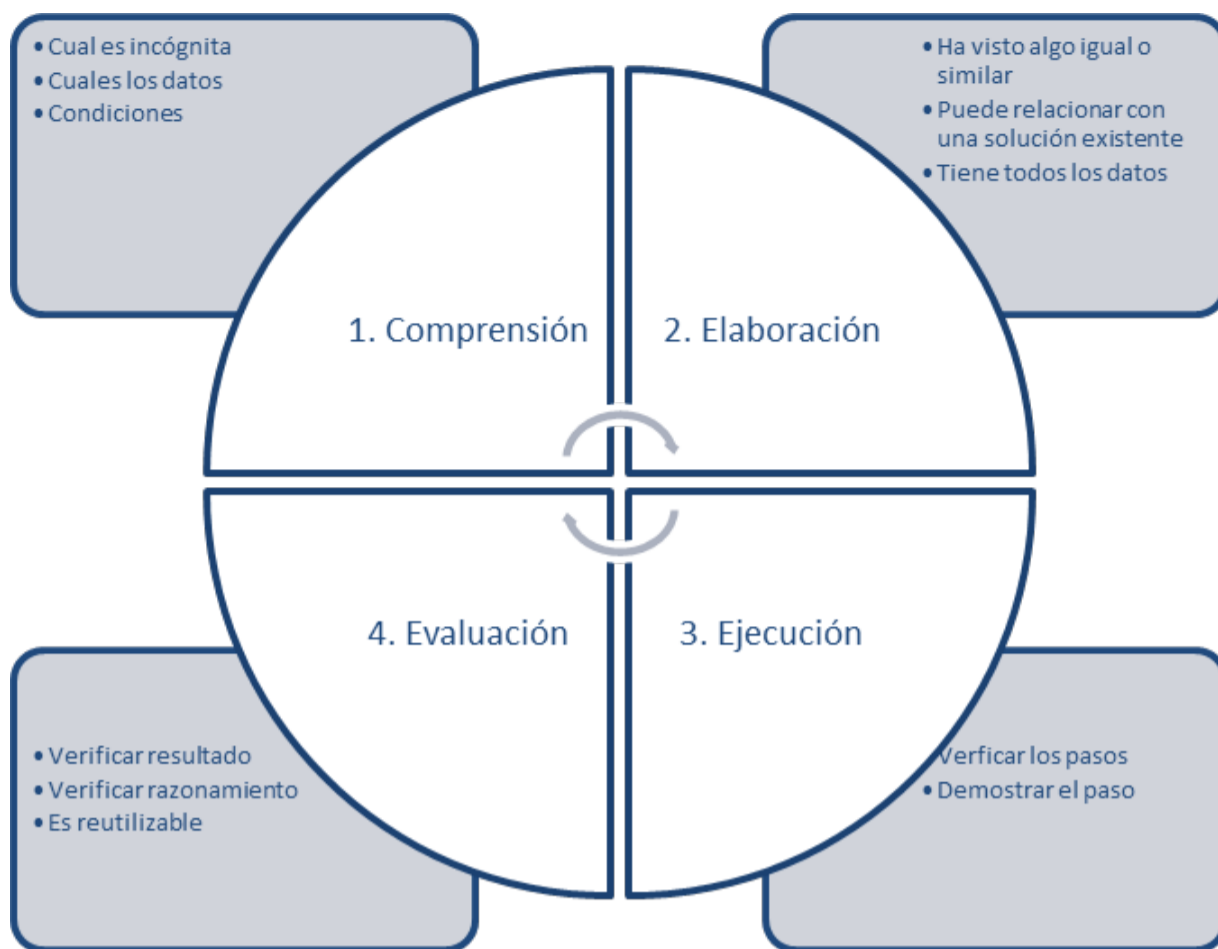


Figura 2. Pasos en la solución de problemas según Polya

Fuente: Elaboración Propia

## Diseño Procedimental

### Naturaleza de la Investigación

Analizar la influencia del pensamiento computacional en el desarrollo de las habilidades cognitivas para solucionar problemas a través del aprendizaje de la programación por parte de los estudiantes del curso de programación, necesitó de un enfoque cualitativo y un diseño fenomenológico, puesto que el objetivo principal de dicho enfoque es reconstruir la realidad según la observación que realizan los autores y la interpretación que hacen sobre los comportamientos de los individuos objeto de investigación. En este tipo de investigación, las técnicas de recolección de información incluyeron la observación participante, la entrevista semiestructurada y la evaluación de experiencias e interacción con grupos o comunidades.

### Informantes

Los informantes de la investigación fueron 10 estudiantes de los cursos de programación estructurada I y II en el periodo 2021-I, de los programas de ingeniería ambiental, civil, eléctrica, telecomunicaciones, mecánica y mecatrónica de la sede principal de la Universidad de Pamplona, seleccionados de manera intencional, teniendo en cuenta los criterios de inclusión: que no estuviesen repitiendo la materia para programación I y que estuviesen cursando los programas antes mencionados.

Tabla 2. Universo e Informantes

Grupo	Programación I	Programación II
Matriculados	30	48
Informantes clave (Estudiantes)	5	5

Fuente: Elaboración Propia

### Fases de la Investigación

Tal como lo presenta Martínez Miguelez en su libro Ciencia y Arte en la Metodología Cualitativa, el método fenomenológico consta de tres (3) etapas diferentes: etapa descriptiva, etapa estructural y etapa de discusión de resultados. La figura 3 representa las diferentes etapas de la investigación.



Figura 3. Fases de la Investigación

Fuente: Elaboración propia

En la etapa descriptiva se realizó la recolección y revisión bibliográfica, ubicación de los antecedentes investigativos relacionados con el pensamiento computacional y la solución de problemas, que permitieron describir claramente lo que se pretendía estudiar. Posteriormente se ubicó el método a aplicar y las formas de recolección de información, que correspondieron a la aplicación de la prueba diagnóstica, la observación directa, la entrevistas semiestructurada y la aplicación de

la prueba validadora. En la etapa estructural se revisó la información obtenida; se aplicó la prueba inicial; se hizo la observación y se aplicó la entrevista a los estudiantes del curso de programación I y II. Luego se aplicó la prueba final, se hizo la observación y la entrevista final a los estudiantes del curso programación II. Después se realizó la categorización de los elementos a evaluar según los resultados obtenidos, se presentaron los resultados según las fases y se obtuvieron los resultados generales. En la etapa de resultados y discusión se relacionaron los resultados obtenidos con estudios similares de otros autores, contrastando los logros con el fin de enriquecer el conocimiento en el área. Los resultados permitieron confrontar las posturas de los autores plasmadas en el marco referencial con el fin de corroborar o controvertir las conceptualizaciones dadas por ellos y de esta manera validar la aplicabilidad de los conceptos en el contexto objeto de estudio.

## Técnicas de recolección de información

Para obtener la información conducente al logro de los objetivos, se seleccionaron como técnicas e instrumentos, los siguientes:

### Prueba Diagnóstica y Prueba Validadora.

Los instrumentos prueba diagnóstica y prueba validadora constaron de un conjunto de problemas planteados a los cuales el estudiante respondió según su interpretación. Las preguntas y problemas allí planteados se tomaron de las iniciativas internacionales “International Bebras Contest” y “Computer Olympiad Talent Search”. Dichos instrumentos buscaron establecer el uso de habilidades del pensamiento computacional en estudiantes con ningún, medio y alto conocimiento de programación, permitiendo determinar la existencia, la mejora o la adquisición de estas habilidades a medida que obtenían conocimientos de programación y desarrollaban el pensamiento computacional.

La prueba diagnóstica se aplicó al inicio del semestre y consistió en ocho preguntas que se aplicaron a cinco (5) estudiantes de programación I y cinco (5) estudiantes de programación II. Estos informantes representaban los estudiantes con bajo y medio conocimiento de programación respectivamente.

La prueba validadora tenía el mismo formato que la prueba diagnóstica, sin embargo, sus preguntas eran diferentes y se aplicó a cinco (5) estudiantes al finalizar el curso de programación II. Este grupo representa aquellos informantes que cuentan con un conocimiento superior de programación respecto a los anteriores. Dichos informantes también participaron de la aplicación de la prueba diagnóstica que se aplicó al inicio del semestre, lo que permite conocer el avance de los estudiantes al adquirir conocimientos avanzados de programación y por ende un mayor desarrollo del pensamiento computacional.

### La observación participante.

Es la técnica más usada en los tipos de investigación cualitativa y permite al investigador participar de las actividades de la población muestra, tomando nota de los detalles para posteriormente preguntarse y responder el quién, qué, cuándo, cómo, dónde y por qué alguien hizo algo (Martinez Miguelez, 2004).

Para la presente investigación, cada problema planteado en las pruebas diagnósticas y validadoras aplicadas requerían de una breve explicación respecto a la solución aportada, lo que junto con las dudas que durante la presentación de las pruebas surgían por parte de

los informantes, sirvieron como mecanismo de observación. Con la observación, los investigadores obtuvieron datos adicionales en un contacto directo con la realidad y en el medio natural donde los informantes desarrollaron los problemas, estando al tanto de la situación y el contexto, lo que permitió interactuar con ellos, observar y analizar su comportamiento al momento de resolver los problemas y de esta manera determinar la forma en que piensan y actúan frente a una situación determinada, la interpretación que dan sobre los problemas presentados y el análisis hecho sobre los resultados obtenidos.

### Entrevista Semiestructurada.

La entrevista es una técnica de gran utilidad en la investigación cualitativa para recabar datos; se define como una conversación que se propone un fin determinado distinto al simple hecho de conversar, es un instrumento técnico que tiene consonancia con los tipos de investigación cualitativa en su componente metodológico y epistemológico (Martinez Miguelez, 2004).

En la presente investigación, la entrevista buscó obtener información relevante de las percepciones de los estudiantes sobre el uso, existencia y desarrollo de habilidades cognitivas y metacognitivas al iniciar y finalizar los cursos de programación y de la manera como estas influyen al momento de solucionar los problemas planteados. Su aplicación se realizó en el momento en que cada estudiante completó la prueba diagnóstica y la prueba validadora y estuvo compuesta de un conjunto de preguntas abiertas que tenían relación con los elementos que utilizaron para desarrollar cada pregunta o con la forma en que dichos elementos aportan al desarrollo de los problemas de las pruebas aplicadas.

La entrevista diagnóstica buscó obtener información sobre las habilidades que tienen los estudiantes antes de iniciar los cursos de programación y la percepción sobre la forma en que afrontan los problemas de las pruebas aplicadas; por su parte, la entrevista validadora buscó obtener información de la percepción del estudiante sobre la forma en que el curso de programación aportó para obtener nuevas habilidades que permitieran solucionar los problemas planteados.

Preguntas como ¿Qué elemento o elementos se le dificultaron en mayor medida al resolver los ejercicios planteados?, ¿Cuál de las preguntas del test anteriormente aplicado le pareció más compleja contestar, cuál más sencilla?, ¿Por qué?, ¿Qué aptitudes o habilidades cognitivas considera usted que le ayudaron a desarrollar de manera adecuada los problemas planteados?, ¿Qué aptitudes o habilidades cognitivas considera usted que le hacen falta adquirir o perfeccionar para obtener mejores resultados en la prueba? ó ¿Cómo espera que el aprendizaje de la programación le ayude en el desarrollo de este tipo de problemas? fueron aplicadas una vez desarrollaron la prueba diagnóstica, mientras que preguntas como ¿Qué habilidades cognitivas considera usted que pudo adquirir o mejorar durante el curso de programación?, ¿Qué aspectos cree que mejoraron en el desarrollo de la prueba final una vez visto el curso de programación?, ¿De las habilidades cognitivas adquiridas o perfeccionadas, cuáles y como cree que le van a servir para mejorar el rendimiento académico y en el desarrollo de su vida personal? se aplicaron después de realizar la prueba validadora.

### Cursos de Programación

Es válido precisar, para efectos de contextualización, que los cursos de programación sometidos al presente estudio son teórico-prácticos, con una intensidad horaria de 2 horas teóricas y 3 prácticas semanales (3 créditos) y una duración total de 16 semanas. Su objetivo princi-



pal está en utilizar el concepto de función como principal herramienta para la abstracción y descomposición en un problema complejo y como fundamento de la programación estructurada, así como adquirir habilidades en la solución de problemas por medio del computador, mediante el desarrollo del pensamiento algorítmico. En el contacto directo, el docente se encarga de impartir los conocimientos teóricos necesarios para la comprensión de los contenidos y a través de problemas reales ilustra la aplicación de los mismos; es habitual el uso del modelo de aula invertida, siendo un mecanismo de autoaprendizaje que le permite al estudiante tener información suficiente previa al contacto directo con el docente. Una vez el estudiante recibe la conceptualización, aplica lo aprendido a través de problemas prácticos que desarrollan en pares en las sesiones de laboratorio, de manera que sean capaces de obtener diferentes alternativas de solución al problema planteado, retroalimentando el conocimiento aprendido. Aproximadamente se alcanzan a desarrollar 14 sesiones de laboratorio durante el transcurso del semestre.

## Resultados

Para determinar si el estudiante utilizó o no cierta habilidad, a cada uno de los problemas planteados en las pruebas diagnóstica y validadora se le asociaron uno o más elementos del pensamiento computacional, relación que se encuentra previamente establecida en las iniciativas “International Bebras Contest” y “Computer Olympiad Talent Search”, tal como se resume en la tabla 3. Cuando el estudiante dio respuesta a los problemas de la prueba aplicada, se hizo la clasificación de las habilidades utilizadas según la relación mencionada y al final se determinó cuáles fueron las que más se usaron y cuales no fueron tan utilizadas.

De esta manera, los instrumentos permitieron evaluar los siguientes elementos: descomposición (DE), abstracción (AB), reconocimiento de patrones (PR), algoritmos (AL), evaluación (EV), razonamiento Lógico (RL).

**Tabla 3. Habilidad por Pregunta**

Pregunta	(DE)	(AB)	(PR)	(AL)	(EV)	(RL)
P1 DV		X	X		X	
P2 DV		X				X
P3 DV				X		X
P4 D			X	X		
P5 D	X	X		X	X	
P6 D	X				X	
P7 D				X		
P8 D	X	X				
P4 V	X	X		X		
P5 V			X	X		
P6 V				X		X
P7 V	X	X		X		
P8 V	X	X	X	X		

Nota. Fuente Propia

Las tablas 4 y 5 muestran los resultados obtenidos por informante en la prueba diagnóstica.

**Tabla 4. Resultados Prueba Diagnóstica Programación I**

Pregunta	IEP1_1	IEP1_2	IEP1_3	IEP1_4	IEP1_5
P1	NO	NO	NO	SI	SI
P2	NO	NO	SI	SI	SI

Pregunta	IEP1_1	IEP1_2	IEP1_3	IEP1_4	IEP1_5
P3	SI	SI	SI	SI	SI
P4	-	-	SI	-	SI
P5	NO	NO	NO	SI	NO
P6	SI	-	-	SI	SI
P7	NO	-	-	NO	SI
P8	SI	NO	SI	NO	SI

Fuente: Elaboración Propia

**Tabla 5 Resultados Prueba Diagnóstica Programación II**

Pregunta	IEP2_1	IEP2_2	IEP2_3	IEP2_4	IEP2_5
P1	SI	NO	SI	SI	NO
P2	SI	NO	SI	SI	NO
P3	SI	SI	SI	SI	SI
P4	NO	SI	-	SI	SI
P5	SI	SI	NO	SI	SI
P6	SI	SI	SI	SI	NO
P7	SI	SI	NO	SI	NO
P8	SI	SI	SI	SI	SI

Fuente Elaboración Propia

Los elementos presentados en las tablas 4 y 5 reflejan el comportamiento de los estudiantes con poco o mediano conocimiento de programación y los resultados obtenidos muestran la debilidad de los estudiantes que inician el curso de programación en la mayoría de las habilidades computacionales, evidenciando falencias notorias en más de la mitad de ellos, en elementos como la abstracción, el reconocimiento de patrones, la algoritmia y la evaluación de los resultados, todos fundamentales al momento de solucionar problemas de esta índole. Estos elementos fueron mejorados en algunos de los estudiantes que contaban con un conocimiento previo de programación, revertiendo las fallas al solucionar los problemas y obteniendo mejores resultados. Por su parte, el uso de la descomposición es un elemento que tiene un equilibrio en cuanto a la debilidad o la fortaleza de los estudiantes sin conocimientos de programación, una habilidad que permite simplificar la complejidad de los problemas y que redundante en la efectividad de la solución cuando es aplicada de manera adecuada; dicha habilidad presenta una significativa mejora en la mayoría de los estudiantes con conocimientos medios de programación.

Como resultado de la observación, resultados presentados en la tabla 6, se evidencia desconocimiento de alternativas para casi todos los estudiantes del grupo 1. Los estudiantes del grupo 2 se esmeran por tener una solución basada en modelos matemáticos existentes lo que denota un uso de problemas similares para dar solución. De manera general, los estudiantes de ambos grupos son capaces de aplicar elementos matemáticos básicos, seguramente adquiridos previamente. Inicialmente, los estudiantes demostraron una alta aceptación de ejercicios que solo requieren seguir pasos, complementados con elementos de observación y secuenciación que permiten obtener un resultado adecuado. Las personas que dieron respuesta a los problemas afirman comprender las instrucciones dadas y dan cuenta de la capacidad espacial que tienen para ubicarse en un plano determinado. Se observa que las soluciones dadas por los estudiantes que inician los cursos de programación se relacionan con una alta capacidad para seguir secuencias, es decir, son capaces de ejecutar un algoritmo, sin embargo, otras habilidades que revisten mayor complejidad no son capaces de ejecutarlas de una manera adecuada o simplemente no se utilizan.

Tabla 6. Interpretación de la Observación

P1. Suma de Números	
Prueba Diagnóstica	Prueba Validadora
<p>El resultado de la observación denota desconocimiento de alternativas para casi todos los estudiantes del grupo 1 y su falta de interés para querer solucionar problemas de los cuales no es evidente o sencilla su solución. Los estudiantes del grupo 2 se esmeran por tener una solución basada en modelos matemáticos existentes lo que denota un uso de problemas similares para dar solución.</p> <p>Esta pregunta se relaciona con abstracción, reconocimiento de patrones y evaluación</p>	<p>Al revisar la forma en que el estudiante plantea la solución para este ítem se observa que aplican el método de gauss, método matemático para el cálculo de series que es muy útil para este tipo de ejercicio. Algunos de los estudiantes siguen sin presentar esfuerzo para buscar alternativas más eficientes a las inicialmente planteadas.</p>
P2. Nota promedio	
Prueba Diagnóstica	Prueba Validadora
<p>De manera general los estudiantes de ambos grupos son capaces de aplicar elementos matemáticos básicos, seguramente adquiridos previamente. Hay que recalcar que, aunque traten de hacer cálculos matemáticos, no realizan una comprobación de su resultado para validar la respuesta lo que causa que, aunque expliquen el procedimiento, no aciertan en el resultado. Hay errores operacionales, todos se sienten familiarizados con este tipo de problemas.</p> <p>Esta pregunta se relaciona con abstracción y razonamiento lógico.</p>	<p>El estudiante es capaz de utilizar procedimientos matemáticos simples. A diferencia de los resultados de la prueba diagnóstica, los estudiantes aparentemente comprenden completamente el enunciado del ejercicio y su planteamiento está correctamente enfocado. Sigue existiendo deficiencias en procesos de evaluación que permitan validar la respuesta obtenida.</p>
P3. Comida de Animales	
Observación	
<p>Mecanismos simples de observación y comparación le permiten al estudiante organizar los datos de manera tal que lleguen a conclusiones. Ninguno falla al dar la solución correcta y tanto estudiantes de programación I como de programación II desarrollan el ejercicio de una manera muy similar. El planteamiento y la estrategia utilizada son acordes a lo solicitado en el ejercicio. No es difícil para el estudiante comparar y seguir indicaciones, lo que refleja el resultado del ítem.</p> <p>Esta pregunta evalúa algoritmos y razonamiento lógico.</p>	<p>En este ítem no cambió la dinámica presentada en la prueba diagnóstica, los estudiantes aciertan en la solución y su planteamiento incluye elementos de comparación y observación. No se encuentra diferencias significativas en el planteamiento de la solución respecto a la prueba inicial.</p>
P4D. Dibujo del robot	P5V. Dibujo Rectángulo
Prueba Diagnóstica	Prueba Validadora
<p>Al igual que en el ejercicio anterior, los estudiantes demostraron una alta aceptación a ejercicios que solo requieren seguir pasos, complementados con elementos de observación y secuenciación que permiten obtener un resultado adecuado. Las personas que dieron respuesta al ejercicio comprendieron las instrucciones dadas y dan cuenta de la capacidad espacial que tienen para ubicarse en un plano determinado.</p> <p>Un poco menos de la mitad de los estudiantes del grupo 2 describen que siempre se gira a la derecha, lo que demuestra la capacidad que tienen para solo actuar sobre elementos relevantes, una característica importante de la abstracción, elemento central del pensamiento computacional. Este elemento no se evidenció en los estudiantes que no tienen conocimientos de programación.</p> <p>Este ejercicio contiene habilidades de reconocimiento de patrones y algoritmos</p>	<p>Este ítem es importante porque permite evaluar la capacidad que tiene el estudiante para describir una secuencia de pasos a seguir, demostrando habilidad en el pensamiento algorítmico, además del reconocimiento de patrones que permitan simplificar las soluciones. Algunos estudiantes presentan notorias deficiencias al interpretar el enunciado para llevarlo a un algoritmo y muchas veces su planteamiento no es claro lo que desencadena en soluciones erróneas. Otro grupo de estudiantes es capaz de describir claramente lo que se debe hacer. No se reconoce el patrón como elemento diferenciador para simplificar la solución.</p>
P5D. Tablero con 4 piezas	P6V. Estatura Ardillas
Observación	
<p>Un alto porcentaje de estudiantes del grupo 1 fallaron al solucionar el problema y la observación permite determinar que no buscaron siquiera alternativas para plantear la solución. Solo un informante utiliza el descarte como estrategia, utilizando un mecanismo de observación para verificar la cantidad de elementos utilizados en el tablero lo que le permitió disminuir la cantidad de probabilidades.</p> <p>Algo similar sucede con los estudiantes del segundo grupo y aunque los resultados son mejores solo dos presentan una solución más racional acorde a lo planteado en el ejercicio. Estos estudiantes hacen una validación de las alternativas presentadas para saber que tanto filas como columnas son conectadas según la descripción del ejercicio.</p> <p>Los elementos a tener en cuenta son abstracción, descomposición, algoritmos y evaluación.</p>	<p>Para el estudiante no es clara la forma en que debe iniciar el planteamiento de la solución y esto causa que no se llegue a una correcta solución. Aunque parece ser un ejercicio de simple observación se requiere de habilidades asociadas con la abstracción de manera que se logre tener el punto adecuado de partida, puede ser esta la causa que genera no obtener una solución adecuada. Las personas que solucionaron el problema de manera adecuada lograron identificar cual era la ardilla más alta en primera instancia.</p>

Fuente: Elaboración Propia

Por su parte y al igual que lo refleja la prueba y la observación diagnóstica, la entrevista inicial deja al descubierto problemas generales de interpretación y comprensión en los estudiantes que inician o han visto un curso de programación, así como deficiencias o desconocimiento de herramientas y técnicas que les permitan desarrollar de una manera más adecuada los problemas planteados. Al indagar sobre

qué problemas se les facilitaron en la prueba diagnóstica presentada, las respuestas dadas permitieron determinar que los problemas con contenidos de algoritmos o descomposición son fáciles de resolver por los estudiantes, mientras que los que requieren habilidades cognitivas más complejas como evaluación, abstracción y síntesis se les dificultan. También se logró establecer que el estudiante espera que el

aprendizaje de la programación y el desarrollo del pensamiento computacional le aporte a la consecución de mejores resultados, le proporcione los elementos necesarios para resolver los problemas y mejore las habilidades de observación, comparación, comprensión y análisis.

La tabla 7 relaciona la interpretación por parte de los investigadores de algunas de las respuestas dadas por los estudiantes entrevistados y que iniciaban los cursos de programación.

**Tabla 7. Interpretación de la Entrevista Diagnóstica**

**PED\_1. ¿Qué elemento o elementos se le dificultaron en mayor medida al resolver los ejercicios planteados?**

Problemas como la comprensión, la interpretación o desconocimiento de procedimientos elementales son el mayor inconveniente que tienen los estudiantes al iniciar el curso para enfrentarse a este tipo de problemas, elementos que persisten en estudiantes de programación II pero que se suman a conceptos como la abstracción y la velocidad de respuesta.

**PED\_3. ¿Cuál de las preguntas de la prueba anteriormente desarrollada le pareció más compleja contestar, cuál más sencilla? ¿Por qué?**

Estudiantes que apenas afrontan el curso de programación no muestran una inclinación bien definida sobre los elementos que dificultan el desarrollo de problemas, sin embargo, perciben que elementos que se relacionan con la comprensión como comparar, comprender o interpretar los resuelven más fácilmente. Entre las observaciones dadas, se acomodan a ser operativos, es decir a seguir instrucciones u ordenar elementos, pero se les es difícil obtener alternativas para dar soluciones

**PED\_4. ¿Cuáles aptitudes o habilidades cognitivas considera le ayudaron a desarrollar de manera adecuada los problemas planteados?**

Al igual que lo identificado en la prueba, los estudiantes de los dos grupos seleccionados se inclinan por mencionar habilidades de orden inferior para la solución de los problemas. Elementos como la observación, comparación, la interpretación, la memoria y el uso de conceptos anteriores aparecen identificados por parte de los informantes. Cuando se consulta a los estudiantes con conocimientos de programación muchos incluyen nuevas habilidades, relacionadas con procesos cognitivos de orden superior, dentro de los que encontramos la recomposición y la evaluación. Los elementos mencionados por los informantes permiten decantar los siguientes niveles: Análisis, comprensión, conocimiento, evaluación

**PED\_5. ¿Cuáles aptitudes o habilidades cognitivas considera debe adquirir o mejorar para obtener mejores resultados al desarrollar pruebas de este tipo?**

Es evidente que todos los procesos requieren de mejora, así lo reflejaron los estudiantes de ambos grupos. Sus respuestas muestran actividades relacionadas con habilidades ya adquiridas y de las cuales indicaron deben mejorar, una apreciación común en los dos grupos de programación. También dejan ver la necesidad de adquirir unas nuevas que, aunque no son explícitas para ellos, indican el desarrollo de habilidades de orden superior como la síntesis.

Fuente: elaboración Propia

La tabla 8 muestra los resultados obtenidos por informante en la prueba validadora

**Tabla 8. Resultados Prueba Validadora**

	Pregunta	IEP2_1	IEP2_2	IEP2_4	IEP2_5	IEP2_6
1	Suma de Números	SI	SI	SI	SI	SI
2	Nota promedio	SI	NO	NO	SI	SI
3	Comida de Animales	SI	SI	SI	SI	SI
4	Servicio de Transporte	SI	SI	SI	SI	SI
5	Dibujar Rectángulo	SI	NO	NO	SI	SI
6	Estatua Ardillas	SI	NO	NO	SI	NO
7	Programa Agitar Muestra	NO	SI	SI	SI	SI
8	Cartas Boca Arriba	SI	SI	SI	SI	SI

Fuente: Elaboración Propia

La asertividad de las respuestas evidencia mejora en los elementos asociados al pensamiento computacional cuando se tiene algún conocimiento de programación. El análisis que inicialmente plantearon

algunos de los estudiantes de programación I fue generalizado en los estudiantes de programación II, lo que evidencia un buen desarrollo de las habilidades que se pretendían evaluar cuando se desarrolla el pensamiento computacional a través del aprendizaje de la programación.

Como resultado de la observación asociada al desarrollo de la prueba validadora se encuentra que el estudiante es capaz de utilizar procedimientos matemáticos simples. A diferencia de los resultados de la prueba diagnóstica, los estudiantes aparentemente comprenden completamente el enunciado del ejercicio y su planteamiento está correctamente enfocado. Como se había evidenciado en la observación diagnóstica, el comportamiento de los estudiantes que tienen algún conocimiento de programación al resolver los problemas planteados sigue mejorando. Los resultados de la prueba y la observación hecha por los investigadores permiten evidenciar mayor efectividad en el planteamiento y análisis realizado para resolver cada problema, puesto que son más acordes con lo que el estudiante quiere representar en su solución. En esta prueba se pudo evidenciar un alto contenido de elementos asociados al pensamiento computacional, elementos que eran escasos en la primera prueba. Los resultados presentados y las observaciones realizadas también permitieron identificar que muy pocos estudiantes del curso de programación II tuvieron dificultades al desarrollar problemas con elementos asociados a la descomposición, el pensamiento algorítmico sigue siendo una fortaleza y además se incluyen elementos de evaluación.

En la entrevista, el estudiante percibe que el aprendizaje de la programación y el desarrollo del pensamiento computacional mejora los elementos asociados con el análisis y la comprensión, así como ayuda al desarrollo del razonamiento y la memoria. Al aprender y utilizar nuevos métodos y herramientas el estudiante siente que puede desarrollar los problemas de una manera más eficiente y creativa. Finalmente, el estudiante expresa la importancia del aprendizaje de la programación y el desarrollo del pensamiento computacional en los elementos que permiten tener alternativas en la solución de problemas.

Algunas interpretaciones de las respuestas dadas por los estudiantes entrevistados que finalizaron el curso de programación II son presentadas en la tabla 9

**Tabla 9. Interpretación Entrevista Validadora**

**PEV\_2. ¿Qué aspectos mejoraron en el desarrollo de la prueba después de haber visto el curso de programación?**

Al utilizar nuevos métodos y herramientas el estudiante siente que puede desarrollar los problemas de una manera más eficiente. La comprensión del ejercicio permite interpretarlo de mejor manera y buscar alternativas de solución.

**PEV\_3. ¿Cómo influye el conocimiento adquirido en el curso de programación en la forma como ahora soluciona los problemas planteados en esta y las demás asignaturas?**

Los estudiantes expresan la importancia del aprendizaje de la programación y el desarrollo del pensamiento computacional en los elementos que permiten tener alternativas en la solución de problemas. La secuenciación y el ordenamiento como componentes del pensamiento algorítmico y elementos de diseño y planeación como ejes fundamentales para conseguir buenos resultados fueron mencionados por los informantes en este apartado.

**PEV\_4. ¿Cómo aportó el curso de programación en el desarrollo académico de las materias vistas este semestre manera general?**

Nuevamente expresan la capacidad que se desarrolla para visualizar soluciones de manera diferente, permitiendo tener alternativas que consigan el mismo resultado. Se sigue identificando el desarrollo del pensamiento computacional como un factor diferencial al solucionar problemas, ayudando en elementos de habilidades inferiores como análisis y comprensión, pero también muy influyente al momento de realizar una planificación y la utilización de alternativas en las soluciones. Aparecen términos como creatividad, innovación, alternativas.

Fuente: Elaboración Propia

Un análisis entre los resultados obtenidos en la prueba diagnóstica y los resultados de la prueba validadora, el análisis de las observaciones hechas y la interpretación de las respuestas dadas en las entrevistas, mostró una alta tendencia de mejora en cuanto a la solución de los problemas para los estudiantes que han obtenido conocimientos de programación en todas las habilidades cognitivas asociadas con el pensamiento computacional, recalando avances significativos en procesos de descomposición, abstracción, generalización y evaluación. Como elemento particular se debe mencionar que el razonamiento lógico sigue mostrando un buen desempeño, es una habilidad que permanece a lo largo del tiempo en los estudiantes y que no se afecta a causa del desarrollo de habilidades asociadas al pensamiento computacional.

centrales de un lenguaje en particular, como su semántica y su sintáctica. Sin embargo, el aprendizaje de la programación en su etapa temprana no requiere destrezas específicas ni mucho menos complejas. Como efecto colateral, el aprendizaje de la programación y los aspectos que de allí se desprenden, como el uso del diseño, la división en funciones, la simulación, la experimentación, el desarrollo de la creatividad y el aprendizaje basado en problemas, entre otros, permite desarrollar el pensamiento computacional, como lo expresaron Voskoglou y Buckley (2012) en su artículo “Problem Solving and Computers in a Learning Environment”, situación que facilita la adquisición de elementos adicionales como la algoritmia, la descomposición y la generalización, entre otras. Estos elementos ayudan al estudiante a desarrollar nuevas habilidades asociadas con niveles cognitivos superiores que se relacionan con el análisis, la síntesis y la evaluación (Voskoglou & Buckley, 2012). Como aspecto retroalimentador, el desarrollo de estas habilidades cognitivas mejora el proceso de aprendizaje, no solo de la programación, sino de las demás materias, tal como se representa en la figura 4.

## Discusión

Para Herrera (2001), las habilidades cognitivas se pueden clasificar en cognitivas y meta cognitivas, donde las primeras son facilitadoras del conocimiento y operan directamente sobre los datos, incluyendo actividades de recolección, análisis, comprensión, procesamiento y almacenamiento que posteriormente son recordadas para utilizarlas según la necesidad; las segundas son las responsables de la calidad de ese conocimiento y la forma como lo aplicamos en la resolución de problemas (Herrera Clavero, 2001).

De esta manera, y en consonancia con lo expresado por algunos autores referenciados en esta investigación, se puede decir que las habilidades cognitivas inferiores sirven para resolver problemas elementales y son el soporte para aprender a programar con los elementos básicos de un lenguaje de programación. Sin embargo, dicho aprendizaje conlleva a desarrollar habilidades cognitivas de orden superior, algunas asociadas al desarrollo del pensamiento computacional, que mejoran la forma en que se da solución a los problemas y contribuyen a la obtención de un aprendizaje significativo.

Aprender a programar requiere de algunas habilidades cognitivas básicas previas que le permitan al estudiante adquirir los elementos

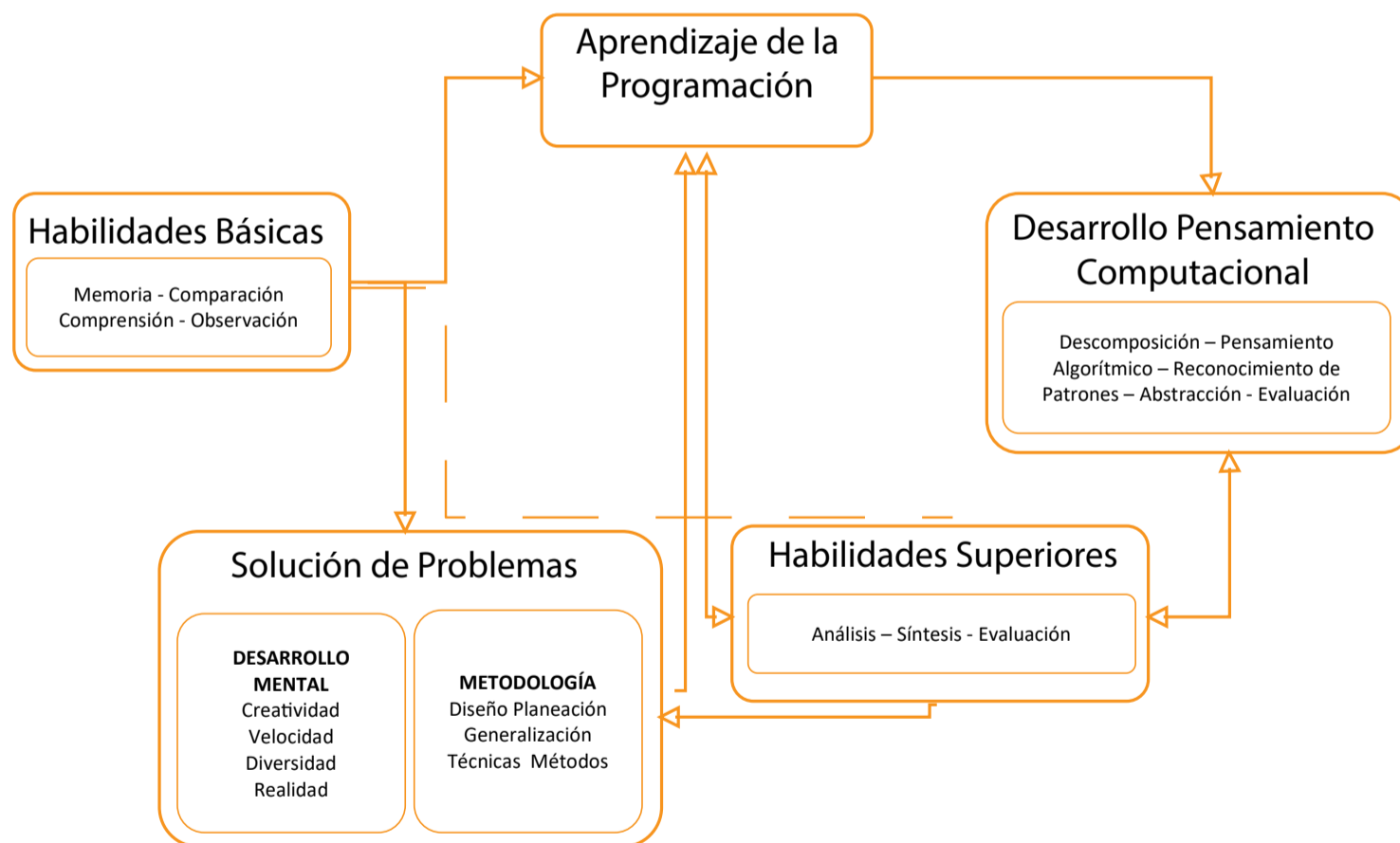


Figura 4. Mejoramiento de las Habilidades Cognitivas

Fuente: Elaboración Propia

Una síntesis final de los resultados obtenidos deja en evidencia las necesidades y fortalezas cognitivas que presentan los estudiantes antes y después de ver el curso de programación y cómo con el desarrollo del pensamiento computacional, a través del aprendizaje de la programación, estas se van potenciando. El resultado de las pruebas aplicadas, la interpretación de las respuestas dadas en las entrevistas y la observación de la forma como los estudiantes afrontaron los problemas, dejan claro que el desarrollo del pensamiento computacional mejora los elementos cognitivos superiores que contribuyen eficaz-

mente a la solución de problemas y, tal como lo mencionaba Herrera (2001), a desarrollar habilidades metacognitivas que ayudan a tener un conocimiento perdurable y de calidad.

En concordancia con los resultados de investigaciones anteriores, como la realizada por Voskoglou y Buckley (2012) en el artículo “Problem Solving and Computers in a Learning Environment”, esta investigación refuerza la teoría que una manera de potenciar el pensamiento computacional es a través del aprendizaje de la programación y que

una vez este se adquiere se desarrollan destrezas mentales de orden superior que propician la mejora de las habilidades propias para la resolución de problemas. Otro elemento que refuerza los estudios previos tiene que ver con la forma en que se afronta la solución de los problemas, es decir, las estrategias y procedimientos utilizados, planteado por Zúñiga, Rosas, Fernández et al. (2014) en su investigación. El presente estudio permitió identificar que los estudiantes expresan mejoras en la creatividad, la habilidad y la velocidad, elementos que fueron concluyentes también y que se presentaron en el artículo “Estrategia metodológica de la enseñanza de la programación para la permanencia de los alumnos de primer año de Ingeniería Electrónica” (Arellano, Rosas, Zuñiga, Fernández, & Guerrero, 2014).

## Conclusiones

Esta investigación permitió identificar que, si bien es cierto que los estudiantes inician con un conjunto básico de habilidades catalogadas como inferiores que le permiten solucionar problemas elementales, el desarrollo del pensamiento computacional, a través del aprendizaje de la programación, les aporta elementos para adquirir nuevas habilidades y destrezas, así como herramientas para mejorar las existentes, lo que se refleja en una mejora al momento de solucionar problemas. Se evidenció que el estudiante al iniciar el curso de programación tiene habilidades cognitivas básicas, como la memoria, la comprensión, el análisis y la observación; sin embargo, estos elementos se van mejorando a medida que ven los contenidos del curso y desarrollan el pensamiento computacional. Por otro lado, el estudio permitió conocer las habilidades desarrolladas por el estudiante, las cuales están asociadas a niveles cognitivos de orden superior, según los expresado por Bloom, y que corresponden a las características del pensamiento computacional, como lo son la abstracción, la síntesis y la evaluación; también que existe la adquisición de otros elementos propios del pensamiento computacional como la descomposición y la generalización (algoritmia y patrones), que el estudiante va adquiriendo a medida que desarrolla ejercicios en el computador.

Adicional a las habilidades cognitivas antes mencionadas, existen otros elementos identificados en el estudio que no están catalogados propiamente dentro de los niveles del pensamiento propuestos por Bloom, sin embargo, son aspectos que generan valor a la forma en que el estudiante soluciona problemas y que se derivan del desarrollo del pensamiento computacional. Dichos elementos se pueden clasificar como destrezas y se van adquiriendo en mayor medida con la experiencia. Dentro de estos elementos se encuentra la creatividad, la agilidad, la relación con la realidad, la diversidad y la generalización. También es importante mencionar que, además de desarrollar nuevas habilidades, el estudiante mostró mejoras en los hábitos para resolver problemas, introduciendo elementos de diseño y planeación, que le permitieron establecer un conjunto ordenado de pasos que le condujeron a un mejor resultado, evitando errores, automatizando soluciones y disminuyendo el tiempo de solución.

En cuanto a la relación que existe entre el pensamiento computacional y las habilidades que adquieren o mejoran los estudiantes al resolver problemas, se debe dejar claro que tanto el pensamiento computacional como las habilidades cognitivas superiores utilizan procesos mentales similares, en sus definiciones se encuentran componentes semejantes y fácilmente identificables como el análisis y la evaluación. Sin embargo, no todos los elementos están claramente asociados, pero se pueden relacionar a través de los procesos metacognitivos como la abstracción y la síntesis, tal como se encuentra definido en la taxonomía de Bloom.

Es claro, como se expresó en la presentación de los resultados, que las habilidades cognitivas elementales como el conocimiento, la memoria, la comprensión, la observación o la comparación, permiten solucionar problemas simples; pero su perfeccionamiento y madurez, así como la adquisición de otras más complejas, contribuyen a que dichas habilidades vayan en procura de soluciones a problemas de mayor envergadura. Una manera de ayudar a la consolidación de estas habilidades se encuentra en el pensamiento computacional que se convierte en una herramienta útil al momento de solucionar los problemas. Entonces, el desarrollo del pensamiento computacional es una manera de potenciar el desarrollo de dichas habilidades y, aunque en esencia parezca lo mismo que una simple habilidad cognitiva, se puede decir que uno es catalizador del otro y que se complementan de manera biyectiva. No se puede concebir, entonces, el desarrollo del pensamiento computacional sin la influencia de las habilidades de orden superior, pero tampoco se puede ignorar que el desarrollo del pensamiento computacional contribuye de manera directa con la mejora de dichas habilidades.

De manera general, se puede indicar que el aprendizaje de la programación y el desarrollo del pensamiento computacional le permitió al estudiante mejorar la forma en que resuelve los problemas a través de la adquisición y el mejoramiento de las diferentes habilidades, utilizando nuevos métodos y estrategias, generando alternativas de solución de manera creativa, ágil y eficiente. Son los estudiantes conscientes que el uso adecuado del conocimiento y los elementos adquiridos en los cursos de programación, así como el mejoramiento de habilidades previas como análisis y la comprensión, van a permitir optimizar la forma en que solucionan problemas.

Desarrollar el pensamiento computacional a través del aprendizaje de la programación permitirá al estudiante, no solo apropiarse los contenidos de este curso, sino mejorar en todos los aspectos académicos e inclusive utilizarlos para su desempeño personal y profesional. El pensamiento computacional es una herramienta poderosa para explorar aquellas habilidades que tienen los individuos, que no siempre son desarrolladas y que se puede aplicar de manera general en cualquier área.

## Referencias Bibliográficas

- Aho, A. V. (July de 2012). Computation and Computational Thinking. *The Computer Journal*, 55(7), 832–835. doi:10.1093/comjnl/bxs074
- Araya, V., Alfaro, M., & Andonegui, M. (2007). Constructivismo: Orígenes y Perspectivas. *Laurus, Revista de Educación*, 13(24), 76-92. Obtenido de <https://www.redalyc.org/articulo.oa?id=76111485004>
- Arellano, N., Rosas, M. V., Zuñiga, M. E., Fernández, J., & Guerrero, R. (2014). Una experiencia en la enseñanza de la programación para la permanencia de los alumnos de Ingeniería Electrónica. *Revista de Experiencias Docentes en Ingeniería de Computadores*(4), 69-80. Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=5396705>
- Fernández, J. M., Zúñiga, M. E., Rosas, M. V., & Guerrero, R. A. (2018). Experiences in Learning Problem-Solving through Computational Thinking. *Journal of Computer Science and Technology*, 18(2), 136-142. doi:10.24215/16666038.18.e15
- Gutiérrez Ávila, J. H., de la Puente Alarcón, G., Martínez González, A. A., & Piña Garza, E. (2012). *Aprendizaje Basado en Problemas: un camino para aprender a aprender*. (U. A. México, Ed.) México, DF.
- Herrera Clavero, F. (2001). *Habilidades cognitivas. Notas del departamento de Psicología Evolutiva y de la educación*. (E. Universidad de Granada, Ed.) Obtenido de Open AIU Courses: <https://cursos.aiu.edu/Desarrollo%20de%20Habilidades%20del%20Pensamiento/PDF/Tema%203.pdf>

- Martínez Miguelez, M. (2004). *Ciencias y Arte en la Metodología Cualitativa*. México: Trillas.
- Ortiz Granja, D. (2015). El constructivismo como teoría y método de enseñanza. *Sophia: colección de Filosofía de la Educación*, 19(2), 93-110. doi:10.17163/soph.n19.2015.04
- Pozo, J. I., Pérez, M., Angel, G. M., & Postigo, Y. (1994). *La solución de problemas*. Madrid: Santillana.
- Rojas, A., & García, F. J. (2020). Evaluación del pensamiento computacional para el aprendizaje de programación de computadoras en educación superior. *Revista de Educación a Distancia (RED)*, 20(63). doi: <https://doi.org/10.6018/red.409991>
- antimateo, D., Nuñez, G., & González, E. (2018). Estudio de dificultades en la enseñanza y aprendizaje en los cursos básicos de programación de computadoras en Panamá. *Revista de Investigación en Tecnologías de la Información: RITI*, 6(11), 13-18. Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=7107365>
- Terceros, I. (2019). Programación creativa: pensamiento computacional y constructivismo desde contextos interculturales. *Analysis. Claves de Pensamiento Contemporáneo*, 22, 121-125. Obtenido de <https://studiahumanitatis.eu/ojs/index.php/analysis/article/view/67>
- Velásquez Burgos, B. M., Remolina de Cleves, N., & Calle Márquez, M. G. (2013). Habilidades de pensamiento como estrategia de aprendizaje para los estudiantes universitarios. *Revista De Investigaciones UNAD*, 12(2), 23-41. doi:10.22490/25391887.1174
- Voskoglou, M. G., & Buckley, S. (2012). *Problem Solving and Computational Thinking in a Learning Environment*. The PS process: A review. *Egyptian Computer Science Journal*, 28-46. Obtenido de <https://arxiv.org/abs/1212.0750v1>
- Wing, J. M. (06 de 03 de 2011). *School of Computer Science*. Obtenido de Carnegie Mellon University: <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- Zapotecalt, J. (2018). *Introducción al pensamiento computacional: conceptos básicos para todos*. (A. Academia Mexicana de Computación, Ed.) Obtenido de <http://www.pensamientocomputacional.org/index.php/curso-pc>
- Zúñiga, M. E., Rosas, M. V., Fernández, J., & Guerrero, R. A. (2014). El desarrollo del pensamiento computacional para la resolución de problemas en la enseñanza inicial de la programación. XVI *Workshop de Investigadores en Ciencias de la Computación*, 340-343. Obtenido de <http://sedici.unlp.edu.ar/handle/10915/41352>