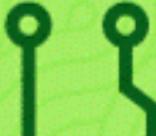


Retos avanzados de pensamiento computacional

Grado sugerido: Once

Jonny Alberto Campo Herrera

Publicado en el Banco Virtual de Recursos de Colombia Programa en el año 2025.



Este material se comparte bajo la licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0). Puede copiar y redistribuir el material en cualquier medio o formato, siempre que dé el crédito adecuado al autor, no lo use con fines comerciales, y no remezcle, transforme o cree a partir del material.

Para más información, consulte la licencia completa en [Deed - Attribution-NonCommercial-NoDerivatives 4.0 International - Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Para contactar al autor/a de este recurso, escriba a: ing.jhonnycampo@gmail.com

RETO BEBRAS
Retos Avanzados de Pensamiento Computacional Grado 11° IE San Miguel
Aguachica – Cesar.

Instrucciones para quien desarrolla el reto	<p>Se presentan a continuación 5 retos para ser resueltos haciendo uso de los conocimientos previos y aplicando pensamiento computacional para conseguir el objetivo.</p> <p>Puedes usar papel para tomar notas, dibujar. Escoge la respuesta correcta de acuerdo al razonamiento lógico que realices.</p>
Nivel de dificultad	<i>Difícil</i>
Preguntas, desafíos o retos	<p>1. “Imagina que eres el programador para un robot de exploración. Este sólo podrá avanzar si se siguen unas reglas matemáticas programables” ¿eres capaz de recorres todos los lugares sin perderte?</p> <p>Un robot está en un tablero de 5x5 casillas. Solo puede pararse en casillas donde la suma de los dígitos de la fila y la columna es par.</p> <p>Pregunta 1: ¿Cuántas casillas puede visitar el robot?</p> <p>a) 10 b) 13 c) 15 d) 20</p> <p>Pregunta 2: ¿Cuál de los siguientes pseudocódigos permite al robot recorrer todas las casillas posibles sin repetir ninguna?</p> <p>a. Para i de 0 a 4: <i>Para j de 0 a 4: Si $(i + j) \% 2 == 0$: Visitar (i, j)</i></p> <p>b. Para i de 0 a 4: <i>Para j de 0 a 4: Visitar (i, j)</i></p> <p>c. Para i de 0 a 4: <i>Si $i \% 2 == 0$: Visitar (i, i)</i></p>

2. “Eres un detective de patrones numéricos. Observa con lupa la secuencia y descifra la regla secreta que la genera. ¿Podrás predecir el siguiente número y explicar el misterio?”

Observa la secuencia:

2, 4, 12, 48, 240, _____,

Pregunta 1: ¿Cuál es el siguiente número?

- a) 720
- b) 1,440
- c) 1,200
- d) 2,400

Pregunta 2: ¿Cuál es la regla general de la secuencia?

- a) Multiplicar por 2 cada vez
- b) Multiplicar por el número de posición (empezando en 2)
- c) Sumar el doble del anterior
- d) Multiplicar por un número que aumenta en 1 cada vez

3. “Ponte en el lugar de un criptógrafo: tienes sobres con sumas y relaciones ocultas.

Usa tu lógica para descubrir cuánto hay en cada uno y diseña un método que funcione para cualquier cantidad total.”

Tres sobres:

- A tiene el doble que B
- B tiene la mitad que C
- A + B + C = 84

Pregunta 1: ¿Cuánto hay en el sobre B?

- a) 14
- b) 16.8
- c) 21
- d) 28

Pregunta 2: ¿Cuál es el pseudocódigo correcto para calcular B si la suma total es T?

- a. $B = T / 5$
- b. $B = T / 3$
- c. $B = T / 2$

4. “Hoy eres un “cazador de bugs”. Un algoritmo tiene un error escondido: encuéntralo, corrígetlo y explica cómo lo harías funcionar correctamente. ¡No dejes que el error te engañe!”

Un algoritmo intenta calcular el factorial de un número n:

```
resultado = 0
para i desde 1 hasta n:
    resultado = resultado * i
retornar resultado
```

Pregunta 1: ¿Cuál es el error?

- a) El ciclo está mal
- b) El resultado inicial debe ser 1
- c) Falta una suma
- d) No hay error

Pregunta 2: ¿Cuál es el algoritmo corregido?

- a.

```
resultado = 1
para i desde 1 hasta n:
    resultado = resultado * i
retornar resultado
```
- b.

```
resultado = 0
para i desde 1 hasta n:
    resultado = resultado + i
retornar resultado
```
- c.

```
resultado = n
para i desde 1 hasta n:
    resultado = resultado * i
retornar resultado
```
- 5. “Te conviertes en un estratega de rutas. Debes encontrar el camino más eficiente para visitar todas las casas, usando las reglas de movimiento permitidas.”
¿Podrás optimizar tus pasos y justificar tu estrategia?

Un repartidor debe visitar 6 casas alineadas. Solo puede avanzar a casas adyacentes.

Pregunta 1: ¿Cuál es el menor número de pasos para visitar todas las casas al menos una vez?

- a) 4
- b) 5
- c) 6
- d) 7

Ahora, el repartidor puede saltar hasta dos casas en un solo paso.

Pregunta 2: ¿El mínimo de pasos cambia?

- a) Sí, ahora puede hacerlo en 3
- b) No, sigue siendo 5
- c) Sí, ahora puede hacerlo en 4
- d) No, sigue siendo 6

Respuestas correctas y retroalimentación

Reto 1: El Laberinto de Condiciones

- **Pregunta 1:**
Respuesta correcta: b) 13
Retroalimentación:

El robot puede estar en las casillas donde la suma de los dígitos de la fila y la columna es par. Si recorres la matriz 5x5 y sumas los dígitos de

	<p>cada posición, verás que hay exactamente 13 casillas que cumplen esta condición. Este ejercicio desarrolla la habilidad de descomposición y análisis lógico, fundamentales en pensamiento computacional.</p> <ul style="list-style-type: none"> • Pregunta 2: Respuesta correcta: A Retroalimentación: <i>El pseudocódigo A recorre todas las posiciones del tablero y verifica la condición de suma par antes de visitar la casilla. Así, garantiza que el robot solo visite las casillas válidas y no repita ninguna, aplicando pensamiento algorítmico y lógica condicional.</i> <p>Reto 2: Secuencia Misteriosa</p> <ul style="list-style-type: none"> • Pregunta 1: Respuesta correcta: b) 1,440 Retroalimentación: <i>La secuencia multiplica cada número por un valor que aumenta en 1 cada vez (2, 3, 4, 5, 6...). Así, $240 \times 6 = 1,440$. Identificar patrones numéricos es clave en la abstracción y la generalización, habilidades esenciales en computación y matemáticas.</i> • Pregunta 2: Respuesta correcta: d) Multiplicar por un número que aumenta en 1 cada vez Retroalimentación: <i>Reconocer cómo evoluciona la secuencia permite deducir la regla general. Este tipo de razonamiento ayuda a abstraer información relevante y construir fórmulas o algoritmos.</i> <p>Reto 3: Variables Encapsuladas</p> <ul style="list-style-type: none"> • Pregunta 1: Respuesta correcta: b) 16.8 Retroalimentación: <i>Si planteas las ecuaciones según las relaciones dadas ($A = 2B$, $B = 0.5C$, $A + B + C = 84$), descubrirás que $B = 16.8$. Resolver problemas con variables y relaciones fomenta la abstracción y el modelado algorítmico.</i> • Pregunta 2: Respuesta correcta: A Retroalimentación: <i>El pseudocódigo correcto es $B = T / 5$, ya que al sustituir las relaciones en la suma total se obtiene esta fórmula. Saber traducir relaciones matemáticas a pseudocódigo es una habilidad clave para la programación.</i> <p>Reto 4: Depuración de Algoritmos</p> <ul style="list-style-type: none"> • Pregunta 1: Respuesta correcta: b) El resultado inicial debe ser 1 Retroalimentación: <i>El factorial se calcula multiplicando por cada número desde 1 hasta n,</i>
--	--

	<p>por lo que el valor inicial debe ser 1 y no 0. Detectar y corregir errores en algoritmos es fundamental para el desarrollo de software confiable.</p> <ul style="list-style-type: none"> • Pregunta 2: Respuesta correcta: A Retroalimentación: <i>El algoritmo corregido inicia resultado en 1 y multiplica sucesivamente por cada i. Esta capacidad de depuración y mejora es esencial para cualquier programador.</i> <p>Reto 5: Optimización de Rutas</p> <ul style="list-style-type: none"> • Pregunta 1: Respuesta correcta: b) 5 Retroalimentación: <i>Para visitar todas las casas alineadas moviéndose solo a casas adyacentes, el mínimo de pasos es 5. Este reto ejercita la descomposición y la optimización de procesos.</i> • Pregunta 2: Respuesta correcta: b) No, sigue siendo 5 Retroalimentación: <i>Aunque el repartidor puede saltar hasta dos casas, el mínimo de pasos para cubrir todas las casas sigue siendo 5. Analizar restricciones y buscar rutas óptimas es una habilidad clave en la resolución de problemas complejos.</i>
Adaptaciones	<p>Podríamos ajustar la actividad de acuerdo a algunos factores:</p> <p><i>Zona rural:</i> <i>En lo posible material físico o impreso, uso de programación desconectada, proyectos se contextualicen a su entorno.</i></p> <p><i>Población con discapacidad:</i> <i>Personalizar las actividades y ajustarlas a su ritmo y desarrollo de trabajo colaborativo.</i></p> <p><i>Sin internet:</i> <i>Promover el uso de software offline.</i></p>
Referencias	<ul style="list-style-type: none"> - Chicos.net. (2023). <i>Desafío Bebras - Pensamiento Computacional en la escuela [PDF]</i>. https://www.chicos.net/bebras/assets/Gu%C3%ADa%202023%20-%20C3%91and%C3%BA%20final.pdf - Castillo, E. L. (2023). <i>Estrategias educativas para la enseñanza del pensamiento computacional: Una revisión sistemática</i>. <i>Ciencia Latina Revista Científica Multidisciplinaria</i>, 7(4), 8833-8851. https://doi.org/10.37811/cl_rcm.v7i4.7590