

**UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA**  
**FACULTAD DE FILOSOFÍA Y LETRAS**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

---



**IMPLICACIONES DE LA TRANSFERENCIA DEL PENSAMIENTO  
COMPUTACIONAL EN INTERVENCIONES EDUCATIVAS**

POR:

**SANDRA ERIKA GUTIÉRREZ NÚÑEZ**

**TESIS PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO DE  
DOCTORA EN EDUCACIÓN, ARTES Y HUMANIDADES**

**CHIHUAHUA, CHIH. MÉXICO**

**SEPTIEMBRE, 2022**



Implicaciones de la transferencia del pensamiento computacional en intervenciones educativas. Tesis presentada por Sandra Erika Gutiérrez Núñez como requisito parcial para obtener el grado de Doctora en Educación, Artes y Humanidades. Ha sido aprobado y aceptado por:

Javier H. Contreras O

Dr. Javier Horacio Contreras Orozco  
Director de la Facultad de Filosofía y Letras

Dr. Jorge Alan Flores Flores  
Secretario de Investigación y Posgrado

Dr. Erslem Armendáriz Núñez  
Coordinador Académico

Dr. Pedro Barrera Valdivia  
Presidente

Fecha: 02 Septiembre de 2022

Comité:

Vocal 1: Dr. Arturo Olivarez  
Vocal 2: Dra. Aixchel Cordero Hidalgo  
Vocal 3: Dra. Paola Margarita Chaparro Medida  
Secretario: Dr. Efraín Alfredo Barragán Perea

© Derechos Reservados

Sandra Erika Gutiérrez Núñez,  
(Circuito de Terranova, 2886).

Fecha: Septiembre, 2022

# Implicaciones de la Transferencia del Pensamiento Computacional en Intervenciones Educativas

Sandra Erika Gutiérrez Núñez

Universidad Autónoma de Chihuahua

## Notas del Autor

Sandra Erika Gutiérrez Núñez  <https://orcid.org/0000-0002-4246-620X>

Los datos y el contenido de esta tesis se comparten en acceso abierto en el repositorio de la Universidad Autónoma de Chihuahua: <http://repositorio.uach.mx/>

En esta tesis no existe conflicto de intereses.

Proyecto financiado por el programa de becas de posgrados nacionales del Consejo Nacional de Ciencia y Tecnología (CONACYT), a través del Programa Nacional de Posgrados de Calidad (PNPC).

La correspondencia relacionada con esta tesis debe dirigirse a Sandra Gutiérrez. Correo electrónico: [sandraerigutierrez@gmail.com](mailto:sandraerigutierrez@gmail.com)

**Resumen**

*Palabras clave:* Pensamiento computacional, pensar como computadora, pensamiento para la informática, resolución de problemas, educación en ciencias computacionales, transferencia de conocimiento.

La presente tesis constituye un intento por entender cómo funciona la Transferencia del Conocimiento (T) y cómo podemos aplicar este mecanismo para lograr que los niños de 12 y 13 años transfieran el Pensamiento Computacional (PC) a contextos distintos y distantes de las Ciencias Computacionales (CC).

Para lograrlo, se realizó una investigación con enfoque cuantitativo de corte experimental y diseño factorial 2x2. De este modo se ofreció una intervención en Pensamiento Computacional al cien por ciento de los participantes en el estudio, y tras aplicarles una prueba de pensamiento computacional se les clasificó en grupos con alto o bajo nivel de pensamiento computacional. Posteriormente la primera mitad de cada uno de estos grupos se asignó al grupo de control y la segunda mitad al grupo de tratamiento.

Al grupo de tratamiento se le ofreció un Entrenamiento en Transferencia del Pensamiento Computacional (ETPC) y finalmente se reunió nuevamente a todos los participantes (tanto del grupo control como del grupo de tratamiento) para evaluar el Nivel de Transferencia del Pensamiento Computacional (NTPC) que logró cada uno de ellos.

El análisis de los resultados se realizó mediante métodos paramétricos y no paramétricos y fueron discutidos bajo la luz de la Taxonomía de Transferencia Lejana de Barnett y Ceci (2002) a partir de los pilares del pensamiento computacional propuestos por Jeanette Wing (2006; 2011).

Los resultados de la investigación demostraron que el factor esencial para lograr la Transferencia del Pensamiento Computacional es el Nivel de Pensamiento Computacional de los participantes y que la Transferencia fue lograda por el cien por ciento de los participantes en el estudio en diferentes elementos del pensamiento computacional, aunque la mayoría de ellos solo logró la transferencia cercana y los menos lograron la transferencia lejana.

Estos descubrimientos aportan evidencia de que la habilidad de resolución de problemas del Pensamiento Computacional a ámbitos distintos de las ciencias computacionales es posible si se ofrece un entrenamiento intensivo tanto en los elementos constitutivos del Pensamiento Computacional, como en las técnicas y estrategias que faciliten la Transferencia, por lo que las

planeaciones de clase o talleres instruccionales que se realicen en este sentido deben plantearse la transferencia como objetivo.

**Abstract**

*Keywords:* computational thinking, computer-like thinking, computational-informatics thinking, problem solving, computer science education, and knowledge transfer.

This thesis constitutes an attempt to understand how Knowledge Transfer (T) works and how we can apply this mechanism to get 12- and 13-year-old to transfer Computational Thinking (CT) to different and distant contexts of Computer Science (CS).

To achieve this, an investigation was carried out with a quantitative approach of an experimental nature and a 2x2 factorial design. In this way, an intervention in Computational Thinking was offered to one hundred percent of the study participants, and after applying a computational thinking test, they were classified of participants with a high or low level of computational thinking. The first half of each of these groups was then assigned to the control group and the second half to the treatment group.

The treatment group was offered Computational Thinking Transfer Training (CTTT) and finally all the participants (both control and treatment group) were brought together again to assess their Computational Thinking Transference Level (CTTL).

The analysis of the results was carried out using parametric and non-parametric methods and they were discussed under the light of the Far Transfer Taxonomy of Barnett and Ceci (2002) based on the pillars of computational thinking proposed by Jeanette Wing (2006; 2011).

The results of the research showed that the essential factor to achieve the Transfer of Computational Thinking is the Level of Computational Thinking of the participants and that the Transfer was achieved by one hundred percent of the participants in the study in different elements of computational thinking, although most of them only achieved the near transfer and the fewest achieved the far transfer.

These discoveries provide evidence that the problem-solving skills from Computational Thinking can be transferred into different knowledge areas if intensive training is provided in the constituent elements of Computational Thinking, as well as in the techniques and strategies that facilitate Transfer. Therefore, the class planning or instructional workshops that are carried out in this sense should consider transfer as an objective.

**Agradecimientos y Reconocimientos**

Esta tesis doctoral es la culminación de un intenso periodo de trabajo al cual me dediqué con mucha pasión y solo me fue posible llegar a este momento gracias a la ayuda de las personas que estuvieron conmigo en esta etapa.

Mi agradecimiento en primer lugar a mi directora de tesis, la Dra. Aixchel Cordero Hidalgo, por todo el acompañamiento, asesoría, ayuda y retroalimentación que me brindó desde el día en que me convertí en su tesista ya que sin ella no me hubiera sido posible llegar a hasta aquí.

Gracias también al Dr. Arturo Olivarez, mi co-director, por guiarme a través del análisis estadístico para obtener los resultados de los ejercicios realizados, así como por su pronta atención y su disponibilidad para colaborar con mi trabajo de tesis. Su valiosa retroalimentación me permitió presentar el día de hoy esta investigación de manera más completa.

Gracias a mi comité tutorial formado por el Dr. Pedro Barrera Valdivia, la Dra. Paola Margarita Chaparro Medina y el Dr. Efraín Alfredo Barragán Perea por sus valiosas observaciones y los consejos que me dieron para mejorar mi trabajo de tesis, pero principalmente por el aliento que siempre me infundieron para seguir trabajando en él.

Gracias a mi hija preciosa, Mía Simoné Nacif Gutiérrez, porque ella me dio todos los días toda la motivación que necesitaba para seguir trabajando y hacer todo lo que fuera necesario al pensar que eso contribuiría a su bienestar, y gracias a mi papá, Enrique Gutiérrez Magallanes, y a mi mamá, Sandra Luz Núñez Soto, por interesarse siempre en las cosas que hago.

Gracias, por supuesto, a Conacyt por la beca que me otorgó y que me permitió dedicarme exclusivamente a trabajar en mi tesis y gracias a todos los profesores del Doctorado en Educación, Artes y Humanidades (DEAH) que me dieron clase por su excelente calidad humana y alto nivel académico.

Mi agradecimiento muy especial al Mtro. Gerónimo Mendoza porque fue quien me motivó en primer lugar para realizar este posgrado y me brindó acompañamiento personalizado, como solía hacer con todos los estudiantes, hasta que culminé mi proceso de inscripción al DEAH y más tarde me apoyó cuando decidí hacer cambios radicales en mi tema de investigación.

Gracias incluso a la pandemia de Covid-19 porque me dio la oportunidad de hacer un doctorado y al mismo tiempo estar en la casa con mi pequeña hija de 3 años. Fue una etapa

complicada a nivel mundial, pero a mí me permitió ser muy productiva y salir casi todas las tardes a jugar al parque con mi hija.



**Contenido**

Introducción ..... 11

Capítulo 1. Diseño de la Investigación..... 12

    Antecedentes ..... 12

    Problema de Investigación ..... 14

    Objetivos del estudio ..... 16

    Hipótesis ..... 16

    Preguntas de investigación ..... 17

    Justificación y Delimitación del Estudio..... 18

Capítulo 2. Marco Teórico..... 22

    Conceptos esenciales sobre PC..... 22

    Antecedentes Históricos del PC ..... 23

    Marcos de Referencia para PC ..... 24

    PC como Habilidad de Resolución de Problemas ..... 25

    PC y su Relación con Otros Tipos de Pensamiento..... 27

        Pensamiento Convergente y Divergente ..... 28

        Pensamiento Lógico-Creativo ..... 28

        Pensamiento Abstracto..... 29

        Pensamiento Crítico-Racional-Creativo ..... 29

        Pensamiento Analógico ..... 29

        Pensamiento Disciplinar o por Área del Conocimiento ..... 30

        Pensamiento Matemático..... 30

        Pensamiento Algorítmico..... 30

    Iniciativas Mundiales sobre Programas de PC..... 30

        Estados Unidos..... 30

        Europa ..... 31

        Francia..... 32

    Experiencias en Latinoamérica..... 32

        México..... 32

        Argentina..... 33

        Chile ..... 33

    Políticas y Prácticas Relacionadas con el PC..... 34

Transferencia del PC ..... 37

Técnicas que favorecen la Transferencia del PC ..... 48

Estudios Sistemáticos Realizados en Este Campo ..... 49

Capítulo 3. Método..... 55

    Variables.....57

        Dimensiones e indicadores de la VI1 NPC ..... 57

        Dimensiones e indicadores de la VI2 ETPC ..... 57

        Covariable..... 60

Contexto de la investigación ..... 61

Población y Muestra..... 61

Procedimiento ..... 62

Instrumentos ..... 65

Actividades para evaluar la transferencia del PC..... 66

Análisis estadísticos ..... 74

Capítulo 4. Análisis de Resultados..... 76

    Resultados obtenidos ..... 76

    Desempeño del Grupo de Tratamiento en las actividades de T..... 79

    Corroboración de resultados paramétricos..... 83

    Resumen de los resultados observados..... 84

Capítulo 5. Discusión y Conclusiones..... 86

    Discusión..... 86

    Conclusiones ..... 91

    Recomendaciones ..... 93

    Limitaciones..... 93

Referencias..... 95

**Lista de Tablas**

Tabla 1. Matriz Metodológica .....	17
Tabla 2. Modelo DigComp .....	24
Tabla 3. El PC como Habilidad de Resolución de Problemas .....	26
Tabla 4. Interés Mundial en la Enseñanza de Informática en Educación Primaria y Secundaria	31
Tabla 5. Niveles de Transferencia .....	39
Tabla 6. Tipos de Transferencia por Tipo de Conocimiento .....	40
Tabla 7. Taxonomía de Transferencia Lejana .....	42
Tabla 8. Niveles 1-3 de la Taxonomía de Transferencia Lejana correspondientes al Contenido..	43
Tabla 9. Niveles 4 -9 de la Taxonomía de Transferencia Lejana correspondientes al Contexto...	43
Tabla 10. Tipos de Transferencia entre Tareas .....	44
Tabla 11. Tipos de Transferencia según Robertson .....	44
Tabla 12. Clasificación de los Tipos de Transferencia .....	45
Tabla 13. Taxonomía de Transferencia .....	45
Tabla 14. Taxonomía de Transferencia Lejana del PC como se aplicó en este estudio. ....	52
Tabla 15. Transferencia Lejana de las Habilidades de Identificación del Problema.....	53
Tabla 16. Operacionalización de la Variable NPC.....	57
Tabla 17. Operacionalización de la Variable NTPC .....	58
Tabla 18. Taxonomía de Transferencia Lejana aplicada a la primera actividad de T.....	67
Tabla 19. Taxonomía de Transferencia Lejana aplicada a la segunda actividad de T. ....	69
Tabla 20. Taxonomía de Transferencia Lejana aplicada a la tercera actividad de T. ....	71
Tabla 21. Taxonomía de Transferencia Lejana aplicada a la cuarta actividad de T. ....	72
Tabla 22. Taxonomía de TL aplicada a la actividad final de T.....	73
Tabla 23. Conformación de los grupos para el Análisis de Covarianza (ANCOVA).....	75
Tabla 24. Prueba de efectos Inter Sujetos.....	77
Tabla 25. Medias de los grupos conformados por la interacción de todos los niveles de la variable Grupo A con todos los niveles de la variable Grupo B .....	78
Tabla 26. Medias marginales estimadas para el grupo de participantes basado en los resultados del Examen de PC.....	79
Tabla 27. Medias marginales estimadas para el Grupo de participantes asignados al Tratamiento .....	79
Tabla 28. Homogeneidad de varianza, supuestos de Levene.....	80
Tabla 29. Traza de Hotelling para la prueba multivariante .....	80
Tabla 30. Medias de los resultados en las actividades de T para Grupo Alto y Grupo Bajo .....	82
Tabla 31. Pruebas de efectos dentro de sujetos usando la prueba de Huynh-Feldt.....	83
Tabla 32. Prueba U de Mann- Whitney que rechaza la hipótesis nula del primer factor .....	84
Tabla 33. Prueba de U de Mann-Whitney que conserva la hipótesis del segundo factor .....	84
Tabla 34. Medias de aprovechamiento en las actividades de Transferencia .....	89

**Lista de Figuras**

Figura 1. Representación Gráfica del Diseño Factorial 2 x 2 ..... 60

Figura 2. Especificación de los 28 ítems de la Prueba de PC ..... 66

Figura 3. Matriz de evaluación de las actividades de T..... 74

Figura 4. Variable Resultado Final ajustado por la covariable Promedio General ..... 78

Figura 5. Medias marginales estimadas para las cinco actividades de T..... 81

Figura 6. Desempeño del Grupo de Tratamiento en las actividades de T..... 82

## **Introducción**

El presente trabajo de investigación versa sobre las Implicaciones de la Transferencia del Pensamiento Computacional para el diseño de Intervenciones Educativas en Educación Básica toda vez que el Pensamiento Computacional ha sido reconocido como una competencia esencial para el siglo XXI y gracias a la Transferencia es posible trasladar la habilidad de resolución de problemas que este desarrolla a otros contextos fuera de las ciencias computacionales.

La tesis consta de cinco capítulos con la siguiente estructura y contenido:

El capítulo uno se refiere al diseño de la investigación y plantea los diferentes elementos metodológicos que sustentan la tesis. Sus planteamientos comprenden los antecedentes del problema estudiado en relación con la Transferencia del Pensamiento Computacional (TPC), así como su respectiva justificación, objetivos, hipótesis y demás variables que preocupan para su análisis en este trabajo.

En el capítulo dos se profundiza en la definición de los conceptos centrales del trabajo de investigación: Pensamiento Computacional (PC) y Transferencia (T), y se hace un recuento histórico de las principales posturas teóricas surgidas en torno a ellos desde sus inicios hasta la actualidad. También se presentan las principales taxonomías que se han creado para medir la T y los principales marcos conceptuales para el PC.

En el capítulo tres se detallan el carácter y tipo de investigación de la presente tesis, así como la metodología, el procedimiento y los instrumentos utilizados para recabar los datos, y los métodos estadísticos que se utilizaron para comprobar las hipótesis.

En el capítulo cuatro se reportan los resultados obtenidos en la intervención en PC y del Entrenamiento en Transferencia de Pensamiento Computacional (ETPC), para lo cual se retomaron las hipótesis de la investigación.

En el capítulo cinco se discute a detalle el significado de los resultados encontrados. En primer lugar, se explica cómo influye el nivel de pensamiento computacional (NPC) de los participantes en el Nivel de Transferencia del PC (NTPC) que logran; luego, cómo el NTPC cambia a partir de un ETPC; en seguida se explican los resultados de la investigación a partir de la interacción entre el NPC y el ETPC. Por último, se discuten los distintos NTPC que logran los participantes según la Taxonomía de Transferencia Lejana (TTL).

**Capítulo 1. Diseño de la Investigación**

Este capítulo referido al diseño de la investigación plantea los diferentes elementos metodológicos que sustentan la tesis. Sus planteamientos comprenden los antecedentes del problema estudiado en relación con la Transferencia del Pensamiento Computacional (TPC), así como su respectiva justificación, objetivos, hipótesis y demás análisis de variables que preocupan para su estudio en este trabajo.

**Antecedentes**

El ritmo de avance hacia una mayor inclusión tecnológica en el mundo del trabajo solo se ha acelerado desde 2018 (Minnian & Montroy, 2018) y se acentuó de manera dramática con la pandemia de SARS-COV2 (Weller, 2020; ONU, 2021). La automatización y la transformación digital en el ámbito laboral consiste en la adopción de tecnologías de la información (TI) (Universidad de Valladolid, 2018), incluida la inteligencia artificial, por parte de las organizaciones para alterar fundamentalmente sus funciones y procesos internos y externos, fomentar el cambio organizacional y adoptar nuevas prácticas comerciales habilitadas por TI (Van der Linden, 2019).

En el reporte 2020 sobre El Futuro del Empleo el World Economic Forum señala que en los próximos años el mundo experimentará una división del trabajo muy distinta a la que prevalece actualmente, donde las máquinas podrían desplazar 85 millones de puestos de trabajo y conllevarían a la creación de 97 millones de nuevos puestos laborales. “Para 2025, las capacidades de las máquinas y los algoritmos se emplearán más ampliamente que en años anteriores, y las horas de trabajo realizadas por las máquinas coincidirán con el tiempo dedicado a trabajar por los seres humanos” (WEF, 2020, p. 8).

Este informe proyecta que, a mediano plazo, la destrucción de puestos de trabajo probablemente se verá compensada por el crecimiento del empleo en los 'puestos de trabajo del mañana': en la economía verde; la economía de datos y de inteligencia artificial; nuevos roles en ingeniería, computación en la nube y desarrollo de productos; la economía del cuidado; marketing, ventas y producción de contenido; así como roles a la vanguardia de la cultura (WEF, 2020).

También habrá que monitorear cómo evoluciona el empleo en ocupaciones de baja calificación que utilicen competencias poco complejas y habilidades manuales, pues los ocupados en estos empleos pueden ser desplazados si las tendencias de digitalización y

automatización se acentúan y deberán disponer de programas de recalificación para reengancharse en el mercado de laboral (Torres, 2015; Vargas, 2020; ONU, 2021).

En Europa, las últimas dos décadas han visto una mayor demanda de trabajadores con habilidades cognitivas, no rutinarias y de alto nivel en Tecnologías de la Información y la Comunicación (TIC) (OCDE, 2017). Según las estadísticas de la Secretaría del Trabajo de los Estados Unidos de América, los empleos relacionados con la computación y la tecnología de la información crecerán un 12% del 2018 al 2028, lo que representa un crecimiento mucho más acelerado que el promedio de otras ocupaciones, lo que generará un total de 546 mil 200 nuevos puestos de trabajo y la mayor concentración de ellos estará relacionado con la computación en la nube, la recolección y el almacenaje de Big Data y la Ciberseguridad. Mientras que el sueldo anual promedio de dichas ocupaciones fue de 86 mil 320 dólares americanos en mayo de 2018, el ingreso medio anual para el resto de las otras ocupaciones fue de 38 mil 640 dólares americanos (U.S. Bureau of Labor Statistics, 2021).

Ante este panorama, la pandemia también provocó que empleados y desempleados buscaran herramientas y cursos online para incrementar, mejorar y/o cambiar sus habilidades profesionales. Los empleados buscando principalmente la actualización y los desempleados una nueva salida laboral acorde a la demanda del mercado, pues la falta de alineación entre la oferta y la demanda contribuye a los problemas del mercado laboral que experimentan muchos graduados recientes (Álvarez & Romero, 2015; Van der Linden, 2019).

No solo el crecimiento en el número de recién graduados universitarios difiere del crecimiento de las vacantes de trabajo potencialmente disponibles para ellos, pero también un gran número de personas terminan completando su grado universitario en asignaturas para las que hay poca demanda o para las que existe exceso de oferta en relación con la demanda (Van der Linden, 2019).

Estas tendencias implican retos para las instituciones educativas que deberán cambiar su enfoque formativo para incentivar la capacidad de aprendizaje permanente en sus estudiantes sobre una base sólida de conocimiento que les permita adaptarse a los cambios, y tender a la transversalidad de los contenidos porque la digitalización conlleva una fuerte homogenización de las tareas. (Laviña et al., 2019).

El WEF (2020) también señala que las principales habilidades y grupos de habilidades que van a aumentar en importancia en el período previo a 2025 son el pensamiento crítico y de análisis, la resolución de problemas y las habilidades en la autogestión, el aprendizaje activo, la

resiliencia, la tolerancia al estrés y la flexibilidad. Paralelamente, la Agenda Educativa 2030 de la Organización de las Naciones declara que el desarrollo de competencias básicas de lecto-escritura y aritmética hoy en día no es suficiente para definir una educación de calidad, sino que es necesario el desarrollo “de aptitudes analíticas, de solución de problemas y otras habilidades cognitivas, interpersonales y sociales de alto nivel” (UNESCO, 2017a; UNESCO, 2017b).

Previo a la crisis sanitaria de 2020, el Instituto de Estadísticas de la UNESCO, desarrolló el “Marco referencial global de habilidades para la alfabetización digital para el indicador 4.4.2” que se desprende del objetivo 4, Educación de calidad, de la Agenda 2030 de la Organización de las Naciones Unidas que fija los Objetivos de Desarrollo Sostenible. Dicho marco tiene como meta guiar la creación y desarrollo de planes y programas de estudio, así como de evaluación en el campo educativo (Law et al., 2018). Ante este panorama, distintos países impulsan ya reformas educativas cuyo enfoque sea el STEAM, es decir, la alfabetización en Ciencia, Tecnología, Ingeniería, Artes y Matemáticas y hacen de la programación y el PC una prioridad para la educación (Manches & Plowman, 2017).

Debido a esta nueva realidad surge la necesidad de realizar investigación en torno a los beneficios de la incorporación del PC en las aulas de educación básica en México, para que los estudiantes desarrollen las llamadas habilidades del siglo XXI de manera integrada con el actual plan de estudios definido por la Secretaría de Educación Pública para dicho nivel educativo, lo cual puede lograrse si se busca dicha integración a partir de la Transferencia de conocimiento, de la que se hablará más adelante.

### **Problema de Investigación**

Como PC se entiende la “habilidad de resolución de problemas que toma elementos de los conceptos fundamentales de las ciencias computacionales (CC) para generar soluciones que puedan ser ejecutadas por un ordenador (humano o electrónico)” (Wing, 2006). En 1960, Seymour Papert creó LOGO, el primer lenguaje de programación para enseñar a los niños conceptos básicos de geometría y aritmética, y llegó a señalar que a través de la experiencia con la programación los niños desarrollaban una poderosa habilidad de pensamiento para la resolución de problemas que llamó Pensamiento Computacional (PC) (Denning, 2017) y que estos beneficios cognitivos podían extenderse a otras áreas académicas, con lo que inició una serie de investigaciones que buscaban probar la TPC y cuyos resultados no aportaron evidencia concluyente al respecto.



En 1960 el PC no logró permear todos los niveles ni ambientes educativos y fue hasta 2006 cuando el interés público se volvió hacia él nuevamente. Desde entonces la investigación realizada de manera conjunta por especialistas en CC y pedagogía busca definir con precisión qué es el PC, cuáles son sus elementos constitutivos, cómo dosificarlo y graduarlo para incorporarlo en los planes de estudio dependiendo de la edad de la población estudiantil objetivo, así como desarrollar y estandarizar los instrumentos idóneos para su evaluación.

De acuerdo con ISTE (International Society of Technology in Education [Sociedad Internacional para la Tecnología en Educación]), el PC es una competencia que va más allá del enfoque tradicional que se da en las escuelas a la enseñanza de las competencias digitales y en el 2011 creó una definición operacional de lo que es y cómo debe enseñarse el PC en los planteles de educación básica que incluye los siguientes elementos: (i) formulación de problemas de manera que puedan resolverse usando computadoras u otros dispositivos digitales, (ii) organización lógica y análisis de la información, (iii) representación de la información por medio de modelos y simulaciones, (iv) automatización de soluciones, (v) identificación, análisis e implementación de posibles soluciones con el propósito de lograr la más eficiente combinación de pasos y recursos, (vi) *generalizar y transferir el proceso de solución de problemas a una amplia gama de problemas* (ISTE, 2018).

Uno de los principales argumentos de los que abogan por integrar el PC en los planes de estudio de educación básica es la transferencia de conocimiento (T), en este caso, de la habilidad de resolución de problemas que se desarrolla a través del PC y la programación, tal como lo sugería Papert (punto seis de los listados arriba). Sin embargo, las evidencias empíricas en ese sentido han arrojado resultados contradictorios porque no se ha logrado operacionalizar la manera correcta de lograr dicha T.

La T se puede definir como la habilidad de aplicar exitosamente en un contexto diferente aquello que ha sido aprendido anteriormente (Byrnes, 1996) y viene a ser la respuesta para quienes acusan que el modelo de educación por competencias se basa en una retórica economicista cuya meta es preparar personal trabajador para las economías del conocimiento (Del Rey & Sánchez-Parga, 2011; Guzmán, 2017)

Sin embargo, en los establecimientos dedicados a la educación no se aborda este aspecto de forma sistemática hasta ahora y los resultados son observables cuando el estudiante es incapaz de aplicar los conocimientos adquiridos para resolver situaciones problemáticas. Esta inaplicabilidad de los conocimientos previos se debe a la práctica didáctica que tiene lugar actualmente en las instituciones educativas donde los maestros dedican más del 70 por ciento

del tiempo de instrucción a la enseñanza de hechos y contenidos, descuidando la parte práctica que permitiría la aplicación de estos conocimientos y la promoción específica de la capacidad de resolver problemas (Bohl, 2000; Pätzold et al., 2003).

Entender cómo se logra que los estudiantes transfieran su conocimiento a situaciones nuevas es un tema de importancia teórica y práctica. Teóricamente, toca temas centrales relacionados con la representación del conocimiento, el razonamiento lógico, la generalización, la cognición encarnada y el concepto de formación. Prácticamente, el aprendizaje sin T de lo que se ha aprendido es casi siempre improductivo e ineficiente (Goldstone & Day, 2012), pero a pesar de las serias consecuencias que eso implica, existe poco acuerdo entre los expertos acerca de la naturaleza de la T, la medida en la que ocurre, y sus mecanismos subyacentes (Barnett & Ceci, 2002) y sólo un 1% de los trabajos de las revistas más prestigiosas se han ocupado de su estudio (Chen & Klahr, 2008).

Por lo tanto, esta investigación pretende aportar elementos a considerar para el diseño, implementación y evaluación de intervenciones educativas que tengan como propósito desarrollar el PC de manera transversal en todas las asignaturas escolares a partir de la T.

### **Objetivos del estudio**

- a) Establecer el impacto que tiene el Nivel de Pensamiento Computacional (NPC) en el Nivel de Transferencia de Pensamiento Computacional (NTPC) en niños de 12 y 13 años.
- b) Estimar el efecto que tiene el Entrenamiento en Transferencia de Pensamiento Computacional (ETPC) en el Nivel de Transferencia de Pensamiento Computacional (NTPC) que logran los niños de 12 y 13 años.
- c) Determinar el impacto de la interacción del NPC y el ETPC en el NTPC en niños de 12 y 13 años.
- d) Identificar los distintos tipos y niveles de Transferencia en Pensamiento Computacional (TPC) que logran los niños de 12 y 13 años a través de cinco actividades de evaluación.

### **Hipótesis**

H1. Los niños de 12 y 13 años que presentan mayor NPC reportan un NTPC más alto.

H2. Al contar con ETPC los niños de 12 y 13 años logran un NTPC más alto.

H3. Los niños de 12 y 13 años que cuentan con un NPC alto y reciben ETPC logran mayor NTPC debido a la interacción entre estos.

H4. El NPC de los niños de 12 y 13 años y el ETPC generan diferentes tipos y niveles de TPC a través de cinco actividades de evaluación.

**Preguntas de investigación**

Esta problemática plantea las siguientes preguntas de investigación:

- a) ¿Qué impacto tiene NPC en el NTPC en niños de 12 y 13 años?
- b) ¿Qué efecto tiene el ETPC en el NTPC que logran los niños de 12 y 13 años?
- c) ¿Qué resultado se obtiene la interacción del NPC y el ETPC en el NTPC logrado por los niños de 12 y 13 años?
- d) ¿Qué tipo y/o nivel de transferencia logran los niños de 12 y 13 años a partir de su NPC y a través de cinco actividades de evaluación?

En la matriz metodológica que se muestra a continuación, es posible relacionar de manera más clara los elementos que componen este trabajo (ver Tabla 1).

Tabla 1. Matriz Metodológica

Preguntas	Objetivos	Hipótesis	Variables independientes	Variable dependiente
¿Qué impacto tiene NPC en el NTPC en niños de 12 y 13 años?	Establecer el impacto que tiene el nivel de Pensamiento Computacional (NPC) en el nivel de transferencia de Pensamiento Computacional (NTPC) en niños de 12 y 13 años.	Los niños de 12 y 13 años que presentan mayor NPC reportan un NTPC más alto.	NPC (Grupo A)	NTPC
Pregunta 2	Objetivo 2	Hipótesis 2		
¿Qué efecto tiene el ETPC en el NTPC que logran los niños de 12 y 13 años?	Estimar el efecto que tiene el Entrenamiento en Tráferencia de Pensamiento Computacional (ETPC) en el Nivel de Tráferencia de Pensamiento Computacional	Al contar con ETPC los niños de 12 y 13 años logran un NTPC más alto.	ETPC (Grupo B)	NTPC

Preguntas	Objetivos	Hipótesis	Variables independientes	Variable dependiente
	(NTPC) que logran los niños de 12 y 13 años.			
<u>Pregunta 3</u>	<u>Objetivo 3</u>	<u>Hipótesis 3</u>		
¿Qué resultado se obtiene de la interacción del NPC y el ETPC en el NTPC logrado por los niños de 12 y 13 años?	Determinar el impacto de la interacción del NPC y el ETPC en el NTPC en niños de 12 y 13 años.	Los niños de 12 y 13 años que cuentan con un NPC alto y reciben ETPC logran mayor NTPC debido a la interacción entre estos.	NPC*EPC (Grupo A*B)	NTPC
<u>Pregunta 4</u>	<u>Objetivo 4</u>	<u>Hipótesis 4</u>		
¿Qué tipo y/o nivel de transferencia logran los niños de 12 y 13 años a partir de su NPC y a través de cinco actividades de evaluación?	Identificar los distintos tipos y niveles de TPC que logran los niños de 12 y 13 años a través de cinco actividades de evaluación.	El NPC de los niños de 12 y 13 años y el ETPC generan diferentes tipos y niveles de TPC en ellos a través de cinco actividades de evaluación.	NPC*EPC (Grupo A)	NTPC

Nota: Esta tabla demuestra los objetivos y preguntas de la investigación, así como sus hipótesis.

### Justificación y Delimitación del Estudio

La Comisión Europea ha asumido un roll activo para promover una agenda digital en la cual establece que la programación es el nuevo alfabetismo (Siu & Lam, 2005). Y aunque numerosos países reconocen la importancia de introducir los principios de las Ciencias Computacionales (CC) a edad temprana, prevalece el desconocimiento del camino adecuado para hacerlo (Furber, 2012). De igual modo, en los marcos curriculares para PC que existen actualmente se ha obviado la T, lo cual reduce el impacto que logran los programas de PC en contextos fuera de la informática.

Para poder ubicar adecuadamente el PC dentro de los currículos escolares es menester considerar las diferentes concepciones que se tienen sobre él y revisar como estas se traducen en diferentes maneras de valorar la importancia de impartir dichos contenidos en los centros

educativos de todos los niveles. Según Denning y Tedre, (2019) el PC se encuentra en algún punto entre las competencias digitales y las habilidades STEM, entre la enseñanza de la programación y la robótica y puede ser añadido a otras áreas de estudio, pero no es un reemplazo para ningún tipo de pensamiento y no es una meta-habilidad para todos los campos.

El PC ha significado además una revolución en la epistemología de la ciencia, pues los científicos de todas las áreas afirman que el PC es un nuevo método de hacer ciencia, diferente de los métodos clásicos basados en la teoría y el experimento. Zapata- Ros (2015) considera que la enseñanza del PC se debe iniciar en las etapas formativas en la primera infancia tal como se hace con la lecto-escritura y el pensamiento matemático.

Aun cuando existe consenso a nivel mundial sobre la necesidad de desarrollar en los estudiantes el PC al considerarlo una herramienta esencial para el ejercicio de la ciudadanía digital en el siglo XXI (Valencia, 2019) los intentos que se han realizado en diversos países para determinar cuál es la mejor estrategia o vía para hacerlo no han llegado a resultados concluyentes que puedan convertirse en parámetros para el desarrollo de este internacionalmente (Royal Society, 2012).

Algunos países han optado por introducirlo como talleres o actividades extraescolares, otros lo han incorporarlo directamente en el plan de estudios como un campo formativo más y otros más han buscado desarrollarlo de manera transversal en todas las asignaturas (Brackmann et al., 2016), pero aún se desconoce cómo organizar los contenidos curriculares de las ciencias computacionales y cómo dosificar los elementos del PC de acuerdo con el nivel de desarrollo cognitivo de cada etapa escolar.

Las investigaciones realizadas concuerdan en señalar que la enseñanza del PC debe realizarse teniendo en cuenta que los estudiantes de hoy serán los analistas de información del mañana, así como los especialistas en inteligencia artificial y *Machine Learning*, los desarrolladores de software y aplicaciones, los técnicos que harán posible la automatización, los científicos de la informática cuántica y los expertos en ciberseguridad. Sin embargo, en el contexto actual de acelerado crecimiento tecnológico resulta pertinente centrar los esfuerzos en transferir la habilidad de resolución de problemas asociada al PC para que tenga aplicaciones en todos los campos del conocimiento.

Wing (2006) sostiene que “el PC influencia prácticamente a todas las disciplinas, tanto a las ciencias como a las humanidades y se inserta en un círculo virtuoso de retroalimentación entre los cuestionamientos científicos, la innovación tecnológica y las demandas sociales”, por lo que recomienda su inclusión en los currículos de educación básica a partir de la primera infancia

que permita construir en los niños bases sólidas e inclusivas para la aplicación del PC. A partir de esta declaración los gobiernos de Europa principalmente y Estados Unidos han buscado intensamente la incorporación del PC en sus respectivos currículos escolares.

La importancia de este estudio radica en atraer la atención sobre un tema que ha sido obviado por la actual investigación para la inclusión del PC en el currículo de educación básica, que prioriza el desarrollo de plataformas, dispositivos e instrumentos de evaluación del PC pero que descuida los procesos cognitivos subyacentes en este.

La investigación realizada hasta el día de hoy en torno al PC ha encontrado que las intervenciones educativas realizadas en la primera infancia tienen menor costo y su efecto es más duradero que el de aquellas que se realizan más tarde (Cunha & Heckman, 2006) y aunque autores como Brown (1989) encontraron que la T puede lograrse incluso en niños tan pequeños como cuatro años de edad dependiendo de la perspectiva de evaluación que se considere para medir el logro de esta; para fines prácticos en esta investigación se estudiará la T del PC en estudiantes de 12 y 13 años de edad, que de acuerdo con la Teoría Psicogenética de Piaget, se encuentran en el estadio de operaciones formales. Recordemos que, de acuerdo con Piaget, existen 4 estructuras cognitivas primarias o 4 estadios de desarrollo cognitivo que son: Estadio sensoriomotor, estadio preoperatorio, estadio de operaciones concretas y estadio de operaciones formales. El estadio sensoriomotor comprende de los cero a los dos años de edad, cuando la inteligencia toma la forma de acciones motoras; en la etapa preoperatoria que comprende de los 3 a los 7 años, la inteligencia es intuitiva; de los 8 a los 11 años se presenta la etapa de las operaciones concretas cuando la estructura cognitiva es lógica pero depende de referencias concretas; y de los 12 a los 15 años se presenta el estadio de la operaciones formales, cuando el pensamiento involucra abstracciones (Piaget, 1988).

También es importante señalar en este punto que las ideas de Piaget, impulsor del constructivismo en educación, influyeron significativamente en Papert, creador del construccionismo (Vielma & Salas, 2000; Badilla & Chacón, 2004; Saldarriaga-Zambrano et al., 2016). Tanto el constructivismo como el construccionismo parten del supuesto de que el sujeto en formación construye su conocimiento a partir de sus esquemas previos, los cuales construye y reconstruye para adquirir nuevos aprendizajes. No obstante, el diferenciador entre ambas teorías pedagógicas radica en que el construccionismo destaca la importancia de la acción y de la creación de artefactos en el proceso de aprendizaje al considerar éste no se puede simplemente transmitir.

A la luz de lo expuesto, podemos inferir que la base cognitiva del constructivismo y del construccionismo es la T, pues es el gran fundamento del aprendizaje, pensamiento y resolución de problemas (Pellegrino & Hilton, 2012).

La T no solo facilita la integración de los distintos campos formativos y los principios fundamentales de las CC al convertir al PC en un conocimiento fluido esencial para el aprendizaje permanente y la homogenización de tareas en muchas áreas de formación, sino que resulta esencial para los sistemas educativos de todo el mundo porque equivale en significado a aprendizaje Detterman (1993).

En el capítulo siguiente se abordarán a profundidad los conceptos centrales de esta tesis y se expondrá su evolución a través del tiempo, así como las posturas de los principales teóricos en torno a ellos y los marcos conceptuales y las taxonomías que han desarrollado para su estudio.

## Capítulo 2. Marco Teórico

En este capítulo se profundiza en la definición de los conceptos centrales del trabajo de investigación: PC y T, y se hace un recuento histórico de las principales posturas teóricas surgidas en torno a ellos desde sus inicios hasta la actualidad. También se presentan las principales taxonomías que se han creado para medir la T y los principales marcos conceptuales para el PC.

### Conceptos esenciales sobre PC

Aunque Wing (2006), lo definió como una habilidad de resolución de problemas, el PC tiene muchas interpretaciones (Aho, 2011; Barr & Stephenson, 2011; Cuny et al, 2010; Grover & Pea, 2013; National Research Council, 2010). Román- González (2015) distingue entre tres tipos de definiciones del PC: a) las genéricas; b) las operacionales; c) las educativas y para uso en los currículos escolares. Al primer grupo pertenece la definición de Jeannette Wing (2006; 2011), dentro de las definiciones operacionales podemos considerar la elaborada por CSTA e ISTE (2011), y a las definiciones para uso educativo corresponden los “frameworks” o marcos conceptuales a partir de los cuales se han desarrollado propuestas curriculares y pedagógicas, como el modelo DigCom creado por el Centro Conjunto de Investigación de la Comisión Europea para la Ciencia y el Conocimiento (Carretero, et al., 2017; Kluzer & Pujol, 2018).

Wing (2006), explica que “el PC incluye la resolución de problemas, el diseño de sistemas y el entendimiento del comportamiento humano”. Sin embargo, debido a la falta de consenso sobre esta definición, en el 2011 Wing complementó su definición agregando que el PC es un proceso de pensamiento que permite la identificación y planteamiento de problemas de manera tal que una computadora o procesador pueda expresar la solución. Wing plantea por lo tanto que el PC tiene cuatro pilares:

- a) Descomposición: consiste en identificar un problema complejo y dividirlo en subproblemas más fáciles de manejar.
- b) Reconocimiento de patrones: Cada subproblema puede ser resuelto a partir de sus similitudes con problemas similares que se resolvieron previamente.
- c) Abstracción: Cada uno de estos problemas puede ser resuelto enfocándose únicamente en los detalles que son relevantes.
- d) Algoritmo: Se le llama al conjunto de reglas que son creadas para resolver cada uno de los subproblemas encontrados.



Una revisión reciente sobre PC en contextos educativos lo describe como la "base conceptual necesaria para resolver problemas de manera eficaz y eficiente (es decir, algorítmicamente, con o sin la ayuda de computadoras) con soluciones que son reutilizables en diferentes contextos" (Shute et al., 2017). Esta definición holística caracteriza al PC como una capacidad cognitiva en lugar de ser simplemente una habilidad práctica en un contexto específico y, por lo tanto, enfatiza la amplia aplicabilidad del PC (Tsarava et al., 2019).

### **Antecedentes Históricos del PC**

El PC no surge recientemente junto con la actual era de la computadora digital. Desde 1600, los matemáticos más eminentes contrataron asistentes conocidos como "computadores", que eran un grupo de personas dedicadas a hacer laboriosos "cálculos" matemáticos, asistidos con el uso de lápiz y papel, gis y pizarra, ábaco, reglas de cálculo y primitivas calculadoras mecánicas. Por esa razón, a las primeras computadoras electrónicas del siglo XX se les llamó "computadoras automáticas" para diferenciarlas de los humanos. De manera que los humanos ya trabajaban con habilidades de pensamiento matemático y computacional para resolver laboriosos problemas de cómputo fisicomatemático antes de que existieran las calculadoras y computadoras electrónicas.

Puede considerarse, en un sentido más amplio, que el PC es tan antiguo como la civilización misma (Denning & Tedre, 2019). Los egipcios eran capaces de resolver problemas geométricos donde calculaban dimensiones, áreas y ángulos para sus cosechas y construcciones monumentales como las pirámides; también los babilonios dejaron por escrito procedimientos generales para resolver problemas matemáticos.

A partir de la época de la Ilustración, el método científico comienza a representar simbólicamente y matemáticamente sus procedimientos (García-Carmona, 2002; Vega, 2012). Durante la Revolución Industrial, los matemáticos pusieron su experiencia al servicio de operadores no expertos (Gutarra, 2015) que simplemente tenían que seguir las instrucciones para resolver operaciones matemáticas en un orden concreto para resolver problemas de índole mecánica, industrial, construcción, navegación, militar o espacial.

A partir de contribuciones pioneras como fue la aportación de la noción de "programa" de la condesa Ada Lovelace para automatizar las primeras calculadoras mecánicas se afianza el concepto de algoritmo (concepto esencial en la programación) adaptando el uso de tarjetas perforadas de los telares para especificar programas compuestos por secuencias de instrucciones codificadas y pregrabadas en dichas tarjetas (Molero, 2016; Heredero de Pedro & Muñoz, 2011).

Durante el siglo XX con el surgimiento de la Computación y la Informática, se hace indispensable un PC para diseñar procedimientos con el fin de programar las primeras computadoras analógicas y digitales. Siendo hasta 1950 con el desarrollo de las primeras computadoras electrónicas programables cuando los diseñadores y programadores comienzan a sistematizar cuestiones derivadas del PC que eran necesarias en el ejercicio de su profesión (Denning y Tedre, 2017).

En 1960 inicia formalmente la enseñanza del PC en las universidades debido al surgimiento de los departamentos de CC, con el construccionismo como su precedente teórico. Dicha teoría enlaza corrientes clásicas del aprendizaje con la tecnología siendo sus iniciadores Seymour Papert, Cynthia Solomon, Daniel Bobrow y Wally Feurzeig, creadores del primer lenguaje y entorno de programación para niño, al que llamaron LOGO y que tenía como mascota a una tortuga (Zapata-Ross, 2019). Años más tarde Aho (2011) se unió a la definición de Wing en la que describen el PC como los procesos de pensamiento involucrados en la formulación de problemas para que sus soluciones puedan ser representadas algorítmicamente; y Denning y Tedre (2017) afirmarían que el PC es una habilidad más que un conjunto particular de conocimientos aplicables, la cual se adquiere con el tiempo y la práctica.

**Marcos de Referencia para PC**

El modelo DigCom describe cuales son las competencias necesarias hoy en día para usar la tecnología digital de manera segura, crítica, colaborativa y creativa para lograr las metas relacionadas con el trabajo, el aprendizaje y el entretenimiento, y la inclusión y participación en la sociedad digital (ver Tabla 2).

Tabla 2. Modelo DigComp

Área de Competencia	Competencias
Información y alfabetización	Navegación, búsqueda y filtrado de datos, información y contenido digital Evaluación de datos, información y contenido digital Gestión de datos, información y contenido digital
Comunicación y colaboración	Interactuar a través de tecnologías digitales Compartir a través de tecnologías digitales Participar en la ciudadanía a través de tecnologías digitales Netiquetas Gestión de la identidad digital
Creación de contenido digital	Desarrollo de contenido digital Integración y reelaboración de contenido digital Copyright y licencias Programación

Área de Competencia	Competencias
Seguridad	Protección de dispositivos Protección de datos personales y privacidad Protección de la salud y el bienestar Protección del medio ambiente
Resolución de problemas	Resolución de problemas técnicos Identificación de necesidades y respuestas tecnológicas Uso creativo de tecnologías digitales Identificación de brechas de competencias digitales

Nota: Elaboración propia basada en Kluzer & Pujol (2018)

Barefoot (2014) enfatiza que el PC involucra seis conceptos diferentes y cinco aproximaciones. Los conceptos son: (i) razonamiento lógico, (ii) algoritmos, (iii) descomposición, (iv) patrones, (v) abstracción, (vi) evaluación; y las aproximaciones: probar, crear, depurar, perseverar y colaborar. Mientras que Barr et al., (2011) y Shute et al., (2017) consideran que el PC comprende una variedad de conceptos y capacidades tales como la recopilación, organización y análisis de datos, el pensamiento algorítmico, la abstracción, generalización y evaluación.

Shein (2014) asegura que al desarrollar el PC la gente aprende a pensar abstractamente y puede dividir un problema en piezas pequeñas para resolverlo, resaltando que el desarrollo de las habilidades de resolución de problemas en las Ciencias Computacionales (CC) es más relevante que la simple enseñanza de escribir código. Los componentes que articula el pensamiento computacional, según Valverde (2015), son el análisis ascendente y descendente, la heurística, el pensamiento divergente, la creatividad, la resolución de problemas, el pensamiento abstracto, la recursividad e iteración, los métodos por aproximaciones sucesivas de ensayo-error, los métodos colaborativos, los patrones, la sinéctica y la metacognición.

Las habilidades antes mencionadas son potenciadas por un conjunto de actitudes periféricas o disposiciones que son dimensiones importantes del PC. Estas incluyen resistencia a la ambigüedad, determinación para trabajar en problemas cuyas soluciones no son obvias, confianza para enfrentar la complejidad, la habilidad de cooperar y comunicarse con colegas para cumplir metas comunes y encontrar las soluciones a los problemas presentados (Barr et al., 2011).

### **PC como Habilidad de Resolución de Problemas**

De acuerdo con Holyoak y Morrison (2012) el proceso para resolver problemas consta de dos fases principales: i) representación y ii) ejecución, las cuales a su vez que se corresponden con las cuatro etapas del Enfoque de Sistemas Complejos Múltiples para la resolución de problemas: i) exploración y entendimiento, ii) representación y formulación, iii) planeación y

ejecución, y iv) Monitorización y reflexión (Greiff, Wüstenberg y Funke, 2012); mientras que el marco de referencia de PISA 2012 define cuatro procesos que forman parte de la habilidad de resolución de problemas: i) recuperación de información, ii) construcción de un modelo, iii) predicción, y iv) monitoreo y reflexión (National Research Council, 2012). También Polya (1957) contempla estas etapas cuando propone su metodología para la resolución de problemas y a cada una de ellas se asocia algún elemento del PC (ver Tabla 3).

Tabla 3. El PC como Habilidad de Resolución de Problemas

Fases de la resolución de problemas (Holyoak y Morrison, 2012)	Enfoque de Sistemas Complejos Múltiples (Greiff, Wüstenberg y Funke, 2012)	Pisa 2012	Metodología de Polya (1957)	Procesos cognitivos del PC propuestos por la autora	Componentes que articula el PC (Valverde, 2015)
Planeación	Etapa 1. Exploración y entendimiento	1. Recuperación de la información	Comprender el problema	1. Identificación del problema 2. Abstracción 3. Descomposición	Análisis ascendente Análisis descendente Pensamiento abstracto
	Etapa 2. Representación y formulación	2. Construcción de un modelo	Concebir un plan	4. Reconocimiento de patrones 5. Creación de algoritmos	Heurística Pensamiento divergente Creatividad Recursividad Métodos colaborativos Patrones Sinéctica
Ejecución	Etapa 3. Planeación y ejecución	3. Predicción	Ejecutar el plan	6. Ejecución	Pensamiento crítico Métodos por aproximaciones sucesivas ensayo-error
	Etapa 4. Monitorización y reflexión	4. Monitoreo y reflexión	Verificar el resultado	7. Evaluación y depuración	Iteración Metacognición

## PC y su Relación con Otros Tipos de Pensamiento

Muchos de los conceptos, habilidades, actitudes y disposiciones mencionadas arriba no son exclusivos del PC. De hecho, ni siquiera son nuevos. De ahí que surja la interrogante respecto a qué diferencia al PC de otros tipos de pensamiento como el matemático y el crítico. Para responder a esta pregunta, Barr et al. (2011) explican que el PC combina componentes distintivos de las habilidades de pensamiento, que cuando se reúnen, adquieren la condición de una forma avanzada de resolución de problemas que está orientado al uso de herramientas. Además de que utiliza efectivamente estrategias de resolución de problemas como la prueba y el error, la iteración e incluso la adivinanza que resultan no prácticas en el contexto de otras formas de pensamiento pero que adquieren sentido cuando las condiciones requieren automatización y alta velocidad de respuesta. No obstante, el PC comparte similitudes con otros tipos de pensamiento. Por ejemplo, con el matemático comparte las bases y procesos presentes en la resolución de problemas, con el de ingeniería se identifica en los elementos correspondientes al diseño y evaluación de los procesos y con el pensamiento científico se asimila en lo concerniente al análisis sistemático (González-González, 2019).

Al respecto, Wing (2016), explica que:

“Las CC toman de las matemáticas sus fundamentos subyacentes, tales como las herramientas y el lenguaje para expresar los fundamentos de la computabilidad... el PC obviamente se superpone con el pensamiento matemático (PM) en términos de modelado, abstracción, razonamiento, lógica y más... El PC también se superpone con el pensamiento de las ingenierías (PI) porque después de todo en las CC se construyen cosas para funcionar en el mundo real y por lo tanto tienen que trabajar con las limitaciones del mundo real como cualquier otra ingeniería.

Y concluye que lo que diferencia el PC de otras áreas de pensamiento es el software, el cual es fácil de crear, ser copiado, distribuido, editado y manipulado; y permite crear cualquier cosa que pueda ser concebida por el pensamiento humano.

Aunque el PC no requiere el uso de una computadora para su desarrollo, si requiere del razonamiento necesario para capitalizarlo en el uso de herramientas computacionales, de modo que uno puede considerar que el PC es un conjunto benéfico de habilidades universales que constituyen el centro de las CC.

A continuación, se presentan las características de otros tipos de pensamiento que interactúan con el PC.

Antes de hablar de tipos de pensamiento es necesario decir que es el pensamiento. Al respecto Arboleda (2013) explica que el pensamiento es una función psíquica que le permite al individuo experimentar el mundo por medio de la cognición. Es decir, que el pensamiento le permite al sujeto representar la realidad y utilizar estrategias y operaciones frente a las situaciones que vive, sean esas reales, ideales o imaginarias. De acuerdo con este autor la dimensión mental abarca la inteligencia, las emociones, la voluntad, la memoria, la atención, la imaginación, la motivación, la cognición y el aprendizaje.

Haciendo una revisión de las teorías pedagógicas fundamentales de Piaget, Vigotsky y Ausbel, Jaramillo y Puga (2016) encontraron que el pensamiento se concibe como la capacidad que tienen los seres humanos para asimilar, crear y producir ideas en momentos determinados (Jaramillo & Puga, 2016).

Margarita Amestoy (2002), sugiere que el pensamiento se compone de tres dimensiones: el conocimiento, las operaciones y las disposiciones. A su vez, clasifica las operaciones en cognitivas y metacognitivas. Las primeras se refieren a los procesos mentales del pensamiento que dan lugar a la formación de procesos lógicos, mientras que las segundas permiten dirigir y controlar la producción de significados, procesos y productos del pensamiento y la planificación, supervisión y evaluación de este.

### ***Pensamiento Convergente y Divergente***

Pascual (2006) clasifica el pensamiento en dos grandes grupos: el convergente y el divergente. Al pensamiento convergente lo relaciona con la solución directa, obvia y más típica de un problema, y lo define con lo racional, lógico, vertical o convencional. Mientras que al pensamiento divergente lo relaciona con la creatividad e imaginación y a las soluciones que un sujeto aplica para la resolución de un problema a partir de su propia intuición.

### ***Pensamiento Lógico-Creativo***

El pensamiento lineal o lógico es de los primeros tipos de pensamiento que desarrollan los individuos al inicio de la escolarización, y si es debidamente guiado puede contribuir al desarrollo de seres altamente reflexivos (Jaramillo & Puga, 2016). El razonamiento lógico es la base para la resolución de problemas porque permite el análisis, la argumentación, clasificación, justificación y comprobación de las hipótesis planeadas. Como complemento a esto, Oliveros (2002) resalta el carácter altamente deductivo del razonamiento lógico y Jaramillo y Puga sugieren complementarlo con el pensamiento creativo porque permite fortalecer el análisis, la

síntesis, el argumento de saberes, y la abstracción de conclusiones de textos desde otra perspectiva.

### ***Pensamiento Abstracto***

El pensamiento abstracto permite identificar la esencia de las cosas y es muy útil para deducir, sintetizar, interpretar y analizar los distintos fenómenos que experimentamos. Los principales beneficios de este tipo de pensamiento son que facilita el análisis y síntesis de nuevos aprendizajes de manera rápida y asociativa, y que mejora la velocidad con que nuestras capacidades cognitivas operan (Ferreira y Pedrazzi, 2007).

### ***Pensamiento Crítico-Racional-Creativo***

El pensamiento crítico se relaciona principalmente con la racionalidad, aunque también trabaja de manera fructífera con el pensamiento creativo. Su función principal es revisar, evaluar y repasar las ideas y lo que se genera a partir de otros tipos de pensamiento. Entre sus objetivos está el describir el mundo o la realidad de la manera más precisa posible (Shermer, 1997). Al combinarse con el pensamiento creativo se vuelve más ingenioso y flexible porque estos se retroalimentan mutuamente (Lipman, 1998).

Furedy & Furedy (1985) revisaron cómo se operacionaliza el pensamiento crítico en las investigaciones educativas y encontraron que maneja dimensiones como identificar argumentos y supuestos, reconocer relaciones importantes, realizar inferencias correctas, evaluar la evidencia y la autoridad, y deducir conclusiones.

Ennis (2011) describe quince capacidades del pensamiento crítico, doce de ellas consideradas medulares y tres más de tipo auxiliar. Las capacidades medulares se centran en el análisis y valoración de las preguntas, los argumentos, las fuentes, las inducciones y deducciones, las definiciones, los supuestos, las acciones realizadas en cada situación, la interacción con los demás y las disposiciones, y en general de todos los elementos que permitan tomar una decisión y defenderla. Mientras que entre las capacidades auxiliares se cuentan las habilidades blandas como la empatía con los sentimientos y las capacidades cognitivas de los otros, así como la competencia comunicativa y la autorregulación del discurso.

### ***Pensamiento Analógico***

El pensamiento analógico busca las semejanzas estructurales y funcionales entre dos situaciones, problemas, ideas o nociones complejas con la finalidad de facilitar su comprensión y es muy utilizado tanto en el contexto escolar como en el lenguaje. "...busca hacer más

asequible a otras personas una determinada idea o noción, que se considera compleja, a través de otra que resulta más conocida y familiar” (Oliva, 2004).

### ***Pensamiento Disciplinar o por Área del Conocimiento***

Este tipo de pensamiento se utiliza al trabajar con actividades específicas de las distintas áreas del saber: “Cada tipo específico de actividad transmite- por así decirlo- al pensamiento peculiaridades distintivas” (Labarrere, 1996).

### ***Pensamiento Matemático***

El pensamiento matemático utiliza un tipo de razonamiento propio que se caracteriza por ser secuenciado, lógico y exacto que busca la concordancia óptima entre lo formal y lo lógico (List, 1982), siendo la resolución de problemas su característica distintiva, por lo que es considerado una actividad eminentemente humana que las máquinas no pueden realizar Campistrous (2000).

### ***Pensamiento Algorítmico***

El pensamiento algorítmico es la habilidad de entender, ejecutar, evaluar y crear procedimientos computacionales. Un pensador algorítmico debe ser primero capaz de entender y ejecutar un procedimiento computacional paso a paso y ser capaz de crear nuevos algoritmos. Dada una tarea, un pensador algorítmico bien logrado puede desarrollar una secuencia precisa de instrucciones paso a paso que siempre resuelve completamente la tarea, de manera correcta y eficiente (Lamagna, 2015).

## **Iniciativas Mundiales sobre Programas de PC**

El PC además de ser benéfico para la Sociedad, es visto por algunos países como una estrategia para su desarrollo y para abrirse camino hacia nuevos mercados (Brackmann et al. 2016).

### ***Estados Unidos***

El comité para la enseñanza de la ciencia, tecnología, ingeniería y matemáticas del Consejo Nacional de Ciencia y Tecnología de los Estados Unidos establece tres objetivos en esta dirección: 1. Promover la alfabetización digital y la ciberseguridad. 2. Hacer del PC un elemento integral de todo el sistema educativo. 3. Expandir las plataformas digitales para la enseñanza y el aprendizaje.



Así como la alfabetización fue considerada en los siglos pasados como una habilidad básica que permitía acceder a mejores oportunidades, la alfabetización digital es un factor crítico para el éxito de las personas en la sociedad contemporánea, pues empodera a los ciudadanos con las herramientas para encontrar y discernir información, hacer uso de dicha información para responderse cuestiones y para compartir ideas que promueven la colaboración (National Science & Technology Council, 2018). Aún más, dicho comité afirma que la demanda de carreras en ciberseguridad continuará al alza, toda vez que de ellas depende que los Estados Unidos mantengan a largo plazo su ventaja en este campo.

Como consecuencia, el PC y los fundamentos de la programación empezaron a enseñarse en algunas escuelas estadounidenses desde el año 2014. (Rich et al., 2013; Grover & Pea, 2013; K-12 Computer Science Framework Steering Committee 2016); Repenning et al., 2010; Smith, 2016). Para el año 2016, 54% de las escuelas habían adoptado las ciencias computacionales como sustituto de alguna otra materia y los estados participantes adoptaron el marco de referencia para la enseñanza del PC propuesto por la Asociación de Maestros de Ciencias Computacionales (CSTA, por sus siglas en inglés), denominado Modelo Curricular de Ciencias Computacionales para el K-12, donde se indica que la enseñanza del PC debe iniciarse en el kínder y continuar hasta el último grado de preparatoria. A esta iniciativa se suma el impacto del proyecto Code.org que ha logrado que el país en general tenga en gran aprecio las Ciencias Computacionales (Brackmann, 2016).

**Europa**

La integración de la programación en el currículo escolar ha tenido amplia aceptación a nivel nacional, regional y local (Balanskat & Engelbart, 2015), y dieciocho países miembros de la Unión Europea planeaban integrar la programación en sus planes de estudio para el 2020.

A continuación, se muestra una tabla de los países que han incluido de manera obligatoria u optativa la enseñanza del PC en las escuelas de educación básica (ver Tabla 4).

Tabla 4. Interés Mundial en la Enseñanza de Informática en Educación Primaria y Secundaria

País	Educación Primaria	Educación Secundaria
Australia	Obligatoria	Obligatoria
Inglaterra	Obligatoria	
Estonia	Obligatoria	Obligatoria
Finlandia	Obligatoria	
Nueva Zelanda		Optativa
Noruega		Optativa
Suecia	Obligatoria	Optativa
Corea del Sur	Obligatoria	Optativa

Estados Unidos  
Macedonia

Obligatoria

Optativa

---

Nota: Elaboración propia basada en Brackmann (2016).

### ***Francia***

En el informe titulado *Es urgente no esperar más*, presentado el 6 de marzo de 2013 por la Academia de Ciencias sobre el estado de la enseñanza de la informática en Francia, se reconoce que el país (y Europa en general) está significativamente atrasado en términos de concepto e industria en el área del desarrollo de software, el internet de las cosas y la inteligencia artificial en comparación con los países más dinámicos, como los Estados Unidos y ciertos países asiáticos, y asegura que este retraso está parcialmente relacionado con las deficiencias en la enseñanza de la informática que se ha estancado durante mucho tiempo o se ha reducido al aprendizaje solo de los usos de los productos básicos (L'Académie des Sciences, 2013). Como consecuencia, recomienda establecer una educación en ciencias de la computación desde la escuela primaria hasta la secundaria, orientada a comprender y dominar las ciencias de la computación; incluir la enseñanza de las TIC en la capacitación inicial de los maestros de escuela y capacitar a los maestros activos a través del desarrollo profesional proactivo para que todos puedan presentar a sus estudiantes esta disciplina; así como reclutar maestros de disciplina de TI en la escuela secundaria con un requisito de nivel y diploma idéntico al de otras ciencias en la escuela secundaria. Por último, aconseja la participación de investigadores para el desarrollo de una mejor comprensión de lo que debe incluir un plan de estudios en este campo.

### **Experiencias en Latinoamérica**

#### ***México***

La Ley General de Educación contempla la educación digital y la impulsa a través de la Agenda Digital que transforma las Tecnologías de Información y Comunicación (TIC) en Tecnologías de la información, comunicación, conocimiento y aprendizaje digitales (TICCAD), como resultado de la propia evolución conceptual del término TIC y de la aplicación y uso en los campos del aprendizaje, adquisición, construcción y divulgación del conocimiento (SEP, 2020). Desde 2017, se incluyó a la programación dentro del apartado de Autonomía Curricular, para que las escuelas que así lo decidan, incorporen intervenciones educativas sobre programación y robótica (Cárdenas, 2020). Cuenta además con la plataforma @prende 2.0 que forman parte de

la Estrategia Digital en Educación de la Secretaría de Educación Pública (2020) para fomentar el uso de las TIC, el desarrollo de las habilidades digitales y el PC.

### *Argentina*

La Fundación Sadosky publicó en el 2013 un manifiesto denominado “CC-2016: Una propuesta para reformar la enseñanza de la computación en las escuelas argentinas”, cuya meta es concientizar a la comunidad científica respecto a la importancia de un cambio profundo en la enseñanza de la educación secundaria y media superior con la introducción de los principios computacionales en su estructura. El documento explica que la enseñanza de la computación es esencial para lograr mayores oportunidades provenientes de estas tecnologías. Además, la organización advierte sobre la necesidad de los estudiantes de desarrollar habilidades esenciales para la vida moderna.

En el 2018 la provincia de Buenos Aires incorporó en el diseño curricular de educación primaria las ideas del PC con la intención de incorporarlo transversalmente en las diferentes áreas curriculares (Dirección General de Cultura y Educación de la Provincia de Buenos Aires, 2018; Queiruga et al. 2019). Mientras que en la educación secundaria se incorporó desde el 2010 la asignatura “NTIC” ubicada en cuarto año, que es el único espacio curricular donde se abordan algunos conceptos relacionados a las Ciencias Informáticas (Dirección General de Cultura y Educación de la provincia de Buenos Aires, 2010).

Actualmente, la enseñanza de la Informática en la escuela forma parte de la agenda de trabajo del Ministerio de Educación de la Nación y de varios ministerios provinciales, con programas de políticas educativas como el proyecto “Program.AR” (“Program.AR”, s.f), la creación del “Programa Conectar Igualdad” y más recientemente el “Plan Nacional Integral de Educación Digital” (PLANIED, sf), el proyecto “Secundaria 2030” (Secundaria 2030, sf). (Queiruga et al. 2019).

### *Chile*

La introducción de los recursos de la tecnología digital en la secundaria y la educación media superior comenzó hace 20 años, y culminó con la creación del examen SIMCE-TIC (Sistema de Medición de la Calidad de la Educación) que se aplica desde 2011. De acuerdo con el “Reporte Global de la Tecnología de la Información” de 2014, publicado por el Foro Económico Mundial, Chile es líder en Latinoamérica en la Información, Comunicación y Tecnología de la Información, con una posición en el lugar 35 a nivel global y de acuerdo con el Banco Interamericano de Desarrollo, también posee la mejor infraestructura escolar en Latinoamérica.

Pero incluso con esas iniciativas, los estudiantes chilenos están lejos de acceder a los conceptos de computación en las escuelas, especialmente los jóvenes que viven en contextos vulnerables donde hay menos acceso a la tecnología (González-González, 2019).

También existen campañas e iniciativas ONG alrededor del mundo que promueven el aprendizaje informal de competencias computacionales desde edades tempranas (Bocconi et al., 2016), tales como: CS Unplugged, Bebras, Scratch, PBS KIDS Scratch Jr., Minecraft, Kodable, codeSpark Academy, Sphero, Khan Academy, Code Club, Code Avengers, EU Code Week, Code.org: The Hour of Code, CoderDojo, Code Club, Made with Code, European Coding Initiative, Computing at School, CompuThink, Code for Change, Google CS First, Epic Queen, Cuantrix.mx, Programamos.es, Program.ar, entre otras.

### **Políticas y Prácticas Relacionadas con el PC**

La enseñanza de la programación en educación básica puede rastrearse hasta 1960 cuando Seymour Papert desarrolló LOGO, el primer lenguaje de programación diseñado especialmente para niños y pensado como encuadre metodológico para la enseñanza de las matemáticas (Feurzeig & Papert, 2011). Al programar el ordenador, el niño adquiere un sentido de maestría frente a uno de los tipos de tecnología más potente y novedosa, y establece un contacto íntimo con algunas de las ideas más profundas de las ciencias, las matemáticas y la construcción de modelos intelectuales (Papert, 1980). Sin embargo, Logo no se incorporó en la mayoría de las escuelas estadounidenses en 1980 debido probablemente, a la incompatibilidad entre su enfoque habilitado para el descubrimiento y el enfoque más convencional de la cultura escolar de ese entonces (Agalianos et al., 2001).

Fue hasta el año 2006 cuando Jeannette Wing tomó el puesto como director asistente de la National Science Foundation (NSF) de los Estados Unidos y propuso que el PC era un estilo de pensamiento que todos necesitaban aprender en la era de la computación. En concordancia con esto, destinó grandes cantidades de dinero para capacitar a los educadores, diseñó los primeros cursos para las universidades sobre los nuevos principios de la ciencia computacional, definió el PC para el K-12 e hizo recomendaciones para el currículo de las mismas escuelas del K-12. A esto se le conoce como el movimiento de “ciencia computacional para todos” que ha tenido mucho éxito en llevar la computación a las escuelas (Denning & Tedre, 2019).

Los primeros intentos de llevar la ciencia computacional a las escuelas de educación básica en los Estados Unidos datan del año dos mil, cuando los educadores de la ciencia computacional trabajaron junto a los educadores del K-12 (que en México equivale a los años escolares que van del kínder a la primaria) para desarrollar cursos de alfabetización informática,

sin embargo, estos no han tenido mucho éxito alrededor del mundo. Por ejemplo, Furber (2012), señala que la educación computacional que se imparte en las escuelas en el Reino Unido es altamente insatisfactoria, pues a pesar de que la currícula para la enseñanza de las tecnologías de la información y la comunicación (TIC) son amplias y dan lugar a que los maestros inspiren a los alumnos y los ayuden a desarrollar el interés en la Computación, muchos alumnos no son inspirados por lo que se les enseña y solo desarrollan las competencias digitales básicas como el uso de un procesador de texto o una base de datos.

Entre las causas que Furber encuentra para dicha situación se cuentan las siguientes: El currículo nacional para la enseñanza de las TIC puede ser ampliamente interpretado y es necesario reducirlo a un nivel donde los maestros que no son especialistas en el tema puedan impartirlo; escasean los maestros con capacidad de enseñar Ciencias Computacionales (CC) más allá de la literatura digital básica; se carece de programas que den continuidad al desarrollo profesional de maestros de CC; y la infraestructura de las escuelas inhiben la enseñanza efectiva de las CC (Furber, 2012).

Según Román- González (2015) si bien no hay consenso en cuanto a la mejor manera de incorporar el PC en la educación básica: obligatoria, optativa o transversal a otras áreas, en Europa hay 13 países que han optado por integrar la enseñanza de la programación en la Educación Secundaria, mientras que otros diez han optado por incorporarlo en la etapa de Educación Primaria. Esta incorporación se realizada esencialmente o como nueva disciplina dentro del programa escolar o bien, de manera trans, multi o interdisciplinaria (Brackmann et. al., 2016).

Mientras en países europeos se desarrollan planes para la incorporar el PC en el ciclo de formación básica, en la mayoría de los países latinoamericanos no se considera incluirlo en el currículum porque las clases de computación en estos países aún tienen el enfoque puesto en el desarrollo de las TIC para el uso eficiente de dispositivos informáticos, que dista por mucho del enfoque que se requiere para el desarrollo de las habilidades cognitivas implícitas en el PC que incluyen la resolución de problemas a partir de procedimientos sistemáticos y orientados a la computación (Balanskat & Engelhardt, 2015).

En 41.38% de las investigaciones educativas que se realizaron en la última década para validar los métodos de incorporación del PC en las escuelas se efectuaron en Estados Unidos (Heintz, Mannila y Farnqvist, 2016). De estas investigaciones, el 51.72% de las intervenciones tomaron contenidos curriculares de otras materias para realizar actividades que permitieran la aplicación y desarrollo del PC, como es el caso de las investigaciones de Arastoopour Irgens et

al., (2020) en el área de ciencias y la de Dickes, Farris y Sengupta (2020) en matemáticas. Dichos trabajos tienen a la programación como común denominador porque es la estrategia didáctica más empleada para desarrollar el PC y la que cuenta con mayor diversidad de instrumentos para la evaluación de sus productos desde diferentes perspectivas (Gutiérrez, Cordero y González, 2020).

Otras de las estrategias utilizadas por los investigadores del PC son la robótica y el PC desenchufado. La primera con un 31.03% y la segunda con un 6.90%. Entre los investigadores que han utilizado la robótica se encuentran Barr, Harrison, y Conery (2011); Saritepeci y Durak, (2017); Muñoz-Repiso y Caballero-González (2019); y Del Olmo-Muñoz et al., (2020), quienes explican que el principal beneficio de esta como estrategia didáctica es que ofrece una retroalimentación inmediata sobre el desempeño del programa que se acaba de crear, de modo que los niños pueden ver en el mundo real a través del hardware las consecuencias de su programación (software). Autores como Zapata-Ross (2015) y Bell y Vahrenhold (2018) prefieren estrategias desenchufadas para desarrollar el PC porque este permite que los estudiantes comprendan y apliquen los principios del PC y de la programación sin utilizar computadoras ni ningún otro tipo de dispositivo electrónico, para lo cual recurre materiales del Método Montessori.

La decisión sobre qué tipo de estrategia didáctica utilizar depende de varios factores a considerar, entiéndase estos como la edad del sujeto en formación y su correspondiente madurez cognitiva, así como del entorno donde se realizará la intervención y de la infraestructura con la que cuentan las escuelas, así como del entusiasmo de las autoridades educativas y de los maestros. De este modo, para las intervenciones en preescolar generalmente se utiliza el PC desenchufado que recurre a los mismos objetos ya presentes en el aula para el desarrollo de los conceptos y habilidades del PC, así como a juegos de mesa de tamaño real (Da Cruz Alves, Gresse Von Wangenheim, y Hauck, 2019); en primaria menor (primer a tercer grado) se recurre preferentemente a la robótica y entre los kits favoritos para esto se encuentran KIBO y Bee-Bot, mientras que en primaria mayor (quinto a sexto grado) se introduce el uso de las plataformas de programación visual, siendo la más popular Scratch que permite también trabajar con microcontroladores como Arduino (Brennan y Resnick, 2012; Burke, 2012; Lye & Koh, 2014), y para robótica en este nivel educativo la herramienta favorita es Robot Virtual Worlds. En secundaria se continúa con las plataformas de programación visual y se agregan recursos como ASPIDE para SPARC (Gutiérrez, Cordero y González, 2020).

Sin embargo, también hay investigaciones que han llevado la robótica al preescolar utilizando BOTS y Lego Education WeDo 2.0; en primaria fue Scratch, Scratch para Arduino, Robot Virtual Worlds; y en secundaria ASPIDE para SPARC (Gutiérrez, Cordero y González, 2020).

### **Transferencia del PC**

Concepto y desarrollo de la TPC hacia las diversas disciplinas académicas. En 1980, Papert aseguraba que la experiencia de programar con LOGO podía desarrollar poderosas habilidades intelectuales de pensamiento entre los niños; sin embargo, la evidencia empírica de los estudios realizados en relación con la programación en LOGO no encontró evidencia concluyente de ello (Pea et al., 1985).

De acuerdo con los postulados de la Disciplina Formal (Binet, 1899) la T fue vista inicialmente como un tipo de propagación global de habilidades lograda por el entrenamiento de las facultades mentales básicas (lógica, atención, memoria) en el ejercicio de materias como el latín o la geometría. La teoría de la T basada en los elementos comunes tuvo su origen en la psicología americana pre-conductista con Thorndike y Woodworth (1901), quienes realizaron una serie de experimentos en busca de especificidad en las funciones mentales, así como la difusión o transferencia de la competencia adquirida a otras funciones (estrechamente relacionadas); concluyeron que la T de competencia no va mucho más allá de los efectos prácticos directos. Thorndike y sus colegas informaron que numerosos experimentos encontraron una T pobre o desigual a través de tareas dispares que implican operaciones similares. Por lo tanto, la T se considera limitada por y para los elementos idénticos compartidos en las situaciones de aprendizaje. Esta afirmación de "elementos comunes" ha estimulado gran parte de la investigación de T en las últimas décadas, aunque la naturaleza de los elementos ha cambiado. Las versiones modernas del enfoque de T de elementos comunes difieren en varias formas del original, dado que se centran en una variedad de unidades de T y proporcionan una descripción más detallada de las unidades que se cree que coinciden entre la fuente de transferencia (FT) y el objetivo de transferencia (OT) (Helfenstein, 2005).

La crítica que Judd (1908), realizó a la Teoría de los Elementos Comunes tuvo gran impacto en los Gestaltistas y dio origen a una tradición alternativa de investigación de la T. Judd sostuvo que el aprendizaje como una combinación inadvertida de elementos y su transferencia mecanicista a situaciones no puede sostenerse, porque incluso los animales pueden seleccionar y distinguir aspectos relevantes de los irrelevantes en función de su valor para el logro de un objetivo situacional. Los partidarios de las ideas de Judd argumentaron que

este hallazgo apoya la opinión de que los maestros deberían centrarse en enseñar principios y generalizaciones amplias en lugar de hechos, habilidades y creencias específicas. Selz (1913, 1922) también argumentó en contra de la teoría de la asociación clásica y preparó el campo para los investigadores de la T esquemática con su interpretación del pensamiento como un proceso sensato y dirigido, que constantemente busca completar las estructuras de conocimiento para lograr un significado completo. Dentro de la investigación sobre T basada en esquemas, el énfasis se ha puesto principalmente en el razonamiento analógico y los modelos mentales.

Sin embargo, ni la teoría de los elementos comunes ni la teoría basada en los esquemas ha permitido conocer el proceso mental que se involucra en el fenómeno mismo de la T (Helfenstein, 2005). A partir de los 60's, la psicología cognitiva estudió la mente como si fuese una computadora, aplicando la arquitectura básica de los modelos de procesamiento de información: entrada-proceso-salida. En los procesos cognitivos esto equivale a codificación, almacenamiento y recuperación (Doyle, 2004). Dicha tendencia investigativa descuidó el estudio de las habilidades cognitivas, pero condujo los trabajos hacia la T analógica (aprendizaje con el ejemplo) en la resolución de problemas, con énfasis en los mecanismos de T. En la resolución analógica de problemas, primero se le proporciona al alumno un problema y su solución (la fuente o el ejemplo). Posteriormente, se les presenta un problema similar (Doyle, 2004).

La ciencia cognitiva comparte con la tradición más antigua de la psicología del desarrollo una preocupación por cómo el nuevo aprendizaje debe integrarse con el conocimiento previo, pero trasciende el trabajo anterior en el análisis de la resolución de problemas y los procesos de aprendizaje para dominios de conocimiento específicos, y encuentra poco papel para los principios estructurales generales que invocan "etapas" (Pea & Kurland, 1984).

En la literatura sobre T de aprendizaje, los enfoques de resolución de problemas surgen en dos esferas vinculadas: en primer lugar, en el estudio de los procesos cognitivos en relación con el aprendizaje y la T (por ejemplo, Bransford et al., 1999); en segundo lugar, como una estrategia de instrucción para mejorar el aprendizaje y la T (Albanese & Mitchell, 1993). Otro enfoque para el aprendizaje o la adquisición de conocimientos dentro del campo cognitivo es el de la cognición situada. La investigación sobre resolución analógica de problemas enfatiza los mecanismos de T, mientras que la investigación sobre cognición situada y T enfatiza el contexto (Doyle, 2004). Greeno et al., (1993) distinguen entre las explicaciones empiristas y racionalistas de la T. El empirista busca los elementos o componentes superpuestos de las dos situaciones,



mientras que el racionalista busca una estructura o representación compartida que el alumno lleva de la situación original a la situación de T.

El debate suscitado entre los defensores de la disciplina formal y los defensores de la teoría de los elementos idénticos continúa en el ámbito de la psicología educativa sobre la enseñanza para la T (Fogarty et al., 1992), Un proponente de la segunda línea es Detterman (1993), quien sugiere que los enfoques de enseñanza para T son variantes de la doctrina de la disciplina formal.

La mayor parte de los estudios que se han realizado abordan la T de habilidades cognitivas y del conocimiento. Entre ellos se encuentran los trabajos de Gick & Holyoak, 1980, 1983; Gentner & Gentner, 1983; Holland et al., 1986; Robertson, 2001) y las fuentes de evaluación de similitud utilizadas en las taxonomías de T han sido tradicionalmente estímulos concretos, contexto y dominio del conocimiento, y tareas psicológicas (Helfenstein, 2005).

Fue necesario desarrollar taxonomías más específicas que permitieran medir los diferentes aspectos y elementos que se involucran en el proceso de T. Así, Haskell (2000), elaboró una clasificación de seis niveles de T y una taxonomía de catorce tipos de esta. La clasificación de los seis niveles de T se basa en juicios de similitud (ver Tabla 4).

Tabla 5. Niveles de Transferencia

Nivel de Transferencia	Definición
Nivel 1. No específica	De la idea constructivista de que todo el aprendizaje ocurre a partir del conocimiento presente
Nivel 2. Aplicada	Consiste en la aplicación de lo que se ha aprendido a una situación específica
Nivel 3. Contextual	T libre de contexto entre tareas similares
Nivel 4. Cercana	Cuando el conocimiento previo se transfiere a nuevas situaciones que son muy similares, pero no idénticas a las situaciones anteriores
Nivel 5. Lejana	Esto se refiere a aplicar el aprendizaje a situaciones que son bastante diferentes al aprendizaje original. Caracterizada por el razonamiento analógico

Nivel 6. Creativa o de desplazamiento	El pensamiento analítico permite la síntesis de la experiencia pasada y presente y permite el surgimiento de “algo” nuevo, llámese concepto, proceso, metodología, etc.
---------------------------------------	---

Nota: Elaboración propia basada en Haskell (2000).

Para efectos de la construcción de esta taxonomía, Haskell (2000), usó la definición del psicólogo Walter Weimar donde explica que los estímulos son equivalentes cuando tienen el mismo significado a pesar de sus características diferenciadas. Además, considera que los niveles 1 y 2 son un aprendizaje esencialmente simple, no una T adecuada; el nivel 3 es simplemente la aplicación del aprendizaje, reservando el nivel 4 como T cercana y los niveles 5 y 6 como T remota. También señala que lo que considera una T significativa es la que requiere el aprendizaje de algo nuevo para realizarla. Los niveles 4, 5 y 6 generalmente requieren ese nuevo aprendizaje. Sin el requisito de un nuevo aprendizaje, la T no es tal sino simplemente la aplicación del mismo aprendizaje.

Así mismo, el autor clasificó la T en catorce tipos dependiendo del conocimiento en cuestión: i) declarativo, ii) procedimental, iii) estratégico, iv) condicional y v) teórico (ver Tabla 6).

Tabla 6. Tipos de Transferencia por Tipo de Conocimiento

Tipo de Transferencia	Características	Tipo de Conocimiento
De contenido a contenido o de declaración a declaración	Consiste en hacer uso de lo que se sabe en un área temática para el aprendizaje de otra área. También se refiere al aprendizaje de nuevos conocimientos que pueden ser algo diferentes del aprendizaje original	De contenido o de declaración
De procedimiento a procedimiento ó de habilidad la habilidad	Se refiere al uso de los procedimientos aprendidos en un área de habilidades en otra área de habilidades	Procesal
De declaración a procedimiento	Ocurre cuando aprender acerca de algo ayuda a hacer algo	Declarativo a procesal
De procedimiento a declaración	Cuando la experiencia práctica en un área ayuda a aprender conocimiento más abstracto de dicha área	Procesal a declarativo
Estratégica	Cuando el conocimiento sobre nuestros procesos mentales, como la forma en que aprendemos o recordamos, se obtiene mediante el monitoreo de nuestras actividades mentales durante	Estratégico

Tipo de Transferencia	Características	Tipo de Conocimiento
	el aprendizaje	
Condicional	Es el conocimiento de cuándo es apropiado usar el conocimiento aprendido en otro contexto	Condicional
Teórica	Es comprender las relaciones profundas de causa y efecto en un área que pueden transferirse a otra	Teórico
General o no específica	Es cuando el conocimiento previo que no es específico de la situación de capacitación se transfiere a otras situaciones a pesar de que no existen similitudes aparentes entre las situaciones antiguas y las nuevas	Teórico, estratégico, condicional, procesal y declarativo
Literal	Consiste en usar conocimiento o procedimientos directamente en la nueva situación de aprendizaje. También puede interpretarse como T cercana	Declarativo y procesal
Vertical	Se refiere al aprendizaje previo transferido a un nuevo aprendizaje que es más alto en una jerarquía de conocimiento, o aprendizaje que presupone el aprendizaje previo	Teórico, declarativo, condicional, procesal
Lateral	Es cuando el aprendizaje previo se transfiere al mismo nivel en una jerarquía	Condicional
Reversa	La T inversa, a veces llamada T de retroceso, ocurre cuando el conocimiento existente (anterior) se modifica y se vuelve a ver en términos de sus similitudes con la nueva información. La T hacia atrás invierte la dirección de la vista típica de la T de proceso	Estratégico
Proporcional	Es una T más abstracta. Reconocer una melodía tocada en una octava o tecla diferente es un ejemplo	Condicional
Relacional	Consiste en ver la misma estructura entre dos cosas	Teórico

Nota: Elaboración propia basada en Haskell (2000).

El conocimiento procesal es conocimiento práctico; el conocimiento estratégico es el conocimiento de nuestros procesos mentales, como la forma en que aprendemos y recordamos;

el conocimiento condicional es el conocimiento de cuándo aplicar nuestro conocimiento de manera apropiada al contexto; el conocimiento teórico es nuestra comprensión de las relaciones de nivel profundo, de causa y efecto, y otras conexiones explicativas sobre los fenómenos; y el conocimiento declarativo es el conocimiento de algo. Haskell (2000), señala por último que ninguno de los tipos de T es necesariamente mutuamente excluyente.

Por su parte, Barnett y Ceci (2002) desarrollaron la Taxonomía de la Transferencia Lejana (TL) la cual puede considerarse como el tipo de T prototípica a la que aspiran todos los sistemas educativos y que está estrechamente relacionada con el estudio del razonamiento analógico. En la taxonomía de nueve niveles elaborada por los autores, los tres primeros se configuran a partir del factor del contenido y los seis restantes del de contexto (ver Tabla 7).

Tabla 7. Taxonomía de Transferencia Lejana

Factor	Nivel
Contenido	La especificidad-generalidad de la habilidad aprendida La naturaleza del cambio de rendimiento evaluado Las demandas de memoria de la tarea de T
Contexto	Campo de conocimiento Contexto físico Contexto temporal Contexto funcional Contexto social Modalidad

Nota: Elaboración propia basada en Barnett y Ceci (2002).

Tradicionalmente, los experimentos para demostrar T se constituían de tal forma que sólo permitían afirmar o negar la presencia de esta, sin embargo, un recurso como la taxonomía propuesta por Barnett y Ceci (2002) permite definir de manera más precisa el nivel de logro del estudiante en la tarea de T considerando las características del contenido y el contexto.

En la Tabla 8 se explican los niveles de T posibles según dichos autores, donde la T se divide para su análisis en 9 aspectos o niveles. Los tres primeros corresponden al contenido que se transfiere, siendo estos i) la especificidad-generalidad de la habilidad aprendida, ii) la naturaleza del cambio de rendimiento evaluado y iii) las demandas de memoria de la tarea de T. Dependiendo del contenido transferido, este puede mostrar tres comportamientos diferentes que van de la TC a la TL.

Por ejemplo, si hablamos de la demanda de memoria (ver elemento tres de la Tabla 8), cuando la tarea de transferencia solo demanda ejecución por parte del estudiante se considera

TC, en cambio, cuando la tarea obliga al estudiante a recordar, reconocer y ejecutar se trata de TL.

Tabla 8. Niveles 1-3 de la Taxonomía de Transferencia Lejana correspondientes al Contenido

Niveles 1-3 de la Taxonomía de Barnett y Ceci ejemplificados			
A. Contenido: Qué se transfiere			
Habilidad aprendida	Procedimiento	Representación	Principio o heurística
Cambio de rendimiento	Velocidad	Precisión	Enfoque
Demanda de Memoria	Sólo ejecución	Reconocimiento y ejecución	Recordar, reconocer y ejecutar

Nota: La ejemplificación es la propuesta por los autores de la taxonomía.

De igual modo, atendiendo al contexto de la T, cuando la tarea inicial y la tarea de T pertenecen a campos de conocimiento muy cercanos (nivel uno de la taxonomía) se trata de TC, y conforme se va alejado un campo del otro, la T también incrementa gradualmente hasta llegar a la TL (ver Tabla 9).

Tabla 9. Niveles 4 -9 de la Taxonomía de Transferencia Lejana correspondientes al Contexto

Niveles 4-9 de la Taxonomía de Barnett y Ceci ejemplificados					
B. Contexto: Cuándo y dónde ocurre la T					
	Cercana			Lejana o remota	
Campo de conocimiento	Cálculo vs. Álgebra	Biología vs. Botánica	Biología vs. economía	Ciencia vs. historia	Ciencia vs. arte
Contexto físico	Mismo espacio en la escuela	Diferente espacio en la escuela	Escuela vs. laboratorio de investigación	Escuela vs. hogar	Escuela vs. playa
Contexto temporal	Misma sesión	Día siguiente	Semanas después	Meses después	Años después
Contexto funcional	Ambos claramente académicos	Ambos académicos, pero uno no evaluable	Académico vs. llenar formatos de impuestos	Académico vs. cuestionario informal	Académico vs. de juego
Contexto social	Ambos individual	Individual vs. parejas	Individual vs. grupo pequeño	Individual vs. grupo grande	Individual vs. Sociedad
Modalidad	Ambos escritos, mismo formato	Ambos escritos, de opción	Leer un libro vs. examen oral	Lectura vs. cata de vino	Lectura vs. tallado de madera

Niveles 4-9 de la Taxonomía de Barnett y Ceci ejemplificados	
B. Contexto: Cuándo y dónde ocurre la T	múltiple vs. ensayo

Nota: La ejemplificación es la propuesta por los autores de la taxonomía.

Tabla 10. Tipos de Transferencia entre Tareas

Tipo	Definición
Transferencia dentro de la tarea	Cuando se realiza a una tarea o situación, que es tan similar a una experiencia previa que en realidad puede considerarse como la misma tarea
Transferencia entre tareas	Se aplica a una situación superficialmente diferente, pero funcionalmente equivalente a la anterior
Transferencia inventiva	Los estudiantes tienen que desarrollar una nueva solución sobre la base de similitudes y diferencias críticas de la tarea de origen y destino

Nota: Elaboración propia basada en Butterfield y Nelson (1991)

Butterfield y Nelson (1991), hacen la distinción entre tres tipos de T según las características de la tarea (Ver Tabla 9). Una fortaleza típica de la teoría orientada a tareas es la especificación objetiva de su mecanismo objetivo, que produce manipulaciones instruccionales precisas. Por lo que los autores de esta taxonomía concluyen que: 1) es más probable que se observe la T en estudios que producen grandes ganancias educativas; 2) considerar las diferencias en el tipo de T podría traer orden al campo; 3) la T es más probable cuando se agregan métodos diseñados específicamente para promoverla a la instrucción diseñada para producir aprendizaje. Los investigadores orientados al comportamiento han ofrecido evidencia a favor de agregar capacitación en múltiples ejemplos, reducir la discriminación entre las pruebas y los entornos de T, recompensar la respuesta generalizada, etc. (Albin & Horner, 1988; Engelmann & Carnine, 1982; Stokes & Baer, 1977).

Por otra parte, Robertson (2001), distingue dos tipos de T y explica que, si dos problemas son muy similares, podemos decir que un problema encontró su mejora, pero no se puede hablar de T (ver Tabla 11).

Tabla 11. Tipos de Transferencia según Robertson

Tipo	Definición
Reproductiva o de Conocimiento	Aplicación del conocimiento a una tarea novedosa
Productiva o de resolución de problemas	Implica la adaptación, es decir, la mutación y mejora, de la información retenida

(Conocimiento fluido)

Nota: Tabla de elaboración propia basada en Robertson (2001).

Cuando alguien puede usar exitosamente un procedimiento que usó en el pasado para resolver una situación nueva, entonces se habla de T positiva. En cambio, cuando el procedimiento aprendido en el pasado impide el aprendizaje de un nuevo procedimiento se trata de T negativa. (Robertson, 2001). Mayer y Wittrock (1996) plantean una clasificación de los tipos de T donde agrupan distintas posturas teóricas desarrolladas desde 1899 (ver Tabla 12).

Tabla 12. Clasificación de los Tipos de Transferencia

Tipo	Teoría	Ejemplo
Transferencia general de habilidad general	Disciplina formal	Binet, 1899
Transferencia específica de habilidad específica	Elementos idénticos	Thorndike, 1901
Transferencia específica de habilidad general	Teoría de transferencia de los gestaltistas	Judd, 1908
Control metacognitivo de habilidades generales y específicas	Combinación de las tres vistas anteriores	Brown, 1989

Nota: Elaboración propia basada en Mayer y Wittrock, (1996).

Salomon y Perkins (1987) dicotomizan la T de manera diferente, según la cantidad de esfuerzo cognitivo que requiere cada tipo (ver Tabla 13).

Tabla 13. Taxonomía de Transferencia

Tipo	Características
Transferencia de baja trayectoria (TBT)	Implica la activación de rutinas bien practicadas por condiciones de estímulo similares a las del contexto de aprendizaje
Transferencia consciente o de alta trayectoria (TAT)	Implica una abstracción deliberada y laboriosa y una búsqueda de conexiones

Nota: Elaboración propia basada en Salomon y Perkins (1987).

Perkins (2009), señala que las similitudes estructurales profundas de los procesos de pensamiento de las diferentes disciplinas, aunado a una explicación explícita de dichas similitudes, son componentes esenciales para lograr la T y diferencia entre la Transferencia de Baja Trayectoria (TBT), que se produce cuando una actuación practicada casi hasta la automaticidad en un contexto se activa espontáneamente por condiciones de estímulo en otro contexto; y la Transferencia de Alta Trayectoria (TAT) que implica abstracción consciente

deliberada de un contexto y aplicación a otro. Por lo tanto, la T de capacidades cognitivas abstractas y generales puede ocurrir a través de la automatización.

Larsen-Freeman (2013) explica que en lugar de ver a los alumnos “exportar” lo que han aprendido de una situación a otra, se propone que los alumnos transformen su aprendizaje bajo la premisa de que, y así como las experiencias pasadas pueden influir en las actuales, las actividades actuales también pueden alterar la calidad de las habilidades y recuerdos adquiridos previamente. Para lograrlo es necesario enseñar muchas formas de hacer algo, y la adaptación como un proceso, por lo tanto, la iteración y la adaptación son componentes críticos esenciales para la T porque permiten la movilización del aprendizaje más allá del aula (OCDE, UNICEF & UNESCO, 2016).

Las personas a menudo usan ejemplos familiares relevantes o principios generalizados de un grupo de problemas que comparten una solución subyacente para resolver problemas desconocidos (objetivo) (Holyoak & Thagard, 1997; Reed, 1993). Cuando los problemas fuente y destino difieren en el dominio semántico, la resolución exitosa de problemas analógicos depende en gran medida de la construcción de un esquema abstracto (Gick & Holyoak, 1983). Un esquema abstracto en la resolución de problemas se refiere a una representación mental general de una estructura de objetivos compartida por un conjunto de problemas o un principio de solución que se puede aplicar para resolver el conjunto de problemas. La inducción de esquemas en la cognición humana implica la extracción de información general y esencial de instancias individuales que se encuentran.

Gick y Holyoak (1987) identifican cuatro clases de variables que pueden afectar a estos procesos de T, como son: (i) la similitud entre la estructura de las tareas durante la formación y la T; (ii) los factores de la instrucción que afectan el nivel, la especificidad y la interconexión de las tareas que se encuentran codificadas en la memoria; (iii) lo apropiado de claves de recuerdos; y (iv) el impacto de la historia pasada del sujeto en el aprendizaje y en la T. Butterfield y Nelson (1991) afirman que la T es más probable cuando se agregan métodos diseñados específicamente para promoverla a la instrucción diseñada para producir aprendizaje. Mientras que Sternberg y Frensch (1992) identificaron mecanismos que podrían facilitar la T exitosa a través de la instrucción como la codificación de la especificidad, la organización del conocimiento, la práctica de discriminaciones y el desarrollo de un conjunto mental para la T.

Existe una aceptación cada vez mayor entre investigadores y profesionales de que las habilidades, los conocimientos y las actitudes específicas se transfieren a contextos similares. El problema radica en la T de habilidades más generales o genéricas a nuevos contextos, por



ejemplo, habilidades de resolución de problemas y comunicación. Examinar cómo aprender a programar se transfiere a las capacidades cognitivas de los niños (por ejemplo, la planificación) y aumenta el desarrollo cognitivo, debería ser una de las dimensiones esenciales en la evaluación de la eficacia de los programas educativos de PC (Chen et al., 2017).

Algo de ayuda para dar sentido a las circunstancias proviene de distinguir entre lo que puede transferirse de la programación y cómo ocurre la T. Si bien muchos se han ocupado de lo primero, se sabe poco sobre lo último. Sin embargo, es el conocimiento de los "cómo" de esta lo que puede permitir predecir cuándo ocurrirá y sugerir formas de promoverla. Pea y Kurland (1984), ofrecen una visión general al respecto, concluyendo entre otras cosas que: (i) diferentes niveles de competencia en programación permiten la T de diferentes conceptos y habilidades; (ii) la T de la resolución y planificación de problemas requiere una metacognición considerable; (iii) la T no ocurre espontáneamente; requiere orientación y modelado.

Dos catalizadores clave subyacen a las creencias de que la programación disciplinará el pensamiento: el primero desde la perspectiva de la inteligencia artificial, donde la construcción de programas modelando la complejidad de la cognición humana es vista como una manera de entender ese comportamiento. La segunda influencia es la amplia asimilación de las epistemologías constructivistas del aprendizaje más conocidas por el trabajo de Piaget (Pea & Kurland, 1984).

Papert (1972, 1980) fue un gran defensor de la explicación piagetiana de la adquisición de conocimiento a través de experiencias de resolución de problemas autoguiadas e influyó en las concepciones de los beneficios de aprender a programar a través de un proceso que tendría lugar sin una enseñanza deliberada u organizada. Por eso, quienes esperaban ver la T manifestarse de manera espontánea obtuvieron resultados negativos en sus experimentos.

Pea y Kurland (1984) sugieren que aprender a programar trae acerca de siete cambios fundamentales en el pensamiento: (i) provoca un pensamiento riguroso y una expresión precisa (ya que las computadoras ejecutan algoritmos específicos); (ii) comprensión de conceptos generales como procedimiento formal, variable, función y transformación (ya que se usan en programación), (iii) mayor facilidad con el arte de la "heurística", (iv) enfoques explícitos útiles para resolver problemas en cualquier dominio, como planificar, encontrar un problema relacionado, resolver el problema descomponiéndolo en partes, etc.; (v) aplicar la "depuración" de errores a cualquier tipo de resolución de problemas; (vi) inventar pequeños procedimientos como bloques de construcción para construir gradualmente soluciones a grandes problemas

(modularización); (vii) mejora de la autoconciencia y alfabetización sobre el proceso de resolver problemas.

La teoría psicogenética de Piaget que enuncia 4 estadios de desarrollo cognitivo que van de los 0 a los 12 años (en adelante) permite plantear la TPC a partir de cierta madurez cognitiva alrededor de los 12 años. Dicha teoría influyó significativamente también en Seymour Papert y su filosofía educativa constructivista en la que sobresale el papel activo del estudiante no solo en la construcción de su aprendizaje sino en la construcción, por sus propios medios, de un objeto tangible representativo para ellos del nuevo aprendizaje adquirido (Papert, 1980).

El proceso de adquisición de las habilidades cognitivas sigue un esquema de tres fases: i) la inicial, ii) la intermedia, y iii) la final (Fitts, 1964; VanLehn, 1996) y al brindar al estudiante experiencia con el uso de variables, un conflicto cognitivo, metacognición y conocimiento de estrategias se facilita el aprendizaje (Shayer y Adey, 1993). En un inicio el conocimiento se adquiere eminentemente ligado al contexto donde se adquiere y no tiene aplicación general hasta que el individuo hace las representaciones mentales de la instrucción recibida y el conocimiento se vuelve flexible, lo que lo prepara para la T Hatano y Inagaki (1992).

### **Técnicas que favorecen la Transferencia del PC**

La T de aprendizaje entre dominios generalmente requiere conexiones estructurales profundas entre los dominios y una instrucción explícita sobre cómo aplicar los conceptos de una disciplina en la otra (Bransford & Schwartz, 1999; Perkins, 2009; Rich et al., 2013).

Para alcanzar la T se requiere poseer ciertas habilidades que permitan reconocer la similitud entre dos situaciones, así como promover el uso de diferentes tipos de conocimiento para que el estudiante identifique y active el conocimiento necesario y pueda aplicarlo correctamente (Haskell, 2000). La T lejana o de alta trayectoria, requiere profundidad en la comprensión de los conceptos, así como su generalización, lo que obliga a cambiar entornos y construir relaciones, en lugar de reproducir las ya existentes entre tareas regulares o comunes (Lobato, 2006).

Así los procesos de T demandan que el estudiante desarrolle un razonamiento flexible que se puede lograr utilizando una amplia gama de estrategias para diferentes situaciones, tales como: la construcción de explicaciones, el modelado, la indagación y los enfoques basados en proyectos y resolución de problemas (Gómez, Sanjosé y Solaz-Portolés, 2012), y se debe iniciar por establecer vínculos conceptuales tanto dentro como entre disciplinas temáticas (Bloom,

2010). Cuando el objetivo es la T entre dominios, se debe incluir contextos de todos los ámbitos viables para el aprendizaje de un cuerpo de conocimiento y su aplicación en situaciones reales.

Otras estrategias que son ampliamente recomendadas para contribuir a la T se basan en proporcionar a los estudiantes lo siguiente: (1) tiempo adecuado para explorar los conceptos subyacentes y generar conexiones; (2) variedad de ejemplos resueltos con una solución detallada y explicada; (3) equilibrio entre actividades concretas dependientes del contexto y aquellas de conocimiento abstracto transcontextualizado; (4) hacer comparaciones estructuradas de diferentes problemas, buscando características profundas y significativas; (5) discutir las condiciones de aplicabilidad de una u otra metodología en diferentes escenarios; y (6) situaciones de autocontrol sobre el aprendizaje (metacognición) (Lodi, 2020)

La retroalimentación a estudiantes sobre su proceso de aprendizaje se vuelve fundamental para lograr la TPC, así como fomentar el razonamiento analógico y la articulación de relaciones causales proporcionándoles una guía con indicaciones precisas que les permitan recordar un ejercicio diferente resuelto con estrategias similares, pues las evidencias indican que la T es más efectiva en una instrucción en la que el docente guía al estudiante (*instrucción directa*), en comparación con aquella donde el estudiante descubre por sí mismo lo que tiene que aprender (*aprendizaje por descubrimiento*) (Klahr & Nigam, 2004).

Las estrategias para lograr TPC, toman como base las situaciones iniciales de aprendizaje que plantea el docente, en donde, se recomienda: (1) buscar similitud entre lo que se enseña y el nivel T que se desea lograr; (2) proporcionar experiencia y práctica suficiente al estudiante; (3) ofrecer diversos ejemplos para apoyar los conceptos y principios que se enseñan; e (4) identificar las características importantes de la tarea para la comprensión de los principios generales. Además, se vuelve fundamental conducir las clases de manera que enfatizan la atención plena de los estudiantes en los procesos de orden superior de resolución de problemas (Wenzelburger, 2020).

### **Estudios Sistemáticos Realizados en Este Campo**

El libro *Transferencia en Juicio [Transfer on Trial]* (Detterman & Sternberg, 1993) hizo visible que la T es un tema que necesitaba ser aclarado. En él se la define como "el grado en que un comportamiento se repite en una nueva situación" (p. 4), y afirma que los estudios que proclaman haberla encontrado solo pueden haberlo hecho en el sentido más generoso de los criterios, lo que no cumpliría con la definición clásica de T.

Los estudios muestran que el alcance de la T y su dirección (es decir, positiva o negativa) depende en gran medida de la definición de T empleada, el nivel de referencia durante el entrenamiento (es decir, el inicio o el final del entrenamiento) y la medida dependiente utilizada para evaluar el desempeño (por ejemplo, tiempo de iniciación o tiempo de ejecución) (Healy & Wohldmann, 2012).

Los estudios de T analógica implican capacitación en una tarea seguida de pruebas en una tarea nueva que es análoga a la primera, para ver si la capacitación se transfiere a la tarea analógica y han arrojado innumerables resultados aparentemente contradictorios. Muchos de estos estudios han tenido éxito en mostrar la T, pero varios estudios similares han fallado (Barnett & Ceci, 2002).

Los métodos típicos utilizados para medir la T consisten en comparar el rendimiento o cambio de desempeño entre la prueba que se aplicó previo al entrenamiento y la tarea nueva que se introdujo por primera vez en la prueba posterior a este, o bien utilizándolo como valor de referencia el desempeño de un grupo de control de participantes que no recibieron entrenamiento, dependiendo del diseño experimental (Healey y Wohldmann, 2012).

El grupo Jardín de Niños y Más Allá del Aprendizaje a lo largo de la Vida [Kindergarten and Beyond LifeLong Learning (KGB L3)], formado por investigadores de la Universidad Rey Juan Carlos y la Universidad Nacional de Educación a Distancia, en España, ha desarrollado diversas investigaciones que muestran que el desarrollo del PC a través de la programación tiene un impacto positivo en el aprendizaje de distintas disciplinas, como las matemáticas, los idiomas, las ciencias o la narrativa (Meseguer et al., 2015; Moreno et al., 2017). De igual modo estos autores afirman que los alumnos de los últimos cursos de primaria, que se acercan a la adolescencia, tienen unos niveles de madurez cognitiva que pueden aumentar las posibilidades de T.

Moreno et al. (2017), concluyeron además que la integración de la programación en Matemáticas fue más dura cognitivamente que en las ciencias sociales, lo que se tradujo en que los estudiantes disfrutaron más y mostraron mayor autonomía y motivación en estas últimas; mientras que Schiff y Vakil (2015) señalan que la transferencia lograda varía según la edad y el consiguiente desarrollo cognitivo de los participantes en la investigación, siendo menor en los participantes más jóvenes.

Al respecto, la investigación de Brown (1989), demostró que los niños pequeños si son capaces de lograr la T cuando poseen el conocimiento suficiente y que transfirieren con mayor éxito cuando entienden los eventos a un nivel causal en lugar de simplemente aprender a

replicar comportamientos particulares. Los niños no pueden aplicar esquemas causales aplicados a un tema sobre el cual no saben nada. Cuando se cumplieron estos criterios, Brown (1989) descubrió que los niños transfirieron con éxito un principio general a un contexto novedoso.

Arfé et al. (2020), también examinaron cómo el desarrollo de habilidades de programación en alumnos de primer grado de primaria se transfirió a dos importantes funciones ejecutivas: planeación e inhibición de la respuesta. Los resultados demostraron que la práctica con la plataforma Code.org no solo mejoró la habilidad de los niños para resolver problemas, sino que causó un incremento en el tiempo que los niños pasan planeando, su habilidad para resolver tareas de planeación y para inhibir respuestas prepotentes.

Ortega-Ruipérez y Asensio-Brouard (2018) realizaron un estudio para lograr la transferencia del PC a contextos reales a través de una estrategia para la enseñanza de la programación y la robótica, en el cual se muestra una mejora significativa del grupo experimental respecto al grupo control en todas las fases de la resolución de problemas.

De manera similar, el trabajo de Salehi et al. (2020), presenta evidencia de que el entrenamiento en PC es transferible a la resolución de problemas en el dominio de otras ciencias e hizo manifiesto que los estudiantes de Ciencias Computacionales (CC) de grados avanzados poseen mejores habilidades para la resolución de problemas que los alumnos de CC de primer grado y que los estudiantes de todos los grados de las otras carreras.

Molina-Ayuso (2020) realizó una investigación para medir cómo la habilidad de resolución de problemas matemáticos se incrementa en estudiantes de secundaria gracias a la experiencia educativa con la programación, para la cual siguió un diseño cuasi experimental con pretest y post test y encontró que un 72.2 % del alumnado consigue mejor puntuación en el postest.

Schanzer et al. (2015) utilizaron Bootstrap, un plan de estudios informático diseñado para reforzar objetivos específicos de aprendizaje en álgebra que permite a los estudiantes aplicar directamente los procesos de resolución de problemas para programar a problemas estándar en álgebra, sin embargo, los autores concluyeron que si bien la programación ha tenido cierto éxito en el desarrollo otras competencias matemáticas, la transferencia al álgebra ha sido difícil de establecer.

Por último, Adey y Shayer (1993) encontraron evidencia de T de la clase de ciencias a los exámenes de inglés tomados 2 años después de su intervención para entrenar habilidades

metacognitivas y de pensamiento de orden superior. Por lo tanto, hay alguna evidencia de T exitosa de habilidades de razonamiento relacionadas con la inteligencia de orden superior.

En la presente investigación se trabajó con la Taxonomía de Transferencia Lejana de Barnett y Ceci, (2002), porque permite identificar los diferentes niveles de transferencia de la habilidad de resolución de problemas a través de los elementos principales del PC (identificación del problema, descomposición, abstracción, reconocimiento de patrones y elaboración de algoritmos) que lograron los participantes.

Con la intención de alcanzar la TL de cada uno de los componentes del PC se ofreció el ETPC al grupo de tratamiento, sin embargo, el contexto físico, temporal y social fueron fijados como TC porque debido a las restricciones sanitarias impuestas por el COVID-19 todas las sesiones se realizaron de manera online (contexto físico), todas las actividades de T se evaluaron en la misma sesión (contexto temporal) y todos los ejercicios se realizaron de manera individual (contexto social) (ver Tabla 14).

Tabla 14. Taxonomía de Transferencia Lejana del PC como se aplicó en este estudio.

Habilidad Transferida: Elementos del PC					
	Identificación del problema	Abstracción	Descomposición	Reconocimiento de patrones	Diseño de algoritmos
1. Habilidad aprendida	Lejana				
2. Cambio de rendimiento	Lejana				
3. Demanda de memoria	Lejana				
4. Campo de conocimiento	Lejana				
5. Contexto físico	Cercana				
6. Contexto temporal	Cercana				
7. Contexto funcional	Lejana				
8. Contexto social	Cercana				
9. Modalidad	Lejana				

Enseguida se ejemplifica cómo se aplica la Taxonomía de Transferencia Lejana al primer elemento del PC, la identificación del problema (ver Tabla 15).

Tabla 15. Transferencia Lejana de las Habilidades de Identificación del Problema

A. Contenido: Identificación del problema					
1.Habilidad aprendida	Principio o heurística				
2.Cambio de rendimiento	Velocidad	Precisión		Enfoque	
3.Demanda de Memoria	Recordar, reconocer y ejecutar				
B. Contexto: Cuándo y dónde ocurre la T					
Cercana			Lejana o remota		
4.Campo de conocimiento	CC vs. Desarrollo de Software	CC vs. Ingeniería	CC vs. Economía	CC vs. Historia	CC vs. arte
5. Contexto físico	Salón de clases	Diferente salón de clases	Salón vs. laboratorio de computación	Salón vs. hogar	Salón vs. playa
6. Contexto temporal	Misma sesión	Día siguiente	Semanas después	Meses después	Años después
7. Contexto funcional	Ambos claramente académicos	Ambos académicos, pero uno no evaluable	Académico vs. llenar formatos de impuestos	Académico vs. cuestionario informal	Académico vs. de juego
8. Contexto social	Ambos individual	Individual vs. parejas	Individual vs. grupo pequeño	Individual vs. grupo grande	Individual vs. sociedad
9. Modalidad	Ambos con la herramienta de programación 1, reto 1	Herramienta 1, reto 2	Herramienta 2, reto 2	Herramienta 3, PC desenchufado	Herramienta 3 vs. situación de la vida cotidiana

Nota: Ejemplo de la aplicación de la Taxonomía de Barnett y Ceci (2002), al primer pilar del PC. Niveles 4-9. Elaboración propia.

De acuerdo con esta taxonomía, la TL de la identificación del problema se logra cuando los niños son capaces de precisar cuál es el problema que necesita ser resultado en cualquier contexto y hace un cambio en el enfoque de resolución del problema para reducir el tiempo de solución y aumentar la precisión de las estrategias aplicadas, pudiendo trabajar colaborativamente.

A menudo, la T espontánea es suprimida en los principiantes debido a su organización mental de la información. Para solucionar esta situación es necesario ayudar a los alumnos a identificar las partes estructurales de los procedimientos de resolución de problemas y

permitirles conectarlos con una variedad de casos (Robins et al., 2019). Desde la perspectiva de la cognición, aprender contenido no es lo mismo que aprender a transferir contenido (Bransford et al., 1999; van Merriënboer et al., 2002). Para lograr este último se requiere de la metacognición, que implica funciones como: (i) planificación del aprendizaje, su supervisión durante la marcha (o monitoreo) y evaluación del éxito del aprendizaje y de la aplicación de diferentes estrategias, y depende de las variables (i) persona, (ii) tarea y (iii) estrategia (Flavell, 1987; Valenzuela, 2019).

De acuerdo con la literatura revisada, es importante destacar que la presencia o no de la T en los experimentos para probarla depende de plantear el diseño adecuado para encontrar el nivel y tipo de T deseada, de lo contrario, los experimentos realizados en este sentido fallarán por un error en el diseño. Y también es muy importante diseñar la intervención de PC considerando la habilidad que se desea transferir para crear las condiciones de infraestructura y las pautas de instrucción necesarias para que esta ocurra.

Gracias a esta revisión de literatura hemos logrado una comprensión más completa del fenómeno de la Transferencia y sus desafíos, así como del Pensamiento Computacional y de los retos que su desarrollo implica para los estudiantes y para los gobiernos que desean incorporarlo en sus planes de estudio. A continuación, en el capítulo 3, se explica la metodología de investigación que se aplicó en la realización de esta tesis.



### Capítulo 3. Método

En el capítulo anterior se explicaron los constructos teóricos en los que se sustenta la presente investigación, siendo estos el PC y la T, y se presentó una revisión de las iniciativas que existen alrededor del mundo para incorporar el PC en los currículos de educación básica, así como el resultado de algunas investigaciones que se han realizado en los últimos años en torno al proceso de la T del conocimiento. En este capítulo se detallan el carácter y tipo de investigación de esta tesis, así como la metodología, el procedimiento y los instrumentos utilizados para recabar los datos.

La investigación será de tipo cuantitativo porque (i) tiene su objeto de estudio en el planteamiento positivista, (ii) las concepciones que se proponen tienen su origen en el método científico, (iii) el conocimiento sólo se logra a través de la objetividad, (iv) utiliza el método hipotético-deductivo, (v) explica la correlación entre la causa-efecto de los fenómenos en la investigación, (vi) utiliza mecanismos de categoría estadística para procesar las variables del proceso (Campoy, 2018).

Dentro de la metodología cuantitativa se eligió el enfoque experimental para el desarrollo del estudio porque permite manipular con determinada intención una o más variables independientes (supuestas causas antecedentes), para analizar las consecuencias que la manipulación de una de estas variables ocasiona en una variable dependiente (supuestos efectos consecuentes), dentro de una situación creada por el investigador (Fleiss, 2013; O'Brien, 2009 y Green, 2003) Citados por Hernández- Sampieri (2018).

La investigación experimental busca determinar si una intervención influye en el resultado del grupo estudiado. El investigador evalúa esto realizando una intervención en un grupo y ocultándola a otro y luego determina cómo ambos grupos puntúan en un resultado (Creswell, 2013).

Los experimentos también pueden ser llamados estudios de intervención, porque un investigador genera una situación para tratar de explicar cómo afecta a quienes participan en ella en comparación con quienes no lo hacen (Creswell, 2013), por lo tanto, una investigación que implica una intervención (variable independiente) para establecer sus efectos en (variable dependiente) en comparación con un grupo control, se considera un estudio experimental.

Los requisitos para un experimento son: (i) la manipulación intencional de una o más variables independientes, (ii) medir el efecto que la variable independiente tiene en la variable dependiente, y (iii) control o la validez interna de la situación experimental (Hernández-

Sampieri, 2018). En el caso presente, las variables independientes se manipularon a través de intervenciones y luego se midió el efecto de dichos eventos en la variable dependiente.

La investigación es de tipo aplicada, la cual se caracteriza por abordar un problema en su etapa inicial para luego proponer alternativas de solución. En este caso el problema radica en descubrir cómo el PC, como habilidad de resolución de problemas, se transfiere a otras situaciones académicas y de la vida diaria.

El diseño de la presente investigación es factorial. Los diseños factoriales permiten manipular dos o más variables independientes con dos o más niveles o modalidades de presencia en cada una de ellas (Hernández-Sampieri, 2018). Se les considera estudios explicativos por que analizan las relaciones entre una o más variables independientes y una o más dependientes para establecer los efectos causales de las primeras sobre las segundas o las correlaciones entre ellas. Se trata de diseños que se fundamentan en el enfoque cuantitativo y en el paradigma deductivo (Hernández-Sampieri, 2018).

Se emplea un diseño factorial cuando se usan valores seleccionados para dos o más variables independientes en todas las combinaciones posibles. El análisis de los datos de la variable dependiente arroja información sobre: a) la influencia de cada variable independiente sobre la variable dependiente; y b) la interacción entre las dos variables independientes. Se puede afirmar que existe una interacción entre las dos variables independientes si el valor de la variable dependiente de una variable independiente está determinado por el valor específico asumido por la otra variable independiente (McGuigan, 1996). En el diseño factorial puro se coloca a cada participante en una sola condición; es decir, una combinación de los niveles de ambas VI (Clark-Carter, 2002).

En esta investigación se estudió el efecto de dos variables independientes variando dando a cada una de ellas dos niveles. La cantidad de números empleados en la designación indica el número de variables independientes estudiadas en el experimento y el tamaño de los números indica la cantidad de valores de las variables independientes. El 2 x 2 tiene cuatro condiciones experimentales (McGuigan, 1996). En la representación acostumbrada de un experimento factorial se traza una matriz con un factor en cada lado y a los cuadros de la matriz se les llama celdas (Martín, 2008).

Una de las hipótesis de esta tesis sugiere que el NTPC depende del NPC y de que la T se entrene durante una intervención (ETPC). Por lo tanto, las variables independientes son NPC y ETPC, y la variable dependiente es el NTPC (Ver Tabla 16).

**Variables**

***Dimensiones e indicadores de la VI1 NPC***

Una variable independiente (VI) es la que se considera que puede afectar a otra (Clark-Carter, 2002). En este diseño factorial tenemos dos variables independientes, la VI<sub>1</sub> es el NPC; y la VI<sub>2</sub> es el entrenamiento en transferencia del pensamiento computacional (ETPC) (ver Tabla 18).

Aunque en el estudio principal de esta tesis, el NPC del estudiante se considera como variable independiente, en el estudio preliminar puede ser considerada ella misma como variable dependiente, pues surge como resultado de la intervención en PC que se realizó con los estudiantes. Es decir, la intervención de PC funciona como variable independiente y el resultado del postest de PC sería en este punto la variable dependiente.

Tabla 16. Operacionalización de la Variable NPC

Conceptualización	Dimensiones	Indicadores	Ítems	Técnicas/ Instrumentos
El PC es un “proceso de resolución de problemas...” ISTE, 2018 (ver definición completa en la página 13).	Conceptos computacionales	Secuencias Loop o bucles Eventos Paralelismos Condicionales Operadores Datos		Prueba de PC (Román-González, 2015)
	Prácticas computacionales	Experimentar e iterar Probar y depurar Reutilizar y mezclar		
	Perspectivas computacionales	Expresar Conectar Preguntar		

Para la operacionalización de la variable NPC se utilizó la definición de ISTE (2018), las dimensiones del PC propuestas por Brennan y Resnick (2012) y los ítems del examen de PC de Román-González (2015).

***Dimensiones e indicadores de la VI2 ETPC***

Una variable dependiente (VD) es aquella donde una independiente puede tener un efecto, es decir, el valor de la VD depende del nivel de la VI (Clark-Carter, 2002). Las dimensiones que mide la VD son tres de contenido y seis de contexto (ver Tabla 19).

Tabla 17. Operacionalización de la Variable NTPC

Conceptualización	Dimensiones	Indicadores	Técnicas/Instrumento
Estaremos tratando de la T cuando estudiamos el efecto del aprendizaje pasado en el aprendizaje presente. (Ausubel & Robinson, 1969, p. 136).	La especificidad-generalidad de la habilidad aprendida	Procedimiento Representación Principio o heurística	Rúbrica
	La naturaleza del cambio de rendimiento evaluado	Velocidad Precisión Enfoque	Rúbrica
	Las demandas de memoria de la tarea de T	Solo ejecución Reconocimiento y ejecución Recordar, reconocer y ejecutar	Rúbrica
	Campo de conocimiento	Materias hermanas (CC y Matemáticas) Materias cercanas (Matemáticas y física) Materias medianamente cercanas (Matemáticas y economía) Materias medianamente lejanas (CC e Historia) Materias lejanas (CC y Artes)	Rúbrica
	Contexto físico	Mismo salón en la escuela Diferente salón en la escuela Escuela vs. laboratorio de investigación Escuela vs. casa Escuela vs. playa	Rúbrica

Conceptualización	Dimensiones	Indicadores	Técnicas/Instrumento
	Contexto temporal	Misma sesión Día siguiente Semanas después Meses después Años después	Rúbrica
	Contexto funcional	Ambos contextos académicos Ambos contextos académicos, pero uno es no evaluable Académico vs. llenar facturas Académico vs. cuestionario informal Académico vs. lúdico	Rúbrica
	Contexto social	Ambas tareas individuales Individual vs. parejas Individual vs. grupo pequeño Individual vs. grupo grande Individual vs. sociedad	Rúbrica
	Modalidad	Ambos escritos en el mismo formato Ambos escritos, pero uno de opción múltiple y otro tipo ensayo Leer un libro vs. examen oral Lectura vs. cata de vino Lectura vs labrado en madera	Rúbrica

---

Conceptualización	Dimensiones	Indicadores	Técnicas/Instrumento
-------------------	-------------	-------------	----------------------

---

Para la operacionalización de la variable dependiente se utilizó la Taxonomía de Transferencia Lejana de Barnett y Ceci (2002)

A continuación, se muestra la representación gráfica del diseño factorial 2 x 2 que se utilizó en esta investigación, con dos factores de dos niveles cada uno (Ver Figura 1).

Figura 1. Representación Gráfica del Diseño Factorial 2 x 2

Diseño factorial 2 x 2		VI <sub>1</sub> Desarrollo de PC	
		Alto	Bajo
VI <sub>2</sub> Entrenamiento en T	Si	G <sub>1</sub>	G <sub>2</sub>
	No	G <sub>3</sub>	G <sub>4</sub>

Nota: Diseño y elaboración propios.

La principal ventaja de un experimento factorial es que permite estudiar las interacciones entre las variables independientes. Ocurre una interacción cuando la relación entre una variable independiente y el comportamiento del participante (variable dependiente) depende del nivel de la segunda variable independiente. Los experimentos factoriales son los únicos que permiten investigar las interacciones entre las variables.

La validez interna del estudio factorial esta dado por el grado en que un diseño demuestra con éxito que los cambios de una VD obedecen a los de una VI (Clark-Carter, 2002). Para satisfacer el requisito de la validez interna del experimento se realizarán dos intervenciones con grupos similares asignados aleatoriamente, pues según Hernández-Sampieri (2018) la validez interna se alcanza mediante: (i) varios grupos de comparación, y (ii) la equivalencia de los grupos en todo, excepto en la manipulación de la o las variables independientes.

**Covariable**

Con la finalidad de ajustar el modelo de análisis, se incorporó el promedio general de los participantes en el estudio como covariable (Ver Anexos). Esta variable se obtuvo usando el promedio de cinco materias de aprovechamiento académico por cada estudiante que participo

en el estudio. El propósito principal del uso del promedio general como covariable fue para proveer ajustes estadísticos que cada uno de los participantes proveyó como una habilidad académica y de esa manera mantener el equilibrio inicial de cada una de las muestras obtenidas en cada grupo de estudio.

### **Contexto de la investigación**

El trabajo de investigación se realizó en la ciudad de Chihuahua, Chih., durante los meses de abril y mayo de 2021, a través de la plataforma Meet para la parte instruccional de la intervención y la herramienta de programación visual Scratch para desarrollar las tareas de programación y de T. La intervención se estructuró en torno a los conceptos y prácticas computacionales como son descritos en el marco para el desarrollo del PC creado por Brennan y Resnick (2012), y constó de 8 sesiones. La mayoría de los participantes (90%) fueron estudiantes de una escuela secundaria pública ubicada en una zona considerada de alto riesgo por el índice delictivo que presenta y de nivel socioeconómico bajo. El departamento de trabajo social de la institución reporta que los jóvenes reciben poca atención de los padres y pasan gran parte del tiempo solos porque los progenitores deben trabajar.

### **Población y Muestra**

La población de interés con la que se trabajó fueron niños y niñas de entre 12 y 13 años que viven en la ciudad de Chihuahua, con o sin experiencia previa en programación y/o computación, por lo tanto, el tipo de muestreo fue no probabilístico constituido por sujetos voluntarios.

La muestra fue por conveniencia, no representativa, debido a las limitaciones para la investigación que supone la crisis sanitaria por Covid-19.

Dentro del muestreo no probabilístico está el muestreo a propósito, en el que los expertos hacen una selección a propósito cumpliendo el objetivo de que los integrantes de la muestra cumplan algún requisito (León y Montero, 2003). En este caso lo que interesa es elegir a los sujetos para que cumplan con ciertas características en función de las necesidades de la investigación. Además, en investigación a veces se trabaja con muestras incidentales. “Se denomina muestra incidental a aquella que se emplea porque está a disposición del investigador en un momento determinado”, (Pereda, 1987, pág. 127).

Así, la muestra para esta investigación fue incidental y a propósito, constituida por los sujetos voluntarios que respondieron a la convocatoria efectuada en la Escuela Secundaria Técnica No. 29 turno matutino y que mostraron interés en aprender programación en el taller

“LET´S C ♥ DE”. Participaron 11 niños y 9 niñas para hacer una muestra total de 20 participantes. El 90% de ellos cursaban el primer grado de educación secundaria (K7 en la nomenclatura internacional), el 5% el quinto grado de primaria y otro 5% el sexto grado. El 95% de los participantes cursaba sus estudios en escuelas de sostenimiento público y el 5% restante en instituciones de carácter privado. Del total de participantes solo el 10% tenía experiencia previa en programación de uno a tres meses en sus respectivas escuelas de origen.

### **Procedimiento**

La investigación constó de dos momentos y cada uno dio lugar a un estudio por separado. El primer momento constituyó el estudio preliminar y el segundo momento el estudio principal. El estudio preliminar permitió dividir la muestra en Grupo Alto y Grupo Bajo para conformar la variable Grupo A que funcionaría como variable independiente en el estudio principal.

De esta manera, en el primer momento se realizó una intervención en PC de 8 sesiones de 80 min de duración cada una, es decir, 2 sesiones semanales durante 4 semanas, con la finalidad de desarrollar el PC en los participantes. Posteriormente se les aplicó un examen de PC para determinar el nivel de PC que cada participante alcanzó gracias a la intervención. Como ya se mencionó anteriormente, esto dio lugar a la conformación del Grupo Alto y Bajo (variable Grupo A en el análisis estadístico).

Más tarde, en el segundo momento de la investigación, se asignó aleatoriamente a miembros del Grupo A al Grupo de Tratamiento para recibir una intervención en T de 4 sesiones de 60 min de duración cada una, es decir, 2 sesiones semanales durante 2 semanas. Al concluir la intervención en T, se reunió nuevamente a todos los participantes y se les aplicó la tarea de T número cinco que fungió como la evaluación final de T representada como Resultado Final (Y5) en el análisis estadístico.

Durante el ETPC se siguió la técnica de la T analógica aplicada en la resolución de problemas, en la cual se presenta un problema al estudiante y su solución y posteriormente se le pide que resuelva un problema similar (Doyle, 2004; Albanese & Mitchell, 1993). El diseño de las actividades del entrenamiento en T buscó lograr una T de conocimiento procesal a declarativo (Haskell, 2000) con la intención de que la experiencia práctica de la programación en el entorno de Scratch pudiera ayudar a los participantes a aprender conocimiento más abstracto sobre la lógica de la programación que posteriormente pudieran utilizar de manera general en contextos diversos.



Siguiendo las recomendaciones de Sternberg y Frensch (1992) en el ETPC se trató de desarrollar un conjunto mental de acciones que los niños pudieran concientizar y generalizar, de modo que pudieran utilizar los pasos aprendidos para la resolución de problemas en contextos variados.

En cada sesión se instruyó a los participantes a completar un formato físico donde plasmaran su proceso mental para la resolución del problema en cuestión, con la finalidad de que lograsen dar sentido a la información nueva e identificaran los elementos, pasos o etapas que conforman la habilidad de resolución de problemas (Polya, 1945; Chi, Bassok, Lewis, Reitman y Glaser, 1989; Chi, de Leeuw, Chiu y LaVancher, 1994).

Para el diseño de las cinco actividades de T se utilizó como guía la Taxonomía de Transferencia Lejana e Barnett y Ceci (2002) de nueve niveles (Ver Capítulo 2) y considera dos factores: contenido y contexto.

En la Tarea No 1 se mostró a los niños cómo crear un programa en Scratch para calcular el área de un rectángulo y posteriormente se les pidió que crearan un programa en Scratch para calcular el área de un triángulo equilátero. El concepto de programación utilizado en esta actividad es el de secuencias (Brennan & Resnick, 2012). Se considera una actividad de TC en cuanto al contenido porque tiene como objetivo la ejecución de un procedimiento y un cambio positivo en el rendimiento representado por la disminución del tiempo de ejecución del procedimiento.

De igual modo, en cuanto al contexto, la actividad uno es de TC porque implica la aplicación del procedimiento exactamente dentro del mismo campo de conocimiento (geometría), en el mismo entorno de programación (Scratch), dentro de la misma sesión, con objetivos claramente académicos, de manera individual, y ambos en línea. Esta T también podría considerarse una Transferencia dentro de la tarea (Butterfield & Nelson, 1991) porque la tarea de T que se solicitó a los participantes es tan similar a la demostración previa que puede considerarse como la misma tarea.

La Tarea No 2 se llamó Bufete y el reto consistió en elaborar un programa en Scratch que calculara la cuenta del bufete en función del número de personas, aplicando el concepto computacional de Condicionales. De inicio se mostró a los estudiantes cómo calcular el costo de una comida bufete donde los precios variaran en función de la edad de los comensales.

Se considera una actividad de TC porque lo que se busca transferir es un procedimiento, se espera que después de realizar con la maestra un programa similar puedan realizar esta tarea en menor tiempo y en cuanto a la demanda cognitiva no requiere que el alumno recuerde, pues

la distancia temporal entre la actividad previa y la actividad de T es mínima, de modo que el niño solo necesita reconocer las similitudes entre los dos planteamientos y ejecutar la solución.

En cuanto a los contextos que maneja la Taxonomía de TL también se clasifica como de TC por tratarse de una actividad dentro del campo de conocimiento de las matemáticas y se mueve ligeramente en la misma línea hacia la contabilidad. En cuanto a los contextos físico, temporal, funcional y social y la modalidad, tanto la actividad previa como la actividad de T permanecen dentro de los mismos parámetros.

Para la Tarea No 3, el entrenamiento en T se movió a la plataforma de Code.org. con la finalidad de hacer la actividad de El Artista utilizando el concepto de programación de Ciclos. Esta es la tercera y última actividad de TC y su objetivo fue iniciar una transición suave hacia las actividades de TL al presentarle al estudiante el desafío de realizar una animación en Code.org como las que se realizaron previamente en Scratch.

Primero se mostró a los estudiantes como hacer una “obra de arte” visual (animación) con una figura geométrica y utilizando ciclos de programación y luego se le pidió que realizara su propia obra de arte. Esta es una actividad de TC con elementos de TL porque, aunque en contenido se transfiere una fórmula, se busca una mejora en el rendimiento y se requiere solo la ejecución, en el campo de conocimiento obliga al estudiante a desplazarse desde las matemáticas hasta las artes y utilizar un entorno de programación distinto por primera vez y aplicar en él todo lo aprendido en Scratch.

La Tarea No 4, titulada Viajes de Cristóbal Colon es la primera actividad de TL, ya que se desplaza del campo de la programación y las matemáticas al campo de la historia y consiste en aplicar el concepto de programación de Secuencias para realizar un mapa interactivo en Scratch que muestre los tres viajes de expedición de Cristóbal Colón y que despliegue las fechas de cada uno.

De inicio se mostró a los estudiantes las herramientas y recursos creados y cargados por la maestra en Scratch para el proyecto y se ejemplificó como realizar el primer recorrido interactivo. De igual manera se les proporcionó material de lectura para que identificaran la información correspondiente a los tres viajes de Colón y que fuera útil para la realización de la tarea encomendada.

En esta tarea se busca la transferencia de los cinco pilates del PC como un principio o heurística e implica que el estudiante deba recordar, reconocer y ejecutar procedimientos realizados anteriormente, además de que lo obliga a un cambio de enfoque. Mientras el contexto físico y social, temporal y la modalidad permanecen inalterables, el contexto funcional va de la lectura de un texto a la realización de una animación.

La Tarea No 5 se aplicó tanto al Grupo Control como al Grupo de Tratamiento y consistió en plantear un reto matemático a los estudiantes para que lo resolvieran por medio de un programa en Scratch. El reto consistía en plantear el problema correctamente haciendo uso de los conceptos de programación trabajados en las sesiones anteriores y se considera una tarea de elevado nivel de dificultad porque posee elementos distractores que pudieron confundir a los participantes, además de que implica que sean capaces de recordar los procedimientos y conceptos vistos en las sesiones previas, y reconocer cuales podrían resultar útiles para resolver el problema en cuestión. Finalmente debían ejecutar el algoritmo que diera la respuesta. En el contexto temporal el cambio fue de un día para otro y en el contexto funcional se llevó al estudiante de un contexto académico del PC al contexto de una situación hipotética en el campo.

### **Instrumentos**

Para medir la VI1 se eligió el Computational Thinking Test [Prueba de Pensamiento Computacional] de Román-González (2015) porque es un instrumento autónomo que se ha revelado confiable para la evaluación del PC en Educación Primaria y Secundaria y puede ser administrado como pre-test y post-test en investigaciones que lo requieran. La prueba se centra en evaluar el desarrollo de los conceptos computacionales del marco tridimensional de PC propuesto por Brennan y Resnick (2012).

La prueba de pensamiento computacional (PPC) se diseñó con el objetivo de medir el nivel de desarrollo del pensamiento computacional a partir de la siguiente definición operacional de PC: capacidad de formular y resolver problemas basándose en los conceptos fundamentales de las ciencias computacionales y utilizando la sintaxis lógica de los lenguajes de programación y los conceptos de programación como las secuencias básicas, los bucles, la iteración, los condicionales, las funciones y las variables. Fue diseñada y destinada a estudiantes españoles de entre 12 y 13 años (denominados K7 y K8 por la nomenclatura estadounidense), bajo un formato de opción múltiple con 4 opciones de respuesta (solo una correcta), un total de 28 ítems y un tiempo estimado de finalización de 45 minutos (Román- González, 2015) (Ver Figura 3).

Figura 2. Especificación de los 28 ítems de la Prueba de PC

	Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida	Opción correcta
			Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)				
				Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros			
Item 1	Laberinto	Visual por flechas	Si	No	No	No	No	No	No	No	No	Secuenciación	B
Item 2	Laberinto	Visual por flechas	Si	No	No	No	No	No	No	No	No	Completamiento	C
Item 3	Laberinto	Visual por bloques	Si	No	No	No	No	No	No	No	No	Depuración	D
Item 4	Lienzo	Visual por bloques	Si	No	No	No	No	No	No	No	No	Secuenciación	D
Item 5	Laberinto	Visual por flechas	Si	Si	No	No	No	No	No	No	No	Secuenciación	C
Item 6	Laberinto	Visual por flechas	Si	Si	No	No	No	No	No	No	No	Completamiento	D
Item 7	Lienzo	Visual por bloques	Si	Si	No	No	No	No	No	No	No	Depuración	A
Item 8	Laberinto	Visual por bloques	Si	Si	No	No	No	No	No	No	Si	Secuenciación	B
Item 9	Laberinto	Visual por flechas	Si	No	Si	No	No	No	No	No	No	Secuenciación	D
Item 10	Laberinto	Visual por bloques	Si	No	Si	No	No	No	No	No	No	Completamiento	C
Item 11	Laberinto	Visual por flechas	Si	Si	Si	No	No	No	No	No	Si	Depuración	C
Item 12	Lienzo	Visual por bloques	Si	Si	Si	No	No	No	No	No	Si	Secuenciación	A
Item 13	Laberinto	Visual por flechas	Si	No	Si	Si	No	No	No	No	Si	Secuenciación	B
Item 14	Laberinto	Visual por bloques	Si	No	Si	Si	No	No	No	No	Si	Secuenciación	A
Item 15	Laberinto	Visual por flechas	Si	Si	Si	Si	No	No	No	No	Si	Completamiento	D
Item 16	Laberinto	Visual por bloques	Si	No	Si	Si	No	No	No	No	Si	Depuración	D
Item 17	Laberinto	Visual por bloques	Si	No	Si	No	Si	No	No	No	Si	Secuenciación	B
Item 18	Laberinto	Visual por bloques	Si	No	Si	No	Si	No	No	No	Si	Secuenciación	A
Item 19	Laberinto	Visual por bloques	Si	No	Si	No	Si	No	No	No	Si	Depuración	B
Item 20	Laberinto	Visual por bloques	Si	No	Si	No	Si	No	No	No	Si	Completamiento	C
Item 21	Laberinto	Visual por bloques	Si	Si	No	No	No	Si	No	No	Si	Secuenciación	A
Item 22	Laberinto	Visual por flechas	Si	Si	No	No	No	Si	No	No	Si	Secuenciación	B
Item 23	Laberinto	Visual por bloques	Si	No	No	Si	No	Si	No	No	Si	Completamiento	A
Item 24	Laberinto	Visual por bloques	Si	No	No	Si	No	Si	No	No	Si	Completamiento	C
Item 25	Lienzo	Visual por bloques	Si	Si	No	No	No	No	Si	No	Si	Secuenciación	B
Item 26	Lienzo	Visual por bloques	Si	Si	No	No	No	No	Si	No	Si	Completamiento	B
Item 27	Laberinto	Visual por bloques	Si	Si	No	No	No	No	Si	No	Si	Secuenciación	A
Item 28	Laberinto	Visual por bloques	Si	Si	No	No	No	No	Si	No	Si	Completamiento	C

Nota: Tabla de reactivos de la Prueba de PC elaborado por Román- González (2015).

La PPC se ha administrado a una muestra de 1.251 estudiantes españoles de quinto a décimo grado.

Para medir el NTPC se diseñó una actividad basada en los conceptos y prácticas del PC y de acuerdo con la TTL propuesta por Barnett y Ceci (2002), la cual permitió determinar el nivel de T con base en nueve elementos, tres de ellos relacionados con el contenido que se transfiere y seis con el contexto en el que ocurre la T (Ver Tablas 7). Entre las habilidades de PC que se pueden transferir se encuentran los procedimientos, las representaciones y los principios generales o heurística.

### Actividades para evaluar la transferencia del PC

Las actividades del ETPC y de la evaluación para determinar el NTPC logrado se diseñaron a partir de la TTL y se consideraron los elementos del PC: i) identificación del problema, ii) descomposición, iii) abstracción, iv) reconocimiento de patrones o generalización, v) algoritmo.

**Actividad 1.** Calcula el área de un triángulo equilátero sabiendo la medida de su base (Secuencias).

Desarrollo. Primero se me muestra al estudiante cómo se calcula el área de un rectángulo utilizando los bloques de Scratch y luego se le pide que aplique el conocimiento adquirido para resolver el problema planteado.

Tabla 18. Taxonomía de Transferencia Lejana aplicada a la primera actividad de T.

Niveles 1-3 de la Taxonomía de Barnett y Ceci ejemplificados		
<b>A. Contenido: Qué se transfiere</b>		
Habilidad aprendida	Procedimiento	Representación
Cambio de rendimiento	Velocidad	Precisión
Demanda de Memoria		Reconocimiento y ejecución
Niveles 4-9 de la Taxonomía de Barnett y Ceci ejemplificados		
<b>B. Contexto: Cuándo y dónde ocurre la T</b>		
Cercana		Lejana o remota
Campo de conocimiento	Matemáticas → Matemáticas	
Contexto físico	Entorno de programación de Scratch → Entorno de programación de Scratch	
Contexto temporal	Misma sesión	
Contexto funcional	Ambos claramente académicos	
Contexto social	Ambos individual	

---

Niveles 4-9 de la Taxonomía de Barnett y Ceci ejemplificados

---

**B. Contexto: Cuándo y dónde ocurre la T**

---

Modalidad	Ambos en Scratch, mismo formato
-----------	---------------------------------------

---

**Actividad 2.** Banquete en un restaurante (Decisiones). En un restaurante que organiza banquetes colectivos cobran 400 pesos por persona si el número de comensales es inferior a 50; 350 pesos por persona si está entre 50 y 100 y 300 pesos por persona si es superior a 100 invitados. Calcula el presupuesto total en función del número de personas.

Desarrollo. Primero se muestra a los estudiantes cómo calcular el costo de una comida bufete donde los adultos pagan 450 pesos por persona, los niños menores de 12 años 200 pesos por persona y los adultos mayores de 65 años 250 pesos por persona.

Tabla 19. Taxonomía de Transferencia Lejana aplicada a la segunda actividad de T.

Niveles 1-3 de la Taxonomía de Barnett y Ceci ejemplificados			
A. Contenido: Qué se transfiere			
Habilidad aprendida	Procedimiento	Representación	Principio o heurística
Cambio de rendimiento			Enfoque
Demanda de Memoria		Reconocimiento y ejecución	
Niveles 4-9 de la Taxonomía de Barnett y Ceci ejemplificados			
A. Contexto: Cuándo y dónde ocurre la T			
Cercana o remota			Lejana
Campo de conocimiento		Matemáticas → Contabilidad	
Contexto físico	Mismo entorno de programación de Scratch		
Contexto temporal	Misma sesión		
Contexto funcional	Ambos en el mismo contexto (Restaurante)		
Contexto social	Ambos individual		
Modalidad	Ambos en Scratch,		

---

Niveles 4-9 de la Taxonomía de Barnett y Ceci ejemplificados

---

A. Contexto: Cuándo y dónde ocurre la T

---

mismo

formato

---



**Actividad 3.** Actividad del artista en Code.org. (Ciclos). El estudiante debe realizar una obra de arte haciendo uso de figuras geométricas y ciclos de programación.

Desarrollo. Se inicia sesión en la plataforma Code.org y se muestra a los estudiantes como hacer “obras de arte” visual (animaciones) con una figura geométrica y utilizando ciclos de repetición.

Tabla 20. Taxonomía de Transferencia Lejana aplicada a la tercera actividad de T.

Niveles 1-3 de la Taxonomía de Barnett y Ceci ejemplificados	
A. Contenido: Qué se transfiere	
Habilidad aprendida	Principio o heurística
Cambio de rendimiento	Velocidad
Demanda de Memoria	Sólo ejecución
Niveles 4-9 de la Taxonomía de Barnett y Ceci ejemplificados	
A. Contexto: Cuándo y dónde ocurre la T	
Cercana o remota	Lejana
Campo de conocimiento	Matemáticas vs. artes
Contexto físico	Diferente entorno de programación visual Scratch vs. Code.org
Contexto temporal	Misma sesión
Contexto funcional	Académico vs. artístico
Contexto social	Ambos individual
Modalidad	Ambos en línea

**Actividad 4.** Viajes de Cristóbal Colón. Realiza un mapa interactivo en Scratch que muestre los tres viajes de expedición de Cristóbal Colón y que despliegue las fechas de cada uno.

Desarrollo. Se muestra los estudiantes las herramientas necesarias para realizar el proyecto y se ejemplifica como realizar el primer recorrido interactivo.

Tabla 21. Taxonomía de Transferencia Lejana aplicada a la cuarta actividad de T.

Niveles 1-3 de la Taxonomía de Barnett y Ceci ejemplificados			
<b>A. Contenido: Qué se transfiere</b>			
Habilidad aprendida			Principio o heurística
Cambio de rendimiento			Enfoque
Demanda de Memoria			Recordar, reconocer y ejecutar
Niveles 4-9 de la Taxonomía de Barnett y Ceci ejemplificados			
<b>A. Contexto: Cuándo y dónde ocurre la T</b>			
Cercana		Lejana o remota	
Campo de conocimiento			Programación vs. historia
Contexto físico	Mismo entorno de programación		
Contexto temporal		Día siguiente	
Contexto funcional			Lectura vs. realizar animación
Contexto social	Ambos individual		
Modalidad	Ambos en línea		

**Actividad 5.** Un granjero necesita tener 50 gallinas en su granja para vender los huevos y poder mantener la granja. Cada huevo lo vende en 20 centavos. Sin embargo, si tiene más de 50 gallinas, estas se comen el maíz que también necesita vender el granjero. Cada cuatro semanas cada gallina pone 3 huevos que tardan 2 semanas en convertirse en gallos y gallinas. De los huevos que nacen, son 1 macho y 2 hembras. El granjero trae zorros a la granja una semana al mes para controlar la población de gallinas y mantenerla en 50. Si cada zorro se come 10 gallinas a la semana, ¿Cuántos zorros necesita traer el granjero? Elabora un programa en Scratch que te diga cuantos zorros necesita el granjero dependiendo del número de gallinas que tenga.

Nota: para la solución de este problema se asume que las gallinas se reproducen todas al mismo tiempo.

Desarrollo. Únicamente se instruye a los estudiantes a realizar la actividad aplicando los conocimientos adquiridos previamente.

Tabla 22. Taxonomía de TL aplicada a la actividad final de T

Contenido	Habilidad aprendida	Pensamiento computacional
	Cambio de rendimiento	Enfoque
	Demanda de memoria	Recordar, reconocer y ejecutar
Contexto	Campo de conocimiento	Ciencias computacionales
	Contexto físico	En línea
	Contexto temporal	Día siguiente
	Contexto funcional	Académico → Situación hipotética en el campo
	Contexto social	Individual
	Modalidad	Ambos en línea

Figura 3. Matriz de evaluación de las actividades de T.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	
<b>2. Descompuso el problema en subproblemas</b>			
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
<b>4. Reutilizó elementos de su propio código</b>			
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			
Funcional (2)	No funcional (1)	No formuló algoritmo (0)	
<b>T O T A L</b>			

**Análisis estadísticos**

Se utilizó el programa estadístico SPSS para corroborar dichas hipótesis y se establecieron como variables de trabajo Grupo A y Grupo B. La variable Grupo A se conformó con los grupos establecidos a partir del examen de PC de Román-González (2015), el cual mostró una confiabilidad muy alta al obtener un .865 en la correlación de Pearson elaborada como parte del presente análisis (ver Correlaciones en anexos) lo cual permitió clasificar a los estudiantes en Grupo Bajo (con 9 sujetos) y Grupo Alto (con 11 sujetos) dependiendo de su nivel de PC; mientras que la variable Grupo B hace alusión tanto al Grupo Control (con 10 sujetos) que no recibieron entrenamiento en T, como al Grupo Tratamiento (con 10 sujetos) que si recibió el entrenamiento en T (ver Tabla 21).

Tabla 23. Conformación de los grupos para el Análisis de Covarianza (ANCOVA)

**Factores Inter sujetos**

	Etiqueta de valor	N
Grupo de participantes basado en los resultados del Examen de Román- González (Grupo A)	Grupo Bajo	9
	Grupo Alto	11
Grupo de participantes asignados al tratamiento (Grupo B)	Control	10
	Tratamiento	10

El primer análisis estadístico que se realizó para responder las preguntas planteadas en esta investigación fue de tipo descriptivo y permitió conocer la media, la desviación estándar y la correlación a través de los grupos de investigación. En seguida se realizaron los análisis necesarios para verificar que los datos cumplieran con los supuestos de homogeneidad de varianza y homogeneidad de correlación o esfericidad. Para el supuesto de homogeneidad de varianza se utilizó la prueba de Levene y para el supuesto de esfericidad se utilizó la prueba de Mauchly.

El resultado de los supuestos permitió proceder con un análisis ANCOVA factorial 2x2 entre grupos para determinar el impacto del ETPC en el Resultado Final y un ANCOVA con medidas repetidas para los datos aportados por el Grupo de Tratamiento en cada una de las actividades de T, utilizando Promedio General como la covariable.

Finalmente, se utilizaron métodos no paramétricos para corroborar los resultados del análisis paramétrico, como el de Mann-Whitney para comparar diferencias entre grupos usando la mediana como índice de comparación.

### Capítulo 4. Análisis de Resultados

En el capítulo anterior se presentaron los métodos estadísticos que se utilizaron para comprobar las hipótesis de la investigación y en este capítulo se reportan y analizan los resultados obtenidos de la intervención en PC y del ETPC, para lo cual es necesario retomar las hipótesis de la investigación:

- H1. Los niños de 12 y 13 años que presentan mayor NPC reportan un NTPC más alto.
- H2. Al contar con ETPC los niños de 12 y 13 años logran un NTPC más alto.
- H3. Los niños de 12 y 13 años que cuentan con un NPC alto y reciben ETPC logran un NTPC más alto debido a la interacción entre estos elementos.
- H4. El NPC de los niños de 12 y 13 años y el ETPC generan diferentes tipos y niveles de TPC.

Para determinar los estadísticos que se utilizarían para el análisis de los resultados se realizaron previamente las pruebas de supuestos incluyendo la prueba de homogeneidad de varianza de Levene. Dicho análisis produjo un resultado no favorable debido al resultado obtenido  $F(3, 16) = 4.07, p = .025$  por tratarse de una muestra pequeña. A pesar de que este supuesto no se cumplió, estudios de simulación tipo Monte Carlo apoyan la solidez del análisis de varianza y covarianza en este tipo de casos (Maxwell y Delaney, 2004). Debido a que las muestras para cada grupo son muy similares se determinó proseguir con el análisis y contestar las preguntas de este estudio.

#### Resultados obtenidos

El análisis ANCOVA permitió aceptar la hipótesis alternativa que afirma que los niños de 12 y 13 años que presentan mayor NPC reportan un NTPC más alto, pues el estadístico mostró significancia  $F(1,15) = 16.91, p < .01., \eta^2 = .53$ . Ver Tabla 24 y Figura 4.

Por el contrario, el mismo análisis obligó a rechazar la hipótesis alternativa de que al contar con ETPC los niños de 12 y 13 años logran un mayor NTPC, pues el estadístico asociado dio como resultado no significancia  $F(1,15) = .32, p > .05, \eta^2 = .32$ . Los niveles críticos indican que los participantes del Grupo de Tratamiento no obtuvieron resultados significativamente mejores que los del Grupo Control y por lo tanto el tratamiento en T no fue un factor decisivo en los resultados de la actividad final de T o Resultado Final (ver Tabla 24).

La hipótesis que expresa que los niños de 12 y 13 años que cuentan con un alto NPC y reciben ETPC logran mayor NTPC debido a la interacción entre estos tuvo que ser rechazada

porque mostró no significancia dado que el estadístico F correspondiente al efecto de la interacción entre Grupo A (NPC) y Grupo B (ETPC) lleva asociado un nivel crítico de .083. Esto indica que la interacción entre ellos no posee un efecto significativo en el Resultado Final para determinar el NTPC (ver Tabla 24).

Tabla 24. Prueba de efectos Inter Sujetos

Variable dependiente: Resultado Final

Origen	Tipo III de suma		Media	
	de cuadrados	Gl	cuadrática	F
Prom Gen.	2.50	1	2.50	1.06
Grupo_A	40.00	1	40.00	16.91**
Grupo_B	1.27	1	1.27	0.32
Grupo_A * Grupo_B	.196	1	.196	.083
Error	35.50	15	2.37	
<b>Total corregido</b>	<b>76.550</b>	<b>19</b>		

\*\*p < .01 \* p < .05 a. R al cuadrado = .504 (R al cuadrado ajustada = .53)

En la Tabla 25 aparecen representadas las medias para Resultado Final o NTPC calculadas en cada subgrupo resultante de combinar cada nivel de la variable Grupo A (Grupo Alto y Grupo Bajo) con cada nivel de la variable Grupo B (Grupo Tratamiento y Grupo Control). Esta interacción pone de manifiesto que las medias más altas fueron alcanzadas por el grupo surgido de la interacción entre Grupo Alto y Tratamiento con una media de 5.2, seguido de cerca por el Grupo Alto y Control con una media de 5.0, luego aparece el Grupo Bajo y Tratamiento con una media de 2.6, y finalmente el Grupo Bajo y Control con una media de 2.0 (ver Tabla 24 y Figura 4).

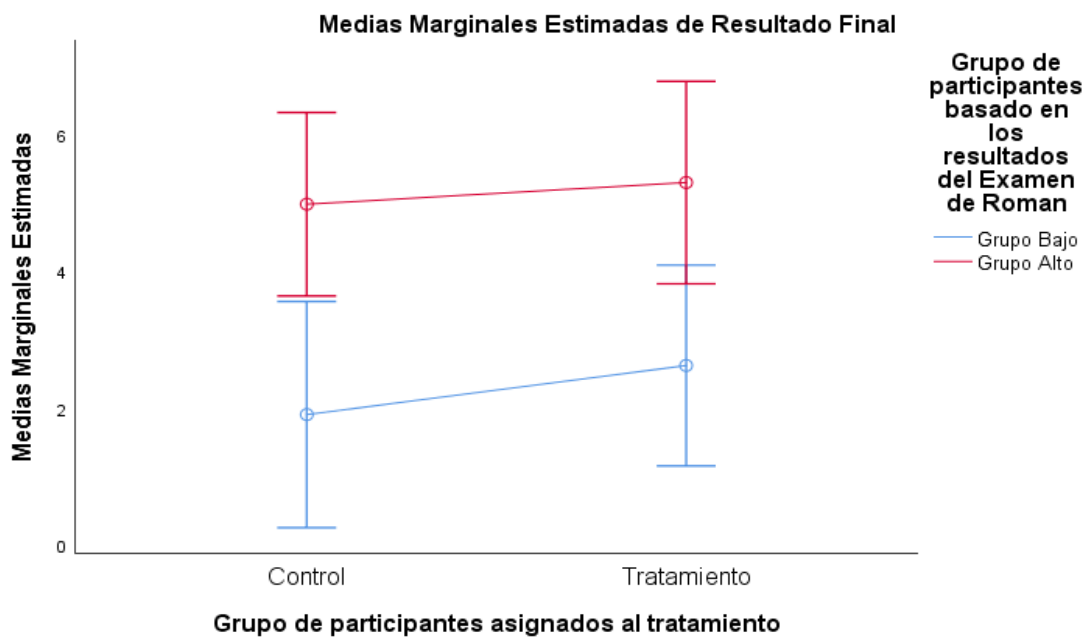
Con la incorporación de la covariable Promedio Académico General en el análisis de covarianza, se hizo visible que el Grupo Bajo se benefició más con el ETPC que el Grupo Alto al mostrar que la diferencia de medias en Resultado Final (NTPC) dentro del Grupo Bajo es mayor en el grupo que recibió el Tratamiento con un 2.6, contra el 2.0 del grupo control (ver Tabla 24 y Figura 4).

Tabla 25. Medias de los grupos conformados por la interacción de todos los niveles de la variable Grupo A con todos los niveles de la variable Grupo B

Variable dependiente: Resultado Final

Grupo de participantes basado en los resultados del Examen de PC	Grupo de participantes asignados al tratamiento	Media	Error estándar	Intervalo de confianza al 95%	
				Límite inferior	Límite superior
Grupo Bajo	Control	2.000	.771	.367	3.633
	Tratamiento	2.600	.689	1.139	4.061
Grupo Alto	Control	5.000	.629	3.666	6.334
	Tratamiento	5.200	.689	3.739	6.661

Figura 4. Variable Resultado Final ajustado por la covariable Promedio General



Las covariables que aparecen en el modelo se evalúan con los siguientes valores: Promedio General = 8.366666666666667

Barras de error: 95% CI



Centrando el análisis únicamente en el Grupo A se aprecia que los participantes con alto PC (Grupo Alto) logran una media de 5.1 que representa casi el doble de la media de 2.3 del Grupo con bajo (Grupo Bajo) (ver Tabla 26).

Tabla 26. Medias marginales estimadas para el grupo de participantes basado en los resultados del Examen de PC

Variable dependiente: Resultado Final

Grupo de participantes basado en los resultados del Examen de PC		Intervalo de confianza al 95%			
		Media	Error estándar	Límite inferior	Límite superior
Grupo Bajo		2.300	.517	1.204	3.396
Grupo Alto		5.100	.467	4.111	6.089

Sin embargo, al revisar la siguiente tabla podemos apreciar que el Grupo de Tratamiento apenas logró un 0.4 por arriba del 3.5 obtenido por el Grupo Control en la actividad final de T o Resultado Final, (ver Tabla 27).

Tabla 27. Medias marginales estimadas para el Grupo de participantes asignados al Tratamiento

Variable dependiente: Resultado Final

Grupo de participantes asignados al tratamiento		Intervalo de confianza al 95%			
		Media	Error estándar	Límite inferior	Límite superior
Control		3.500	.497	2.446	4.554
Tratamiento		3.900	.487	2.867	4.933

### **Desempeño del Grupo de Tratamiento en las actividades de T**

Para responder a la hipótesis cuatro se analizó el desempeño de los participantes del Grupo de Tratamiento a través de las cinco actividades de T. Con este propósito se utilizó el Modelo Lineal General para medidas repetidas.

Previo a realizar el análisis comparativo del desempeño del Grupo de Tratamiento a través de estas actividades, se condujo las pruebas de supuestos: homogeneidad de varianzas y homogeneidad de correlaciones (esfericidad de Mauchly). Los resultados de la prueba de Levene produjeron resultados favorables en 4 de las 5 medidas de interés y solo la actividad quinta presentó una leve violación a este supuesto. En relación con el supuesto acerca de la esfericidad, se reportó una leve violación,  $W(9) = .034$ , aprox.  $\chi^2 = 18.26$ ,  $p = .04$ . Esto indica que la varianza en los resultados obtenidos por el Grupo Tratamiento en Resultado Final fue mayor que en las otras actividades, sin embargo, no se consideró una violación grave al tratarse de una diferencia de .002 puntos para alcanzar el .05 (ver Tabla 28).

Tabla 28. Homogeneidad de varianza, supuestos de Levene

Prueba de igualdad de Levene de varianzas de error<sup>a</sup>

	F	df1	df2	Sig.
Primer resultado	.559	1	8	.476
Segundo resultado	.928	1	8	.364
Tercer resultado	2.647	1	8	.142
Cuarto resultado	1.906	1	8	.205
Resultado Final	5.422	1	8	.048

$p < 0.05$

Se observó también que no hubo ajuste estadístico de parte de la covarianza, sin embargo la interacción entre las cinco actividades y el grupo A fue significativa  $F(4, 4) = 6.48$ ,  $p = .049$  usando la prueba de la Traza de Hotelling (ver Tabla 29 y figura 5). Lo cual muestra que el Grupo A pudo resolver las tareas de T gracias a su NPC.

Tabla 29. Traza de Hotelling para la prueba multivariante

Medida: Actividades

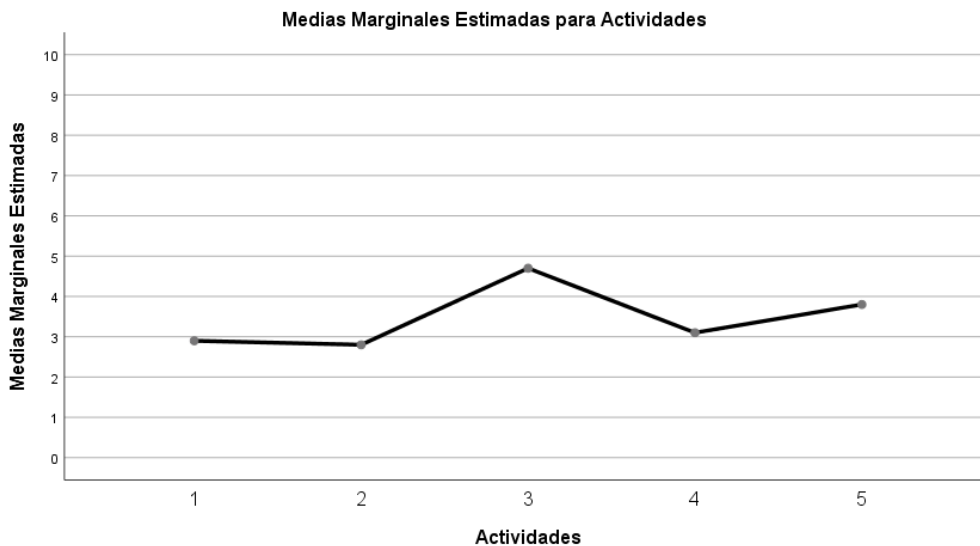
Efecto	Valor	F	Gl de hipótesis	gl de error	Sig.
actividades	Traza de Hotelling .228	.228	4.000	4.000	.910
actividades * X	Traza de Hotelling .245	.245	4.000	4.000	.899

actividades *	Traza de Hotelling	6.482	6.482	4.000	4.000	.049
Grupo_A						

Así, la Figura 5 muestra el desempeño del Grupo de Tratamiento a lo largo de las cinco actividades que se realizaron con la finalidad de entrenarlos para alcanzar la TL, donde las tres primeras actividades corresponden a la TC y las dos últimas a la TL. Actividades representa las cuatro actividades del entrenamiento en T y la Actividad Final de T.

Como se puede apreciar en el gráfico el comportamiento de los participantes fue estable en lo que respecta a las primeras dos actividades de TC con una media de 2.9 en la actividad uno y de 3.0 en la actividad dos (ver Tabla 28). En la actividad tres la media se eleva 1.7 puntos en comparación con la actividad anterior para dar un total de 4.7. Sin embargo, al realizar la actividad cuatro, la primera de TL, la media baja a 3.1 y recupera .7 en la actividad final para dar una media de 3.8.

Figura 5. Medias marginales estimadas para las cinco actividades de T



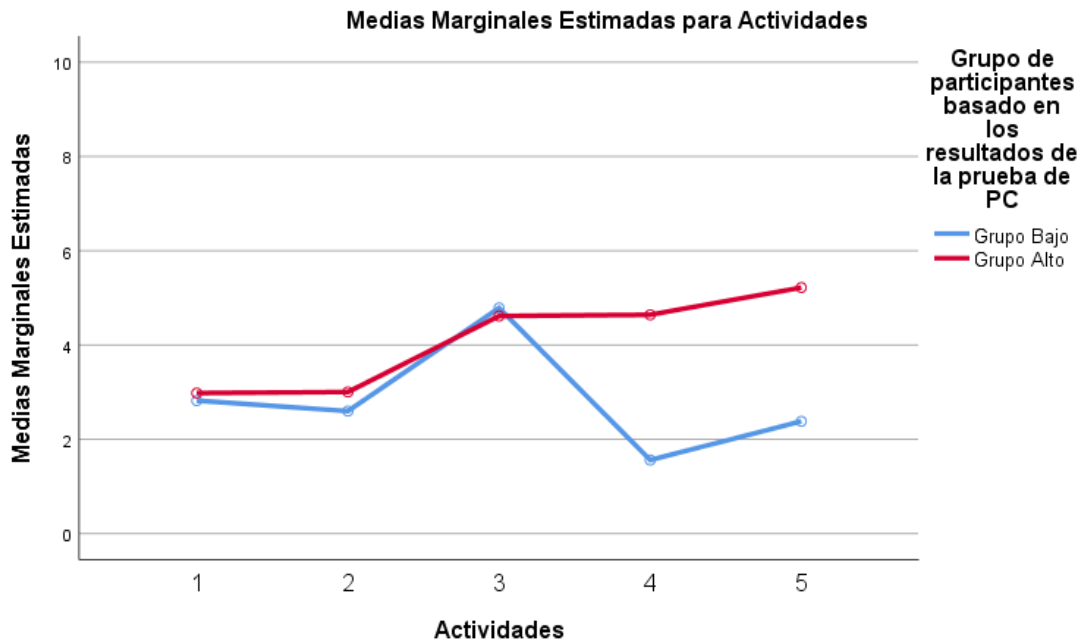
Las covariables que aparecen en el modelo son evaluadas bajo los siguientes valores: Promedio General = 8.188888888888890

Al comparar el desempeño dentro del Grupo A en el tratamiento se observa que mientras el Grupo Alto mantiene una tendencia constante al alza a través de las cinco actividades de transferencia, el Grupo Bajo tiene altas y bajas con especial incremento en la actividad tres de

TC y el mayor descenso en la actividad cuatro, que es la primera de TL para finalmente recuperarse un poco en actividad final de T.

En la Figura 6 se puede ver claramente la diferencia de aprovechamiento entre el Grupo Alto y el Grupo Bajo con respecto a NTPC que logró cada uno (ver Figura 6).

Figura 6. Desempeño del Grupo de Tratamiento en las actividades de T



Las covariables que aparecen en el modelo son evaluadas por los siguientes valores: Promedio General = 8.188888888888890

La siguiente tabla muestra las estadísticas descriptivas para estos dos grupos a través de cada una de las actividades (ver Tabla 30).

Tabla 30. Medias de los resultados en las actividades de T para Grupo Alto y Grupo Bajo

	Grupo de participantes basado en los resultados del Examen de PC	Media	Desviación Estándar	N
Primer resultado	Grupo Bajo	2.80	2.04	5
	Grupo Alto	3.00	1.22	5
	Total	2.90	1.59	10
Segundo resultado	Grupo Bajo	2.60	1.14	5
	Grupo Alto	3.00	1.41	5
	Total	2.80	1.22	10
Tercer resultado	Grupo Bajo	4.80	.83	5

	Grupo Alto	4.60	1.34	5
	Total	4.70	1.05	10
Cuarto resultado	Grupo Bajo	1.60	1.14	5
	Grupo Alto	4.60	3.36	5
	Total	3.10	2.84	10
Resultado Final	Grupo Bajo	2.40	1.14	5
	Grupo Alto	5.20	2.16	5
	Total	3.80	2.20	10

En la prueba de Huynh-Feldt el estadístico F correspondiente al efecto que tiene el NPC (Grupo A) en las actividades de T, tiene un valor de  $F(3,1, 21,7) = 4.54$ ,  $p = .012$ , lo que significa que el NPC es determinante del NTPC que logran los niños de 12 y 13 años que participaron en la investigación (ver Tabla 31).

Tabla 31. Pruebas de efectos dentro de sujetos usando la prueba de Huynh-Feldt

Medida: Actividades

Origen	Tipo III de suma de cuadrados	gl	Media cuadrática	F	Sig.
actividades	2.011	3.104	.648	.376	.778
actividades * X	2.150	3.104	.693	.402	.759
actividades * Grupo_A	24.269	3.104	7.820	4.536	.012
Error(actividades)	37.450	21.725	1.724		

### Corroboración de resultados paramétricos

Dado que en el estudio se violaron algunos de los supuestos de homogeneidad de varianza, se decidió realizar una comprobación de los resultados a través de análisis no paramétricos. A continuación se exponen los resultados.

La variable nula para el primer factor indica que el NPC del estudiante no afecta el NTPC (Resultado Final). De acuerdo con la Prueba U de Mann-Whitney para muestras independientes se rechaza la hipótesis nula por lo que se puede asegurar que el NPC afecta positivamente el NTPC (ver Tabla 32).

Mientras que el mismo análisis para la hipótesis del factor dos que afirma que el ETPC no impacta en el NTPC, la prueba U de Mann-Whitney indica que se debe aceptar la hipótesis nula, es decir, concluye que efectivamente el ETPC no impactó en el NTPC de los participantes (ver Tabla 33).

Tabla 32. Prueba U de Mann-Whitney que rechaza la hipótesis nula del primer factor

Hipótesis nula	Prueba	Sig.	Decisión
La distribución de resultados para la Actividad 5 (Resultado Final) es la misma entre las categorías de Grupo A (NPC)	Prueba U de Mann-Whitney para muestras independientes.	.000*	Rechace la hipótesis nula.

Se muestran significaciones asintóticas. El nivel de significación es .05.

\*Se muestra la significación exacta para esta prueba.

Tabla 33. Prueba de U de Mann-Whitney que conserva la hipótesis del segundo factor

Hipótesis nula	Prueba	Sig.	Decisión
La distribución de resultados para la Actividad 5 (Resultado Final) es la misma entre las categorías de Grupo B (ETPC)	Prueba U de Mann-Whitney para muestras independientes.	1.000*	Conserve la hipótesis nula.

Se muestran significaciones asintóticas. El nivel de significación es .05.

\*Se muestra la significación exacta para esta prueba.

### Resumen de los resultados observados

En este capítulo se mostraron los resultados de los análisis estadísticos realizados, descriptivos, de covarianza y de medidas repetidas, así como de los supuestos correspondientes

de homogeneidad de varianza y de esfericidad. Dichos estadísticos dieron como resultado significancia en la pregunta sobre el impacto que tiene el NPC en el NTPC en niños de 12 y 13 años, mientras que mostró no significancia en las preguntas sobre el efecto que tiene el ETPC en el NTPC que logran los niños de 12 y 13 años, y la que se plantea qué la interacción del NPC y el ETPC generan un alto NTPC en los niños de 12 y 13 años.

Mientras que con el estudio de medidas repetidas se pudo observar el comportamiento del Grupo A en las cinco actividades del ETPC y se concluyó que el NPC es el factor principal que permite lograr la TL a los participantes de este estudio.

En el siguiente capítulo se discutirán los resultados aquí expuestos.

## Capítulo 5. Discusión y Conclusiones

### Discusión

En el capítulo anterior se mostraron los resultados del experimento realizado para probar las hipótesis del presente trabajo de investigación y a continuación se discute a detalle el significado de los resultados encontrados.

En primer lugar, se explica por qué el NPC de los participantes se convierte en el factor principal para determinar el NTPC que alcanzan estos en la actividad final de Transferencia. En seguida se aborda la interacción entre el NPC y el ETPC y su impacto en el NTPC nombrado Resultado Final en el análisis estadístico. Luego se explican los resultados del Grupo de Tratamiento en cada una de las cinco actividades de Transferencia.

Los principales teóricos de la Transferencia señalan que la dificultad para probar la TPC radica en el diseño metodológico de los experimentos que se han diseñado para ese fin, pues muchos de ellos apuestan por un diseño cerrado enfocado en probar sólo un tipo de Transferencia y bajo ciertos parámetros, cuando lo necesario es un diseño de amplio espectro que permita percibir distintos tipos y niveles de Transferencia.

Al realizar el análisis del NTPC de acuerdo con la Taxonomía de Transferencia Lejana de Barnett y Ceci (2002) se pudo constatar que la TPC se logró por parte de todos los niños de 11 y 12 años que participaron en la investigación, aunque en distintos niveles y en distintos conceptos de programación, dependiendo del NPC que desarrollaron previamente en la intervención en PC. Este fenómeno es explicado por Pea y Kurland (1984) cuando señalan que diferentes niveles de competencia en programación permiten la Transferencia de diferentes conceptos y habilidades.

Los análisis estadísticos efectuados en el capítulo anterior permitieron observar que el dominio temático, en este caso el NPC, es el factor primordial para lograr la Transferencia tal como lo afirmaron Brown (1989) y Haskell (2000) al señalar que el desconocimiento de los estudiantes sobre los distintos temas planteados puede constituir la primera barrera para la Transferencia.

En este caso se comprobó que la implementación de la metodología de resolución de problemas basada en el PC, en particular lo que corresponde a la descomposición del problema, requiere que los niños tengan conocimiento sobre la temática del problema en cuestión para que puedan descomponerlo efectivamente en subproblemas, dado que les resulta imposible



descomponer en partes un problema con cuyo contexto y/o contenido disciplinar no están familiarizados.

Ortega-Ruipérez y Brouard (2018) afirman que es necesario lograr un dominio amplio de los procesos cognitivos implicados en el PC para facilitar la resolución de problemas complejos ajenos al contexto de programación o de lo contrario el PC se limitará a facilitar la resolución de problemas relacionados con la computación evidenciando una falta de T de PC.

Aunque la similitud entre los elementos de la actividad inicial y la actividad de Transferencia son esenciales para la correspondencia entre ambas situaciones y reconocer los mecanismos de resolución que necesitan activarse, esta similitud puede ser superficial o profunda, considerando similitud superficial las mismas fórmulas y mismos planteamientos, y similitud profunda cuando comparten los mismos principios estructurales.

Considerando esto, en cada sesión se ofreció a los estudiantes un ejemplo de la solución de un problema, haciendo corresponder todas sus partes con el formato presentado para el desarrollo del PC y se esperaba que por analogía lograran acomodar los elementos de un problema similar en el mismo formato, pues una forma de facilitar la generalización es la comparación de dos ejemplos, tratando de encontrar su estructura común (Anderson y Schunn, 2000).

Sin embargo, los resultados de este trabajo mostraron que los niños tuvieron problema no solo para aplicar los principios del PC que les fueron enseñados durante la intervención en este, sino que además, no fueron capaces de crear rutas mentales por analogía para resolver los problemas planteados ni de verbalizar el proceso de pensamiento que seguían al resolver un problema, a pesar de haber sido entrenados para ello.

En el ETPC se trabajó de manera especial el desarrollo de la metacognición atendiendo a las recomendaciones de los investigadores que la desatacan como un factor indispensable para la Transferencia, y aunque si se pudo apreciar que los niños se familiarizaron con los elementos que conforman el PC no les resultó fácil aplicarlos en problemas de distinta índole presumiblemente por que el proceso de resolución de problemas del PC no llegó a la automatización – generalización (Salomon y Perkins, 1987).

Kuhn y colaboradores (1992) encontraron que el desarrollo de la metacognición es muy importante para establecer el razonamiento causal entre dos dominios, lo cual es imprescindible para alcanzar la T porque finalmente se transfiere a partir de las estrategias y representaciones

creadas por el sujeto (Singley & Anderson, 1989). Por lo mismo, autores como Albin & Horner, 1988; Engelmann & Carnine, 1982; y Stokes & Baer, 1977, se han manifestado a favor de agregar capacitación en múltiples ejemplos, reducir la discriminación entre las pruebas y los entornos de Transferencia, y recompensar la respuesta generalizada.

Aunque estas recomendaciones se consideraron en el diseño de las tareas de Transferencia, para algunos de los participantes resultó complejo descomponer los problemas más novedosos en subproblemas (segundo pilar del PC) y a pesar de que algunos demostraron identificar en la práctica los primeros elementos del PC no pudieron explicar si el problema actual les recordaba otro realizado durante la intervención en PC o el ETPC.

Así, mientras los elementos básicos del PC como la identificación del problema y la descomposición fueron aquellos en los que la mayoría de los niños logró la TC, entre más complejos se volvían tanto el concepto de programación estudiado como el contexto del problema planteado, más niños fallaban en lograr tanto la TL como la TC.

Los participantes también presentaron dificultades para entender cómo plasmar su proceso cognitivo en el formato que les fue proporcionado, lo que coincide con los resultados de Molina (2020) cuando indica que sus estudiantes no reconocen la importancia de reflexionar sobre las propias estrategias de resolución de problemas y plasmar en el proceso de resolución el procedimiento correspondiente a la elaboración y ejecución del plan.

La Transferencia lograda a partir de la automatización es considerada por Perkins (2009) como una Transferencia de Baja Trayectoria (TBT), contraria a la Transferencia de Alta Trayectoria (TAT) a través de cinco actividades de evaluación que implica la abstracción consciente y deliberada de un principio general, heurística o procedimiento de un contexto para aplicarlo en otro, pero Anderson y Schunn (2000) afirman que la generalización dista de ser un proceso automático puesto que demanda el establecimiento de la correspondencia entre un principio y un problema dado a partir de encontrar la estructura común a ambos.

A partir de los resultados de este trabajo de investigación se puede comprobar que tanto la TBT como la TAT se integran en la Transferencia Vertical (TV) (Haskell, 2000) dado que la generalización es un escalón para lograr la TL y al mismo tiempo, se requiere un profundo conocimiento y estudio deliberado y consciente de las estructuras similares entre la heurística y las diversas situaciones problemáticas para lograr la automatización.

De este modo, el desempeño del Grupo de Tratamiento en las cinco actividades de Transferencia se puede explicar a partir de la teoría de la transferencia basada en los elementos

comunes o idénticos (Thorndike y Woodworth, 1901), de acuerdo con la cual la Transferencia se presenta de manera limitada a los elementos idénticos compartidos en las situaciones de aprendizaje, lo cual no es limitante para lograr la TL (Barnett & Ceci, 2002) o la Transferencia Inventiva (Butterfield & Nelson, 1991) que implica que los estudiantes desarrollen una solución y/o procedimiento nuevo a partir de las similitudes y diferencias críticas encontradas entre la situación desafío actual y las situaciones de aprendizaje previas.

Tabla 34. Medias de aprovechamiento en las actividades de Transferencia

Actividad	Media	Desviación Estándar	N
Primera	2.9	1.59	10
Segunda	2.8	1.22	10
Tercera	4.7	1.05	10
Cuarta	3.1	2.84	10
Quinta	3.8	2.20	10

En el caso de la actividad No 1 de Transferencia, podemos asegurar que la TC se logró sólo en los primeros dos pilares del PC, identificación del problema y descomposición, porque los niños no fueron capaces de discernir la información relevante para resolver el problema (abstracción) y la mayoría de ellos no pudo reconocer los elementos del código creado por la maestra que pudieran ser reutilizados para resolver la tarea (reconocimiento de patrones) y no lograron recrear el algoritmo para calcular el área del triángulo (creación del algoritmo). Solo el diez por ciento del Grupo de Tratamiento logró la TC en Abstracción (ver Tabla 34).

En la actividad No 2, la media grupal fue 0.1 más baja que en la actividad número uno. Nuevamente la TC se logró solo en los dos primeros pilares del PC (Identificación del problema y descomposición) y solo el diez por ciento de los participantes se aproximó a lograr la TC en el tercer pilar del PC (Abstracción) (ver Tabla 34).

En la actividad No 3, a pesar de ser una actividad de transición, los resultados obtenidos fueron favorables, pues es la única actividad en la que la media grupal ronda los cinco puntos y al analizar por separado el comportamiento del Grupo Alto y el Grupo Bajo en dicha actividad se constata que ambos grupos obtuvieron buen puntaje en ella y un treinta por ciento de los niños alcanzó la TC en los tres primeros pilares del PC (Identificación del problema, Descomposición del problema y Abstracción). Esto puede explicarse por la acumulación de la fuerza del aprendizaje debido a la práctica (Singley & Anderson, 1989) (ver Tabla 34).

Este resultado se debe posiblemente a que esta actividad a pesar de moverse hacia las artes, un campo de conocimiento aparentemente muy alejado de la geometría y la programación, y del cambio en el contexto físico al pasar de *Scratch* a *Code.org*, en realidad recupera los principios aplicados en las actividades uno y dos. Las habilidades de las personas para recuperar conocimiento relevante pueden variar de ser “con esfuerzo” a “relativamente sin esfuerzo” (fluido) a “automático” (Schneider y Shiffrin, 1977), por lo tanto la recuperación automática y fluida del conocimiento son características importantes de la experiencia.

En la actividad No. 4 el reto planteado a los niños ocasionó que la media de los resultados cayera a los tres puntos. Sin embargo, al analizar por separado los resultados del Grupo Alto y el Grupo Bajo se aprecia que el Grupo Alto mantuvo una media de 4.6 mientras que el Grupo Bajo cayó hasta el 1.6. Y solo el diez por ciento del Grupo de Tratamiento alcanzó la TL en todos los pilares del PC. El diez por ciento de los participantes no logró identificar el problema, el cuarenta por ciento apenas logró identificar el problema, el treinta por ciento logró descomponer el problema y el diez por ciento logró abstraer la información necesaria para resolver el problema (ver Tabla 34).

El resultado de la actividad No. 5 de Transferencia fue ligeramente mejor que el de la Tarea No 4, sin embargo, en el análisis diferenciado entre Grupo Alto y Grupo Bajo se hace visible que esta media relativamente alta se mantiene por la influencia del Grupo Alto. Así la media del Grupo Alto es de 5.2 mientras que la media del Grupo Bajo es de 2.4 (ver Tabla 34).

De manera global, considerando tanto al grupo control como al grupo de tratamiento, el 20% de los participantes logró la TL en cuatro de los cinco pilares del PC evaluados, 5% logró la TL en los primeros tres pilares del PC, el 45% logró la TL en los dos primeros pilares, y el 30% logró la TL solo en el primer pilar.

La covariable promedio general permitió demostrar que las diferencias de desarrollo de PC logrado por los niños tuvieron su origen en diferencias individuales anteriores al tratamiento y no vinculadas con su desempeño escolar, pues alumnos con buenas calificaciones en materias como matemáticas y ciencias obtuvieron bajos niveles de PC, mientras que alumnos con altas calificaciones en diversas materias no relacionadas típicamente con las CC obtuvieron tanto alto como bajo nivel de PC.

Esto indica que algunos elementos del PC se han desarrollado en los niños por causa de diversas situaciones personales a las que han estado expuestos y que resultan no detectables en el presente estudio.

Tanto el enfoque empirista como el enfoque racionalista (Greeno et al., 1993) fueron utilizados por el Grupo de Tratamiento para la resolución de las actividades de Transferencia. Mientras las actividades una, dos y tres se podían resolver a partir de los elementos superpuestos entre las tres utilizando un enfoque empirista, las actividades cuatro y cinco demandan un enfoque racionalista para su resolución porque implican el reconocimiento de una estructura profunda o representación compartida entre la situación original y la situación de Transferencia.

Butterfield y Nelson (1991) afirman que es más probable lograr la Transferencia cuando se agregan métodos diseñados específicamente para lograrla a los métodos que se utilizan para producir aprendizaje, por ello durante el entrenamiento en PC se presentó y trató a éste como un proceso de cinco pasos para la resolución de problemas y cada situación de aprendizaje de PC se abordó a partir de estos cinco momentos de modo que se lograra cierta automatización del pensamiento en los estudiantes. A esta automatización debe seguir el desarrollo de la metacognición del estudiante porque la resolución y la planificación de problemas requiere altos niveles de ésta (Pea y Kurland, 1984).

Desde la perspectiva de la cognición, aprender contenido no es lo mismo que aprender a transferir contenido (Bransford et al., 1999; van Merriënboer et al., 2002), por eso, desde el inicio del entrenamiento en PC los niños ya estaban siendo entrenados para transferir. Sin embargo, aquellos que no lograron interiorizar los elementos ni los conceptos del PC no pudieron reconocer las similitudes entre los diferentes problemas planteados ni trazar un plan para resolver la tarea en cuestión, de modo que solo lograron la TC en los elementos más básicos del PC.

### **Conclusiones**

El diseño experimental utilizado en este trabajo plantea una aproximación diferente al estudio de la TPC porque permite identificar los elementos del PC que representan mayor reto para su comprensión y apropiación por parte de los estudiantes, siendo estos principios la descomposición del problema y el diseño del algoritmo, y permitió comprobar que la Transferencia es posible gracias a la existencia de elementos comunes entre la tarea inicial y la de Transferencia, pudiendo ser estas similitudes generales o específicas, y la logran con más facilidad las personas consideradas expertas en el campo de conocimiento que se desea transferir.

También permitió identificar el nivel de transferencia que lograron los niños en cada uno de los elementos del PC a través de las cinco actividades de Transferencia, donde los niveles más

altos se lograron en torno a la identificación del problema y a la abstracción, y los niveles más bajos en Transferencia corresponden al principio de descomposición y al diseño del algoritmo.

La manifestación de la Transferencia se logra gracias a la interacción positiva de dos elementos clave: el conocimiento disciplinar y el entrenamiento en Transferenciar. Mientras el estudiante no domine el contenido (declarativo, procedimental, etc.,) que se desea que transfiera, la Transferencia no tendrá lugar. Por lo tanto, para lograr la TPC se debe empezar por diseñar cursos de PC que logren que sus conceptos y pasos sean interiorizados y automatizados por los niños.

El NTPC que logran los estudiantes depende principalmente de los siguientes factores: i) de que el reto cognitivo planteado por la situación de aprendizaje sea adecuado a la edad del participante, ii) de que el estudiante cuente con la cantidad suficiente de práctica en la resolución de problemas, iii) de que se le provea una gran cantidad de ejemplos de problemas resueltos acompañados con la explicación explícita de cómo se resolvieron esos problemas, iv) de que se aborden los problemas de manera trans contextualizada, v) de que se ejercite la metacognición sobre el proceso personal de resolución de problemas una vez que se internalicen los elementos del PC, vi) de que el estudiante sea capaz de establecer la correspondencia analógica entre la estructura de un problema o situación aprendida previamente y una nueva, y vi) del dominio que el estudiante posea sobre el contenido que desea transferir.

Los niños con alto NPC que dominaban elementos más complejos de éste como la abstracción, el reconocimiento de patrones y el diseño de un algoritmo (pensamiento secuencial), pudieron lograr la transferencia de estos.

A pesar de que se siguió la receta de la Transferencia analógica para la resolución de problemas, por el resultado obtenido en la primera actividad que requería hacer una sustitución simple de fórmula y de datos utilizando la misma estructura y contexto de la actividad previa, se entiende que los niños necesitan más entrenamiento en TC con demostraciones de cómo resolver el mismo tipo de problema incluso en el mismo contexto aunque solo se requiera sustituir fórmulas y datos, y más horas de práctica antes de plantearles una situación de transferencia que implique un cambio de contexto o un incremento de demanda cognitiva.

Entre mayor nivel de PC poseen los participantes, más fácil les resulta aplicarlo en nuevas situaciones y para la resolución de nuevos problemas, pues los participantes con bajo nivel de PC carecen de elementos para entender los problemas e identificar las estrategias que podrían implementar para su resolución.

Durante el tiempo que duró la investigación se logró identificar el caso de dos jovencitos con habilidad nata para el PC que no habían programado ni recibido instrucción previa sobre programación y desarrollaban los ejercicios con solvencia y creatividad. Por eso, la socialización del conocimiento adquirido y de los procesos realizados son tan importantes como la misma resolución de problemas dado que permite fijar el conocimiento en el estudiante que la ejecuta y que los estudiantes que experimentan dificultades puedan conocer, entender y apropiarse de las herramientas que utilizan los compañeros más adelantados para resolver las situaciones planteadas.

### **Recomendaciones**

Para futuras investigaciones sobre la Transferencia del Pensamiento Computacional la recomendación principal es que se trabaje por separado con cada uno de los elementos de este y se busque su aplicación en un amplio abanico de situaciones cada vez más alejadas del mundo de las ciencias computacionales para medir el impacto que tiene en las capacidades cognitivas de los niños.

También se recomienda que los entrenamientos para desarrollar tanto el Pensamiento Computacional como la Transferencia tengan una duración mínima de 6 meses con dos sesiones de 60 min por semana y acompañar el proceso con varias sesiones de evaluación y retroalimentación. También es importante dar espacio suficiente a los niños para que expresen tanto sus dudas como los razonamientos que les ayudaron a resolver ciertos problemas.

El ETPC es muy importante para asegurar la TPC, por lo que las intervenciones en PC que se ofrezcan en un futuro deberán incorporar la metodología para el desarrollo de la metacognición y la auto explicación tanto del problema como de los principios de programación y de los elementos del PC.

Desarrollar programas de PC propedéuticos o de nivelación para los alumnos en evidente desventaja en el manejo de herramientas digitales antes de integrarlos a los cursos de PC, con la intención de incrementar su nivel de logro y de satisfacción, de lo contrario podrían sentirse frustrados al no tener los conocimientos básicos para operar un equipo de cómputo.

### **Limitaciones**

La principal limitación del estudio fue el tamaño de la muestra, pues debido al confinamiento ordenado por las autoridades federales como medida de contención de la pandemia del COVID-19, y la suspensión de la actividad en las escuelas fue imposible realizar la

intervención de manera presencial en una institución educativa, lo que hubiera permitido trabajar con una muestra más grande.

La segunda limitación del estudio fue la baja conectividad a internet y el poco equipamiento tecnológico con el que cuenta la gran mayoría de las familias en Chihuahua, pues al menos el 50% de los participantes en este estudio realizó el taller a través de un dispositivo móvil y no en una computadora, lo que implica pérdida de atención y dificultad para seguir detalladamente la explicación de la maestra al tener que cambiar constantemente entre pantallas de la plataforma de instrucción (*Meet*) a la del entorno de programación de *Scratch*.

La tercera limitante fue que los niños no están habituados a recibir instrucciones por medios digitales y se requiere más tiempo de explicación para que asimilen los conceptos y las actividades, por lo que las sesiones pensadas originalmente de 60 min se tuvieron que adaptar para hacerlas de 90 min.



### Referencias

- Arastoopour Irgens, G., Dabholkar, S., Bain, C., Woods, P., Hall, K., Swanson, H., Horn, M., & Wilensky, U. (2020). Modeling and Measuring High School Students' Computational Thinking Practices in Science. *Journal of Science Education and Technology*, 29(1), 137-161. <https://doi.org/10.1007/s10956-020-09811-1>
- AERA, APA and NCME, (2014). Standards for educational and psychological testing, Washington: American Educational Research Association.  
<https://www.apa.org/science/programs/testing/standards>
- Adey, P.S. & Shayer, M. (1993). An exploration of long- term far-transfer effects following an extended intervention program in the high school science curriculum. *Cognition and Instruction*, 11(1), pp. 1-29. <https://bit.ly/3eofWbp>
- Agalianos, A., Noss, R., & Whitty, G. (2001). Logo in mainstream schools: The struggle over the soul of educational innovation. *British Journal of Sociology of Education*, 22(4), 479–500. <https://bit.ly/32DMGIi>
- Aho, A. V. (2011). Ubiquity symposium: Computation and Computational Thinking. *Ubiquity*, 2011(January), 1. <https://doi.org/10.1145/1922681.1922682>
- Albanese, M., & Mitchell, S. (1993). Problem-based learning: A review of literature on its outcomes and implementation issues. *Academic Medicine*, 68(1), 52–81. <https://bit.ly/3xe2mQy>
- Albin, R. W., & Horner, R. H. (1988). *Generalization with precision*. In R. H. Horner, G. Dunlap, & R. L. Koegel (Eds.), *Generalization and maintenance: Life-style changes in applied settings* (pp. 99-120). Brookes. <https://psycnet.apa.org/record/1988-97221-005>
- Álvarez, J. E., & Romero, A. (2015). La empleabilidad de graduados universitarios en el contexto latinoamericano. Realidades de Uniandes, Ecuador. *Atenas* 4(32), 1-15  
<https://www.redalyc.org/articulo.oa?id=478047208001>
- Amestoy, M. (2002). La investigación sobre el desarrollo y la enseñanza de las habilidades de pensamiento. *Revista Electrónica de Investigación Educativa* 4(1).<https://redie.uabc.mx/redie/article/view/55>
- Arboleda, J. (2013). Hacia un nuevo concepto de pensamiento y comprensión. *Boletín Virtual Redipe*, 824. <https://bit.ly/32yXW8P>

- Arfé, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers and Education*, *148*, 103807. <https://doi.org/10.1016/j.compedu.2020.103807>
- Arias, F. (2006). El proyecto de Investigación: Introducción a la metodología científica. Episteme. <https://bit.ly/3NzQb8t>
- Ausubel, D. P. & Robinson, F. G. (1969). *School learning: an introduction to educational psychology*. Holt, Rinehart & Winston.
- Badilla E. & Chacón A. (2004). Construcciónismo: Objetos para pensar, entidades públicas y micromundos. *Revista Electrónica "Actualidades Investigativas en Educación"*, *4*(1), 0. <https://www.redalyc.org/articulo.oa?id=44740104>
- Balanskat, A., & Engelhardt, K. (2015). *Computing our future - computer programming and coding*. European Schoolnet. <https://bit.ly/32wWLH2>
- Barefoot. (2014). *Computing at School (CAS) - Computational Thinking*. <https://www.barefootcomputing.org/>
- Barnett, S. M., & Ceci, S. J. (2002). When and where do we apply what we learn?: A taxonomy for far transfer. *Psychological Bulletin*, *128*(4), 612–637. <https://doi.org/10.1037/0033-2909.128.4.612>
- Barr, D., Harrison, J., & Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. *Learning and Leading with Technology*, *38*(6), 20–23. <https://bit.ly/2QJMbd4>
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48–54. <https://doi.org/10.1145/1929887.1929905>
- Binet, A. (1899). *The Psychology of Reasoning. Based on Experimental Research in Hypnotism*. Open Court. <https://www.apa.org/pubs/books/4320324>
- Bloom, J. W. (2010). Systems thinking, pattern thinking, and adductive thinking as the key elements of complex learning. Annual Meeting of the American Educational Research Association, Denver, Colorado, USA. <http://bit.ly/SystemsThinkng>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice*. EUR 28295 EN. <https://bit.ly/3xdockj>

- Bohl, Th. (2000). Unterrichtsmethoden in der Realschule. Eine empirische Untersuchung zum Gebrauch ausgewählter Unterrichtsmethoden an staatlichen Realschulen in Baden-Württemberg. Ein Beitrag zur deskriptiven Unterrichtsforschung. [Métodos de enseñanza en la escuela secundaria. Un estudio empírico del uso de métodos de enseñanza seleccionados en las escuelas secundarias estatales de Baden-Württemberg. Una contribución a la investigación docente descriptiva. Bad Heilbrunn/Obb. <https://bit.ly/39lHf7P>
- Brackmann, C., Barone, D., Casali, A., Boucinha, R., & Muñoz-Hernandez, S. (2016). *Computational thinking: Panorama of the Americas*. International Symposium on Computers in Education, SIIE 2016: Learning Analytics Technologies, 1–6. <https://doi.org/10.1109/SIIE.2016.7751839>
- Bransford, J. D., Brown, A., & Cocking, R. (1999). *How People Learn: Mind, Brain, Experience, and School*. National Research Council. <https://bit.ly/2Qdwj2Q>
- Bransford, J. D., & Schwartz, D. L. (1999). Rethinking transfer: A simple proposal with multiple implications. *Review of Research in Education*, 24, 61–100. <https://doi.org/10.3102/0091732x024001061>
- Brennan, K., & Resnick, M. (2012). *Nuevos marcos de referencia para estudiar y evaluar el desarrollo del pensamiento computacional*. Documento presentado en el encuentro anual de la American Educational Research Association (AERA) 2012, 1–28. <https://bit.ly/2PeE2x2>
- Brown, A. L. (1989). Analogical learning and transfer: What develops? In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (p. 369–412). Cambridge University Press. <https://doi.org/10.1017/CBO9780511529863.019>
- Butterfield, E. C., & Nelson, G. D. (1991). Promoting Positive Transfer of Different Types. *Cognition and Instruction*, 8(1), 69–102. [https://doi.org/10.1207/s1532690xci0801\\_3](https://doi.org/10.1207/s1532690xci0801_3)
- Burke, B. (2012). Gamification 2020: What Is the Future of Gamification. *Gartner*. <https://www.gartner.com/doc/2226015/Gamification--future-Gamification>
- Byrnes, J. P. (1996). *Cognitive Development and Learning in Instructional Contexts*. Allyn and Bacon. <https://bit.ly/3xAr1Ru>
- Campistrous L. (2000). Tecnología, resolución de problemas y Didáctica de la Matemática, *ICCP*, p. 7

- Campoy, T. (2018). Metodología de la investigación científica. Manual para la elaboración de tesis y trabajos de investigación. Asunción: Marben.
- Cárdenas, M. (2020). *Marco referencial para el Pensamiento Computacional en Educación Básica*. Coordinación general @prende.mx. Secretaría de Educación Pública. <https://bit.ly/2QGdntd>
- Carretero, S., Vuorikari, R., y Punie, Y. (2017). *DigComp 2.1: The Digital Competence Framework for Citizens with eight proficiency levels and examples of use* (No. EUR 28558). Comisión Europea. <https://bit.ly/3gmk8eC>
- Cea, M., Blanco, F., Álvarez, M., Arribas, A., Díaz, E., & Vallés, M. (2012). *Análisis de varianza con medidas repetidas. El procedimiento Modelo lineal general: Medidas repetidas*. SPSS 10 Guía Para El Análisis de Datos, 291–322. <https://bit.ly/30EspVs>
- Chen, Z., & Klahr, D. (2008). Remote transfer of scientific-reasoning and problem-solving strategies in children. In *Advances in Child Development and Behavior* (Vol. 36). [https://doi.org/10.1016/S0065-2407\(08\)00010-4](https://doi.org/10.1016/S0065-2407(08)00010-4)
- Chen, G. H., Shen, J., Barth-Cohen, L., Jiang, S. Y., Huang, X. T., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, 109, 162-175. <https://bit.ly/3xf0hUs>
- Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P. and Glaser, R. (1989). Self-explanations: How students' study and use examples in learning to solve problems. *Cognitive Science*, 13, 145-182.
- Chi, M. T. H., de Leeuw, N., Chiu, M.-H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3), 439–477. <https://psycnet.apa.org/record/1995-16902-001>
- Clark-Carter, D. (2002). *Investigación cuantitativa en psicología: Del diseño experimental al reporte de investigación*. Oxford, University Press. <https://bit.ly/3xlf5BQ>
- Creswell, J. W. (2013). *Qualitative Inquiry & Research Design: Choosing among Five Approaches* (3rd ed.). Thousand Oaks, CA: SAGE. <https://bit.ly/3NEGtBJ>
- CSTA & ISTE. (2011). Operational definition of computational thinking for K-12 education. <https://www.csteachers.org/page/about-csta-s-k-12-nbsp-standards>

- Cunha, F., & Heckman, J. (2007). The technology of skill formation. *American Economic Review*, 97(2), 31-47. <https://bit.ly/32BjCkR>
- Cuny, J., Snyder, L., & Wing, J.M. (2010). Demystifying computational thinking for non-computer scientists. *Educational Research Review*. <https://fla.st/3emfjPB>
- Da Cruz Alves, N., Gresse Von Wangenheim, C., & Hauck, J. C. R. (2019). Approaches to Assess Computational Thinking Competences Based on Code Analysis in K-12 Education: A Systematic Mapping Study. *Informatics in Education*, 18(1), 17-39. <https://doi.org/10.15388/infedu.2019.02>
- del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, 150, 103832. <https://doi.org/10.1016/j.compedu.2020.103832>
- Del Rey, A., & Sanchez-Parga, J. (2011). Crítica de la educación por competencias. *Universitas*, (15), 233. <https://doi.org/10.17163/uni.n15.2011.09>
- Denning, P. J. (2017). *Remaining trouble spots with computational thinking*. *Communications of the ACM*, 60(6), 33–39. <https://doi.org/10.1145/2998438>
- Denning, P. & Tedre, M. (2019). *Computational thinking*. The MIT Press. <https://bit.ly/3xfjfvf>
- Detterman, D. K. (1993a). The case for the prosecution: Transfer as an epiphenomenon. In D. K. Detterman & R. J. Sternberg (Eds.). *Transfer on trial: Intelligence, cognition, and instruction* (p. 1–24). Ablex Publishing.
- Dichanz, H. & Schwittmann, D. (1986). Methoden im Schulalltag. Untersuchung zu Kombinationen und Mustern von Unterrichtsmethoden [Métodos en la vida escolar diaria. Investigación de combinaciones y patrones de métodos de enseñanza]. *La escuela alemana* 78, 3, p. 327-337.
- Dickes, A. C., Farris, A. V., & Sengupta, P. (2020). Sociomathematical Norms for Integrating Coding and Modeling with Elementary Science: A Dialogical Approach. *Journal of Science Education and Technology*, 29(1), 35-52. <https://doi.org/10.1007/s10956-019-09795-7>
- Dirección General de Cultura y Educación de la provincia de Buenos Aires. (2010). *Diseño Curricular para la Educación Secundaria: Marco General para el Ciclo Superior*. <https://bit.ly/3toeb9J>

- Dirección General de Cultura y Educación de la Provincia de Buenos Aires. (2018). *Diseño curricular para la educación primaria: primer ciclo y segundo ciclo*.  
<https://bit.ly/32VsZMj>
- Doyle, S. (2004). *On Transfer: The Distance Learner and the Transfer of Learning* (Working Paper No. 7-04). The Open Polytechnic of New Zealand. <https://bit.ly/3asbYxu>
- Durbin, R. P. (1975). Letter: Acid secretion by gastric mucous membrane. *The American Journal of Physiology*, 229(6), 1726. <https://doi.org/10.1152/ajplegacy.1975.229.6.1726>
- Engelmann, S., & Carnine, D. (1982). *Theory of instruction: Principles and applications*. Irvington. <https://bit.ly/3n6OUcz>
- Ennis, R. H. (2011). *The nature of critical thinking: An outline of critical thinking dispositions and abilities*. Sixth International Conference on Thinking at MIT, Cambridge, MA, July 1994. Last revised May 2011. <https://bit.ly/3xatpwj>
- Ferreira, H. y Pedrazzi, G. (2007). *Teorías y enfoques psicoeducativos del aprendizaje*. Noveduc.
- Feurzeig, W., & Papert, S. A. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*, 19(5), 487–501.  
<https://doi.org/10.1080/10494820903520040>
- Fitts, P. M., & Peterson, J. R. (1964). Information capacity of discrete motor responses. *Journal of Experimental Psychology*, 67(2), 103-112. <https://doi.org/10.1037/h0045689>
- Flavell, J. (1987). *Speculation about the motive and development of metacognition*. Weinert, F. y Klowe, R. (Eds.) *Metacognition, Motivation and Understanding*. Lawrence Erlbaum Associates, Publishers (pp. 21-29).
- Fogarty, R., Perkins, D., & Barrell, J. (1992). *How to teach for transfer*. Hawker Brownlow Education.
- Furber, S. (2012). *Shut down or restart? The way forward for computing in UK schools*. The Royal Society. <https://bit.ly/3sAXEzP>
- Furedy, C., & Furedy, J. (1985). *Critical thinking. Toward research and dialogue*. In Donald, J. y Sullivan, A. (Eds.): *Using research to improve teaching and learning*, No. 23. (pp. 51-69). Jossey-Bass.
- García-Carmona, A. (2002). *Ciencia y pensamiento ilustrado*. *Red Científica: Ciencia, Tecnología y Pensamiento*. 48. <https://bit.ly/3epTi2E>

- García-Valcárcel-Muñoz-Repiso, A., & Caballero-González, Y.-A. (2019). Robotics to develop computational thinking in early Childhood Education. *Comunicar*, 27(59), 63-72.  
<https://doi.org/10.3916/C59-2019-06>
- Gentner, D., & Gentner, D. R. (1983). Flowing waters or teeming crowds: Mental models of electricity. In D. Gentner & A. L. Stevens (Eds.), *Mental models* (pp. 99-129). Lawrence Erlbaum Associates. <https://bit.ly/3eyYb9N>
- Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology*, 12(3), 306-355. [https://doi.org/10.1016/0010-0285\(80\)90013-4](https://doi.org/10.1016/0010-0285(80)90013-4)
- Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology*, 15(1), 1-38. [https://doi.org/10.1016/0010-0285\(83\)90002-6](https://doi.org/10.1016/0010-0285(83)90002-6)
- Gick, M. L., & Holyoak, K. J. (1987). *The cognitive basis of knowledge transfer*. In S. M. Cormier & J. D. Hagman (Eds.), *The educational technology series. Transfer of learning: Contemporary research and applications* (p. 9-46). Academic Press.  
<https://bit.ly/3gocJLO>
- Goldstone, R. L., & Day, S. B. (2012). Introduction to "New conceptualizations of transfer of learning." *Educational Psychologist*, 47(3), 149-152.  
<https://doi.org/10.1080/00461520.2012.695710>
- Gómez, C. B., Sanjosé, V., & Solaz-Portolés, J. J. (2012). Una revisión de los procesos de transferencia para el aprendizaje y enseñanza de las ciencias. *Didáctica de las Ciencias Experimentales y Sociales*, 0(26). <https://doi.org/10.7203/dces.26.1934>
- González-González, C. S. (2019). Estrategias para la enseñanza del pensamiento computacional y uso efectivo de tecnologías en educación infantil: una propuesta inclusiva. *Revista Interuniversitaria de Investigación En Tecnología Educativa*, 85-97.  
<https://doi.org/10.6018/riite.405171>
- Greeno, J. G., Smith, D. R., & Moore, J. L. (1993). Transfer of situated learning. In D. K. Detterman & R. J. Sternberg (Eds.), *Transfer on trial: Intelligence, cognition, and instruction* (pp. 99-167). Ablex <https://bit.ly/3as2gv6>
- Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
- Gutarra, F. (2015). *Introducción a la ingeniería industrial*. Fondo Editorial de la Universidad Continental. <https://bit.ly/3gwFEgL>

- Gutiérrez-Núñez, S.E., Cordero-Hidalgo, A., & González- Quiñones (2021). Revisión sistemática de intervenciones educativas de pensamiento computacional en K-12. In Vega-Caro, L., Vico-Bosch, A., & Recio-Moreno, D. (2021). Nuevas formas de aprendizaje en la era digital: en busca de una educación inclusiva, 237-270. Dykinson. <https://bit.ly/3aLjB53>
- Guzmán, F. (2017). Problemática general de la educación por competencias. *Revista Iberoamericana de Educación*, 74, 107–120. <https://bit.ly/32uWFzE>
- Hage, K., Bischoff, H., Dichanz, H., Eubel, K.D., Oehlschläger, H.J. & Schwittmann, D. (1985). Das Methodenrepertoire von Lehrern. Eine Untersuchung zum Schulalltag der Sekundarstufe I. [El método del repertorio de los profesores. Un estudio de la vida escolar cotidiana en el nivel secundario I]. Opladen: Leske + Budrich. <https://bit.ly/3arB7Zk>
- Haskell, R. (2000). *Transfer of Learning. Cognition, Instruction and Reasoning*. Academic Press. <https://bit.ly/2QDsXWy>
- Hatano, G., & Inagaki, K. (1992). Desituating cognition through the construction of conceptual knowledge. In P. Light & G. Butterworth (Eds.), *Context and cognition: Ways of learning and knowing* (pp. 115–133). Lawrence Erlbaum Associates, Inc
- Healy, A., & Wohldmann, E. (2012). Specificity and Transfer of Learning. *Psychology of Learning and Motivation*, 57,227-253. <https://doi.org/10.1016/B978-0-12-394293-7.00006-6>
- Heintz, F., Mannila, L., & Farnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. *IEEE Frontiers in Education Conference (FIE)*, 1-9. <https://doi.org/10.1109/FIE.2016.7757410>
- Helfenstein, S. (2005). *Transfer: Review, Reconstruction, and Resolution*. [Tesis doctoral, Universidad de Jyväskylä]. Unidad Editorial, Biblioteca Universitaria de Jyväskylä. <https://bit.ly/3n7QoDK>
- Herederero de Pedro, C. & Muñoz, E. (2011). Otras miradas: Aportaciones de las mujeres a las matemáticas. Para integrar en el currículo de Secundaria. Federación de Enseñanza de CCOO. <https://bit.ly/2RZCqZ7>
- Hernández-Sampieri, R. (2018). *Metodología de la Investigación. Las rutas cuantitativa, cualitativa y mixta*. McGraw Hill. México.



- Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P.R. (1986). *Induction: Processes of Inference, Learning, and Discovery*. MIT Press. [https://doi.org/10.1016/0167-2681\(88\)90041-8](https://doi.org/10.1016/0167-2681(88)90041-8)
- Holyoak, K. J., & Morrison, R. G. (Eds.). (2012). *The Oxford Handbook of Thinking and Reasoning* (1.<sup>a</sup> ed.). Oxford University Press. <https://doi.org/10.1093/oxfordhb/9780199734689.001.0001>
- Holyoak, K. J., & Thagard, P. (1997). The analogical mind. *American Psychologist*, 52(1), 35-44. <https://doi.org/10.1037/0003-066X.52.1.35>
- Instituto Nacional de Estadística y Geográfica. Censo de población y vivienda 2010. <https://bit.ly/3fqkluZ>
- ISTE. (2018). *Iste Standards for Education Leaders*. <https://bit.ly/3xe4sQE>
- Jaramillo, L. M & Puga, L. A. (2016). El pensamiento lógico-abstracto como sustento para potenciar los procesos cognitivos en la educación. *Sophia*, colección de Filosofía de la Educación, 21(2), pp. 31-55. <https://bit.ly/3tON9DL>
- Judd, C. H. (1908). The relation of special training to general intelligence. *Educational Review*, 36, 28–42.
- K–12 Computer Science Framework Steering Committee. (2016). *K–12 Computer Science Framework 2016*. <http://www.k12cs.org>
- Keppel, G. (1991). *Design and Analysis A Researcher's Handbook*. 3rd Edition, Prentice Hall.
- Klahr, D., & Nigam, M. (2004). The Equivalence of Learning Paths in Early Science Instruction: Effects of Direct Instruction and Discovery Learning. *Psychological Science*, 15(10), 661-667. <https://doi.org/10.1111/j.0956-7976.2004.00737.x>
- Kluzer S., & Pujol Priego L. (2018). *DigComp into Action - Get inspired, make it happen*. Carretero, S., Punie, Y., Vuorikari, R., Cabrera, M., & O’Keefe, W. (Eds.). JRC Science for Policy Report, EUR 29115 EN, Publications Office of the European Union. <https://bit.ly/3gHvfPB>
- L’Académie des Sciences (2013). *L’enseignement de l’informatique en France - il est urgent de ne plus attendre [Educación informática en Francia: debemos dejar de esperar con urgencia]*. Technical report. Institute de France. <https://bit.ly/3n6wau3>

- Labarrere A. (1996). *Pensamiento. Análisis y autorregulación de la actividad cognoscitiva de los alumnos*. Editorial Pueblo y Educación; p. 4.
- Lamagna, E. A. (2015). Algorithmic thinking unplugged. *Journal of Computing Sciences in Colleges*, 30(6). 45-52. <https://bit.ly/3n47WAm>
- Larsen-Freeman, D. (2013). Transfer of Learning Transformed. *Language Learning*, 63(SUPPL. 1), 107–129. <https://bit.ly/3sKoWgx>
- Laviña, J., León, G., & Valera, J. (2019). *Innovación Tecnológica y Empleo*. Foro Empresas Innovadoras. <https://bit.ly/3uYA5f4>
- Law, N., Woo, D., De la Torre, J., y Wong, G. (2018). A Global Framework of Reference on Digital Literacy Skills for Indicator 4.4.2. UNESCO Institute for Statistics. <https://bit.ly/3n7urEQ>
- León, O. G. y Montero, I. (2003). *Métodos de investigación en psicología y educación* (3ª ed.) McGraw-Hill.
- Lipman, M. (1998). *Pensamiento complejo y educación*. De la Torre. <https://bit.ly/32xJkqm>
- List G. (1982). *Lógica matemática, teoría de conjuntos y dominios numéricos*. Editorial de Libros para la Educación; p. 12. <https://bit.ly/3xaPVVK>
- Lobato, J. (2006). Alternative Perspectives on the Transfer of Learning: History, Issues, and Challenges for Future Research. *Journal of the Learning Sciences*, 15(4), 431-449. [https://doi.org/10.1207/s15327809jls1504\\_1](https://doi.org/10.1207/s15327809jls1504_1)
- Lodi, M. (2020). Introducing computational thinking in k-12 education: historical, epistemological, pedagogical, cognitive, and affective aspects. [Tesis. Universidad de Bologna, Italia].
- López, G. (2013) Pensamiento crítico en el aula. *Docencia e Investigación* (22), pp. 41-60. <https://bit.ly/3egGTOR>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on Teaching and Learning of Computational Thinking through Programming: What Is Next for K-12? *Computers in Human Behavior*, 41, 51-61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Manches, A., & Plowman, L. (2017). Computing education in children's early years: A call for debate. *British Journal of Educational Technology*, 48(1), 191–201. <https://doi.org/10.1111/bjet.12355>

- Martin, A. (2008). Digital Literacy and the “Digital Society”. In C. Lankshear, & M. Knobel (Eds.), *Digital Literacies: Concepts, Policies, and Practices* (pp. 151-176). New York: Peter Lang
- Mayer, R. and Wittrock, M. (1996) Problem-Solving Transfer. In: Berliner, D.C. and Calfee, R.C., Eds., *Handbook of Educational Psychology*, Macmillan, 47-62.
- Maxwell, S. E., & Delaney, H. D. (2004). (2nd ed.). Lawrence Erlbaum Associates Publishers.
- McGuigan, F. J. (1996). *Psicología experimental. un enfoque metodológico*. Ed. Trillas. México.
- Merteens, D. (2010). *Research and Evaluation in Education and Psychology: Integrating Diversity With Quantitative, Qualitative, and Mixed Methods* (3rd ed.) SAGE, (CA).
- Meseguer P., Moreno J., Moreno J., Olco, K., Pimentel E., Toro, M., Velázquez, A., & Vendrell, E. (2015). *Enseñanza de la informática en primaria, secundaria y bachillerato: estado español, 2015*. Technical report, Sociedad Científica Informática de España, Conferencia de Directores y Decanos de Ingeniería Informática.
- Minian, I., & Monroy, Á. M. (2018). The impact of new technologies on jobs in Mexico. *Problemas Del Desarrollo*, 49(195), 27–53.  
<https://doi.org/10.22201/iiiec.20078951e.2018.195.64001>
- Molero, X. (2016). Un viaje a la historia de la informática. Universitat Politècnica de Valencia.  
<https://bit.ly/3eik5xK>
- Molina Ayuso, Á., Adamuz Povedano, N., & Bracho López, R. (2020). La resolución de problemas basada en el método de Polya usando el pensamiento computacional y Scratch con estudiantes de Educación Secundaria. *Aula Abierta*, 49(1), 83-90.  
<https://doi.org/10.17811/rifie.49.1.2020.83-90>
- Moreno, J., Robles, G., & Román, M. (2017) Programar para aprender en educación primaria y secundaria: ¿Qué indica la evidencia empírica sobre este enfoque? *Revista de investigación en docencia universitaria de la informática*, 10(2). <https://bit.ly/3anRWUM>
- National Research Council. 2010. *Preparing Teachers: Building Evidence for Sound Policy*. The National Academies Press. <https://doi.org/10.17226/12882>
- National Research Council. 2012. *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. National Academies Press.  
<https://doi.org/10.17226/13165>

- National Science & Technology Council (2018). *Charting a Course for Success: America's Strategy for Stem Education. A report by the Committee on Stem Education.*  
<https://bit.ly/32DlDwT>
- OCDE, UNICEF, & UNESCO. (2016). La naturaleza del aprendizaje: Usando la investigación para inspirar la práctica. *Entornos* (29). <https://doi.org/10.25054/01247905.1608>
- OCDE. (2017). *Future of work and skills*. 2nd Meeting of the G20 Employment Working Group, 15-17 February. Hamburg, Germany. <https://bit.ly/3tCsHWq>
- Oliva, J. (2004). El pensamiento analógico desde la investigación educativa y desde la perspectiva del profesor de ciencias. *Revista Electrónica de Enseñanza de las Ciencias*, 3(3), 363-384. <https://bit.ly/3eyBEtJ>
- Oliveros, E. (2002). *Metodología de la enseñanza de Matemática*. Ed. Santillana.
- ONU (2021). *Informe sobre tecnología e información 2021*. Conferencia de las Naciones Unidas sobre Comercio y Desarrollo. <https://bit.ly/2P9vck4>
- Piaget, J. (1988). *Psicología evolutiva de Jean Piaget*. Editorial Paidós Mexicana, S. A.
- Papert, S. (1972). Teaching children thinking. *Programmed Learning and Educational Technology*, 9(5), 245-255. <https://doi.org/10.1080/1355800720090503>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Pascual, J. (2006). *Apuntes de Lógica*. La Mancha.
- Pätzold, G., Klusmeyer, J., Wingels, J. & Lang, M. (2003). *Lehr-Lern-Methoden in der beruflichen Bildung. Eine empirische Untersuchung in ausgewählten Berufsfeldern [Métodos de enseñanza-aprendizaje en la formación profesional. Un estudio empírico en campos ocupacionales seleccionados]*. Oldenburg: Universidad de Oldenburg.
- Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology*, 2(2), 137-168. [https://doi.org/10.1016/0732-118X\(84\)90018-7](https://doi.org/10.1016/0732-118X(84)90018-7)
- Pea, R. D., Kurland, D. M. & Hawkins, J. (1985). LOGO and the Development of Thinking Skills. M. Chen and W. Paisley. *Children and Microcomputers: Research on the Newest Medium*, Sage, pp.193-317 <https://bit.ly/32uxr4D>
- Pellegrino, J. & Hilton, M. (2012). *Education for Life and Work: Developing Transferable Knowledge and Skills in the 21st Century*. National Research Council.  
<https://doi.org/10.17226/13398>

- Pereda, S. (1987), *Psicología experimental. I. Metodología*, Madrid, Ediciones Pirámide.
- Perkins, D. (2009). *Making learning whole: How seven principles of teaching can transform education*. Jossey-Bass.
- Pólya., G. (1957). *How to Solve It: A New Aspect of Mathematical Method*. Doubleday.
- Queiruga, C., Banchoff, C., Venosa, P., Martin, S., Aybar Rosales, V., Gomez, S., & Kimura, I. (2019). Escuelas TIC. El pensamiento computacional en la escuela. *Actas Del XXI Workshop de Investigadores En Ciencias de La Computación (WICC 2019)*, (1), 1–6. <https://bit.ly/3tEOXPp>
- Reed, B. J. (1993). Book Reviews. *The American Review of Public Administration*, 23(4), 424–426. <https://doi.org/10.1177/027507409302300409>
- Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. *SIGCSE'10 - Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, (May 2014), 265–269. <https://doi.org/10.1145/1734263.1734357>
- Rich, P. J., Leatham, K. R., & Wright, G. A. (2013). Convergent cognition. *Instructional Science*, 41(2), 431–453. <https://doi.org/10.1007/s11251-012-9240-7>
- Robertson, S. I. (2001). *Problem solving*. Psychology Press. <https://doi.org/10.4324/9780203457955>
- Robins, A. V., Margulieux, L. E., & Morrison, B. B. (2019). *Cognitive Sciences for Computing Education*. Cambridge Handbook of Computing Education Research. <https://doi.org/10.1017/9781108654555.01>
- Román-González, M. (2015). *Test de Pensamiento Computacional: principios de diseño, validación de contenido y análisis de ítems*. I Jornadas de Doctorandos - Programa de Doctorado en Educación (Madrid, 8 de mayo de 2015). <http://euned.es/product/product.php?prdctID=561> y <https://itunes.apple.com/es/itunes-u/ciencias-sociales-y-juridicas/id654947977?mt=10> Presentación disponible en: <https://db.tt/94WjbIr4>
- Royal Society. (2012). Shut down or restart? The way forward for computing in UK schools. *British Journal of Educational Technology*, (January), 789–801. <https://bit.ly/3xrdU3j>

- Saldarriaga-Zambrano, M., Bravo-Cedeño, M., & Loor, M. (2016). La Teoría Constructivista de Jean Piaget y su significación para la pedagogía contemporánea. *Revista Científica Dominio de Las Ciencias*, 2(esp., dic), 127–137. <https://bit.ly/3eiNHej>
- Salehi, S., Wang, K. D., Toorawa, R., & Wieman, C. (2020). Can majoring in computer science improve general problem-solving skills? *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, 156-161. <https://doi.org/10.1145/3328778.3366808>
- Salomon, G., & Perkins, D. N. (1987). Transfer of Cognitive Skills from Programming: When and How? *Journal of Educational Computing Research*, 3(2), 149–169. <https://doi.org/10.2190/6f4q-7861-qwa5-8pl1>
- Sartepeci, M., & Durak, H. (2017). Analyzing the effect of block and robotic coding activities on computational thinking in programming education. *Educational research and practice*, 490-501. <https://bit.ly/3Nozw7T>
- Schneider, W., & Shiffrin, R. M. (1977). Controlled and automatic human information processing: I. Detection, search, and attention. *Psychological Review*, 84(1), 1–66. <https://doi.org/10.1037/0033-295X.84.1.1>
- Selz, O. (1913). *Über die Gesetze des geordneten Denkverlaufs [Sobre las leyes del curso ordenado del pensamiento]*. Stuttgart: Speman.
- Selz, O. (1922). *Zur Psychologie des produktiven Denkens und des Irrtums [Sobre la psicología del pensamiento productivo y el error]*. Bonn: Friedrich Cohen.
- Secretaría de Educación Pública (SEP) (2020). *Agenda Digital Educativa ADE.mx* <https://bit.ly/3n7T9oj>
- Schiff, R., & Vakil, E. (2015). Age differences in cognitive skill learning, retention and transfer: The case of the Tower of Hanoi Puzzle. *Learning and Individual Differences*, 39, 164–171. <https://doi.org/10.1016/j.lindif.2015.03.010>
- Shayer, M., & Adey, P. S. (1993). Accelerating the development of formal thinking in middle and high school students IV: Three years after a two-year intervention. *Journal of Research in Science Teaching*, 30(4), 351-366. <https://doi.org/10.1002/tea.3660300404>
- Shein, E. (2014). Should everybody learn to code? *Communications of the ACM*, 57(2), 16–18. <https://doi.org/10.1145/2557447>
- Shermer, M. (1997). *Why people believe weird things*. W. H. Freeman.

- Singley, M. K., & Anderson, J. R. (1989). The transfer of cognitive skill. Harvard University Press.
- Siu, K. W. M., & Lam, M. S. (2005). Early childhood technology education: A sociocultural perspective. *Early Childhood Education Journal*, 32, (6), 353-358 <https://bit.ly/3sEjsn7>
- Schanzer, E., Fisler, K., Krishnamurthi, S., & Felleisen, M. (2015). *Transferring skills at solving word problems from computing to algebra through bootstrap*. SIGCSE 2015 - Proceedings of the 46th ACM Technical Symposium on Computer Science Education, 616–621. <https://doi.org/10.1145/2676723.2677238>
- Schmidt, G. (1975). Department of Physics annual progress report. <https://doi.org/10.2172/4173983>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22(October), 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Smith, M (2016). Computer Science for All. *The White House*. <https://bit.ly/3x2AWxo>
- Sternberg, R.J., & Frensch P.A. (1992). On Being an Expert: A Cost-Benefit Analysis. *The Psychology of Expertise*. [https://doi.org/10.1007/978-1-4613-9733-5\\_11](https://doi.org/10.1007/978-1-4613-9733-5_11)
- Stokes, T. F., & Baer, D. M. (1977). An implicit technology of generalization. *Journal of Applied Behavior Analysis*, 10, 349-367. <https://doi.org/10.1901/jaba.1977.10-349>
- Bell, T., & Vahrenhold, J. (2018). CS Unplugged—How Is It Used, and Does It Work?. In: Böckenhauer, HJ., Komm, D., Unger, W. (eds) *Adventures Between Lower Bounds and Higher Altitudes*. Lecture Notes in Computer Science(), vol 11011. Springer, Cham. [https://doi.org/10.1007/978-3-319-98355-4\\_29](https://doi.org/10.1007/978-3-319-98355-4_29)
- Tsarava, K., Leifheit, L., Ninaus, M., Román-González, M., Butz, M. V., Golle, J. & Moeller, K. (2019). *Cognitive correlates of computational thinking: Evaluation of a blended unplugged/Plugged-in course*. ACM International Conference Proceeding Series. <https://doi.org/10.1145/3361721.3361729>
- Thorndike, E. L., & Woodworth, R. S. (1901). The influence of improvement in one mental function upon the efficiency of other functions. *Psychological Review*, 8, 247-261. <https://doi.org/10.1037/h0074898>

- Torres, R. (2015). *Perspectivas Sociales y del Empleo en el Mundo: El empleo en plena mutación*. World Employment and Social Outlook. Organización Internacional del Trabajo. <https://doi.org/10.1002/wow3.69>
- UNESCO. (2017a). *Documento de Trabajo E2030: Educación y Habilidades para el Siglo 21*. <https://bit.ly/3gsrldo>
- UNESCO. (2017b). *La Nueva Agenda Educativa Para América Latina*. <https://bit.ly/3ttwuVX>
- Universidad de Valladolid. (2018). *Efectos E Implicaciones De La Transformación Digital a Nivel Legal, Laboral Y Social*. <https://bit.ly/3n1EHhC>
- U.S. Bureau of Labor Statistics (2021). *Occupational Outlook Handbook*. <https://bit.ly/3dyDxaw>
- Valencia, E. & Panaqué, C. (2019). Pensamiento computacional: una nueva exigencia para la educación del siglo XXI [Computational thinking: a new demand for education of the 21st century] *Espacio pedagógico* 26(2), 323–337. <https://bit.ly/2R7aYI8>
- Valenzuela, M. (2019). *¿Qué hay de nuevo en la metacognición? Revisión del concepto, sus componentes y términos afines*. Educ. Pesqui, 45. <https://bit.ly/2QmwiJG>
- Valverde-Berrocoso, J., Fernández-Sánchez, M. R., & Garrido-Arroyo, M. C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia (RED)*, 46. <https://doi.org/10.6018/red/46/3>
- Van der Linden, N. (2019). *Digital Organisational Frameworks and IT Professionalism*. Comision Europea. <https://bit.ly/3edVLNv>
- VanLehn, K. (1996). COGNITIVE SKILL ACQUISITION. *Annual Review of Psychology*, 47(1), 513-539. <https://doi.org/10.1146/annurev.psych.47.1.513>
- van Merriënboer, J. J., Clark, R. E., & de Croock, M. B. (2002). Blueprints for complex learning: The 4C/ Id- model. *Educational Technology Research and Development*, 50(2), 39– 61. <https://bit.ly/3dy5tuR>
- Vargas, F. (2020). *Formación profesional en la respuesta a la crisis y en las estrategias de recuperación y transformación productiva post COVID - 19*. Organización Internacional del Trabajo (OIT) <https://bit.ly/2RN9HXg>
- Vega, M. (2012). Aspectos y avances en ciencia, tecnología e innovación. *POLIS, Revista Latinoamericana*, 11(33),1-12. <https://www.redalyc.org/articulo.oa?id=30525012025>



- Vielma, E. V., & Salas, M. L. (2000). Aportes de las teorías de Vygotsky, Piaget, Bandura y Bruner. Paralelismo en sus posiciones en relación con el desarrollo. *Educere*, 3(9), 30–37. <https://bit.ly/3dXPYYy>
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2011). Computational thinking. *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2011)* (pp. 33-35). IEEE. <https://bit.ly/3xaOKp9>
- Wing, J. M. (2016). What is Computational Thinking? with Jeannette Wing/ Hosted by Martin, F; Phillips, P; Lee, I; y Knoch, J. CSTA'S video interview series on Youtube "What is Computational Thinking". <https://www.youtube.com/watch?v=fSoknljUI4Q&t=668s>
- Weller, J. (2020). *La pandemia del COVID-19 y su efecto en las tendencias de los mercados laborales*. Cepal. <https://bit.ly/3x62Hou>
- WEF. (2020). *The Future of Jobs Report 2020*. World Economic Forum. *Research Report*, (October), 1163. <https://bit.ly/2Q7NDWV>
- Wenzelburger, E. (2020). La transferencia en el aprendizaje. <http://publicaciones.anuies.mx/acervo/revsup/reso61/art4.htm>
- Wüstenberg, S., Greiff, S., & Funke, J. (2012). Complex problem solving—More than reasoning? *Intelligence*, 40(1), 1-14. <https://doi.org/10.1016/j.intell.2011.11.003>
- Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. *Revista de Educación a Distancia (RED)*, (46). <https://doi.org/10.6018/red/46/4>
- Zapata-Ros, M. (2019). Pensamiento computacional desenchufado. *Education in the Knowledge Society (EKS)*, 20(0), 29. <https://bit.ly/3n5yUrA>

Solución de la Actividad 5

The image shows a Scratch project interface. At the top, there is a blue header with the text "Tutoriales", "ACTIVIDAD 5", "Compartir", "Ver página del proyecto", "Guardar ahora", and a user profile icon. The main workspace is divided into two panels: a script area on the left and a stage area on the right.

**Script Area:**

- Starts with a "when green flag clicked" block.
- Followed by a "ask and wait" block: "¿Cuántas gallinas hay?" y esperar.
- Then a "set variable to" block: "gallinas" a "respuesta".
- An "if-then" block: "si" "gallinas > 50" "entonces".
- Inside the "if" block:
  - "set variable to" block: "Excedente" a "gallinas - 50".
  - "set variable to" block: "Zorros requeridos" a "Excedente / 10".
  - "say" block: "TRAER".
  - "wait" block: "1" segundos.
  - "say" block: "Zorros requeridos".
  - "wait" block: "1" segundos.
  - "say" block: "ZORROS".

The stage area displays three variables in orange boxes: "Excedente" with a value of 37, "Zorros requeridos" with a value of 3.7, and "gallinas" with a value of 87. A cat character is positioned on the stage with a speech bubble containing the text "ZORROS".

The object control panel for "Objeto1" shows the following settings:

- Objeto: Objeto1
- mostrar:  (visible)
- Tamaño: 100
- Dirección: 90

Correlaciones de Pearson

Correlations

	PRETEST	POSTEST	Primera resultado	Segundo resultado	Tercer resultado	Cuanto resultado	Resultado Final	Lengua materna	Matemáticas	Lengua extranjera	Ciencias	Historia	Geografía	Civismo y Ética	Educación física	Atles	Promedio General	
PRETEST	1	.865**	.441	.429	.216	.299	.852	.301	.059	.004	.036	.180	.298	.281	-.118	-.102	.130	
POSTEST		1	.203	.216	.402	.002	.000	.197	.806	.987	.880	.448	.201	.239	.619	.667	.585	
Primera resultado			1	.056	.066	.287	.002	.342	.209	.738	.034	.194	.208	-.285	-.063	.012	.195	
Segundo resultado				1	.612	.635	.345	.366	-.308	-.102	.139	-.067	.101	-.008	-.473	-.438	-.142	
Tercer resultado					1	.005	.061	.346	.875	.288	.973	.894	.644	.262	.494	.808	.992	
Cuanto resultado						1	.490	.543	.182	-.522	.340	.179	.534	.204	-.085	.135	.159	
Resultado Final							1	.150	.105	.121	.336	.621	.112	.573	.814	.711	.681	
Lengua materna								1	.608**	.468*	.522*	.736**	.534*	.409	.463*	.439	.760**	
Matemáticas									1	.004	.037	.018	.000	.015	.073	.031	.053	
Lengua extranjera										1	.110	.100	.100	.100	.100	.100	.100	
Ciencias											1	.012	.008	.144	.077	.080	.247	
Historia												1	.892	.691*	.561	.470	.797**	
Geografía													1	.001	.029	.010	.036	
Civismo y Ética														1	.673**	.653**	.705**	
Educación física															1	.039	.193	
Atles																1	.845**	
Promedio General																	1	.745

\*\* Correlation is significant at the 0.01 level (2-tailed).

\* Correlation is significant at the 0.05 level (2-tailed).

Evaluaciones del Grupo de Tratamiento

1.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2
X			
<b>2. Descompuso el problema en subproblemas</b>			2
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
X			
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			2
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
X			
<b>4. Reutilizó elementos de su propio código</b>			1
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
	X		
<b>TOTAL</b>			8

2.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1
	X		
<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
	X		
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
	X		
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			1

Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
	X		
<b>TOTAL</b>			4

3.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1
	X		
<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
	X		
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
X			
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
		X	
<b>TOTAL</b>			3

4.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2
X			
<b>2. Descompuso el problema en subproblemas</b>			2
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
X			
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			2

Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
X			
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
	X		
<b>TOTAL</b>			7

5.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1
	X		
<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
	X		
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
	X		
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
	X		
<b>TOTAL</b>			4

6.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1

	X		
<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
	X		
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			0
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
		X	
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
		X	
<b>T O T A L</b>			2

7.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2
X			
<b>2. Descompuso el problema en subproblemas</b>			0
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
2		X	
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			0
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
		X	
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
		X	
<b>T O T A L</b>			2



8.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1
	X		
<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
		X	
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			0
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
		X	
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
		X	
<b>T O T A L</b>			2

9.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2
X			
<b>2. Descompuso el problema en subproblemas</b>			0
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
		X	
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
	X		
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			1

Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
	X		
<b>TOTAL</b>			4

10.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1
	X		
<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
	X		
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
	X		
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
		X	
<b>TOTAL</b>			3

Evaluaciones del Grupo Control

11.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2
X			
<b>2. Descompuso el problema en subproblemas</b>			2
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
X			
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
	X		
<b>4. Reutilizó elementos de su propio código</b>			1
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
	X		
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
	X		
<b>TOTAL</b>			7

12.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1
	X		
<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
	X		
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
	X		
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	

<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
		x	
<b>T O T A L</b>			3

13.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2
X			
<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
	X		
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
	X		
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		x	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
	X		
<b>T O T A L</b>			5

14.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2
X			
<b>2. Descompuso el problema en subproblemas</b>			2
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
X			
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1

Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
	X		
<b>4. Reutilizó elementos de su propio código</b>			1
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
	X		
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
	X		
<b>TOTAL</b>			7

15.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2
	X		
<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
	X		
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			0
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
		X	
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
	X		
<b>TOTAL</b>			4

16.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2
	X		

<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
	x		
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
	x		
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		x	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
		x	
<b>TOTAL</b>			4

17.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1
	x		
<b>2. Descompuso el problema en subproblemas</b>			0
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
		X	
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			0
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
		X	
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		x	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
		x	
<b>TOTAL</b>			1

18.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1
	X		
<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
	X	X	
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			0
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
		X	
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
		X	
<b>T O T A L</b>			2

19.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1
	X		
<b>2. Descompuso el problema en subproblemas</b>			1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
	X		
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			0
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
		X	
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		X	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0

Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
		x	
<b>TOTAL</b>			2

20.

<b>1. Identificó el problema</b>			<b>PTS</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2
X			
<b>2. Descompuso el problema en subproblemas</b>			0
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)	
		X	
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)	
	x		
<b>4. Reutilizó elementos de su propio código</b>			0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)	
		x	
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)	
		x	
<b>TOTAL</b>			3



Observaciones repetidas Grupo de Tratamiento

1.

<b>6. Identificó el problema</b>			<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1	1	2	2	2
<b>7. Descompuso el problema en subproblemas</b>			1	1	2	2	2
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)					
<b>8. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1	1	2	2	2
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)					
<b>9. Reutilizó elementos de su propio código</b>			0	0	2	1	1
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)					
<b>10. Desarrolló un algoritmo para solucionar el problema</b>			1	1	2	1	1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)					
<b>TOTAL</b>			4	4	10	6	8

2.

<b>1. Identificó el problema</b>			<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1	1	2	2	2
<b>2. Descompuso el problema en subproblemas</b>			1	1	1	1	2
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)					
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			2	2	1	1	2
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)					
<b>4. Reutilizó elementos de su propio código</b>			0	0	1	0	0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)					
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0	1	1	1	1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)					
<b>TOTAL</b>			4	5	6	5	7

3.

<b>1. Identificó el problema</b>			<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1	1	2	1	1
<b>2. Descompuso el problema en subproblemas</b>			0	0	1	1	1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)					
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			0	1	1	1	1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)					
<b>4. Reutilizó elementos de su propio código</b>			0	0	0	0	0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)					
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0	0	0	0	1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)					
<b>TOTAL</b>			1	2	4	3	4

4.

<b>1. Identificó el problema</b>			<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1	1	2	1	1
<b>2. Descompuso el problema en subproblemas</b>			1	1	1	0	1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)					
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1	0	1	0	1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)					
<b>4. Reutilizó elementos de su propio código</b>			0	0	0	0	0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)					
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0	0	0	0	0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)					
<b>TOTAL</b>			<b>3</b>	<b>2</b>	<b>4</b>	<b>1</b>	<b>3</b>

5.

<b>1. Identificó el problema</b>			<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1	1	2	1	1
<b>2. Descompuso el problema en subproblemas</b>			1	1	0	1	1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)					
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1	0	0	1	1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)					
<b>4. Reutilizó elementos de su propio código</b>			0	0	1	0	0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)					
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0	0	1	0	1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)					
<b>TOTAL</b>			<b>3</b>	<b>2</b>	<b>4</b>	<b>3</b>	<b>4</b>

6.

<b>1. Identificó el problema</b>			<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1	2	2	1	2
<b>2. Descompuso el problema en subproblemas</b>			1	1	1	1	0
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)					
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1	0	1	1	1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)					
<b>4. Reutilizó elementos de su propio código</b>			0	0	0	0	0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)					
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0	1	1	0	1
Funcional (2)	Parcialmente funcional (1)	No funcional (0)					
<b>TOTAL</b>			<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>4</b>

7.

<b>1. Identificó el problema</b>			<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2	1	2	1	1
<b>2. Descompuso el problema en subproblemas</b>			1	1	1	1	1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)					
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			1	0	1	0	1
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)					
<b>4. Reutilizó elementos de su propio código</b>			0	0	1	0	0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)					
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0	1	1	0	0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)					
<b>TOTAL</b>			<b>3</b>	<b>3</b>	<b>6</b>	<b>2</b>	<b>3</b>

8.

<b>1. Identificó el problema</b>			<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1	1	2	0	2
<b>2. Descompuso el problema en subproblemas</b>			0	1	0	0	0
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)					
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			0	0	0	0	0
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)					
<b>4. Reutilizó elementos de su propio código</b>			0	0	1	0	0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)					
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0	0	1	0	0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)					
<b>TOTAL</b>			1	2	4	0	2



9.

<b>1. Identificó el problema</b>			<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	2	1	2	1	1
<b>2. Descompuso el problema en subproblemas</b>			1	1	1	0	1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)					
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			2	0	1	1	0
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)					
<b>4. Reutilizó elementos de su propio código</b>			0	0	0	0	0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)					
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			1	1	1	0	0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)					
<b>TOTAL</b>			<b>6</b>	<b>3</b>	<b>5</b>	<b>2</b>	<b>2</b>

10.

<b>1. Identificó el problema</b>			<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>
Rápidamente (2)	Con dificultad (1)	No lo identificó (0)	1	1	2	1	1
<b>2. Descompuso el problema en subproblemas</b>			0	0	1	0	1
Completamente (2)	Parcialmente (1)	No identificó los subproblemas (0)					
<b>3. Fue capaz de discernir entre la información útil e inútil para resolver el problema</b>			0	0	0	0	0
Completamente (2)	Parcialmente (1)	No identificó los distractores (0)					
<b>4. Reutilizó elementos de su propio código</b>			0	0	0	0	0
Todos los necesarios (2)	Algunos de ellos (1)	Ninguno (0)					
<b>5. Desarrolló un algoritmo para solucionar el problema</b>			0	0	1	0	0
Funcional (2)	Parcialmente funcional (1)	No funcional (0)					
<b>TOTAL</b>			1	1	4	1	2