

# ANÁLISIS DE LOS COMPONENTES RELACIONADOS EN PROGRAMACIÓN COMPETITIVA: UN MAPEO SISTEMÁTICO DE LITERATURA

Yuli Esmeralda Sinza Diaz<sup>1</sup>, Jennyfer Estefania Oliva Caipe<sup>2</sup>, Sandra Marcela Guerrero Calva-che<sup>3</sup>

<sup>1</sup>Ingeniería de Sistemas, Universidad de Nariño

<sup>2</sup>Ingeniería de Sistemas, Universidad de Nariño

<sup>3</sup>Ingeniera de Sistemas, Universidad de Nariño

Correo de correspondencia: [yuli06043@gmail.com](mailto:yuli06043@gmail.com)

## RESUMEN

La programación competitiva se aborda como un deporte mental en el que los participantes resuelven problemas algorítmicos y matemáticos mediante la programación. El objetivo principal de este artículo es determinar las características y elementos necesarios para el entrenamiento en programación competitiva, a través de un mapeo sistemático de la literatura basado en el protocolo de Petersen, Feldt y otros. Se identificaron un total de 128 estudios, de los cuales se seleccionaron 17 para el análisis de información, siguiendo el proceso descrito en el presente documento. Como resultado, se logra conocer los aspectos relevantes relacionados con la programación competitiva y su aprendizaje, incluyendo conceptos como jueces, plataformas, veredictos, tipos de ejercicios, metodologías aplicadas y la importancia de la práctica y participación en competencias de programación. Estas investigaciones permitieron responder a seis interrogantes planteados, concluyendo la importancia de establecer un marco de trabajo para el entrenamiento en programación competitiva.

**Palabras clave:** Marco de trabajo, maratones de programación, programación competitiva, plataforma de entrenamiento.

Recibido: 12 de abril de 2023. Aceptado: 19 de Junio de 2023

Received: April 12, 2023. Accepted: June 19, 2023

## ANALYSIS OF RELATED COMPONENTS IN COMPETITIVE PROGRAMMING: A SYSTEMATIC LITERATURE MAPPING.

### ABSTRACT

*Competitive programming is approached as a mental sport in which participants solve algorithmic and mathematical problems through programming. The main objective of this article is to determine the characteristics and elements necessary for training in competitive programming, through a systematic mapping of the literature based on the Petersen, Feldt et al. protocol. A total of 128 studies were identified, of which 17 were selected for information analysis, following the process described in this document. As a result, it is possible to know the relevant aspects related to competitive programming and its learning, including concepts such as judges, platforms, verdicts, types of exercises, applied methodologies and the importance of practice and participation in programming competitions. This research allowed us to answer six questions, concluding the importance of establishing a framework for training in competitive programming.*

**Keywords:** Framework, programming marathons, competitive programming, training platform.

*Cómo citar este artículo: H. Castaño. "Análisis de los componentes relacionados en programación competitiva: un mapeo sistemático de literatura", Revista Politécnica, vol.19, no.38 pp.212-230, 2023. DOI:10.33571/rpolitec.v19n38a14*

## 1. INTRODUCCIÓN

Como parte de la formación académica de los estudiantes de Ingeniería de Sistemas, es importante resaltar todos los procesos de investigación para mejorar el aprendizaje y el desarrollo de nuevas habilidades en la creación de aplicaciones. La programación competitiva se ha convertido en un ámbito significativo en el campo laboral y de investigación, ya que las nuevas tecnologías demandan conocimientos y habilidades en aspectos fuertemente ligados con el desarrollo de software.

La programación competitiva se aborda como un deporte mental donde los participantes resuelven problemas lógicos y matemáticos mediante el uso de la programación. Se combinan dos temas importantes que son: el diseño de algoritmos y la implementación de estos. El diseño consiste en la resolución de problemas, empezando por su análisis, y creando un programa eficiente y correcto. Por otra parte, la implementación de la solución requiere habilidades en programación, ya que se ejecuta en un ambiente de desarrollo de software, donde se pone en marcha el código diseñado con anterioridad. Es necesario contar con un buen estilo de codificación que sea sencillo y conciso, pero a la vez cumpla con los requerimientos [1].

Las maratones de programación son el espacio ideal para la aplicación e implementación de estrategias de aprendizaje colaborativo, ya que brinda a los maratonistas la oportunidad de demostrar sus habilidades en solución de problemas y pensamiento lógico [2]. Por lo general en estas competencias se crean equipos máximos de tres participantes, quienes con guía de un coach resuelven una cantidad de problemas en un límite de tiempo, lo que exige, una buena asignación de roles, capacidad de razonamiento rápido y comunicación asertiva. Estas soluciones son enviadas a un juez virtual, que se encarga de dar un veredicto al código o algoritmo enviado [2]. El resultado depende de factores como; acertar los casos de prueba, que su ejecución no demore más de lo permitido, que no haya errores de compilación entre otros [3].

En todo el mundo a medida que avanzan las tecnologías, las competencias o maratones de programación se realizan cada vez más. Un ejemplo son las maratones ACM/ICPC (Association for Computing Machinery/International Collegiate Programming Contest) patrocinado por IBM con el fin de fomentar la creatividad y la innovación en la creación de nuevos programas de software. Grandes empresas de computación se interesan por vincular buenos programadores a sus equipos de desarrollo. Por lo anterior, equipos de diferentes lugares se inscriben cada año en estas competencias, con el fin de llegar a las grandes ligas. Por lo general existen varias fases para iniciar la clasificación, iniciando con competencias locales, regionales y finalmente a nivel mundial [3].

En Colombia, La Universidad Nacional, la Universidad Francisco de Paula Santander (UFPS) y la Universidad Tecnológica de Pereira, han participado en competencias de programación de gran importancia, para ello han creado sus propias estrategias y técnicas, así como semilleros dedicados a la programación competitiva. Estos espacios de entrenamiento se vuelven indispensables para el aprendizaje y a su vez buscan mejores resultados en futuras competencias, llegando a ser referentes importantes a nivel local y regional [4].

La motivación de la presente investigación es indagar a través de un mapeo sistemático de literatura (SMS) sobre los métodos, materiales, herramientas y marcos de trabajo existentes para el entrenamiento en programación competitiva.

El artículo está organizado de la siguiente manera. En la Sección 2 se describen los trabajos relacionados, en la Sección 3 se presenta el método de investigación utilizado; en la Sección 4 se presentan los resultados obtenidos; y en la Sección 5 se detallan las conclusiones.

---

## 2. TRABAJOS RELACIONADOS

A continuación, se presentan algunos trabajos de investigación que permite explorar la relación con los elementos de programación competitiva y como se desarrolla la formación de esta área de conocimiento en distintas instituciones.

En [5] se propone incentivar a los estudiantes de la UFPS a participar de espacios de programación competitiva a fin de mejorar sus habilidades algorítmicas. El estudio, logró la construcción de un marco de trabajo, siendo su elemento principal un repositorio en GitHub en donde se encuentran la documentación correspondiente a seis de las temáticas trabajadas: grafos, manejo de cadenas, geometría computacional, matemáticas, programación dinámica y optimización de entradas y salidas. Además, contiene los ejercicios propuestos y las soluciones de todos los problemas desarrollados ya sea en competencias o en las sesiones de entrenamiento. Lo anteriormente mencionado se consolidó en una plataforma de web que respalda el proyecto. Los resultados de la implementación del marco de trabajo fueron positivos. Un número significativo de estudiantes se unió al grupo de estudio, logrando gran participación en actividades planteadas, mejorando sus habilidades algorítmicas, y complementando sus conocimientos no solo para competencias si no para sus clases tradicionales.

Se encuentra muchas similitudes entre el antecedente y la presente investigación, ya que se detallan los componentes idóneos para participar en este tipo de competencias, donde su objetivo principal es promover el aprendizaje, y lograr el desarrollo de habilidades para la programación competitiva, lo que lo convierte en un ejemplo claro de que este tipo de entornos funciona y tiene resultados favorables, tanto para estudiantes, docentes y la Universidad en general.

En [6] se explica las generalidades de la programación competitiva, y da a conocer algunos detalles sobre el proyecto que se desarrolla dentro de la Universidad Católica del Oriente (UCO), dando un reporte de todas las participaciones que ha tenido en diversos tipos de competencias; nacionales y regionales; creando así una red institucional, en el cual se forma a los estudiantes para estos eventos. Durante los últimos años la UCO ha iniciado un proceso de participación en maratones de programación, con miras a visibilizar el programa de Ingeniería de Sistemas, tanto a nivel nacional, como internacional. El estudio expone algunos logros alcanzados por la Institución en temas relacionados con la programación competitiva y las metodologías usadas. Frente a esta investigación se rescata el proceso de formación que la institución implementó con sus estudiantes lo cual se podía comparar con un marco de trabajo. Aun así, no se enfoca en la creación de una herramienta propia que complemente su entrenamiento.

En [7] se presenta una investigación donde inicialmente se detalla un estado del arte relacionado con épicos de programación competitiva, destacando aspectos importantes como conceptos, las reglas a tener en cuenta dentro de una competencia, los tipos de veredictos utilizados por los jueces online y tipos de problemas. Posteriormente se describe un análisis frente a la importancia del entrenamiento tomando ejemplos de instituciones y las herramientas que han aplicado en el proceso. Por otra parte, se destaca el camino para iniciar a participar en maratones de programación y cuales serían las ventajas y desventajas de iniciar en estas competencias. Por último, se explica cuál es el papel de las Universidades en maratones tomando como referencia la participación de la Universidad de Nariño planteando la creación de un proyecto de investigación orientado a esta temática, tanto sus objetivos como las fases a llevar a cabo una vez se haya implementado. Este artículo no cuenta aún con resultados ya que se enfoca en el planteamiento, pero aporta varios conocimientos relacionados con los tópicos propios de la programación competitiva, y es a partir de ellos que se crean herramientas y metodologías para lograr mejores estrategias.

## 3. ASPECTOS METODOLÓGICOS

Para efectuar el mapeo sistemático de literatura (SMS) sobre el entrenamiento en programación competitiva, el presente estudio se orienta por el proceso propuesto por [8] (Ver **Figura 1**).

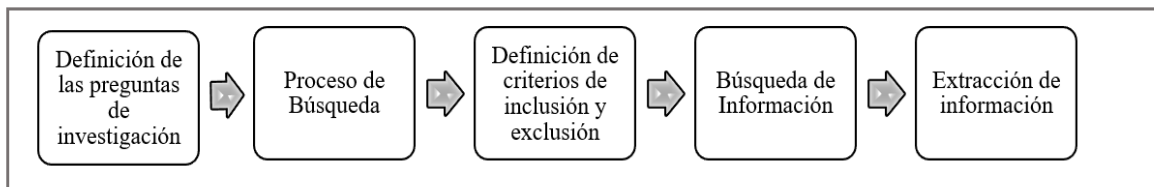


Figura 1. Proceso de investigación del SMS. Fuente: Elaboración propia.

Para la extracción de información de investigación, se consideraron las preguntas y los filtros mencionados previamente en la Figura 1. A continuación, se seguirá el proceso descrito en la Figura 2 para la recopilación de información y obtener resultados sobre los elementos clave en la programación competitiva. Estos elementos incluyen los tipos de ejercicios, los tipos de veredictos, los lenguajes y las herramientas utilizadas en las maratones de programación. Asimismo, se explorarán los métodos y materiales necesarios para la estructuración de un marco de trabajo en este ámbito

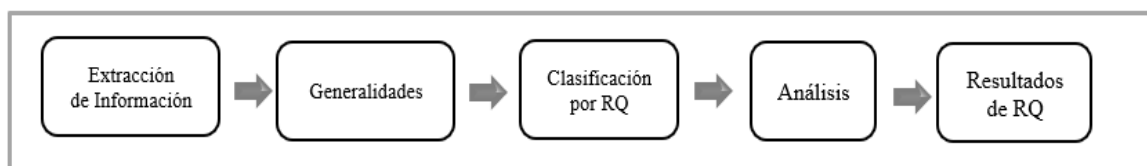


Figura 2. Proceso de recolección de información. Fuente: Elaboración propia.

### Preguntas de investigación

Las preguntas de investigación planteadas buscan información sobre el funcionamiento de jueces o plataformas de evaluación en maratones de programación, tipos de ejercicios, competencias, entre otros tópicos relacionados y poder indagar en cómo se está realizando el proceso el entrenamiento en programación competitiva.

RQ1. ¿Cuáles son los jueces online existentes para la evaluación de ejercicios en programación competitiva y para qué sirven?

RQ2. ¿Cuáles son los tipos y categorías de clasificación de los ejercicios de programación competitiva?

RQ3. ¿Qué métodos y materiales se deben tener en cuenta para la estructuración del marco de trabajo?

RQ4. ¿Qué tipos de veredictos existen en programación competitiva, en los diferentes jueces online?

RQ5. ¿Cómo prepararse para la programación competitiva, en cuanto a metodología, lenguajes, algoritmos y herramientas?

#### 3.1.1. Proceso de Búsqueda

Con base en las preguntas de investigación definidas con anterioridad se consolida una cadena de búsqueda que contemple los términos claves del objeto de estudio. Para ello, se hizo el uso de dos motores de búsqueda que fueron: ACM (Digital Library) e IEEE (Xplore Digital Library) son conocidos por tener una amplia cantidad de información, y los cuales permitieron identificar características y conceptos útiles.

La cadena fue creada para establecer los criterios de búsqueda y con expresiones booleanas OR y AND, en la cual el operador OR se lo utilizo para añadir sinónimos del concepto de búsqueda, mientras que el operador AND, se hace la integración de las palabras relacionadas en la cadena de búsqueda como se muestra a continuación en la **Figura 3**.

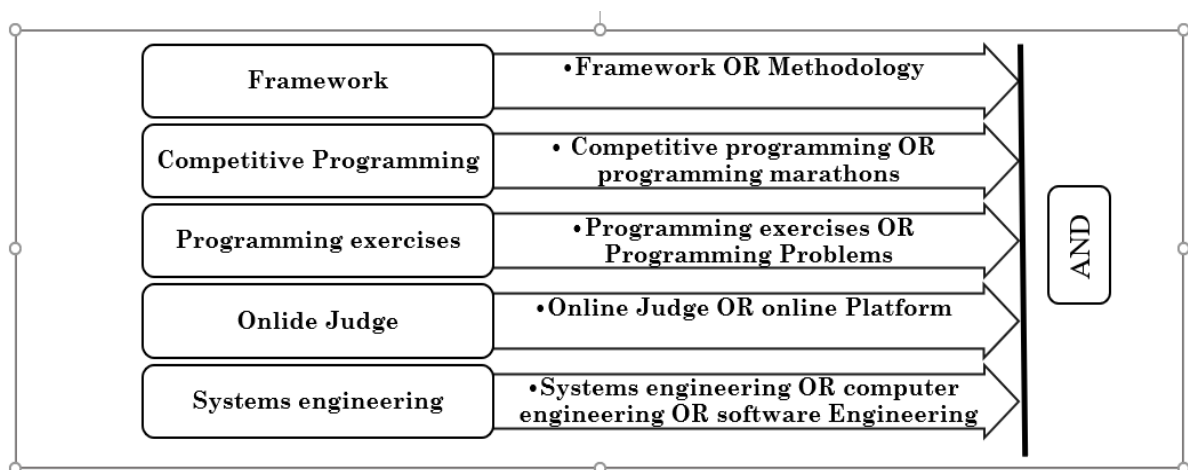


Figura 3. Cadena de búsqueda. Fuente: Elaboración propia.

### 3.1.2. Definición de criterios de inclusión y exclusión

Para la búsqueda y construcción del mapeo sistemático de literatura del presente trabajo se realizó la definición de criterios Inclusión y Exclusión detallados en la Tabla 1.

Tabla 1. Criterios de Exclusión e Inclusión. Fuente: Elaboración propia.

No.	Criterios de inclusión	Criterios de Exclusión
1	Estudios publicados entre 1971 y 2022.	Estudios publicados antes de 1971.
2	Estudios publicados en inglés y español.	Estudios publicados en idiomas diferentes al inglés y español.
3	Estudios cuyas palabras clave estén relacionadas con: Marco de trabajo, marathones de programación, programación competitiva, plataforma de entrenamiento.	Estudios Cuyas palabras clave no estén relacionadas con: Marco de trabajo, marathones de programación, programación competitiva, plataforma de entrenamiento.
4	Estudios relacionados a programación competitiva, veredictos de jueces online, Jueces online, marathones, marco de trabajo, metodologías, tipos de ejercicios.	Estudios no relacionados a programación competitiva, veredictos de jueces online, Jueces online, marathones, marco de trabajo, metodologías, tipos de ejercicios.
5	Estudios citados por otros autores.	Estudios que no esté avalados por otros autores.
6	Estudios completos, publicados en revistas, conferencias, congresos entre otros eventos de igual importancia.	Estudios duplicados o sin reconocimiento.

Además, se consolidaron 5 filtros para la revisión, análisis y extracción de información los cuales se visualizan en la Tabla 2.

Tabla 2. Filtros para el proceso de selección. Fuente: elaboración propia

No.	Criterios de inclusión	Criterios exclusión	
1F	Se hizo una revisión de los Estudios aplicando una cadena de búsqueda en las bases de datos de ACM y IEEE, teniendo en cuenta las fechas de publicación, el idioma, y áreas de conocimiento para quienes eran permitidos.	<b>1,2 y 3</b>	<b>1 y 2</b>
2F	Leer el título, Resumen y palabras clave en el documento de clasificación.	<b>4 y 5</b>	<b>3</b>
3F	Eliminar Duplicados.	<b>-</b>	<b>7</b>

4F	Leer resultados y conclusiones de cada artículo aplicando criterios de inclusión y exclusión.	5, 6 y 7	3, 5 y 6
5F	Leer artículo completo.	6 y 7	3 y

### 3.1.4. Búsqueda de Información

Con la aplicación del primer filtro, se tiene en cuenta algunos criterios de inclusión como las fechas de publicación que se sean entre 1971 y 2021. Se tomó como referencia el año 1971 ya que desde la fecha se originó el concurso más antiguo de programación competitiva: el ICPC International Collegiate Programming Contest. También se tiene presente los temas de investigación que estuvieran ligados con programación, así mismo, las palabras clave incluyan marco de trabajo, maratones de programación y programación competitiva. Terminado este proceso, se obtuvo como resultado veinticinco (25) estudios de IEEE y ciento tres (103) de ACM.

Del mismo modo, se aplica el siguiente filtro el cual consistió en leer título, resumen y palabras clave de cada uno de los resultados obtenidos, obteniendo como resultado dieciséis (16) documentos de IEEE y cuarenta (40) de ACM.

Cuando se aplica el tercer filtro, se hace la comparación entre los estudios encontrados en los dos motores de búsqueda para la eliminación de los documentos duplicados, en este proceso no se encontró ninguno, por lo tanto, en este filtro se mantuvo el mismo número de documentos de investigación.

En el penúltimo filtro se realizó la lectura de resultados y conclusiones de cada documento de investigación con el fin de tener una perspectiva más clara acerca de que estudios serían útiles para la investigación. Una vez aplicado este filtro se obtuvo diez (10) documentos de IEEE y quince (15) de ACM.

Finalmente se realiza la lectura completa de los artículos y conferencias. Al terminar se obtuvieron siete (7) estudios de IEEE y diez (10) de ACM. Las figuras 3 y 4, resumen los resultados mencionados con anterioridad.

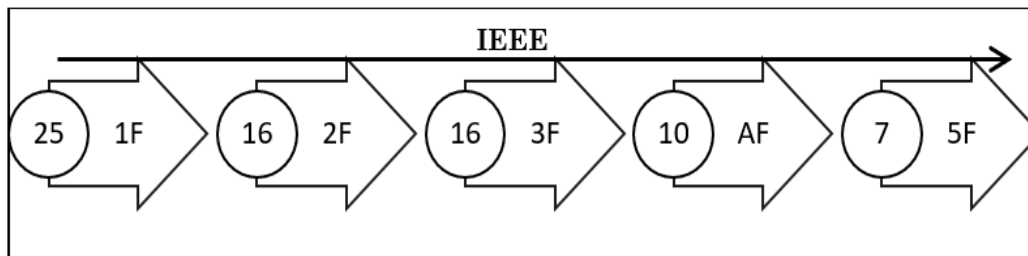


Figura 4. Resultados del motor bibliográfico IEEE Xplor. Fuente: Elaboración propia.

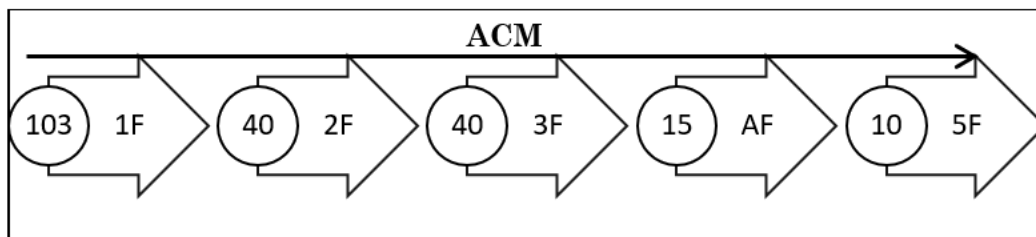


Figura 5. Resultados del motor bibliográfico ACM Digital Library. Fuente: Elaboración propia.

En la Tabla 3 se relaciona el listado de estudios seleccionados para el proceso de extracción de información sobre programación competitiva.

Tabla 3. Listado de estudios seleccionados. Fuente: Elaboración propia.

ID	Nombre de Artículo
A1[9]	Applying Gamification to Motivate Students to Write High-Quality Code in Programming Assignments
A2[10]	FLOP, a Free Laboratory of Programming.
A3[11]	SolMiner: Mining Distinct Solutions in Programs
A4[12]	Analyzing Students' Behavior in Blended Learning Environment for Programming Education
A5[13]	Agile Principles Applied in Learning Contexts
A6[14]	Introductory Programming: A Systematic Literature Review
A7[15]	New Horizons in the Assessment of Computer Science at School and Beyond: Leveraging on the ViVA Platform
A8[16]	The Boss Online Submission and Assessment System
A9[17]	MOOBench: Towards Massive Open Online Workbench
A10[5]	Development and implementation of a framework for training in competitive programming
A11[18]	A Framework for Personalized Competitive Programming Training
A12[19]	Online Judge Support for Programming Teaching
A13[20]	A Programming Contest Strategy Guide
A14[21]	UGrade: Autograder for Competitive Programming Using Contestant PC as Worker
A15[22]	Codeflex: A Web-based Platform for Competitive Programming
A16[23]	Proposal of a gamified virtual learning environment for computer programming courses
A17[24]	Facilitating Course Assessment with a Competitive Programming Platform

## 4. RESULTADOS

Por cada trabajo de investigación se obtuvieron como datos: tipo de estudio, bases de datos bibliográficas, año de publicación, países de afiliación y clasificación por contexto de investigación. En la siguiente sección se pueden detallar los resultados.

### 4.1. Generalidades

Según el tipo de estudio se encontró que el 70% son Artículos, y un 30% son Conferencias. Lo anterior quiere decir que son trabajos de tipo investigativo y apoya al presente estudio en la distribución de documentación relacionada con la innovación de esta área de conocimiento.

En cuanto a la procedencia de los estudios se identificó que el 58.82% son de ACM Digital Library y 41.17% son de IEEE Xplore. Esto indica que son bases de datos bibliográficas que hacen parte de la asociación educativa a nivel mundial, lo que lo convierte en un punto de partida para futuras exploraciones dedicadas a la temática [25].

En cuanto a la frecuencia de publicación de los artículos, se encontró que, el mayor número de estudios van desde el año 2016 hasta la actualidad (82.4%). Lo anterior se puede relacionar a que la programación competitiva se exterioriza en dos ámbitos; el laboral y el educativo y a medida que avanza la tecnología se trata de motivar a los participantes a indagar en el desarrollo de aplicaciones, logrando mejorar sus habilidades. Por otra parte, los resultados arrojan que el año 2019 fue el que destacó en el número de publicaciones con un 29.4%.

En cuanto al país de afiliación a que corresponden los artículos se destaca a Reino Unido con un 17.64%, Estados Unidos con un 11.76% y China con un 11.76% como los más relevantes. Estos países son conocidos por su gran nivel académico, además de ser potencias mundiales y pioneros

en las tecnologías de la información. Se reveló que Estados Unidos ha ganado varias de las competencias de programación competitiva a nivel mundial a través de los años, lo que refleja un interés en el desarrollo de herramientas de entrenamiento para promover competencias de programación en países de primer mundo. Esto sirve como ejemplo para empezar a implementar espacios de estudio en desarrollo de software, cada vez más en distintos lugares.

En cuanto al contexto en el cual se desarrollan las investigaciones relacionadas con programación competitiva y marcos de trabajo, el 76.24% de los estudios pertenecen al entorno académico. Esto representa una percepción positiva respecto a las temáticas de competencias en programación, donde el aprendizaje se convierte en una pieza clave y por lo tanto la creación y aplicación de proyectos basados en estas investigaciones, logran mayor aceptación e interés en programas de distintas instituciones. Además, impulsan la formación de profesionales con una visión hacia el futuro.

Finalmente, algo que fue muy útil e interesante para la investigación hace referencia a que dos de los estudios son de origen colombiano, uno de ellos perteneciente a la Universidad Francisco de Paula Santander UFPS y el otro a la Universidad Nacional de Colombia Sede Medellín, esto representa un porcentaje del 11.76% de publicaciones en relación con otros países. Generando una expectativa de interés real en Latinoamérica, además de sus participaciones en competencias incluso a nivel mundial. En el caso de la UFPS han ocupado el primer y segundo puesto en la VII Maratón de Programación Competitiva organizada por el Semillero de Investigación SILUX de la misma institución en conjunto con la Red Latinoamericana de Programación Competitiva – RPC para el año 2021. [26] Lo anterior, demuestra que el estudio de la programación competitiva trae beneficios interesantes como: una proyección académica y profesional, habilidades en análisis, diseño e implementación de soluciones de tipo algorítmico, superación intelectual y personal, oportunidades laborales, viajes y muchas experiencias, además del reconocimiento a nivel nacional [27].

## 4.2. Resultados de las preguntas de investigación

A continuación, se presenta los resultados encontrados en relación con las preguntas de investigación.

### 4.2.1. RQ1. ¿Cuáles son los jueces online existentes para la evaluación de ejercicios en programación competitiva y para qué sirven?

Al realizar el estudio correspondiente, se encontraron 6 jueces online más activos dentro del entrenamiento y la práctica de la programación competitiva. Destacando a Uva (38.9%), Codeforces (27.8%), Codechef (11.1%), HackerRank (11.1%) y Sphere Online Judge (11.1%) como los jueces con mayor repercusión al ser considerados los más completos y antiguos dentro los concursos de programación.

La ventaja de estos jueces es obtener una retroalimentación que sirve para encontrar y minimizar fallas, que posiblemente existan en los métodos de solución de ejercicios, este proceso se realiza mediante foros de discusión, pautas de entrenamiento, cursos, módulos, entre otros.

### 4.2.2. RQ2. ¿Cuáles son los tipos de ejercicio y categorías que se debe considerar en programación competitiva?

#### a. Tipos de ejercicios

Para el aprendizaje en esta área de conocimiento se consideran los siguientes tipos de ejercicios: Datos Simples (Simple Data) [10], Estructura de Control (Control Structure) [5],[18],[20]. Estructuras de datos (Data Structure) [5],[10],[18], Búsqueda Completa (Complete Search) [10],[18], Divide y Vencerás (Divide & Conquer) [18],[20], Técnicas Codiciosas (Greedy) [18],[20], Programación Dinámica (Dynamic Programming) [18],[20], Grafos (Graph) [5],[18],[20], Problemas Matemáticos (Problems Mathematics) [5],[18],[20], Geometría computacional (Computational Geometry) [18],[20], Procesamiento de Cadenas (String Processing) [5],[18],[20] y Temas avanzados (Some Harder Problems) [18],[20] información sobre programación competitiva . La descripción de cada uno de los tipos de ejercicios se puede visualizar en la **Tabla 3**.

Tabla 3. Tipos de ejercicios en programación competitiva. Fuente: Elaboración propia.



Tipo	Descripción
Simple Data	Son datos que ocupan un espacio de memoria y tiene un único valor en dicha posición, también son llamados variables y además son utilizados en la programación básica, estos pueden ser; enteros, reales, cadenas de caracteres o booleanos (int, float, string, booleano) entre otros [28].
Data Structure	Permite guardar un conjunto de datos en un mismo elemento y que se pueden combinar en enteros, reales, cadena de caracteres, entre otros, además son organizados para información masiva. Se clasifican en estructura de datos estáticas (arrays, registros, archivos) y estructuras de datos dinámica (pilas, colas, listas enlazadas, arboles, grafos etc.) [29].
Control Structure	Utilizados para tener un orden secuencial en la ejecución de algún problema con respecto a las instrucciones que se ejecuten en el programa, también son conocidos como condicionales e iterativas, algunos visto como if, while, for, do...while y bucles anidados [30].

Complete Search	Son problemas que tiene como objetivo buscar un elemento dentro de una estructura de datos de manera recursiva o en iteraciones, ya que son problemas que permite ahorrar tiempo de manera eficiente y por lo general se trabaja con grandes cantidades de datos almacenados [31].
Divide & Conquer	Son problemas que pueden dividirse en subproblemas, donde cada uno de estos se resuelve sistemáticamente para que se combinen las sub-soluciones y así obtener una solución en general. Algunos ejemplos pueden ser Torre de Hanoi (basados en la recursividad), ordenación por fusión, ordenación rápida y Multiplicación de matrices de Strassen [32].
Greedy	Trabaja con problemas de optimización, seleccionando un recurso que le muestra la solución más optima, para aplicarlo de manera general. Un claro ejemplo es el algoritmo de Kruskal (MST), donde elige en cada paso la arista más liviana que no genere un ciclo [33].
Dynamic Programming	Son problemas de optimización sobre la recursividad, teniendo en cuenta las entradas repetitivas, ya que puede utilizar los resultados generados en un subproblema y así no retornar a un resultado anterior. Esto hace que los ejercicios sean más eficientes y reduzcan su complejidad, Un ejemplo es el registro de la sumatoria de la sucesión de Fibonacci [29].
Graph	Ayuda a resolver diversos problemas ya sean de la vida cotidiana (marcando algunas secuencias de tareas o actividades), circuitos, una red social, rutas, etc. Esto debido a que un grafo tiene una conexión de vértices y enlaces que guardan información y están unidos entre ellas, esto permite hacer una búsqueda más rápida y precisa [34].
Problems Mathematics	Generalmente son las aplicaciones de técnicas matemáticas para la solución de problemas algorítmicos, combinando todo tipo de retos/ejercicios [36].
String Processing	Son algoritmos para resolver problemas de aplicación geométricos en la computación gráfica, dado que son de gran complejidad y se utilizan para la salida de un objeto geométrico, estos se aplican en la robótica, el diseño de circuitos integrados, información geográfica, patrones, entre otros [35].
Some Harder Problems	Este tipo de solución de ejercicios son la combinación de todos los problemas vistos con anterioridad, estos pueden ser matemáticas, físicas, casos de la vida diaria, lógica, matemáticas avanzadas u otros [33].

### b. Categorías de clasificación

Con respecto a las categorías de clasificación de ejercicios en programación competitiva que fueron identificadas en los estudios se resaltaron: Nivel de dificultad, Límite de tiempo y Número de Intentos.

**Nivel de Dificultad:** Esta categoría se caracteriza por medir el nivel de conocimiento de cada participante con respecto a la solución de ejercicios, ya que permite identificar que tan eficiente es su desempeño en el desarrollo de algoritmos clasificados en: sencillo, media y superior [10],[18], [20]. Esto permite que los participantes mejoren su capacidad intelectual a medida que avanzan en los cursos de programación competitiva.

**Límite de Tiempo:** Esta categoría se relaciona con el uso del tiempo ha sido importante para la participación en las diversas maratones de programación, porque ha potenciado que los participantes logren trabajar bajo presión y responder en un límite de tiempo [10], [18], [20].

**Número de Intentos:** Esta categoría indica el número de intentos permitidos que tiene cada ejercicio, de modo que le especificará a los estudiantes como realizar la manipulación de los retos que hay en las plataformas online de programación [10], también logrando identificar cuáles son los problemas más frecuentes, y al mismo tiempo verificar como es la interacción con los demás participantes.

#### **4.2.3. RQ3 ¿Qué métodos y materiales se deben tener en cuenta para la estructuración de un marco de trabajo en programación competitiva?**

Un marco de trabajo se define como un conjunto de conceptos, prácticas y criterios que se dedican a enfocar y solucionar un tipo de problemática en particular [5].

##### **a. Métodos para un marco de trabajo en programación competitiva**

Para la definición de un marco de trabajo se necesita de proyectos y actividades de enseñanza que involucren múltiples herramientas, y a la vez tienen diferentes objetivos [13]. Esto se efectúa en maratones de programación con el uso de métodos que sirven para la definición de preguntas, tareas, o retos a través de bodegas o bases de datos [15]. Todo esto con el fin de apoyar y reforzar el progreso de los estudiantes, y que además el uso de una plataforma de trabajo sirva de ayuda para los docentes [5], [15], [17], creando así una comunidad práctica con fines de facilitar el intercambio, búsqueda de conocimiento y colaboración [15]. Para lograrlo se deben definir requisitos de un modelo de aplicación [19] y la realización de etapas importantes como; planificación, seguimiento, control, verificación y validación con el fin de obtener una retroalimentación adecuada [9].

##### **b. Características de un marco de trabajo en programación competitiva**

Un marco de trabajo debe contar con algunas características importantes para su correcto funcionamiento. En primer lugar, se hace indispensable la adaptación a valores y principios que tienen como función orientar y regular las relaciones dentro de las actividades, equipos de trabajo, comunidad estudiantil, etc. [9]. Esto es fundamental para el aprendizaje en conjunto, el cual permite realizar prácticas con mayor capacidad intelectual y de buenas relaciones interpersonales [14].

Como segundo lugar se tiene en cuenta roles y tareas que cada persona involucrada debe desempeñar en el proceso, tanto profesores como estudiantes. Las funciones por parte del profesor deben ser: realizar cursos personalizados, cargar archivos de interés en una plataforma, acompañar la lectura de guías, hacer la demostración de algoritmos y fortalecer el análisis y comparación de ejercicios [5],[17]. Por otra parte, las funciones del estudiante son: Revisar el contenido continuamente, leer las guías, demostrar algoritmos, analizar y estudiar los ejercicios propuestos [5],[17].

En tercer lugar, en cuanto a la plataforma o herramienta a utilizar se debe tener en cuenta ciertas características. Como primer punto, el uso de vista de recursos y gestión de operaciones además un juez de calificación que tenga integrados lenguajes como Python, Java y C++ [5],[17]. También debe contar con un espacio de soporte para búsqueda de usuarios, ejercicios y resultados [15]. Como segundo punto debe tener aspectos importantes como la seguridad en el uso de entornos y espacios aislados para realizar envíos [23] y un buen rendimiento con la implementación de multitareas para el servidor [23].

Por último, la plataforma debe contar con un asistente de navegación que permita la manipulación de los usuarios en la plataforma. A continuación, se mencionan algunas de sus características.

---

## Características del asistente de navegación

**Elección de actividad:** Sí un usuario se encuentra sin estar registrado o haber antes tenido contacto con la aplicación, esta mostrará una interfaz amigable que logre atraer al participante hacia el uso de la plataforma, que tiene las siguientes características:

- **Aprendizaje:** donde el usuario puede observar una introducción a los problemas, y estos estén organizados por categorías, grados de dificultad y con desafíos adecuados para el nivel de cada participante [22].
- **Competencias:** Existen competencias privadas a las cuales se puede acceder de manera restringida y eventos públicos donde la participación es de libre acceso [22].
- **Organización:** visualizar una serie de competencias a las que el nuevo usuario puede inscribirse de manera individual, con amigos o equipos [22].

**Practica de programación:** Una vez dentro de la aplicación el usuario puede encontrarse con una variedad de categorías disponibles; algoritmos, estructuras de datos, aprendizaje automático, inteligencia artificial, etc.

**Problemas y categorías:** En una sección para explorar problemas se le permita al usuario acceder a una lista de retos que estén acorde con su nivel de conocimiento y clasificando el grado de dificultad. También incluye módulos de cómo se soluciona un problema en ciertas categorías y su explicación [22].

**Caracterización del problema:** Los problemas cuentan con características primordiales como descripción del problema, formatos de entrada y salida, puntaje máximo, ejemplos de casos de prueba y también la posibilidad de observar envíos y resultados obtenidos con anterioridad para ese problema en específico [22].

**Resolución de un problema:** La plataforma tiene un programa que proporciona la escritura de código y el envío de su solución [23]

**Evaluación de resultados:** Una vez enviado el código, empieza el proceso de verificación de casos de prueba, la calificación y la entrega de un veredicto [23].

### c. Materiales de un marco de trabajo

Dentro del marco de trabajo en programación competitiva y las metodologías utilizadas para su aplicación se describen los siguientes materiales.

- **Evaluaciones:** Se refiere a todas las actividades que tienen como finalidad evaluar el aprendizaje de los estudiantes en programación competitiva, a través de talleres, laboratorios, quizzes, foros, entre otros [14],[19].
- **Autoinformes:** Contempla documentación individual de los resultados de las actividades propuestas, a través del período de aprendizaje [14],[19].
- **Pruebas:** Evalúa el conocimiento del participante, mediante formatos de preguntas y ejercicios propuestos [14],[19].

Por otra parte, un marco de trabajo se subdivide en categorías [5],[17] y problemas que a su vez hacen parte de la implementación de entornos de desarrollo, [5],[17], bibliotecas de algoritmos [5],[17], bibliotecas de casos de pruebas [5],[17],[19] y módulos de capacitación [15] que sirven para clasificar de manera óptima los tópicos existentes dentro de la programación competitiva.

#### 4.2.4. RQ4. ¿Qué tipos de veredictos existentes en programación competitiva en los diferentes jueces online?

Los veredictos en programación competitiva son el sistema de juzgamiento de los programas enviados por los participantes [9]. El juez online evalúa cada ejercicio e indica el resultado final. De acuerdo con los estudios seleccionados se encontraron los siguientes tipos de veredictos para la evaluación de desarrollo de los estudiantes, como lo indica la Tabla 4.

Tabla 4. Tipos de veredictos. Fuente: Elaboración propia.

Veredictos	Definición
Accepted	Cuando el código enviado pasa todos los casos de prueba guardados en el servidor [22],[23].
Wrong Answer	Los resultados no coinciden con los casos de prueba [22],[23].
Run Time Error	Irregularidad en el tiempo de ejecución de los algoritmos [23].
Compile Error	Error de compilación [23].
Time Limit Exceeded	Excedió el tiempo máximo propuesto para la ejecución del problema [23].

Otra manera de ver los veredictos en la programación competitiva es la clasificación en el estado del estudiante, es decir medir el nivel de conocimiento, en la cual forma parte para incentivar la evolución de cada ejercicio o categorías de las plataformas. Algunos de ellos son:

**Puntos:** Generalmente es el rango de los ejercicios ya desarrollados y tiene una clasificación de 1 a 10 puntos [20],[23],[24].

**Clasificación:** Es la manera de ver como participan cada estudiante bien sea individual, binas o en equipo [20],[23],[24].

**Insignias:** Son los objetivos que obtiene en cada proceso del curso, como, por ejemplo; resolver cierta cantidad de ejercicios [20],[23],[24].

**Niveles:** Se clasifica al estudiante por sus participaciones o avances de los cursos que están desarrollando con un grado de porcentaje [20],[23],[24].

#### 4.2.5. RQ5 ¿Cómo prepararse para la programación competitiva, en cuanto a la metodología, lenguajes, algoritmos y herramientas?

##### a. Herramientas

Para el entrenamiento de la programación competitiva es importante recalcar unos criterios dentro del proceso de aprendizaje y práctica, para ello se define y aplica el Marco de la OII (Marco para el Desarrollo Algorítmico de Competencias); este marco sirve para apoyar a los estudiantes no novatos con la capacitación para los concursos de programación y ayudar a los instructores que administrarán dichos concursos. Se espera que este modelo sea compatible con un sistema donde:

- Se proporcionen problemas de programación (tareas) [19].
- Los participantes presentan una solución compuesta por la especificación un recurso ya realizado, de acuerdo con el modelo y el programa real [19].
- La retroalimentación sobre la especificación de la solución puede ayudar a desarrollar la capacidad del participante para seleccionar estrategias de obtención de buenos resultados [19].
- El aprendiz es modelado por el conjunto de problemas que encontró, por la calidad de las soluciones relacionadas y por las habilidades adquiridas durante la actividad de resolución, representadas por un conjunto de insignias [19].
- El participante puede construir su propio camino de práctica, seleccionando su propia secuencia de problemas, posiblemente aceptando sugerencias provenientes del mismo sistema [19].
- Sugerir un problema al participante individual, basándose en su modelo de estudiante actual y en el objetivo establecido para la actividad general de aprendizaje en el sistema [19].

---

En cuanto a otros criterios para este proceso se debe tener en cuenta algunas prácticas que llevan el proceso de aprendizaje de manera más satisfactoria, entre estas se encuentran: en primer lugar, comentar el código lo que permite tener una organización y entendimiento de lo que se está desarrollando, también da paso a que si se está trabajando en equipo los otros integrantes puedan comprender la sintaxis de mejor manera. En segundo lugar, se debe manejar un solo estilo de código, haciendo correcciones constantes del mismo y sus pruebas correspondientes, así se efectuará una evaluación más precisa y clara de su funcionamiento [16]. Finalmente, no consumir recursos innecesarios como tiempo, código, variables, funciones, etc. [12].

## **b. Metodologías**

A continuación, se describen algunas características importantes dentro de las metodologías de aprendizaje.

- **Clasificación:** Se trata de desarrollar un plan para especificar los resultados de la obtención de conocimiento de manera general, esto podría verse en el crecimiento en el nivel de aprendizaje asociado con los objetivos de enseñanza. De lo anterior depende que cada uno de los pasos a seguir se lleven de manera correcta y se pueda avanzar a las siguientes etapas. [10] [12].
- **Ejecución:** En este punto se ejecuta la planificación, donde se comprenden las actividades a desarrollar; cursos, seminarios, laboratorios, presentaciones, materiales y actividades de aprendizaje; a través de metodologías del aula de clase como: concursos en el aula, enseñanza en tiempo real y dinámicas para incentivar [10],[12].
- **Evaluación:** Se evalúa cada etapa y actividad con el fin de ver el progreso de los participantes a modo de asignar un puntaje y aprendizaje alcanzado [10],[12].
- **Análisis:** Una vez con los resultados de la evaluación se debe empezar por analizar todo el proceso en retrospectiva, inicialmente identificar las cosas que deben cambiarse y luego proponer practicas nuevas, buscando mejorar el proceso de enseñanza y aprendizaje. finalmente elegir niveles de dificultad, tiempo, métodos de evaluación, etc. [13],[5].

## **c. Estrategias de aprendizaje**

Aplicar buenas estrategias de aprendizaje demuestra tener resultados positivos dentro de la participación en competencias, las ventajas se describen a continuación.

**Entrenamiento individual:** Es importante que, a la hora de realizar un entrenamiento en solitario, se repliquen las condiciones de un concurso de programación, especialmente la evaluación automática, para ello existen los jueces de programación quienes ayudan a este tipo de entrenamiento. A medida que se vayan solucionando los problemas se busca que se verifiquen y almacenen los resultados correctos, para hacer uso de ellos recursivamente [21],[22].

**Sesiones de equipo:** El entrenamiento esencial incluye la ejecución de concursos reales, con todas las restricciones y reglas posibles, para así aprender a definir roles, tiempos, tareas etc. Por lo tanto, tener una sesión de discusión grupal después del concurso es de gran ayuda, ya que da paso a más de una opinión y un trabajo significativo y menos demandante. El hecho de trabajar individualmente causa que los estudiantes se sienten mentalmente cansados después de pasar cinco horas trabajando en los problemas, por lo tanto, un apoyo en equipo se vuelve primordial [21],[22].

**Cursos Universitarios:** Es importante que los estudiantes que participan en maratones de programación tengan conocimiento de los conceptos básicos de la informática, por ello es indispensable el cursar un programa Universitario que brinde aprendizaje de muchos conceptos. Los estudiantes deben aprender temáticas como: teorías de planes de estudio, ciencias de la computación, algoritmos y teorías de datos, programación avanzada, estructuras de datos, lógica matemática, teoría de gráficos, cálculos, etc. [21],[22].

**Continuidad e inversión de tiempo:** El entrenamiento se puede ver de forma grupal o individual, así se obtienen buenas estrategias dentro de un grupo de estudio, Aun así, se presenta un problema

cuando los estudiantes terminan la universidad y se gradúan, porque abandonan de manera definitiva los equipos de programación competitiva, es por ellos que se vuelve indispensable alguien que participe activamente en el manejo de los equipos de trabajo, lo que permite que el conocimiento se vaya acumulando de generación en generación, entonces se debe tener a alguien a cargo, ya sean docentes, estudiantes egresados o ex maratonistas que tengan el conocimiento necesario. Por tal razón se involucra a la comunidad universitaria de cada período, para que la participación sea activa y continúa [21],[22].

#### **4.2.6. RQ6. ¿Qué tan efectivo resulta implementar un marco de trabajo para el aprendizaje de programación competitiva?**

Revisando las fuentes bibliográficas se considera la importancia de implementar un marco de trabajo para el aprendizaje de programación competitiva, no solo para los estudiantes de Ingeniería de Sistemas, sino también para la comunidad en general y los interesados en esta área de conocimiento [9], [16]. A continuación, se describirán 4 ítems significativos tales como: Fortalezas, Beneficios, Intereses y Mejoras que se obtienen al adentrarse en el mundo de la programación competitiva.

En cuanto a las fortalezas, se encuentran las oportunidades que proveerán el marco de trabajo de manera dinámica e interactiva para el conocimiento de los estudiantes, entre las cuales se destaca: la capacidad de práctica, innovación e investigación para el desarrollo de habilidades algorítmicas, la solución de problemas y la adquisición de los conocimientos en las ciencias de la computación y programación competitiva [15].

En cuanto a los beneficios de la implementación del marco de trabajo para la programación competitiva se tiene:

- El incremento de las participaciones en las maratones, actividades y concursos de programación competitiva tanto a nivel local, regional, nacional o en las ICPC (International Collegiate Programming Contest) con la colaboración por la asociación ACM-IEEE, donde sus competencias se realizan anualmente alrededor del mundo [14],[20].
- Mayor aporte en la experticia laboral en cuanto a las pruebas técnicas de trabajo [20], ya que algunas organizaciones que buscan perfiles de programadores, entre ellas se encuentran (Google, Facebook, Microsoft) utilizan ejercicios similares a los usados en programación competitiva para el reclutamiento del personal.
- Respecto al nivel educativo se evidencia un apoyo a los estudiantes para mejorar en el desempeño en áreas afines al desarrollo de software.
- Crecimiento de las habilidades de trabajo en equipo [20], y para la resolución de problemas de programación competitiva que involucran áreas de conocimiento en lógica y matemática.

En cuanto a lo que interés se obtiene que el uso de tecnologías para el aprendizaje autónomo. ya que los estudiantes logran resolver dificultades pedagógicas de programación [16] de manera eficiente en los diversos sistemas de jueces en línea.

En las universidades se busca la iniciativa en la preparación y aplicación de seminarios, laboratorios, conferencias y pruebas [22] en programación, para mejorar el desempeño en actividades o concursos de programación competitiva [13]. Por lo general, la formación de los estudiantes se ve afectada por la falta de motivación y apoyo de los entes involucrados, lo que ocasiona el abandono y pérdida de interés en las temáticas y la poca participación en este tipo de eventos.

En los docentes se propicia la incorporación de diversos métodos para el desarrollo de software [10], [11], [12] y la comprensión de algoritmos/ejercicios, logrando mayor participación de los estudiantes en las maratones de programación.

La participación de los estudiantes en las competencias generará la estabilidad en ellas, además permite mejorar las actitudes en programación porque ayuda a la comprensión de ejercicios y la

---

resolución de los mismos considerando aspectos como el tiempo, el nivel de dificultad y la optimización de recursos.

En cuanto a las mejoras se destaca el desarrollo un mayor enfoque de conocimiento en el área de programación para los estudiantes, el cual despierta en ellos el interés por profundizar contenidos sobre estas temáticas [17]. Al mismo tiempo crea provecho en el aprendizaje por parte de los estudiantes en tópicos de programación competitiva [10],[14] Finalmente integra y enseña que los entrenamientos se deben realizar no solo individualmente sino en equipo, de modo que cuando se presenten a un concurso de programación logren sacar provecho a todas las habilidades de un grupo formado con anterioridad y de esta manera tener mejores resultados [17],[18],[20].

## 5. DISCUSIÓN

En el proceso de investigación se encontró gran cantidad de información relacionada con la programación competitiva, teniendo en cuenta los cuestionamientos planteados en un inicio. Lo anterior, sugiere términos y metodologías importantes dentro del mundo de las maratones de programación, que sirve como un punto de partida para futuras investigaciones e incluso la implementación de proyectos educativos y laborales. Se encuentra varios puntos a destacar de la información recolectada, en mayor parte a destacar el uso y beneficio del estudio realizado.

En primer lugar, en cuanto a los jueces online, que han demostrado ser un apoyo para los estudiantes en el proceso de aprendizaje y al mismo tiempo monitorear su avance, se encuentra varios jueces dedicados, donde la ventaja es obtener una retroalimentación que sirve para encontrar y minimizar fallas, que posiblemente existan en los métodos de solución de ejercicios.

En segundo lugar, es importante destacar que existen diferentes tipos y categorías de clasificación de ejercicios en programación competitiva. Esta variedad resulta relevante para orientar y organizar aspectos como la complejidad y temática de cada ejercicio propuesto, con el fin de adaptarse al nivel de aprendizaje y avance de cada participante en el estudio y práctica de la programación competitiva. De esta manera, se puede ampliar el enfoque e interés de los maratonistas, puesto que no solo se enfocarán en una única área de conocimiento, sino en aquella en la que se sientan más cómodos y en la que su rendimiento sea óptimo para asignar roles y tareas dentro del equipo.

En tercer lugar, se destacan métodos y materiales para tener en cuenta, dentro de un marco de trabajo dedicado a la programación competitiva, donde los resultados obtenidos presentan elementos organizados y sustentables dentro de un proceso de formación y aprendizaje. Lo anterior permite desarrollar habilidades en los maratonistas de manera exitosa, además de llevar un orden, logrando el interés y satisfacción de aquellos que decidan hacer parte del proceso. Finalmente propone cumplir con un objetivo propuesto, presto para mejoras, lo que lo convierte en un marco o proceso de aprendizaje adaptativo y escalable.

En cuarto lugar, es importante resaltar el proceso de evaluación dentro de las maratones de programación, ya que en un ámbito educativo y de aprendizaje es la forma en la que se puede medir las capacidades y conocimiento de los participantes o estudiantes. En el caso de la programación competitiva quien se encarga de evaluar es un juez virtual, el cual mediante veredictos da a conocer el resultado del desempeño en la solución de un problema, midiendo factores como, recursividad, tiempo y optimización. Existen otro tipo de calificación que se enfoca más en incentivar y premiar el rendimiento de los participantes, ya que los motiva a escalar y mejorar en cada una de las etapas, donde, a medida que se avanza, los ejercicios son cada vez más complejos. Por ende, el estudiante lo toma como un reto y objetivo a alcanzar. Esta calificación se realiza mediante medallas, insignias y niveles que dan al participante un rango que llega a ser significativo en sus ganas de desempeñarse y mejorar cada vez más.

En quinto lugar, se indaga en la forma y metodologías de los participantes para llevar a cabo el aprendizaje, el cual resulta ser un proceso autónomo, donde existe la guía y acompañamiento, pero la mayor parte del interés nace del estudiante y sus ganas de aprender y mejorar, esto es porque entre más tiempo el estudiante dedique a la práctica y codificación, desarrollará habilidades algorítmicas.

micas y una lógica mucho más ágil y eficaz, que es muy importante en las competencias de programación. Es entonces, que, al unir las metodologías más destacables de la documentación encontrada, se toma lo mejor de cada una y se crea una forma de aprendizaje mucho más eficaz.

Finalmente, es importante destacar que tantos beneficios trae la participación en competencias de programación. Con el análisis y resultados se encuentra que las Fortalezas, Beneficios, Intereses y Mejoras prometen desarrollar en el participante habilidades únicas, que no solo pueden ser aplicadas en el contexto de maratones de programación, sino también en otros ámbitos, ya que genera conocimientos y habilidades en ciencias de la computación, mejora la capacidad de resolución de problemas, da experiencia tanto laboral como educativa y desarrolla habilidades humanas como el trabajo en equipo. Por otra parte, beneficia a los entes encargados de promover los eventos de participación ya que genera avances en la investigación y creación de nuevas tecnologías.

En cuanto a las recomendaciones de tipo investigativo, es importante tener en cuenta que el mapeo sistemático propuesto en este trabajo no abarcó todos los posibles aspectos relevantes. Para abordar esta limitación, se sugiere ampliar la investigación utilizando otras fuentes de datos, como Scopus y ProQuest, o bibliotecas de investigación que puedan proporcionar información más actualizada sobre las maratones de programación. Es evidente que este campo ha experimentado un crecimiento significativo en los últimos años, y se espera que los proyectos dedicados a esta área aumenten considerablemente.

## 5. CONCLUSIONES

El mapeo sistemático de literatura relacionado con el análisis de los componentes en programación competitiva, utilizó dos fuentes de búsqueda de información (ACM Library e IEEE Xplore Digital Library) y permitió ver aspectos positivos frente a esta área de conocimiento, identificando diecisiete estudios ligados a herramientas, conceptos, procesos y proyectos afines, logrando obtener información para la ejecución de futuras investigaciones. Una vez hecho el estudio y al dar respuesta a las seis preguntas de investigación se obtuvo varias conclusiones interesantes.

En RQ1, se pudo conocer que existen varias herramientas dedicadas al entrenamiento en programación competitiva, creadas con distintas características, y muchas veces adaptadas a sus entornos de funcionamiento, destacando que UVA y Codeforces son los jueces online más demandantes para el aprendizaje en programación.

En RQ2, se profundizó en tipos de ejercicios y sus categorías donde se encontró que varias de las temáticas enfocadas en maratones de programación están relacionadas a las matemáticas y lógica; siendo estas áreas de conocimiento cruciales para un profesional en desarrollo de software.

En RQ3, se identificó métodos y materiales para un marco de trabajo en programación competitiva, donde se enfoca en apoyar a los estudiantes en las maratones de programación, con intercambio de ideas, búsqueda y colaboración para fortalecer el aprendizaje. Además, se estableció buenas prácticas para la solución de un problema.

En RQ4, se pudo identificar cinco métodos de calificación que a su vez se convierten en un incentivo para quienes las utilizan, siendo primordial medir el avance de quienes hacen uso de herramientas para el entrenamiento en programación competitiva.

En RQ5, fue posible identificar componentes de un marco de trabajo dedicado al entrenamiento y aprendizaje de la programación competitiva. Se concluye que no solo se trata de herramientas tecnológicas sino también del talento humano y a su vez describe elementos, metodologías y estrategias de aprendizaje que se integran para ser funcionales y tener éxito en un objetivo planteado.

En RQ6, en cuanto a la efectividad de incorporar un marco de trabajo para el entrenamiento en maratones de programación, se pudo identificar cuatro aspectos importantes: Fortalezas, Beneficios, Intereses y Mejoras.



## 6. REFERENCIAS

- [1] W. Di Luigi, G. Farina, L. Laura, U. Nanni, M. Temperini, and L. Versari, "Oii-web: An interactive online programming contest training system," *Olympiads in Informatics*, vol. 10, pp. 207–222, 2016, [doi: 10.15388/oi.2016.13](https://doi.org/10.15388/oi.2016.13).
- [2] T. Wang, X. Su, P. Ma, Y. Wang, and K. Wang, "Ability-training-oriented automated assessment in introductory programming course," *Comput. Educ.*, vol. 56, no. 1, pp. 220–226, 2011, [doi: 10.1016/j.compedu.2010.08.003](https://doi.org/10.1016/j.compedu.2010.08.003) <https://doi.org/10.1016/j.compedu.2010.08.003>
- [3] "Association for Computing Machinery (ACM) - TryEngineering.org Desarrollado por IEEE." <https://tryengineering.org/es/link/association-forcomputing-machinery-acm/> (accessed Jul. 06, 2022).
- [4] "XXXV Maratón Nacional de Programación ACIS REDIS 2021 - Resultados Finales." <https://acis.org.co/archivos/Maraton/2022/Resultados.htm> (accessed Jul. 06, 2022).
- [5] G. Yesid Lázaro Carrillo, A. Melissa Delgado León, and F. Humberto Vera Rivera, "DEVELOPMENT AND IMPLEMENTATION OF A FRAMEWORK FOR TRAINING IN COMPETITIVE PROGRAMMING," No. vol. 20, pp. 69–74, 2016 [doi: 10.1109/ithet.2018.8424620](https://doi.org/10.1109/ithet.2018.8424620)
- [6] J. F. Ca, I. U. Cat, D. Oriente, D. Ca, and S. U. Cat, "on en Maratones de Programación en la Universidad Católica de Oriente," pp. 1–6, 2016.
- [7] F. E. Córdoba, S. M. Guerrero, E. Danilo, and D. Riascos, "La importancia de un Juez Online como herramienta evaluativa para Maratones de Programación .".
- [8] B. Kitchenham *et al.*, "Systematic literature reviews in software engineering-A tertiary study," *Inf. Softw. Technol.*, vol. 52, no. 8, pp. 792–805, 2010, [doi: 10.1016/j.infsof.2010.03.006](https://doi.org/10.1016/j.infsof.2010.03.006).
- [9] R. Kasahara, K. Sakamoto, H. Washizaki, and Y. Fukazawa, "Applying Gamification to Motivate Students to Write High-Quality Code in Programming Assignments," 2019, [doi: 10.1145/3304221](https://doi.org/10.1145/3304221).
- [10] M.-J. Laakso, R. D. McCartney, Itä-Suomen yliopisto, Association for Computing Machinery. Special Interest Group on Computer Science Education, University of Connecticut, and ACM Digital Library., *12th Koli Calling International Conference on Computing Education Research: proceedings: November 15-18, 2012*. <https://doi.org/10.1145/2401796.2401807>
- [11] L. Luo and Q. Zeng, "SolMiner: Mining distinct solutions in programs," in *Proceedings - International Conference on Software Engineering*, May 2016, pp. 481–490. [doi: 10.1145/2889160.2889202](https://doi.org/10.1145/2889160.2889202).
- [12] J. Luo and T. Wang, "Analyzing students' behavior in blended learning environment for programming education," in *ACM International Conference Proceeding Series*, Sep. 2020, pp. 179–185. [doi: 10.1145/3425329.3425346](https://doi.org/10.1145/3425329.3425346).
- [13] J. Budu, "Applying agile principles in teaching undergraduate information technology project management," *Int. J. Inf. Commun. Technol. Educ.*, vol. 14, no. 3, pp. 29–40, Jul. 2018, [doi: 10.4018/IJICTE.2018070103](https://doi.org/10.4018/IJICTE.2018070103).
- [14] A. Luxton-Reilly *et al.*, "Introductory programming: A systematic literature review," in *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, Jul. 2018, pp. 55–106. [doi: 10.1145/3293881.3295779](https://doi.org/10.1145/3293881.3295779).
- [15] D. Giordano, F. Maiorana, A. Csizmadia, S. Marsden, C. Riedesel, and S. Mishra, "New horizons in the assessment of computer science at school and beyond: Leveraging on the ViVA platform," in *ITiCSE-WGP 2015 - Proceedings of the 2015 ITiCSE Conference on Working Group Reports*, Jul. 2015, pp. 117–147. [doi: 10.1145/2858796.2858801](https://doi.org/10.1145/2858796.2858801).

- [16] M. Joy, N. Griffiths, and R. Boyatt, "The BOSS Online Submission and Assessment System."
- [17] W. Wang, L. Huang, D. Guo, Y. Xu, F. Bi, and J. Tang, "Moobench: Towards massive open online workbench," May 2019. [doi: 10.1145/3321408.3322853](https://doi.org/10.1145/3321408.3322853).
- [18] Boğaziçi Üniversitesi and Institute of Electrical and Electronics Engineers, *ITHET 2018: 2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET): April 26-28, 2018, Olhao, Portugal*.
- [19] J. Hidalgo-Cespedes, G. Marin-Raventos, and M. E. Calderon-Campos, "Online Judge Support for Programming Teaching," in *Proceedings - 2020 46th Latin American Computing Conference, CLEI 2020*, Oct. 2020, pp. 522–530. [doi: 10.1109/clei52000.2020.00067](https://doi.org/10.1109/clei52000.2020.00067).
- [20] A. Bloomfield and B. Sotomayor, "A programming contest strategy guide," in *SIGCSE 2016 - Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, Feb. 2016, pp. 609–614. [doi: 10.1145/2839509.2844632](https://doi.org/10.1145/2839509.2844632).
- [21] Universitas Tanjungpura, Institut Teknologi Bandung, Institute of Electrical and Electronics Engineers. Indonesia Section, and Institute of Electrical and Electronics Engineers, *Proceedings of 2019 International Conference on Data and Software Engineering (ICoDSE): Gedung Konferensi Universitas Tanjungpura, Pontianak, Indonesia, November 13th-14th, 2019*.
- [22] A. Rocha, Associação Ibérica de Sistemas e Tecnologias de Informação, and Institute of Electrical and Electronics Engineers, *2019 14th Iberian Conference on Information Systems and Technologies (CISTI): proceedings of CISTI'2019 - 14th Iberian Conference on Information Systems and Technologies: 19 to 22 of June 2019, Coimbra, Portugal*.
- [23] Panepistēmio Peiraiōs, IEEE Education Society, and Institute of Electrical and Electronics Engineers, *Proceedings of 2017 IEEE Global Engineering Education Conference (EDUCON): date and venue: 25-28 April 2017, Athens, Greece*.
- [24] D. Coore and D. Fokum, "Facilitating course assessment with a competitive programming platform," in *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, Feb. 2019, pp. 449–455. [doi: 10.1145/3287324.3287511](https://doi.org/10.1145/3287324.3287511).
- [25] "Association for Computing Machinery (ACM) - TryEngineering.org Desarrollado por IEEE." <https://tryengineering.org/es/link/association-for-computing-machinery-acm/> (accessed Jul. 06, 2022).
- [26] "XXXV Maratón Nacional de Programación ACIS REDIS 2021 - Resultados Finales ." <https://acis.org.co/archivos/Maraton/2022/Resultados.htm> (accessed Jul. 06, 2022).
- [27] "Programación, la carrera que más demandará el mercado laboral en el 2021 | Tendencias | Portafolio." <https://www.portafolio.co/tendencias/programacion-la-carrera-que-mas-demandara-el-mercado-laboral-en-el-2021-548289> (accessed Jul. 06, 2022).
- [28] "Tipos de datos simples - PLEDIN 3.0." <https://plataforma.josedomingo.org/pledin/cursos/programacion/curso/u06/> (accessed Aug. 21, 2022).
- [29] "Data Structures - Dynamic Programming." [https://www.tutorialspoint.com/data\\_structures\\_algorithms/dynamic\\_programming.htm](https://www.tutorialspoint.com/data_structures_algorithms/dynamic_programming.htm) (accessed Aug. 21, 2022).
- [30] "Estructuras de Control · ciencia-de-datos-con-r." [https://rsanchezs.gitbooks.io/ciencia-de-datos-con-r/content/estructuras\\_control/estructuras\\_control.html](https://rsanchezs.gitbooks.io/ciencia-de-datos-con-r/content/estructuras_control/estructuras_control.html) (accessed Aug. 21, 2022).
- [31] "Algoritmos de Búsqueda." <http://artemisa.unicauca.edu.co/~nediaz/EDDI/cap02.htm> (accessed Aug. 21, 2022).

- 
- [32] “Significado del algoritmo divide y vencerás: Explicado con ejemplos.” <https://www.freecodecamp.org/espanol/news/significado-del-algoritmo-divide-y-venceras/> (accessed Aug. 21, 2022).
- [33] “Data Structures - Greedy Algorithms.” [https://www.tutorialspoint.com/data\\_structures\\_algorithms/greedy\\_algorithms.htm](https://www.tutorialspoint.com/data_structures_algorithms/greedy_algorithms.htm) (accessed Aug. 21, 2022).
- [34] “GRAFOS.” <https://ccia.ugr.es/~jfv/ed1/c++/cdrom4/paginaWeb/grafos.htm> (accessed Aug. 21, 2022).
- [35] C. Science, “Geometry in competitive programming,” 2017.
- [36] “Sesión 1.3 - Tipos de problemas.” <https://onlinejudge.inf.um.es/cursos/leccion1.3.html> (accessed Jan. 27, 2023).
- [37] Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, pp. 68–77 (2008). <https://dl.acm.org/doi/10.5555/2227115.2227123>

