



**CENTRO DE CIENCIAS BÁSICAS  
DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN  
DOCTORADO EN CIENCIAS APLICADAS Y TECNOLOGÍA**

**TESIS**

**Mapa de ruta CLT4BP: Desarrollo e implementación de un modelo instruccional integral fundamentado en la teoría de la carga cognitiva como apoyo en la enseñanza – aprendizaje de la programación**

**PRESENTA**

M.I.T.C. Carlos Sandoval Medina

**PARA OBTENER EL GRADO DE  
DOCTOR EN CIENCIAS APLICADAS Y TECNOLOGÍA**

**TUTORES**

Dr. Jaime Muñoz Arteaga  
Dr. Carlos Argelio Arévalo Mercado

**COMITÉ TUTORAL**

Dra. Estela Lizbeth Muñoz Andrade  
Dr. Ángel Eduardo Muñoz Zavala

Aguascalientes AGS., 01 julio de 2026

**MTRO. GUILLERMO DOMÍNGUEZ AGUILAR**  
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS

**P R E S E N T E**

Por medio del presente como **TUTOR** designado del estudiante **CARLOS SANDOVAL MEDINA** con ID **285667** quien realizó la tesis titulado/a: **MAPA DE RUTA CLT4BP: DESARROLLO E IMPLEMENTACIÓN DE UN MODELO INSTRUCCIONAL INTEGRAL FUNDAMENTADO EN LA TEORÍA DE CARGA COGNITIVA COMO APOYO EN LA ENSEÑANZA – APRENDIZAJE DE LA PROGRAMACIÓN**, un trabajo propio, innovador, relevante e inédito y con fundamento en la fracción IX del Artículo 43 del Reglamento General de Posgrados, doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que **él** pueda continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

**A T E N T A M E N T E**

**"Se Lumen Proferre"**

Aguascalientes, Ags., a 19 de 06 de 2026

*Dr. Jaime Muñoz Arteaga*  
**Tutor de tesis**

c.c.p.- Interesado

c.c.p.- Coordinación del Programa de Posgrado

**MTRO. GUILLERMO DOMÍNGUEZ AGUILAR**  
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS

**P R E S E N T E**

Por medio del presente como **CO-TUTOR** designado del estudiante **CARLOS SANDOVAL MEDINA** con ID **285667** quien realizó la tesis titulado/a: **MAPA DE RUTA CLT4BP: DESARROLLO E IMPLEMENTACIÓN DE UN MODELO INSTRUCCIONAL INTEGRAL FUNDAMENTADO EN LA TEORÍA DE CARGA COGNITIVA COMO APOYO EN LA ENSEÑANZA – APRENDIZAJE DE LA PROGRAMACIÓN**, un trabajo propio, innovador, relevante e inédito y con fundamento en la fracción IX del Artículo 43 del Reglamento General de Posgrados, doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que **él** pueda continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

**ATENTAMENTE**

**"Se Lumen Proferre"**

Aguascalientes, Ags., a **19** de **6** de **26**

**Dr. Carlos Argelio Arévalo Mercado**  
**Co-Tutor de tesis**

c.c.p.- Interesado

c.c.p.- Coordinación del Programa de Posgrado

**MTRO. GUILLERMO DOMÍNGUEZ AGUILAR**  
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS

**P R E S E N T E**

Por medio del presente como **ASESORA** designado del estudiante **CARLOS SANDOVAL MEDINA** con ID **285667** quien realizó la tesis titulado/a: **MAPA DE RUTA CLT4BP: DESARROLLO E IMPLEMENTACIÓN DE UN MODELO INSTRUCCIONAL INTEGRAL FUNDAMENTADO EN LA TEORÍA DE CARGA COGNITIVA COMO APOYO EN LA ENSEÑANZA – APRENDIZAJE DE LA PROGRAMACIÓN**, un trabajo propio, innovador, relevante e inédito y con fundamento en la fracción IX del Artículo 43 del Reglamento General de Posgrados, doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que él pueda continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

**ATENTAMENTE**  
"Se Lumen Proferre"

Aguascalientes, Ags., a 22 de junio de 2026.

  
**Dra. Estela Lizbeth Muñoz Andrade**  
**Asesora de tesis**

c.c.p.- Interesado

c.c.p.- Coordinación del Programa de Posgrado

**MTRO. GUILLERMO DOMÍNGUEZ AGUILAR**  
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE

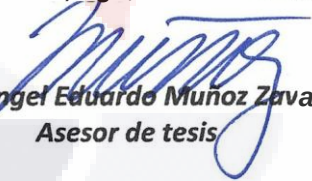
Por medio del presente como **ASESOR** designado del estudiante **CARLOS SANDOVAL MEDINA** con ID **285667** quien realizó la tesis titulada/a: **MAPA DE RUTA CLT4BP: DESARROLLO E IMPLEMENTACIÓN DE UN MODELO INSTRUCCIONAL INTEGRAL FUNDAMENTADO EN LA TEORÍA DE CARGA COGNITIVA COMO APOYO EN LA ENSEÑANZA – APRENDIZAJE DE LA PROGRAMACIÓN**, un trabajo propio, innovador, relevante e inédito y con fundamento en la fracción IX del Artículo 43 del Reglamento General de Posgrados, doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que él pueda continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

**ATENTAMENTE**

**"Se Lumen Proferre"**

**Aguascalientes, Ags, a 26 de 06 de 2026**

  
**Dr. Ángel Eduardo Muñoz Zavala**  
**Asesor de tesis**

c.c.p.- Interesado

c.c.p.- Coordinación del Programa de Posgrado



Fecha de dictaminación (dd/mm/aaaa): 30/06/2026

NOMBRE: Carlos Sandoval Medina

ID 285867

PROGRAMA: Doctorado en Ciencias Aplicadas y Tecnología LGAC (del posgrado): Ciencias de la Computación

MODALIDAD DEL PROYECTO DE GRADO: Tesis ( X ) \*Tesis por artículos científicos ( ) \*\*Tesis por Patente ( ) Trabajo Práctico ( )

TÍTULO: Mapa de ruta CLT4BP: Desarrollo e implementación de un modelo instruccional integral fundamentado en la teoría de carga cognitiva como apoyo en la enseñanza - aprendizaje de la programación

En materia educativa nuevo material pedagógico como apoyo para reducir los índices de reprobación y abandono. Un nuevo modelo para el diseño y desarrollo de material instruccional. Apoyar en la formación de profesionales en el área de la programación.

IMPACTO SOCIAL (señalar el impacto logrado):

INDICAR SEGÚN CORRESPONDA: SI, NO, NA (No Aplica)

| Elementos para la revisión académica del trabajo de tesis o trabajo práctico:                         |                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| SI                                                                                                    | El trabajo es congruente con las LGAC del programa de posgrado                                                                                          |
| SI                                                                                                    | La problemática fue abordada desde un enfoque multidisciplinario                                                                                        |
| SI                                                                                                    | Existe coherencia, continuidad y orden lógico del tema central con cada apartado                                                                        |
| SI                                                                                                    | Los resultados del trabajo dan respuesta a las preguntas de investigación o a la problemática que aborda                                                |
| SI                                                                                                    | Los resultados presentados en el trabajo son de gran relevancia científica, tecnológica o profesional según el área                                     |
| SI                                                                                                    | El trabajo demuestra más de una aportación original al conocimiento de su área                                                                          |
| SI                                                                                                    | Las aportaciones responden a los problemas prioritarios del país                                                                                        |
| SI                                                                                                    | Generó transferencia del conocimiento o tecnológica                                                                                                     |
| SI                                                                                                    | Cumple con la ética para la investigación (reporte de la herramienta antiplagio)                                                                        |
| El egresado cumple con lo siguiente:                                                                  |                                                                                                                                                         |
| SI                                                                                                    | Cumple con lo señalado por el Reglamento General de Posgrados                                                                                           |
| SI                                                                                                    | Cumple con los requisitos señalados en el plan de estudios (créditos curriculares, optativos, actividades complementarias, estancia, predoctoral, etc.) |
| SI                                                                                                    | Cuenta con los votos aprobatorios del comité tutorial                                                                                                   |
| NA                                                                                                    | Cuenta con la carta de satisfacción del Usuario (En caso de que corresponda)                                                                            |
| SI                                                                                                    | Coincide con el título y objetivo registrado                                                                                                            |
| SI                                                                                                    | Tiene congruencia con cuerpos académicos                                                                                                                |
| SI                                                                                                    | Tiene el CVU de la SECIHTI actualizado                                                                                                                  |
| SI                                                                                                    | Tiene el o los artículos aceptados o publicados y cumple con los requisitos institucionales (en caso de que proceda)                                    |
| *En caso de Tesis por artículos científicos publicados (completar solo si la tesis fue por artículos) |                                                                                                                                                         |
|                                                                                                       | Aceptación o Publicación de los artículos en revistas indexadas de alto impacto según el nivel del programa                                             |
|                                                                                                       | El (la) estudiante es el primer autor(a)                                                                                                                |
|                                                                                                       | El (la) autor(a) de correspondencia es el Director (a) del Núcleo Académico                                                                             |
|                                                                                                       | En los artículos se ven reflejados los objetivos de la tesis, ya que son producto de este trabajo de investigación.                                     |
|                                                                                                       | Los artículos integran los capítulos de la tesis y se presentan en el idioma en que fueron publicados                                                   |
| **En caso de Tesis por Patente                                                                        |                                                                                                                                                         |
|                                                                                                       | Cuenta con la evidencia de solicitud de patente en el Departamento de Investigación (anexarla al presente formato)                                      |

Con base en estos criterios, se autoriza continuar con los trámites de titulación y programación del examen de grado:

SI ☒  
No ☐

FIRMAS

Elaboró:

\*NOMBRE Y FIRMA DEL(LA) CONSEJERO(A) SEGÚN LA LGAC DE ADSCRIPCIÓN:

\* En caso de conflicto de intereses, firmará un revisor miembro del NA de la LGAC correspondiente distinto al director o miembro del comité tutorial, asignado por el Decano.

NOMBRE Y FIRMA DEL COORDINADOR DE POSGRADO:

Dr. Francisco Javier Álvarez Rodríguez

Revisó:

NOMBRE Y FIRMA DEL SECRETARIO DE INVESTIGACIÓN Y POSGRADO:

Dra. Iliana Ernestina Medina Ramírez

Autorizó:

NOMBRE Y FIRMA DEL DECANO:

Mtro. Guillermo Domínguez Aguilar

**Nota: procede el trámite para el Depto. de Apoyo al Posgrado**

En cumplimiento con el Art. 24 fracción V del Reglamento General de Posgrado, que a la letra señala entre las funciones del Consejo Académico: Proponer criterios y mecanismos de selección, permanencia, egreso y titulación de estudiantes para asegurar la eficiencia terminal y la titulación y el Art. 28 fracción IX, atender, asesorar y dar el seguimiento del estudiantado desde su ingreso hasta su titulación.

## **Self-Explanation Effect of Cognitive Load Theory in Teaching Basic Programming**

**Carlos Sandoval-Medina, Carlos Argelio Arévalo-Mercado, Estela Lizbeth Muñoz-Andrade, and Jaime Muñoz-Arteaga**

**Recommended Citation:** Sandoval-Medina, C., Arévalo-Mercado, C. A., Muñoz-Andrade, E. L., & Muñoz-Arteaga, J. (2024). Self-Explanation Effect of Cognitive Load Theory in Teaching Basic Programming. *Journal of Information Systems Education*, 35(3), 303-312. <https://doi.org/10.62273/GMIV1698>

**Article Link:** <https://jise.org/Volume35/n3/JISE2024v35n3pp303-312.html>

Received: July 5, 2023  
First Decision: September 25, 2023  
Accepted: December 12, 2023  
Published: September 15, 2024

Find archived papers, submission instructions, terms of use, and much more at the JISE website:  
<https://jise.org>

ISSN: 2574-3872 (Online) 1055-3096 (Print)

---

# Eliciting Knowledge Transfer and Self-management Skill through the Effects of Cognitive Load Theory on Programming Learning

**Carlos Sandoval-Medina\***

Department of Information Systems, Autonomous University of Aguascalientes, Aguascalientes, 20100, México

E-mail: [al285667@edu.uaa.mx](mailto:al285667@edu.uaa.mx)

ORCID iD: <https://orcid.org/0000-0002-6552-4150>

\*Corresponding Author

**Estela L. Muñoz-Andrade**

Department of Electronic Systems, Autonomous University of Aguascalientes, Aguascalientes, 20100, México

E-mail: [lizabeth.munoz@edu.uaa.mx](mailto:lizabeth.munoz@edu.uaa.mx)

ORCID iD: <https://orcid.org/0000-0003-4182-5044>

**Carlos A. Arévalo-Mercado**

Department of Information Systems, Autonomous University of Aguascalientes, Aguascalientes, 20100, México

E-mail: [carlos.arevalo@edu.uaa.mx](mailto:carlos.arevalo@edu.uaa.mx)

ORCID iD: <https://orcid.org/0000-0002-8349-7985>

**Jaime Muñoz-Arteaga**

Department of Information Systems, Autonomous University of Aguascalientes, Aguascalientes, 20100, México

E-mail: [jaime.munoz@edu.uaa.mx](mailto:jaime.munoz@edu.uaa.mx)

ORCID iD: <https://orcid.org/0000-0002-3635-7592>

Received: 01 November, 2024; Revised: 20 March, 2025; Accepted: 26 August, 2025; Published: 08 December, 2025

**Abstract:** Cognitive Load Theory (CLT) is an instructional design theory that aligns with human cognitive architecture for creating instructional materials, through the design guidelines of its 17 instructional effects. However, the Self-Management effect suggests that students can be instructed to manage their learning. The Collective Working Memory effect highlights how a group of students working together can foster a more effective learning environment than an individual student, resulting in better learning outcomes. This research explored applying the Self-Management effect of CLT alongside the Collective Working Memory effect learning data structures in basic programming and measuring their effectiveness regarding essential knowledge acquisition in declarative knowledge, knowledge transfer (near transfer) in procedural knowledge, and developing self-management skills. Cognitive load was measured to determine the difference between groups and to determine the correlation with learning outcomes. The study was carried out through a quasi-experimental design with homogeneous groups, involving students from the Autonomous University of Aguascalientes. The results suggest positive findings in knowledge transfer as well as the development of self-management skills. The cognitive load between the participating groups does not show any significant statistical difference, nor does it show any correlation with the learning results.

**Index Terms:** Cognitive Load Theory, Self-Management Effect, Collective Working Memory Effect, Computer Education, Self-Management Skill, Knowledge Transfer

## 1. Introduction

A key process in education is knowledge transfer (KT), whereby information flows from knowledge providers (teachers, books, video tutorials, online courses) to knowledge recipients (students), who then apply that knowledge to solve problems in the real world [1]. Knowledge transfer in programming refers to the capacity to apply learned



## AGRADECIMIENTOS

A SECIHTI por el apoyo económico brindado a lo largo de estos años de posgrado mediante el programa de becas nacionales para posgrados.

A la Universidad Autónoma de Aguascalientes por brindarme la oportunidad de cursar un posgrado incorporado al sistema nacional de posgrados de calidad.

A mi comité tutorial, Dr. Carlos A. Arévalo Mercado, Dr. Jaime Muñoz Artega, Dra. Estela L. Muñoz Andrade y Dr. Ángel E. Muñoz Zavala por el apoyo y tiempo dedicado durante este periodo.

A todos los estudiantes de las diferentes carreras, que participaron en los diferentes estudios cuasiexperimentales llevados a cabo durante el periodo posgrado.

Al Dr. Héctor Cardona Reyes por el apoyo brindado durante la estancia de investigación, así como a todo el personal de CIMAT Zacatecas por todas las facilidades otorgadas durante este periodo.

## ÍNDICE GENERAL

|                                                                            |           |
|----------------------------------------------------------------------------|-----------|
| <b>LISTA DE FIGURAS . . . . .</b>                                          | <b>4</b>  |
| <b>LISTA DE TABLAS . . . . .</b>                                           | <b>5</b>  |
| <b>RESUMEN . . . . .</b>                                                   | <b>7</b>  |
| <b>ABSTRACT . . . . .</b>                                                  | <b>8</b>  |
| <b>1 INTRODUCCIÓN . . . . .</b>                                            | <b>9</b>  |
| 1.1 Antecedentes del problema de investigación . . . . .                   | 9         |
| 1.2 Problema de investigación . . . . .                                    | 9         |
| 1.3 Propuesta de investigación . . . . .                                   | 10        |
| 1.3.1 Justificación . . . . .                                              | 11        |
| 1.3.2 Planteamiento de objetivos . . . . .                                 | 13        |
| 1.3.3 Preguntas de investigación . . . . .                                 | 14        |
| 1.3.4 Hipótesis . . . . .                                                  | 15        |
| 1.3.5 Alcance y delimitaciones . . . . .                                   | 15        |
| <b>2 ESTADO DEL ARTE . . . . .</b>                                         | <b>16</b> |
| 2.1 Aprendizaje de la programación . . . . .                               | 16        |
| 2.2 Interacción humano-computadora . . . . .                               | 17        |
| 2.3 Realidad virtual en el contexto educativo . . . . .                    | 17        |
| 2.4 Herramientas de apoyo para el aprendizaje de la programación . . . . . | 18        |
| 2.5 Motivación en el proceso de aprendizaje de la programación . . . . .   | 19        |
| <b>3 MARCO TEORICO . . . . .</b>                                           | <b>22</b> |
| 3.1 Teorías cognitivas de aprendizaje . . . . .                            | 22        |
| 3.1.1 Teoría de la codificación dual . . . . .                             | 22        |
| 3.1.2 Diseño instruccional de cuatro componentes 4C/ID . . . . .           | 22        |
| 3.1.3 Diez Pasos para Aprendizaje Complejo (Ten Steps To Complex Learning) | 24        |
| 3.1.4 Teoría cognitiva del aprendizaje multimedia . . . . .                | 26        |
| 3.1.5 Teoría de carga cognitiva . . . . .                                  | 28        |
| 3.2 Metacognición . . . . .                                                | 30        |
| 3.3 Teorías de motivación del aprendizaje . . . . .                        | 31        |
| 3.3.1 Modelo de motivación ARCS . . . . .                                  | 32        |
| 3.3.2 Medición de la motivación basada en el modelo ARCS . . . . .         | 35        |
| 3.3.3 Medición de la motivación . . . . .                                  | 35        |

|          |                                                                                     |           |
|----------|-------------------------------------------------------------------------------------|-----------|
| 3.4      | Protocolos verbales . . . . .                                                       | 36        |
| 3.5      | Evaluación del aprendizaje en programación . . . . .                                | 37        |
| 3.6      | Instrucción diferenciada . . . . .                                                  | 38        |
| <b>4</b> | <b>MODELO CONCEPTUAL DE DESARROLLO DE LA PROPUESTA .</b>                            | <b>39</b> |
| 4.1      | Modelo conceptual y etapas de desarrollo . . . . .                                  | 39        |
| 4.2      | Definición de los efectos de la teoría de carga cognitiva a aplicar . . . . .       | 41        |
| 4.3      | Modelo de diseño instruccional mapa de ruta CLT4BP . . . . .                        | 42        |
| <b>5</b> | <b>IMPLEMENTACIÓN DEL MODELO . . . . .</b>                                          | <b>45</b> |
| 5.1      | Fase de análisis . . . . .                                                          | 45        |
| 5.2      | Fase diseño y desarrollo . . . . .                                                  | 48        |
| 5.3      | Aplicación y evaluación . . . . .                                                   | 50        |
| 5.4      | Estudio piloto . . . . .                                                            | 53        |
| 5.4.1    | Implementación fase análisis . . . . .                                              | 54        |
| 5.4.2    | Implementación fase diseño y desarrollo . . . . .                                   | 56        |
| 5.4.3    | Implementación fase aplicación y evaluación . . . . .                               | 57        |
| 5.4.4    | Resultados . . . . .                                                                | 58        |
| <b>6</b> | <b>EFECTOS DE LA TEORÍA DE CARGA COGNITIVA . . . . .</b>                            | <b>61</b> |
| 6.1      | Diseño instruccional y aplicación en programación básica . . . . .                  | 61        |
| 6.1.1    | Solución libre (Goal-Free effect) . . . . .                                         | 61        |
| 6.1.2    | Ejemplo resuelto (Worked example effect) . . . . .                                  | 63        |
| 6.1.3    | Problema por completar (Completion problem effect) . . . . .                        | 65        |
| 6.1.4    | Atención dividida (Split-attention effect) . . . . .                                | 68        |
| 6.1.5    | Modalidad (Modality effect) . . . . .                                               | 69        |
| 6.1.6    | Autoexplicación (Self-explanation effect) . . . . .                                 | 70        |
| 6.1.7    | Redundancia (Redundancy effect) . . . . .                                           | 73        |
| 6.1.8    | Interactividad (Element interactivity effect) . . . . .                             | 75        |
| 6.1.9    | Variabilidad (Variability effect) . . . . .                                         | 77        |
| 6.1.10   | Imaginación (Imagination effect) . . . . .                                          | 79        |
| 6.1.11   | Elementos aislados (Isolated elements effect) . . . . .                             | 80        |
| 6.1.12   | Reversión de experiencia (Expertise reversal effect) . . . . .                      | 81        |
| 6.1.13   | Desvanecimiento de guía (Guidance-fading effect) . . . . .                          | 82        |
| 6.1.14   | Memoria colectiva de trabajo (Collective working memory effect) . . . . .           | 86        |
| 6.1.15   | Información transitoria (Transient information effect) . . . . .                    | 88        |
| 6.1.16   | Movimiento humano (Human movement effect) . . . . .                                 | 90        |
| 6.1.17   | Autogestión (Self-management effect) . . . . .                                      | 91        |
| <b>7</b> | <b>DISEÑO Y DESARROLLO APLICACIÓN WEB CLT4BP (Producto Mínimo Viable) . . . . .</b> | <b>96</b> |
| 7.1      | IA Generativa en la enseñanza de programación . . . . .                             | 96        |
| 7.2      | Plan de desarrollo e implementación . . . . .                                       | 97        |
| 7.3      | Aplicación web CLT4BP: Visión general . . . . .                                     | 98        |
| 7.4      | Componentes principales del sistema . . . . .                                       | 100       |

|          |                                                                                                     |            |
|----------|-----------------------------------------------------------------------------------------------------|------------|
| 7.4.1    | Autenticación y usuarios . . . . .                                                                  | 100        |
| 7.4.2    | Componente gestión de cursos . . . . .                                                              | 100        |
| 7.4.3    | Componente de evaluaciones . . . . .                                                                | 101        |
| 7.4.4    | Componente Generación de Material con IA . . . . .                                                  | 101        |
| 7.4.5    | Módulo de análisis de datos . . . . .                                                               | 103        |
| 7.4.6    | Motor de procesamiento IA . . . . .                                                                 | 105        |
| 7.4.7    | Modelado de datos . . . . .                                                                         | 106        |
| 7.4.8    | Diseño de interfaces . . . . .                                                                      | 106        |
| 7.4.9    | Flujo de actividades detallado . . . . .                                                            | 108        |
| 7.5      | Estudio piloto: Aplicación web CLT4BP . . . . .                                                     | 110        |
| 7.5.1    | Resultados . . . . .                                                                                | 111        |
| <b>8</b> | <b>DISCUSIÓN . . . . .</b>                                                                          | <b>116</b> |
| 8.1      | Limitaciones y consideraciones metodológicas . . . . .                                              | 117        |
| <b>9</b> | <b>CONCLUSIONES . . . . .</b>                                                                       | <b>118</b> |
|          | <b>REFERENCIAS . . . . .</b>                                                                        | <b>120</b> |
| <b>A</b> | <b>ANEXOS . . . . .</b>                                                                             | <b>129</b> |
| A.1      | Motivated Strategies for Learning Questionnaire (MSLQ). . . . .                                     | 129        |
| A.2      | Computational Thinking Scale (CTS) – Escala de pensamiento computacional . . . . .                  | 132        |
| A.3      | Computer Programming Self-Efficacy Scale (CPSES) – Escala de autoeficacia en programación . . . . . | 134        |
| A.4      | Course Interest Survey (CIS) . . . . .                                                              | 135        |
| A.5      | Instructional Materials Motivation Scale (IMMS) . . . . .                                           | 136        |
| A.6      | Guía de referencia estrategias diferenciadas . . . . .                                              | 139        |
| A.7      | Encuesta sobre el componente de carga cognitiva de CS . . . . .                                     | 141        |



## LISTA DE FIGURAS

|      |                                                                                                                                                |     |
|------|------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 3.1  | Modelo instruccional 4C/ID (Van Merriënboer & Kirschner, 2018). . . . .                                                                        | 23  |
| 3.2  | Vista esquemática de los diez pasos en el proceso de diseño para aprendizaje complejo (van Merriënboer & Kirschner, 2018). . . . .             | 24  |
| 3.3  | Modelo de motivación ARCS de Keller (Bandhu et al., 2024) . . . . .                                                                            | 32  |
| 4.1  | Integración de los fundamentos teóricos en el modelo. . . . .                                                                                  | 39  |
| 4.2  | Etapas para el desarrollo del modelo propuesto . . . . .                                                                                       | 40  |
| 4.3  | Propuesta modelo de diseño instruccional “mapa de ruta CLT4BP”. . . . .                                                                        | 43  |
| 5.1  | Diseño experimental para estudio piloto. . . . .                                                                                               | 53  |
| 5.2  | Extracto de material instruccional desarrollado para estudiantes novatos. . .                                                                  | 57  |
| 5.3  | Extracto de material instruccional desarrollado para estudiantes novatos. . .                                                                  | 57  |
| 5.4  | Comparación promedios en prueba de comprensión. . . . .                                                                                        | 60  |
| 7.1  | Plan de desarrollo aplicando la metodología ágil Scrum. . . . .                                                                                | 97  |
| 7.2  | Arquitectura general del sistema. . . . .                                                                                                      | 98  |
| 7.3  | Diagrama arquitectónico producto mínimo viable. . . . .                                                                                        | 99  |
| 7.4  | Creación de prompt para solicitud de material instruccional. . . . .                                                                           | 105 |
| 7.5  | Diagrama ER diseño base datos CLT4BP. . . . .                                                                                                  | 106 |
| 7.6  | Interfaz de inicio de sesión o registro (login). . . . .                                                                                       | 107 |
| 7.7  | Interfaz de inicio acceso perfil instructor. . . . .                                                                                           | 107 |
| 7.8  | Interfaz cursos perfil instructor. . . . .                                                                                                     | 107 |
| 7.9  | Interfaz de administración de cursos. . . . .                                                                                                  | 108 |
| 7.10 | Interfaz pantalla de inicio perfil estudiante. . . . .                                                                                         | 108 |
| 7.11 | Interfaz de cursos estudiantes. . . . .                                                                                                        | 108 |
| 7.12 | Diagrama de actividades. . . . .                                                                                                               | 109 |
| 7.13 | Ejemplo material generado del componente tareas de aprendizaje. . . . .                                                                        | 110 |
| 7.14 | Ejemplo material generado del componente información procedimental. . . .                                                                      | 111 |
| 7.15 | Resultados promedio grupal motivación Pre-Post. Escala Likert (1 a 7, donde 1 = Totalmente en desacuerdo y 7 = Totalmente de acuerdo). . . . . | 112 |
| 7.16 | Resultados carga cognitiva grupal. . . . .                                                                                                     | 113 |
| 7.17 | Resultados CIS e IMMS. . . . .                                                                                                                 | 115 |

## LISTA DE TABLAS

|      |                                                                                                                                                                                                                                                |    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.1  | Correspondencia Modelo 4C/ID - Diez pasos hacia el aprendizaje complejo (Ten Steps to Complex Learning) desarrollada a partir de:(Frerejean et al., 2021; Kirschner, Sweller, Kirschner, & Zambrano, 2018; van Merriënboer & Kirschner, 2018). | 25 |
| 3.2  | Tres principios de la ciencia cognitiva de aprendizaje de acuerdo con la teoría del aprendizaje multimedia (Mayer, 2014).                                                                                                                      | 26 |
| 3.3  | Procesos cognitivos en el aprendizaje multimedia (Mayer, 2014).                                                                                                                                                                                | 26 |
| 3.4  | Formas de representación en el sistema de procesamiento de información (Mayer, 2014)                                                                                                                                                           | 27 |
| 3.5  | Tipos de procesamiento cognitivo en el aprendizaje multimedia                                                                                                                                                                                  | 27 |
| 3.6  | Revisión de literatura sobre estrategias basadas en la TCC.                                                                                                                                                                                    | 30 |
| 3.7  | Teorías de la motivación y literatura relacionada                                                                                                                                                                                              | 32 |
| 3.8  | Componentes del modelo de motivación ARCS aplicado a la programación (Keller, 2010b)                                                                                                                                                           | 34 |
| 4.1  | Efectos de la Teoría de la Carga Cognitiva (TCC)                                                                                                                                                                                               | 42 |
| 4.2  | Clasificación efectos de la TCC en el modelo 4C/ID.                                                                                                                                                                                            | 44 |
| 5.1  | Planificación de contenidos y objetivos                                                                                                                                                                                                        | 45 |
| 5.2  | Registro de resultados de las pruebas finales de desempeño                                                                                                                                                                                     | 46 |
| 5.3  | Registro de indicadores de motivación y autoeficacia                                                                                                                                                                                           | 46 |
| 5.4  | Evaluación de estrategias cognitivas, metacognitivas y manejo de recursos                                                                                                                                                                      | 46 |
| 5.5  | Resumen estadístico de variables de aprendizaje y motivación                                                                                                                                                                                   | 46 |
| 5.6  | Plan de aplicación de efectos de la carga cognitiva por tema.                                                                                                                                                                                  | 47 |
| 5.7  | Fase de análisis del marco de trabajo.                                                                                                                                                                                                         | 47 |
| 5.8  | Matriz de planificación: Instrucción Diferenciada y Modelo 4C/ID                                                                                                                                                                               | 48 |
| 5.9  | Fase de diseño y desarrollo del marco de trabajo basado en 4C/ID y TCC                                                                                                                                                                         | 49 |
| 5.10 | Fase de aplicación y evaluación final del marco de trabajo                                                                                                                                                                                     | 50 |
| 5.11 | Ejemplos de aplicación modelo ARCS al componente Tareas de Aprendizaje 4C/ID en programación básica.                                                                                                                                           | 51 |
| 5.12 | Ejemplos de aplicación modelo ARCS al componente Información de Soporte 4C/ID en programación básica.                                                                                                                                          | 52 |
| 5.13 | Resumen de diagnóstico del entorno grupal (Fase de Análisis)                                                                                                                                                                                   | 54 |
| 5.14 | Matriz de Objetivos de Aprendizaje y Métodos de Evaluación por Nivel                                                                                                                                                                           | 55 |
| 5.15 | Aplicación de los efectos de la Teoría de la Carga Cognitiva por componente                                                                                                                                                                    | 56 |
| 5.16 | ANOVA multivariante (MANOVA)                                                                                                                                                                                                                   | 59 |
| 5.17 | Estadísticos descriptivos de estrategias cognitivas por grupos                                                                                                                                                                                 | 60 |

|      |                                                                                                        |     |
|------|--------------------------------------------------------------------------------------------------------|-----|
| 7.1  | Arquitectura tecnológica backend producto mínimo viable CLT4BP. . . . .                                | 100 |
| 7.2  | Arquitectura tecnológica frontend producto mínimo viable CLT4BP. . . . .                               | 100 |
| 7.3  | Especificaciones para el tipo de material a crear. . . . .                                             | 102 |
| 7.4  | Resultados de la prueba de normalidad por variable. . . . .                                            | 111 |
| 7.5  | Estadísticos descriptivos según el tipo de carga. . . . .                                              | 112 |
| 7.6  | Comparación por pares mediante la prueba de Friedman. . . . .                                          | 112 |
| 7.7  | Análisis de correlación de Spearman entre las variables de estudio. . . . .                            | 113 |
| 7.8  | Resultados descriptivos por dimensión del Course Interest Survey (CIS). . .                            | 114 |
| 7.9  | Resultados descriptivos por dimensión del Instructional Materials Motivation<br>Survey (IMMS). . . . . | 114 |
| 7.10 | Análisis de confiabilidad (Alfa de Cronbach) para los instrumentos CIS e IMMS.                         | 115 |



## RESUMEN

Aprender a programar es un reto para la mayoría de los estudiantes, puesto que la programación es una competencia compleja que requiere el desarrollo de habilidades cognitivas y metacognitivas, lo que se traduce en altas tasas de fracaso (34 % a nivel mundial, 40 % en Latinoamérica), lo que representa problemas persistentes en la enseñanza de la programación. El aprendizaje efectivo requiere la integración de factores cognitivos, motivacionales y de diferenciación individual, que los enfoques instruccionales tradicionales suelen pasar por alto.

Para apoyar en el proceso de enseñanza - aprendizaje de la programación, este estudio presenta el diseño, creación y validación del modelo instruccional mapa de ruta CLT4BP (Cognitive Load Theory For Basic Programming). Un modelo integral de diseño instruccional que incorpora la Teoría de la Carga Cognitiva (TCC) con marcos y teorías de aprendizaje complementarios como el modelo ARCS, el modelo 4C/ID, estrategias de instrucción diferenciadas y protocolos verbales.

Para la implementación del modelo se realizó una investigación piloto cuasi-experimental con 99 estudiantes universitarios matriculados en un curso introductorio de programación en C++ (grupo experimental  $n=56$  y grupo control  $n=43$ ). El grupo control recibió formación tradicional, mientras que el grupo experimental recibió instrucción utilizando el modelo CLT4BP. Se utilizaron pruebas de recuerdo y comprensión para evaluar los objetivos de aprendizaje, así como cuestionarios sobre carga cognitiva, motivación y estrategias cognitivas. Hallazgos: El grupo experimental obtuvo mejores resultados en las pruebas de comprensión tras ajustar las diferencias iniciales ( $d$  de Cohen = 0,41, lo que indica una mejora del 10,1 %,  $p = 0,044$ ). La carga cognitiva no mostró diferencia significativa entre los grupos. Los factores motivacionales podrían mediar la relación entre los resultados del aprendizaje y el tratamiento instruccional. Los niveles de ansiedad fueron mayores en el grupo experimental, lo que indica que el método estructurado podría causar problemas de adaptación a corto plazo.

Los resultados obtenidos de la implementación del modelo CLT4BP demuestran la importancia de un diseño instruccional integral que aborde simultáneamente las diferencias individuales, la motivación y la cognición. Además, se demostró la importancia de la evaluación orientada al proceso, la personalización mejorada con tecnología y la diferenciación basada en la evaluación.

Sumado al modelo, se diseñó y desarrolló una herramienta de software donde se trasladó el modelo a una plataforma web, la cual sigue todo el proceso propuesto por el modelo en un sistema automatizado apoyando al instructor en el desarrollo del material instruccional. La propuesta novedosa de esta herramienta es la integración con una API de IA, la cual es entrenada con el modelo propuesto y sus teorías base para la generación de material instruccional que el instructor puede utilizar en los procesos de enseñanza.

## ABSTRACT

Learning to program is a challenge for most students, as programming is a complex skill requiring the development of cognitive and metacognitive abilities, resulting in high failure rates (34 % worldwide, 40 % in Latin America). This represents a persistent problem in programming education. Effective learning requires the integration of cognitive, motivational, and individual differences, factors that traditional instructional approaches often overlook.

To support the teaching and learning process of programming, this study presents the design, creation, and validation of the CLT4BP instructional roadmap model. This comprehensive instructional design model incorporates Cognitive Load Theory (CLT) with complementary learning frameworks and theories such as the ARCS model, the 4C/ID model, differentiated instructional strategies, and verbal protocols.

To implement the model, a quasi-experimental pilot study was conducted with 99 university students enrolled in an introductory C++ programming course (experimental group  $n=56$  and control group  $n=43$ ). The control group received traditional training, while the experimental group received instruction using the CLT4BP model. Recall and comprehension tests were used to assess learning objectives, along with questionnaires on cognitive load, motivation, and cognitive strategies. Findings: The experimental group performed better on the comprehension tests after adjusting for initial differences (Cohen's  $d = 0.41$ , indicating a 10.1 % improvement,  $p = 0.044$ ). Cognitive load did not differ significantly between the groups. Motivational factors may mediate the relationship between learning outcomes and the instructional treatment. Anxiety levels were higher in the experimental group, suggesting that the structured method could cause short-term adjustment problems.

The results obtained from implementing the CLT4BP model demonstrate the importance of a comprehensive instructional design that simultaneously addresses individual differences, motivation, and cognition. Furthermore, the importance of process-oriented assessment, technology-enhanced personalization, and assessment-based differentiation was demonstrated.

In addition to the model, a software tool was designed and developed to translate it into a web platform. This platform follows the entire process proposed by the model in an automated system, supporting the instructor in developing instructional materials. The innovative aspect of this tool is its integration with an AI API, which is trained on the proposed model and its underlying theories to generate instructional materials that the instructor can use in their teaching processes.



# TESIS TESIS TESIS TESIS TESIS

## CAPÍTULO 1 INTRODUCCIÓN

### 1.1 Antecedentes del problema de investigación

En la actual era digital, el desarrollo de software se ha vuelto fundamental para el funcionamiento de la mayoría de los dispositivos electrónicos que utilizamos hoy en día en nuestras actividades diarias, tanto de recreación o de trabajo, además el desarrollo de software impulsa áreas como la educación, la investigación y la industria. Lo anterior hace referencia a la necesidad cubrir la demanda generada de especialistas en la industria del software (Kato & Shimakage, 2020). Poco a poco el mundo está siendo controlado por software (Andreessen, 2011), puesto que la mayoría de los aparatos que nos rodean necesitan programas de cómputo para su funcionamiento, y la anterior afirmación se incrementa cada día (Fraser et al., 2015), entonces un proceso en el desarrollo de software como lo es la programación también se vuelve fundamental en la industria.

Las habilidades adquiridas en la programación son importantes no solo en la industria de software sino también en el campo de la educación, así también contribuyen en el avance de las tecnologías y comunicación en el mundo (Rahman, Watanobe, & Nakamura, 2020), como lo son la industria 4.0 mediante las fábricas inteligentes, donde surge la necesidad de un desarrollo continuo de sistemas, así como las tecnologías de ciencia de datos e inteligencia artificial (Nakagawa, Antonino, Schnicke, Kuhn, & Liggesmeyer, 2021). Pero el proceso de enseñanza – aprendizaje de la programación siempre ha representado un reto para los estudiantes novatos así como también para los profesores (Beege et al., 2021) con tasas de fallo en los cursos de 34 % aproximadamente en todo el mundo (Bennedsen & Caspersen, 2019; Cerón Garnica, Archundia Sierra, Cervantes Márquez, & Beltrán Martínez, 2021; Watson & Li, 2014).

Lo anterior ha derivado en una constante investigación por encontrar opciones que apoyen tanto a profesores como estudiantes en el proceso de enseñanza – aprendizaje de la programación. Una de estas alternativas es la Teoría de Carga Cognitiva (TCC) que es una teoría de diseño instruccional, la cual reporta resultados positivos en su aplicación como apoyo en el aprendizaje de la programación.

### 1.2 Problema de investigación

El desarrollo de habilidades en programación históricamente representa un alto grado de dificultad para los estudiantes, por lo cual es complicado la formación de especialistas programadores, esta complejidad genera que los estudiantes abandonen o no aprueben el curso. De ello la necesidad de encontrar alternativas pedagógicas que ayuden a mitigar el problema, apoyando el proceso de enseñanza – aprendizaje, tanto a los estudiantes como a los profesores. Por consiguiente, el desarrollo de esta investigación se enfoca a la búsqueda de alternativas al problema de la enseñanza - aprendizaje de la programación a nivel superior en cursos básicos de programación, limitándose a algunos de los factores que describe la literatura como influyentes en el problema.

Temas de aprendizaje complejos como lo es la programación representan un reto para

los estudiantes, sin embargo el completar un curso básico de programación no garantiza que los estudiantes desarrollen las habilidades fundamentales en programación, puesto que la mayoría solo sabe interpretar programas y no construir programas (Insuasti, 2016), sumado a que dichas competencias necesitan el desarrollo de estrategias cognitivas y metacognitivas complejas como lo son, la resolución de problemas, desarrollar o aplicar modelos mentales o matemáticos, desarrollo de pseudocódigo y algoritmos que puedan resolver un problema, y aprender la sintaxis y semántica del lenguaje de programación (Silva-Maceda, David Arjona-Villicana, & Edgar Castillo-Barrera, 2016).

Los conceptos con alto grado de complejidad como son la programación en el proceso de aprendizaje generan una mayor carga cognitiva al estudiante, esto indica que el cerebro del estudiante debe realizar un esfuerzo mayor para el procesamiento de estos conocimientos, pero de acuerdo con los conceptos de la arquitectura humana nuestro cerebro tiene una capacidad de procesamiento de información limitada, entonces si la carga cognitiva sobrepasa esta capacidad el aprendizaje no se lleva a cabo (Sweller, van Merriënboer, & Paas, 2019).

La anterior afirmación se puede constatar con los altos índices de reprobación presentados en estudios realizados a nivel mundial como Bennedsen and Caspersen (2007) reportan un índice de 33 %, Watson and Li (2014) 32.3 %, Simon et al. (2019) 25 %, Bennedsen and Caspersen (2019) 27 %, si bien la tendencia muestra una ligera disminución los indicadores siguen siendo altos. Sin embargo esta tendencia no aplica en Latinoamérica, esto se puede observar en estudios como Beltrán, Sánchez, and Rico (2015) de la Universidad Central de Ecuador donde reportan un índice de reprobación de 47 %, Juárez, López, and Villareal (2016) en el Instituto Tecnológico de Mexicali 49.7 %, Justo, García, Elizabeth, Salinas, and Lara (2021) en la Universidad Autónoma de Baja California 38.42 %, Cerón Garnica et al. (2021) en la Benemérita Universidad Autónoma de Puebla 46 % y la Universidad Autónoma de Aguascalientes a través del departamento de estadística (DGPD — Departamento de Estadística Institucional [uaa.mx]), publica los índices de reprobación los cuales presentan porcentajes variados donde predominan valores sobre 30 %.

Las opciones encontradas en la literatura enfocadas en el apoyo del proceso de enseñanza – aprendizaje de la programación, la mayoría hace alusión a la integración de la tecnología sin ningún fundamento en alguna teoría de aprendizaje (Medeiros, Ramalho, & Falcao, 2019). La omisión de un enfoque instruccional fundamentado en una teoría de aprendizaje podría generar resultados adversos a los objetivos planteados, así como el diseño y desarrollo de material pedagógico deficiente.

El problema que representa la programación en el proceso de enseñanza – aprendizaje ha llegado al grado de convertirse en un problema académico – social, puesto que aparte de afectar directamente al estudiante, repercute en la formación de profesionistas en el área de programación, creando una necesidad que se incrementa día a día por cubrir en todo el mundo (Margulieux, Morrison, & Decker, 2020), afectando el sector social y laboral.

### 1.3 Propuesta de investigación

De lo anterior, la presente propuesta de investigación se delimita al problema de la enseñanza – aprendizaje de la programación en estudiantes universitarios, delimitado a los factores de diseño de material instruccional y motivación del estudiante, en el contexto de materias o cursos introductorios a la programación, es decir fundamentos o conceptos básicos, en un en-

TESIS TESIS TESIS TESIS TESIS

torno de heterogeneidad en cuestión de conocimientos previos y motivación, con el fin no solo de terminar el curso, sino adquirir las habilidades básicas para programar. En el tema hacia cual paradigma de programación utilizar primero en el proceso de enseñanza – aprendizaje de programación básica, de acuerdo con Medeiros et al. (2019) la programación estructurada se adapta más para la enseñanza – aprendizaje de fundamentos de programación.

Dicho problema como ya se mencionó en el presente documento se puede observar de forma tangible en los índices de reprobación y abandono en materias de programación, no obstante, a que los reportes a nivel mundial indican una disminución en estos, los indicadores continúan siendo altos, en lo que respecta a la comunidad Latinoamérica estos índices permanecen por encima del promedio mundial. Así como en los reportes de aumento de la demanda de programadores a nivel mundial.

### 1.3.1 Justificación

La programación es una competencia compleja que requiere que los estudiantes aprendan procedimientos complejos para el desarrollo de soluciones que aumentan la eficiencia en la vida personal y profesional, siendo el desarrollo de software una carrera muy demanda en la mayoría de los sectores (Margulieux et al., 2020).

La programación conlleva el uso de conocimientos a través de diversas capacidades cognitivas y metacognitivas, la solución de problemas y la puesta en práctica de estas soluciones en un código que la computadora pueda comprender. Todo esto demanda principios matemáticos como la lógica y los algoritmos, así como el dominio del lenguaje de programación, todo esto puede desmotivar al alumno y causar cansancio. Por esta razón, la mayoría de los alumnos que realizan los cursos introductorios (conceptos básicos), no poseen las habilidades requeridas en la programación (Singh, 2017).

En base a lo anterior, atendiendo los conceptos de como nuestro cerebro procesa la información, información con alto grado de complejidad como la programación, generan una mayor carga cognitiva al estudiante, esto indica que el cerebro del estudiante debe realizar un esfuerzo mayor para el procesamiento de estos conocimientos, pero nuestro cerebro tiene una capacidad de procesamiento de información limitada, entonces si la carga cognitiva sobrepasa esta capacidad el aprendizaje no se lleva a cabo (Sweller et al., 2019).

El ¿cómo? el ser humano aprende es el fundamento de las teorías cognitivas, basándose en la arquitectura cognitiva humana y como el aprendizaje se procesa y almacena en esquemas cognitivos, que utiliza para crear conexiones con nuevos conocimientos, creando una estructura esquemática cognitiva nueva (Cáceres & Munévar, 2017). De lo anterior surge el tema de la metacognición, que se puede definir como, el ser consciente de tu propio aprendizaje (Argelio et al., 2018), es decir que el estudiante sea capaz de autogestionar su aprendizaje conociendo sus habilidades, estrategias, y limitaciones, así como áreas de mejora (Bustingorry & Mora, 2008). Así pues, se considera el tema de la metacognición para futuros trabajos, puesto que, esta habilidad se desarrolla en etapas posteriores a la adquisición de las habilidades de programación.

Un factor importante en el proceso de enseñanza – aprendizaje de la programación es la motivación del estudiante. La forma en ¿cómo? se transmite el conocimiento a través del material instruccional tiene un impacto en la percepción de los estudiantes, afectando de manera positiva o negativa su motivación, actitud y nivel de compromiso por aprender

(Luxton-Reilly et al., 2018).

El desarrollo de estudiantes autodidactas es el enfoque actual de los modelos de enseñanza pasando de ser solo receptor de conocimiento a ser participe en el proceso de generación de conocimiento proporcionando recursos para el desarrollo de habilidades (Frerejean et al., 2021). Sin embargo, esto no basta si el estudiante mismo carece de interés en aprender. Es necesario que los estudiantes sean atraídos y motivados en los procesos de enseñanza. De esta manera, se establece una relación de compromiso por parte del estudiante y la consecución de resultados se torna más gratificante para todos (maestros y alumnos).

Sumado a lo anterior Alomyan and Green (2019) señalan que no existe una teoría de aprendizaje que por sí sola tenga la respuesta para diseñar material de aprendizaje efectivo que logre los objetivos de aprendizaje deseados. Sin embargo, las teorías de aprendizaje de las ciencias cognitivas cuentan con mayor evidencia empírica de su efectividad en el desarrollo de material instruccional, así como ser las más utilizadas en las áreas como las matemáticas, física y programación.

A pesar de que se han realizado varios estudios para la creación de herramientas de apoyo, material instruccional, enfoques educativos, todo esto para el apoyo del aprendizaje de la programación; la producción de material instruccional para abordar esta problemática no sido lo suficientemente amplia y efectiva, además de que existe una marcada tendencia en un enfoque instruccional tradicional donde el enfoque es a través de ejemplos y ejercicios subsecuentes (Berssanette & de Francisco, 2021).

El problema del aprendizaje de la programación afecta en diferentes sectores, como lo son la educación, puesto que la programación es parte fundamental en el constante avance de la tecnología, en el sector económico mediante la industria de software como parte fundamental en los procesos de negocio, y en ámbito social, presente en la mayoría de los dispositivos electrónicos que utilizamos hoy en día. De ello la importancia de seguir trabajando para atacar este problema en busca de cubrir la demanda en el área de programación que continúa creciendo conforme la tecnología se vuelve omnipresente en nuestro mundo.

Por lo tanto, el objetivo de este proyecto es diseñar, desarrollar e implementar un modelo de diseño instruccional integral "mapa de ruta" fundamentado en la teoría de carga cognitiva. Este modelo combina los conceptos del modelo de Diseño Instruccional de Cuatro componentes (4C/ID), el Modelo de Diez pasos para Aprendizaje Complejo (Ten Steps to Complex Learning), los protocolos verbales, las teorías de motivación e instrucción diferenciada para la creación de recursos educativos en asignaturas introductorias de programación en carreras afines a esta. Finalmente, diseñar un prototipo de software fundamentado en el modelo creado, para respaldar el proceso de enseñanza y aprendizaje de la programación, en cursos de introducción a la programación de nivel superior.

Con el desarrollo de la presente propuesta se pretende contribuir con:

- Un modelo de diseño instruccional "mapa de ruta" basado en teorías cognitivas de aprendizaje como apoyo en el proceso de enseñanza – aprendizaje de la programación.
- El diseño de un prototipo de software basado en el modelo desarrollado.

Las aportaciones de la solución propuesta a las ciencias de la computación serían:

1. En materia educativa reducir los índices de reprobación y abandono en las carreras afines a estas ciencias.



2. Un nuevo modelo para el diseño y desarrollo de material instruccional.
3. Material pedagógico que complementen al material tradicional instruccional para obtener un aprendizaje significativo.
4. En el contexto socioeconómico apoyar a la formación de profesionales en el área de la programación para atender la creciente demanda de esta.

Este estudio propone que el diseño y desarrollo de un modelo instruccional integral fundamentado en las directrices de las teorías cognitivas de aprendizaje, y su implementación en un estudio piloto, así como el diseño de un prototipo de software asociado al modelo, puede apoyar el proceso de enseñanza - aprendizaje.

### **1.3.2 Planteamiento de objetivos**

Como ya se revisó en el presente documento, la utilización de las teorías cognitivas de aprendizaje en los procesos de enseñanza – aprendizaje, específicamente en programación, ha sido ampliamente documentado, obteniendo en su mayoría con resultados positivos. La presente propuesta de investigación parte de este planteamiento para el diseño y desarrollo de un modelo instruccional “mapa de ruta” basado en dichas teorías con el fin de aportar nuevo conocimiento para la solución del problema planteado.

El auge tecnológico genera un enfoque hacia su uso en todas las actividades diarias, así como en la educación. Lo mismo pasa en el desarrollo de herramientas de software para el apoyo en la enseñanza y aprendizaje, tienden más la utilización de tecnología novedosa que a aplicar una teoría de aprendizaje generando dudas sobre su efectividad. Por lo tanto, se propone diseñar un prototipo de software basado en el modelo creado, y a partir de esto, proporcionar herramientas alternativas para solucionar el problema de investigación.

Por lo tanto, se prioriza la Teoría de Carga Cognitiva como fundamento para el diseño y evaluación del modelo sugerido. La teoría de la carga cognitiva es uno de los escasos enfoques que han originado una amplia gama de diseños instruccionales basados en la arquitectura cognitiva humana, a través de los efectos instruccionales que se han identificado a partir de esta, en su mayoría con resultados positivos respaldados por pruebas empíricas de su eficacia (Sweller, 2020). Se pueden hallar pruebas de su amplia utilización en la literatura (Berssanette & de Francisco, 2021; Duran, Zavgorodniaia, & Sorva, 2022).

En segundo lugar, se decide incorporar el diseño instruccional de cuatro componentes 4C/ID y modelo de Diez pasos para aprendizaje complejo. El 4C/ID y Diez pasos para aprendizaje complejo también se basan en la arquitectura cognitiva humana, enfocados en el proceso de aprendizaje de temas complejos (donde entra el tema de la programación), los cuales requieren desarrollar habilidades o competencias profesionales complejas, demandando un alto nivel de interactividad en el proceso de aprendizaje.

La instrucción diferenciada se incorpora al diseño del modelo, puesto que existen situaciones donde el instructor debe diferenciar el material instruccional aplicado a los estudiantes, dependiendo de ciertos factores, como lo son conocimientos previos para este estudio en particular. De ello se opta por el modelo ID de Tomlinson.

Por último, la incorporación de los protocolos verbales, con el fin de proporcionar al estudiante ejemplos de cómo el instructor y un experto en programación llevarían a cabo la resolución de una tarea de programación.

Con base en las teorías cognitivas de aprendizaje y demás elementos planteados (protocolos verbales, teorías de motivación y contexto de grupo), así como la identificación y planteamiento del problema y su revisión de literatura sobre los estudios relacionados y sus hallazgos, se plantean los objetivos.

### **Objetivo general**

Diseñar, desarrollar e implementar un modelo instruccional integral “mapa de ruta”, así como el diseño de un prototipo de software asociado, basado en componentes de teorías cognitivas de aprendizaje para aprendizaje complejo, incorporando teorías de motivación, instrucción diferenciada y protocolos verbales, que tenga como objetivo orientar el diseño y desarrollo de material instruccional efectivo en un entorno de aprendizaje interactivo que fomente el desarrollo de habilidades básicas en programación y metacognición, al tiempo contribuya a la motivación de los estudiantes en el aprendizaje de programación.

### **Objetivos específicos**

1. Analizar e identificar los efectos instruccionales de la Teoría de Carga Cognitiva a utilizar en el modelo.
2. Analizar el 4C/ID e identificar los componentes a utilizar en el modelo.
3. Analizar el modelo Diez Pasos para Aprendizaje Complejo e identificar los componentes a utilizar en el modelo.
4. Analizar e identificar las teorías de motivación para aplicar en el modelo.
5. Analizar e identificar herramientas para la medición de la motivación de los estudiantes.
6. Analizar e identificar herramientas y/o métodos para mantener o incrementar la motivación de los estudiantes para incorporar al modelo propuesto.
7. Identificar el modelo de instrucción diferenciada a integrar en el modelo.
8. Diseñar y desarrollar el modelo, conjuntando la Teoría de Carga Cognitiva, el 4C/ID, modelo Diez Pasos para Aprendizaje Complejo, los protocolos verbales y teorías de motivación.
9. Probar experimentalmente el modelo, con respecto a los objetivos de aprendizaje.
10. Realizar el diseño de un prototipo de software tomando como base el modelo desarrollado.

### **1.3.3 Preguntas de investigación**

¿Es posible diseñar y desarrollar un modelo de diseño instruccional válido para la aplicación de los efectos instruccionales de la teoría de carga cognitiva, conjuntando elementos de otras teorías cognitivas de aprendizaje en el desarrollo de material instruccional para materias introductorias de programación?

A partir del modelo desarrollado, ¿Es posible el desarrollo de herramientas software que incluyan aspectos de motivación del estudiante?

¿La solución propuesta es más efectiva en términos de objetivos de aprendizaje que la enseñanza tradicional?



#### 1.3.4 Hipótesis

**Hipótesis 1:** Es realizable diseñar, desarrollar e implementar un modelo de diseño instruccional para la aplicación de los efectos de la teoría de carga cognitiva, como apoyo en el proceso de enseñanza - aprendizaje de la programación.

**Hipótesis 2:** Es realizable diseñar y desarrollar herramientas de software tomando como base el modelo desarrollado.

**Hipótesis 3:** El modelo propuesto, así como las herramientas de software desarrolladas a partir de el, promoverán un aprendizaje significativamente mejor en los estudiantes que la enseñanza tradicional.

#### 1.3.5 Alcance y delimitaciones

De acuerdo con lo planteado en el presente documento, la presente propuesta de tesis se define partiendo del problema que existe en los estudiantes universitarios en el tema del aprendizaje de la programación. En busca de desarrollar una herramienta que apoye en atacar el problema planteado, se define la siguiente propuesta:

- Diseñar, desarrollar e implementar un modelo de diseño instruccional nuevo al que se ha nombrado “mapa de ruta” con el propósito de apoyar al docente de programación para el diseño y aplicación de material instruccional para el proceso de enseñanza – aprendizaje de la programación.
- Diseñar un prototipo de software, trasladando los conceptos del modelo desarrollado a este prototipo.

Para el desarrollo del modelo se toma como base la teoría de carga cognitiva, incorporando conceptos del modelo de diseño instruccional de cuatro componentes 4C/ID, modelo Diez pasos para aprendizaje complejo, instrucción diferenciada, protocolos verbales y teorías de motivación. En lo correspondiente al diseño del prototipo se utiliza la metodología ágil Scrum, utilizando un enfoque de desarrollo web cliente - servidor, incorporando en el diseño aspectos de motivación y entorno social del estudiante.

Se propone un diseño cuasi-experimental con un formato PRE-POST, verificando los resultados obtenidos contra la metodología de enseñanza tradicional, tomando como parámetros de medida las calificaciones de los estudiantes. En base a lo anterior, el desarrollo de la siguiente propuesta se delimita a:

1. Atacar el problema del aprendizaje de la programación de estudiantes universitarios en cursos introductorios, delimitado a diseño de material instruccional y motivación del estudiante.
2. El enfoque del paradigma de programación es a la programación estructurada.
3. Utilización de herramientas tecnológicas de apoyo, se diseña un prototipo de una aplicación web fundamentada en el modelo propuesto.

## CAPÍTULO 2

### ESTADO DEL ARTE

#### 2.1 Aprendizaje de la programación

Históricamente la enseñanza – aprendizaje de temas complejos conlleva un reto para docentes y alumnos, debido a las habilidades cognitivas y metacognitivas que el estudiante debe desarrollar, esto sumado a factores internos y externos al estudiante, como lo es la motivación. Ejemplo de ello es la enseñanza – aprendizaje de la programación, llegando a ser un problema académico y social (Eteng, Akpotuzor, Akinola, & Agbonlahor, 2022), provocando elevadas tasas de reprobación y abandono en los estudios relacionados con esta disciplina lo que a menudo genera porcentajes de reprobación cercanos al 34 % a escala global (Bennedsen & Caspersen, 2019; Simon et al., 2019; Watson & Li, 2014), y en el caso de las universidades latinoamericanas el promedio de reprobación y abandono es más alto en comparación al índice a nivel mundial, superando en promedio el 40 % (Cerón Garnica et al., 2021; Justo et al., 2021; Juárez et al., 2016)).

El aprendizaje de programación no solo supone el aprendizaje de un lenguaje en sí mismo, sino que es una habilidad compleja que requiere que el alumno aprenda a razonar, examinar una situación y hallar la respuesta más apropiada (Garces, Vieira, Ravai, & Magana, 2023; Price, Williams, Solyst, & Marwan, 2020). Por esta razón, es necesario que los alumnos de programación desarrollen diversas competencias cognitivas (Silva-Maceda et al., 2016):

- Resolución de problemas.
- Desarrollo y aplicación de modelos mentales o matemáticos.
- Generación de pseudocódigo y algoritmos que puedan resolver el problema.
- El aprendizaje de la sintaxis y semántica necesaria para codificar en un lenguaje de programación.

Programar implica el desarrollo y aplicación de habilidades cognitivas y metacognitivas para solucionar problemas y transformar estas soluciones en estructuras definidas bajo un cierto lenguaje de programación que la computadora pueda interpretar. Compañ-Rosique, Satorre-Cuerda, Llorens-Largo, and Molina-Carmona (2015) sostienen que "para cualquier individuo, la elaboración de la solución a un problema demanda un considerable esfuerzo de abstracción, especialmente si se requiere expresarla en forma de algoritmo".

Y. C. Lin, Liu, and Kalyuga (2022) mencionan que, la sintaxis de los lenguajes de programación, el no poder ver los resultados a la par de la ejecución de un programa, y la falta de motivación en los estudiantes afecta el proceso de aprendizaje. Según Goswami, Senges, Estier, and Cherubini (2023) el factor motivacional es un índice predictivo del aprendizaje puesto que éste influye directamente en el esfuerzo y tiempo que el estudiante invierte en el proceso de aprendizaje.

En resumen, para cualquier aprendizaje complejo, como la programación, el estudiante debe desarrollar habilidades cognitivas y metacognitivas que en un formato de enseñanza tradicional es muy complejo, por ello los métodos de enseñanza empleados son muy importantes en el proceso, lo anterior exige la búsqueda de alternativas para producir recursos pedagógicos que puedan contribuir en el proceso de enseñanza y aprendizaje de la programación.

## 2.2 Interacción humano-computadora

El entorno informático soporta el actual estilo de vida de los humanos, el área de la Interacción Humano-Computadora (IHC) en las ciencias de la computación se encarga de estudiar la interacción entre las personas y las computadoras, ofreciendo métodos para el desarrollo de sistemas que cumplan con los requerimientos de diseño para satisfacer las necesidades del usuario atendiendo factores como usabilidad y experiencia del usuario, entre otros (Morales, Rusu, & Quinones, 2020).

En el proceso de desarrollo de software, la función principal del programador es la codificación, y para ello debe interactuar con la computadora mediante entornos de programación y otras herramientas de desarrollo, convirtiéndose en un tipo de usuario el cual genera también respuestas y percepciones sobre el uso de las herramientas que utiliza para realizar su trabajo de programación (Programmer eXperience, PX) (Morales et al., 2020).

Latoza (2016) afirman que la utilización de métodos HCI centrados en el ser humano, en este caso el programador, pueden mejorar las herramientas de programación, y como consecuencia tener mejores resultados en su tarea de codificación. Proponen los métodos “Natural-programming elicitation” y “rapid prototyping”. El primer método se refiere a la utilización de imágenes o videos de situaciones, en donde se incita a los programadores novatos a expresar como instruirían a la computadora para replicar lo visto en las imágenes o videos. Una pauta clave en HCI es iterar rápidamente en el diseño y a esto hace referencia el método de “rapid prototyping”, creando prototipos rápidos para probar si la solución funcionara.

## 2.3 Realidad virtual en el contexto educativo

El termino Realidad Virtual o VR (Virtual Reality), no es en sí nuevo. Cuando comenzó a desarrollarse esta tecnología no se contaba con la infraestructura actual, lo cual no la hacía tan atractiva como en la actualidad, los últimos avances tecnológicos en materia de equipo de VR han hecho que esta tecnología se vuelva muy atractiva para los estudiantes. Los más nuevos visores de realidad virtual o HDMS, entregan una experiencia al usuario con alto grado de inmersión a los mundos VR (Rahman et al., 2020).

Según el diccionario Oxford, la Realidad Virtual es la simulación producida por computadora de imágenes o ambientes en tres dimensiones, en la que una persona puede “interactuar” de manera visual y física mediante el uso de dispositivo electrónico especial (Frederiksen et al., 2020). “La Realidad Virtual es una recreación de un entorno tridimensional producido por ordenadores, donde el usuario tiene la habilidad de tanto visualizar como manejar los elementos de dicho entorno” (Kim et al., 2019). Según Radianti, Majchrzak, Fromm, and Wohlgenannt (2020) la implementación de realidad virtual inmersiva en la educación universitaria se centra en tres aspectos fundamentales: a) la estructura actual de los contenidos educativos, b) los componentes para la creación de realidad virtual y c) las teorías de aprendizaje empleadas. Obteniendo los resultados siguientes:

- Incorporación de la inmersión de realidad virtual en la educación universitaria. 76 % Observadores de realidad virtual, 20 % de realidad virtual móvil, 2 % de realidad virtual mejorada, 2 % de no especificar, 2 % no especifica.

- Elementos de diseño, contenidos de aprendizaje, y tecnologías inmersivas utilizados para apoyar el aprendizaje basado en realidad virtual. El diseño de desarrollo fue el más popular, seguido del experimental, usabilidad, pruebas de usuario.
- Teorías de aprendizaje aplicadas para guiar el diseño y desarrollo de aplicaciones de realidad virtual para la educación superior. 68 % No menciona, 11 % aprendizaje experimental, 3 % constructivismo, y algunos mencionan aprendizaje por gamificación.
- Métodos de evaluación que han sido aplicados para evaluar los resultados del aprendizaje. 22 % cuestionarios, 12 % actividades dentro de la aplicación, 5 % exámenes, evaluación de expertos, grupos de discusión, el enfoque para la evaluación fue la usabilidad.
- Áreas de la educación superior que han aplicado realidad virtual en la enseñanza aprendizaje. 24 % ingeniería, 10 % ciencias de la computación, 7 % astronomía y 12 % no menciona específica área.”

De acuerdo con Silva-Maceda et al. (2016) el 20 de octubre de 2014, un instituto Checo inaugura las primeras clases de realidad virtual a nivel global, empleando las gafas Oculus Rift y la tecnología Leap Motion. Esta tecnología registra los movimientos de las manos, facilitando la interacción.

En lo que respecta al tema de la enseñanza – aprendizaje de la programación, la utilización de la realidad virtual reporta algunos casos de estudio, (Centeno Bragado, 2018; Frederiksen et al., 2020; Kim et al., 2019; Oberhauser & Lecon, 2017; Singh, 2017; Tanielu, Akau Ola, Varoy, & Giacaman, 2019), en su mayoría con resultados positivos.

## 2.4 Herramientas de apoyo para el aprendizaje de la programación

La literatura muestra una gran variedad de estudios enfocados al desarrollo de alternativas que contribuyan en el proceso de instrucción y aprendizaje de la programación. Algunas de estas opciones han sido los lenguajes y ambientes de programación, los investigadores han enfocado sus esfuerzos en desarrollar una amplia diversidad de ambos con el objetivo de simplificar el aprendizaje de la programación, según indican Keller (2010b) en su estudio, en el que presentan una revisión de lenguajes y ambientes de programación creados para respaldar el aprendizaje de la programación, los autores sugieren dos grupos para los lenguajes y ambientes de programación, aquellos diseñados con el propósito de asistir a las personas en el aprendizaje de la programación y los que se diseñan en base a la noción de que la relevancia de la programación es poder ayudar a las personas a crear o construir cosas con base en sus necesidades.

Otras investigaciones enfocadas en el diseño y contenido del curso, aplicación de estrategias de enseñanza y aprendizaje, con enfoque hacia los objetivos de aprendizaje y métodos de evaluación, aplicación de herramientas de software, método de enseñanza centrados en el estudiante, entre otras (Luxton-Reilly et al., 2018).

La utilización de herramientas novedosas como la realidad virtual como lo aplican Oberhauser and Lecon (2017) para la visualización, navegación y transmisión de datos de estructuras de código de manera inmersiva e interactiva, con el objetivo de respaldar los procesos cognitivos, exploratorios, analíticos y descriptivos de código, a través de un prototipo denominado FlyThruCode basada en realidad virtual (VR-FTC). El propósito de este prototipo fue asis-

TESIS TESIS TESIS TESIS TESIS

tir a los programadores a obtener una mejor representación de las estructuras de código, favoreciendo simultáneamente los procesos de comprensión.

La gamificación es otra las opciones más utilizadas en el desarrollo de estas herramientas como lo presenta Centeno Bragado (2018) mediante la virtualización de un juego serio para la instrucción de principios de programación, utilizando grupos de trabajo en los que se plantea inicialmente un problema que demanda la creación de un programa para su resolución y la redacción de la respuesta en papel. La segunda sección consiste en responder a algunas preguntas teóricas relacionadas con conceptos de programación. A partir de las respuestas, se crea un código o clave que, junto con la respuesta al primer problema, establece la ubicación física en la escuela donde debe seguir las instrucciones del siguiente nivel para seguir adelante.

El desarrollo de entornos de aprendizaje con la inclusión de tecnologías novedosas como la realidad virtual y aumentado tal como lo presenta Kim et al. (2019) en su investigación donde se desarrolló una "plataforma de aprendizaje de programación de realidad mixta", aplicando realidad virtual y aumentada. Además de la inclusión del tema de la gamificación donde los estudiantes tiene que resolver los problemas a manera de juego. Para ejecutar el código, el usuario se involucra trasladando físicamente el avatar de acuerdo con las instrucciones que él mismo ha establecido, mientras que el sistema monitorea los movimientos en tiempo real, produciendo feedback visual de fallos en el código, facilitando a los usuarios su identificación más ágil y disminuyendo la carga cognitiva.

La evaluación y retroalimentación automática es otra de las estrategias aplicadas en algunas investigaciones. En este contexto, el proceso de retroalimentación automática se torna más importante ya que con la cantidad de ejercicios multiplicada por la cantidad de estudiantes, los tiempos de revisión tradicionales son muy extensos, a diferencia de las herramientas automáticas (Yen & Wang, 2017) en su estudio, ofrecen una revisión amplia de herramientas de apoyo para el proceso de enseñanza-aprendizaje de la programación, categorizándolas en cuatro categorías: evaluación automática, multimedia, sistemas inteligentes de tutoría y de aprendizaje visual.

Según lo expuesto, la extensa búsqueda de herramientas de apoyo y la creación de recursos educativos para el aprendizaje de la programación ofrece un gran repertorio que consultar en la literatura. Con el auge tecnológico la mayoría lleva su enfoque a la incorporación de estas tecnologías sin dar importancia a bases teóricas en educación como lo son las teorías enseñanza y aprendizaje, lo que genera que la mayoría de estos materiales no cuenten con un diseño adecuado para un proceso de enseñanza – aprendizaje (Medeiros et al., 2019). La ausencia de un enfoque específico hacia una teoría al diseñar recursos educativos, herramientas u objetos de apoyo en el proceso de enseñanza-aprendizaje puede llevar a la desviación de los objetivos y a la consecución de resultados no ideales.

## 2.5 Motivación en el proceso de aprendizaje de la programación

Las emociones son un componente esencial de los procesos cognitivos en todo proceso de aprendizaje y estas afectan el desempeño en las tareas de aprendizaje de nuestro cerebro, ya sea de manera positiva o negativa (Morris, Grehl, Rutter, Mehta, & Westwater, 2022). La complejidad de los principios de programación causa elevadas tasas de reprobación en cursos de programación. Además, afecta los niveles de motivación de los estudiantes e impacta de manera negativa en el proceso de aprendizaje. Por esta razón, mantener la motivación en



los estudiantes es un componente crucial en el proceso de enseñanza y aprendizaje de la programación (Dawar, 2020).

Mangaroska, Sharma, Gašević, and Giannakos (2022) plantean que el pensamiento convergente y la atención se correlacionan positivamente con el rendimiento del estudiante, en cambio la carga de memoria y la carga cognitiva se correlacionan negativamente con el rendimiento del estudiante. En el tema afectivo, encontraron que la confusión y frustración se correlacionan positivamente con el rendimiento, mientras que el gusto y aburrimiento se correlacionan negativamente con el rendimiento. La frustración y la confusión son estados naturales e inevitables que los estudiantes demuestran cuando participan en un aprendizaje profundo. Por lo tanto, siempre que estos estados sean resultado del procesamiento cognitivo y no estén causados por un estímulo externo, no es necesaria una intervención.

Una de las habilidades requeridas por la programación es la resolución de problemas complejos, esta capacidad se ve afectada por muchos factores como lo son, los conocimientos previos, tipo de problema, entorno de grupo, entre otros, donde la aplicación de esta habilidad requiere de la conjunción de las dimensiones cognitivas, metacognitivas y afectivas (Mangaroska et al., 2022), esto requiere que el estudiante esté en un estado de atención sostenida, es decir concentración, pero para evitar sobrecargar su memoria dirigen su atención a determinadas partes de su interés.

Los enfoques de enseñanza declarativo, saber “que”, y procedimental, saber “como”, no son suficientes en la enseñanza de resolución de problemas y por ende la programación, debido a que estos enfoques ignoran el entorno afectivo del estudiante así como el interés de este en el propio conocimiento, por ello el diseño instruccional debe tomar en cuenta la dimensión afectiva del estudiante, generando un diseño de aprendizaje holístico, integrando enfoque declarativo, procedimental, metacognitivo y afectivo (Mangaroska et al., 2022).

De acuerdo con Dawar (2020) la motivación del estudiante en el proceso de aprendizaje de temas complejos como lo es la programación depende en gran medida del valor que el mismo estudiante le da al aprendizaje, así como sus expectativas sobre el mismo, por ello el nivel de compromiso del estudiante dependerá de su percepción sobre el conocimiento a aprender y de su motivación. Dawar (2020) en su trabajo toman como base el modelo motivacional de valor – expectativa del aprendizaje para la aplicación de un método instruccional con el fin de mantener positivamente la motivación de los estudiantes en el aprendizaje de la programación, y con base a sus resultados creen que es posible afectar positivamente la motivación de los estudiantes con su propuesta.

En el proceso de aprendizaje de la programación, una de las habilidades requeridas que deben desarrollar los estudiantes es la resolución de problemas, Mayer (1998) plantean tres componentes que son necesarios en la resolución de problemas, el primero habilidad, conocimiento específico del tema, segundo meta habilidad, estrategias de cómo usar ese conocimiento, y tercero motivación, sentimientos de interés y creer en la propia capacidad de resolver el problema, por ello afirman que los materiales instruccionales deben atender estos tres componentes.

En su investigación Mayer (1998) presentan tres enfoques teóricos con respecto a la motivación, motivación basada en el interés, la cual plantea que los estudiantes se esforzarán más si tienen interés en el tema, motivación basada en la auto-eficacia, los estudiantes se esforzarán más si confían en sus capacidades, y motivación basada en atribuciones, los estudiantes se esforzarán más si atribuyen los éxitos y fracasos al esfuerzo que a las capacidades.

Por otro lado Ryan and Deci (2000) de acuerdo con la teoría de la autodeterminación, plantean los tipos de motivación intrínseca y extrínseca, la primera hace referencia al interés de realizar algo por mera satisfacción, en cambio la motivación extrínseca se refiere al interés de realizar algo porque conduce a la obtención de un resultado o beneficio.

Al ser la programación un tema de enseñanza – aprendizaje complejo es muy importante tomar en cuenta la motivación del estudiante, por ello se han desarrollado varios trabajos intentando elevar la motivación del estudiante en el proceso de enseñanza – aprendizaje de la programación. De los cuales algunos se basan en teorías de motivación, por ejemplo:

- Y. H. Chang, Song, and Fang (2018) proponen la integración del modelo ARCS y el aprendizaje basado en problemas en un formato de salón de clases invertido como apoyo en el aprendizaje de la programación en C#, reportando resultados positivos.
- Goswami et al. (2023) tomando como base la teoría de la autodeterminación desarrollan una plataforma de aprendizaje colaborativo como apoyo del aprendizaje de la programación, reportan resultados positivos en la co-regulación del aprendizaje, así como en la motivación de los estudiantes.
- Huang and Li (2021) presenta un análisis de la motivación de los estudiantes ante el uso de transmisiones en vivo de codificación en el área de programación por parte de expertos programadores.

Otros más basados en principalmente en la gamificación, estrategias específicas y la utilización de herramientas tecnológicas como realidad virtual y aumentada:

- Ardiana and Loekito (2020) aplican el marco de trabajo de gamificación Marczewski para motivar a los estudiantes en un curso de programación orientada a objetos, reportando resultados positivos de su uso.
- Kazimoglu (2020) a través del diseño de un juego serio que implementan como apoyo en el aprendizaje de la programación y como aliciente de la motivación, reportando resultados positivos en ambos rubros.
- Luik et al. (2020) presentan una investigación sobre los factores de motivación que influyen en los estudiantes para ingresar a cursos MOOCs (Masive Open Online Courses).
- Mohanarajah (2018) desarrollan un prototipo de juego educativo, con el principio de “repara y juega”, es decir el juego le va mostrando al estudiante diferentes errores que tiene que corregir en código para poder jugar.

## CAPÍTULO 3

### MARCO TEORICO

#### 3.1 Teorías cognitivas de aprendizaje

Alomyan and Green (2019) plantean que en cualquier proceso de aprendizaje intervienen procesos cognitivos y metacognitivos, como el pensamiento lógico, abstracción, reflexión, motivación, entre muchos otros. Si estos procesos cognitivos y metacognitivos no son bien aplicados, el aprendizaje se torna complicado. Las teorías cognitivas de aprendizaje siguen el enfoque de desarrollar y potencializar estos procesos cognitivos y metacognitivos en los estudiantes para facilitar el aprendizaje. El cognitivismo se centra en la cognición humana o en los pensamientos existentes, basándose en esto, busca mantener la atención del alumno, haciendo sus actividades mentales menos complejas a través de la incorporación de material educativo donde la información debe ser comprensible y, por ende, fácil de procesar por el estudiante. Algunos ejemplos de estos materiales educativos son las animaciones, explicaciones instructivas, ejemplos ilustrativos y demostraciones.

##### 3.1.1 Teoría de la codificación dual

De acuerdo con la teoría de codificación dual, se basa en la premisa de que la cognición humana se da a través de dos sistemas estructurales/procesamientos diferentes, uno que se encarga de procesar eventos no verbales como lo son imágenes, objetos, etc., y otro que encargado de eventos verbales (Paivio, 2014), de acuerdo con esta afirmación, la memoria de largo plazo almacena la información en forma dual, es decir, visual y verbal, lo que facilita su recuerdo (Alomyan & Green, 2019). Esta información guardada en la memoria de larga duración se compone de unidades representativas, vínculos entre estas unidades y sus vínculos con los sistemas de estímulo y respuesta (Paivio, 2014).

De acuerdo con Paivio (2014), las unidades de representación se refieren a la información esta almacenada en la memoria, representaciones verbal o no verbal, llamadas “logogens and images”, donde estas representaciones se activan cuando recibimos un estímulo, ya sea no verbal, mediante imágenes, objetos o eventos, o mediante información verbal. Para que estas representaciones se activen existen diferentes conexiones como lo son, la conexión de los estímulos externos con sus correspondientes “logogens” o “images”, la conexión entre los “logogens” e “images”, y conexiones entre representaciones. El reconocimiento, el recuerdo libre y las tareas de memoria asociadas son las principales fuentes de evidencia relevantes para los diferentes procesos en la teoría de la codificación dual.

Basándose en la teoría de la codificación dual y sus principios Alomyan and Green (2019) enfocados en el diseño de cursos de aprendizaje en línea plantean un listado de implicaciones a tomar en cuenta al diseñar cursos de aprendizaje en línea, todo ello tomando como base los fundamentos de las teorías cognitivas con enfoque especial en la teoría de la codificación.

##### 3.1.2 Diseño instruccional de cuatro componentes 4C/ID

De acuerdo con Van Merriënboer and Kirschner (2018) el modelo de diseño instruccional de cuatro componentes 4C/ID (four-components instructional design) al igual que la teoría

de carga cognitiva se basa en la arquitectura cognitiva humana, se ha ido desarrollando en paralelo a la teoría de la carga cognitiva, la cual ha sido probada en numerosas investigaciones proporcionando evidencia empírica de resultados positivos de la aplicación de sus principios de diseño en material instruccional, en su mayoría relativamente corto, como lecciones o temas específicos de algún curso. Es por ello que el modelo 4C/ID proporciona una extensión a la teoría de carga cognitiva puesto que se centra en el diseño de cursos completos (Sweller et al., 2019).

Está enfocado en el proceso de aprendizaje de temas complejos, los cuales requieren desarrollar habilidades o competencias profesionales complejas, demandando un alto nivel de interactividad en el proceso de aprendizaje. De acuerdo con el modelo 4C/ID las habilidades complejas incluyen habilidades “recurrentes” y “no recurrentes”, las primeras consisten en tareas y situaciones que pueden convertirse en rutinas, y las no recurrentes que se basan en la resolución de problemas, razonamiento y toma de decisiones.

**1. Tareas de aprendizaje**, se basan preferiblemente en tareas de la vida real, donde los alumnos realizan estas tareas de acuerdo con su grado de complejidad de forma gradual, es decir comienzan con tareas simples y van incrementando la complejidad, de esta forma se gestiona la carga cognitiva intrínseca y los alumnos van adquiriendo habilidades recurrentes y no recurrentes en el proceso.

**2. Información de apoyo**, la teoría, la información de apoyo es preferible presentarla a los alumnos antes de que realicen tareas de aprendizaje para evitar una sobrecarga cognitiva.

**3. Información de procedimientos**, instrucciones prácticas y retroalimentación correctiva, las cuales basadas en la carga cognitiva se tiene que presentar al alumno justo en el momento en que lo necesita durante la tarea de aprendizaje.

**4. Práctica de tareas parciales**, selección de aspectos de tareas recurrentes en tareas parciales de práctica. (Sweller et al., 2019).

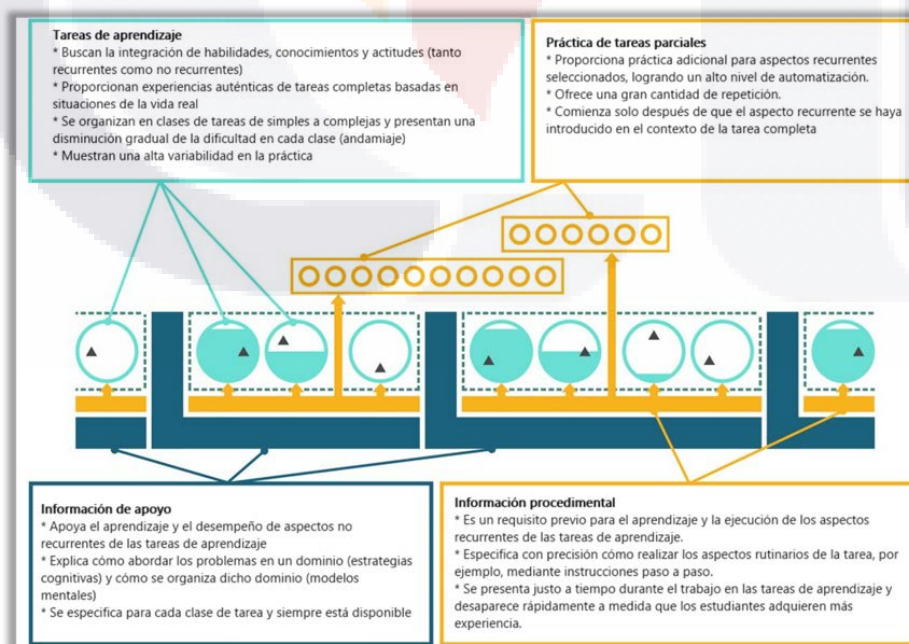


Figura 3.1: Modelo instruccional 4C/ID (Van Merriënboer & Kirschner, 2018).

### 3.1.3 Diez Pasos para Aprendizaje Complejo (Ten Steps To Complex Learning)

El mercado de trabajo evoluciona a la par de los avances e innovaciones tecnológicas, incrementando la complejidad de los actuales y futuros empleos, generando una necesidad de innovación en los programas educativos (Frerejean et al., 2019). De lo anterior surgen varios modelos de diseño instruccional que promueven el aprendizaje de temas complejos, donde el enfoque de todos estos modelos es la utilización de tareas de aprendizaje basadas en tareas del mundo real que un profesional podría encontrar al desempeñar su trabajo (van Merriënboer & Kirschner, 2018).

El principal problema del aprendizaje complejo es la necesidad de aprender y transferir competencias o habilidades complejas a entornos del mundo real (van Merriënboer & Kirschner, 2018), por ello un enfoque de diseño holístico es necesario para que contenga no solo los elementos a aprender sino también su relación entre ellos (Frerejean et al., 2019; Kirschner et al., 2018; van Merriënboer & Kirschner, 2018).

Diez pasos hacia el aprendizaje complejo (Ten Steps to Complex Learning) es una guía práctica, modificada, simplificada y prescriptiva del modelo instruccional de cuatro componentes 4C/ID (ver tabla 3.1) para su aplicación en el diseño instruccional (Kirschner et al., 2018; van Merriënboer & Kirschner, 2018).

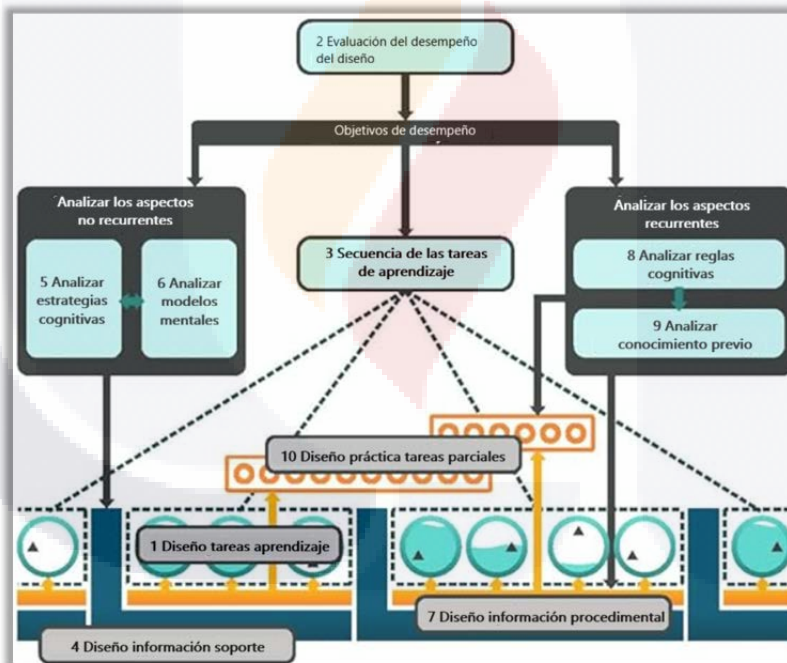


Figura 3.2: Vista esquemática de los diez pasos en el proceso de diseño para aprendizaje complejo (van Merriënboer & Kirschner, 2018).



| Modelo 4C/ID              | Descripción Componentes 4C/ID                                                                                                                                                                                          | Diez pasos hacia el aprendizaje complejo                                                                                                    | Sub-Proceso de aprendizaje | Proceso de aprendizaje     | Ejemplos                                                                                                                                                 |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tareas de aprendizaje     | Tareas integrales basadas en la vida real, para integrar habilidades, conocimientos y actitudes. Con alto nivel de variabilidad y complejidad, organizadas de fácil a difícil, con disminución progresiva del soporte. | 1. Diseño de tareas de aprendizaje<br>2. Secuencia tareas de aprendizaje<br>3. Establecer objetivos de rendimiento (diseño de evaluaciones) | Aprendizaje Inductivo      | Construcción de esquemas   | Proyectos, casos de estudio, problemas, entorno de tareas simuladas por computadora, juegos serios, etc.                                                 |
| Información de soporte    | Información de ayuda para los aspectos no rutinarios de las tareas de aprendizaje, por ejemplo, resolución de problemas. Explica como los temas son organizados y como los problemas deben ser abordados.              | 4. Diseño información de soporte<br>5. Analizar estrategias cognitivas<br>6. Analizar modelos mentales                                      | Elaboración                | Construcción de esquemas   | Libros de textos, presentaciones del instructor, modelos físicos, sistemas multimedia, aprendizaje colaborativo soportado por computadora, micro mundos. |
| Información procedimental | Información de ayuda para los aspectos rutinarios de las tareas de aprendizaje.                                                                                                                                        | 7. Diseño información procedimental<br>8. Analizar reglas cognitivas<br>9. Analizar conocimiento previo                                     | Formación de reglas        | Automatización de esquemas | Guías de referencia rápida, manuales, sistemas de ayuda.                                                                                                 |
| Tareas parciales          | Práctica adicional de aspectos rutinarios para el desarrollo de automaticidad.                                                                                                                                         | 10. Diseño tareas parciales                                                                                                                 | Fortalecimiento            | Automatización de esquemas | Variabilidad de tareas de prácticas isomórficas con base en el mundo real.                                                                               |

Tabla 3.1: Correspondencia Modelo 4C/ID - Diez pasos hacia el aprendizaje complejo (Ten Steps to Complex Learning) desarrollada a partir de:(Frerejean et al., 2021; Kirschner et al., 2018; van Merriënboer & Kirschner, 2018).

### 3.1.4 Teoría cognitiva del aprendizaje multimedia

Mayer and Moreno (1996) propone que la teoría cognitiva de aprendizaje multimedia busca armonizar el diseño de recursos educativos con la estructura cognitiva humana. El enfoque de esta teoría va en el diseño de contenido multimedia educativo y afirma que si este es diseñado basándose en las bases de cómo opera la mente humana en el proceso de aprendizaje tiene más probabilidades de llevar al estudiante a un aprendizaje significativo. Esta teoría se fundamenta en tres principios científicos: el sistema de procesamiento humano (canales duales, visual y auditivo), cada canal tiene una capacidad limitada de procesamiento y el aprendizaje activo implica llevar a cabo un conjunto coordinado de procesos de aprendizaje (Andrade-Lotero, 2012; Mayer, 2014; Mayer & Moreno, 1996) como se puede observar en la tabla 3.2.

| Principio                          | Descripción                                                                                   | Citas relacionadas                 |
|------------------------------------|-----------------------------------------------------------------------------------------------|------------------------------------|
| Canales duales                     | El sistema de procesamiento humano se da a través de dos canales (visual y auditivo)          | Paivio (2014)                      |
| Capacidad limitada                 | Cada canal tiene una capacidad limitada de procesamiento                                      | Sweller, Ayres, and Kalyuga (2011) |
| Procesamiento y aprendizaje activo | El aprendizaje activo implica llevar a cabo un conjunto coordinado de procesos de aprendizaje | Mayer (2014)                       |

Tabla 3.2: Tres principios de la ciencia cognitiva de aprendizaje de acuerdo con la teoría del aprendizaje multimedia (Mayer, 2014).

De acuerdo con Mayer (2014) el diseño de contenido multimedia educativo debe ir enfocado a cinco procesos cognitivos: elegir el texto, elegir imágenes, ordenar las palabras en una representación verbal consistente, ordenar las imágenes en una representación gráfica consistente, e incorporar todas las representaciones, ver tabla 3.3. Existen cinco formas de representación de palabras e imágenes, como se puede observar en la tabla 3.4

| Proceso                | Descripción                                                                                                                          |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Seleccionando palabras | El estudiante presta atención a palabras relevantes en un ambiente multimedia para crear objetos verbales en la memoria de trabajo.  |
| Seleccionando imágenes | El estudiante presta atención a imágenes relevantes en un ambiente multimedia para crear imágenes en la memoria de trabajo.          |
| Organizando palabras   | El estudiante construye conexiones entre las palabras seleccionadas para crear un modelo verbal coherente en la memoria de trabajo.  |
| Organizando imágenes   | El estudiante construye conexiones entre las imágenes seleccionadas para crear un modelo grafico coherente en la memoria de trabajo. |
| Integrando             | El estudiante construye conexiones entre los modelos verbal y gráfico y el conocimiento previo.                                      |

Tabla 3.3: Procesos cognitivos en el aprendizaje multimedia (Mayer, 2014).

| Forma de representación               | Medio/Ubicación                     | Ejemplo                                                                                                                                                   |
|---------------------------------------|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Palabras e imágenes                   | Presentación multimedia<br>Acústica | Ondas de sonido de la bocina de la computadora, patrones de pixeles en la pantalla mostrando alguna figura                                                |
| Representaciones acústicas e icónicas | Memoria sensorial Sonidos           | Sonidos recibidos en los oídos de los estudiantes de la bocina de la computadora, imagen recibida en los ojos de los estudiantes mostrados en la pantalla |
| Sonidos e imágenes                    | Memoria de trabajo                  | Sonidos e imágenes seleccionados                                                                                                                          |
| Modelos verbales y gráfico            | Memoria de trabajo                  | Modelo mental                                                                                                                                             |
| Conocimiento previo                   | Memoria de largo plazo              | Relacionar el modelo mental con el conocimiento previo para crear esquemas.                                                                               |

Tabla 3.4: Formas de representación en el sistema de procesamiento de información (Mayer, 2014)

Durante el proceso de aprendizaje, cuando el estudiante procesa la información, su capacidad cognitiva debe satisfacer tres tipos de exigencias: el procesamiento extraño, que alude al procesamiento de datos que no respalda el objetivo educativo, el procesamiento esencial, que alude al esfuerzo cognitivo para procesar mentalmente la información en la memoria de trabajo y está vinculado con la complejidad del material, y finalmente, el procesamiento generativo, que implica el esfuerzo cognitivo que el estudiante usa para dar sentido a la información (ver tabla 3.5) (Mayer, 2014).

| Nombre                   | Descripción                                                                          | Causado por                | Proceso de aprendizaje   | Ejemplo                                                       |
|--------------------------|--------------------------------------------------------------------------------------|----------------------------|--------------------------|---------------------------------------------------------------|
| Procesamiento extraño    | Procesamiento cognitivo que no está relacionado con la meta instruccional            | Diseño instruccional pobre | Ninguno                  | Centrarse en imágenes irrelevantes                            |
| Procesamiento esencial   | Esfuerzo cognitivo para procesar mentalmente la información en la memoria de trabajo | Complejidad del material   | Seleccionando            | Memorizar la descripción del procesamiento esencial           |
| Procesamiento generativo | Esfuerzo cognitivo que el estudiante destina para dar sentido a la información       | Motivación para aprender   | Organizando e integrando | Explicar el procesamiento generativo con sus propias palabras |

Tabla 3.5: Tipos de procesamiento cognitivo en el aprendizaje multimedia

### 3.1.5 Teoría de carga cognitiva

Sweller et al. (2011); Sweller, van Merriënboer, and Paas (1998); Sweller et al. (2019) afirman que el aprendizaje de temas complejos conlleva a la utilización de habilidades cognitivas y metacognitivas que requieren de un esfuerzo mental adicional por parte del estudiante, y fundamentada en conceptos de la arquitectura humana y como se lleva a cabo la cognición en el cerebro en todo proceso de aprendizaje, la teoría de la carga cognitiva asume que el cerebro humano es un sistema de procesamiento natural de información y que el procesamiento de información en el cerebro humano se realiza a través de dos subsistemas: pasando primero por la memoria de trabajo, que es bastante restringida, para ser almacenada en la memoria de largo plazo, que es inagotable. De ello plantea que todo diseño educativo debería enfocarse en reducir la carga cognitiva que se impone a la memoria de trabajo. Así, hasta ahora se han reconocido 17 efectos/principios para la elaboración de recursos educativos para diversos contextos de aprendizaje (Sweller et al., 2019), con mayoría de ellos con resultados positivos.

En los últimos años, se han reportado múltiples investigaciones sobre el uso de la TCC en el proceso de enseñanza y aprendizaje de la programación. Algunos de estos se detallan en este documento. En la tabla 3.6, se presenta un resumen de la revisión de literatura sobre la TCC en el ámbito de la enseñanza y aprendizaje de la programación.

| Artículo                                | Descripción                                                                                                                                                                                                                 | Resultados                                                                                       | Artefacto o método                                                                                                                        | Efecto TCC                         |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|
| (Aureliano, Tedesco, & Caspersen, 2016) | A través de videos educativos se presentan ejercicios de codificación, donde a la par se incita a los estudiantes a explicar fragmentos de códigos.                                                                         | Estadísticamente no presentan diferencia significativa.                                          | Estudio experimental utilizando enfoque de aprendizaje a través de auto-explicaciones                                                     | Auto-explicación                   |
| (Vieira, Magana, Falk, & Garcia, 2017)  | Se proporcionan ejercicios resueltos con explicaciones en el código.                                                                                                                                                        | Reportan resultados positivos.                                                                   | Estudio experimental utilizando dos enfoques, “the glass box” y “the black box”.                                                          | Auto-explicación, Ejemplo resuelto |
| (Yen & Wang, 2017)                      | Al estar codificando si el estudiante tiene errores de compilación, se le pide explicación del porqué del error, la explicación es comparada con una base de datos, para retroalimentar al estudiante sobre su explicación. | La medición de los resultados es en enfoque a la precisión del prototipo, obteniendo 84 % y 78 % | Prototipo de entorno de programación en C++ utilizando estrategia de auto-explicación y una ontología para proporcionar retroalimentación | Auto-explicación                   |

| Artículo                                            | Descripción                                                                                                                                                                                  | Resultados                                                                                                               | Artefacto o método                                                                                                | Efecto TCC                                                                         |
|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| (Fabric, Mitrovic, & Neshatian, 2017)               | Mediante ejercicios de código por completar, se solicita al estudiante explicar cómo completarían el código.                                                                                 | Reportan diferencias significativas en las puntuaciones en exámenes PRE y POST                                           | Herramienta Py-Kinect, para plataforma móvil, con problemas Parsons, utilizando indicaciones de auto-explicación. | Auto-explicación, Problema por completar                                           |
| (Sands, 2019)                                       | Utilizó el ejemplo resuelto agregando preguntas para provocar la reflexión en el estudiante sobre problema, otra técnica utilizada fue la programación en parejas y la codificación en vivo. | Reportan resultados positivos utilizando los enfoques propuestos.                                                        | Estudio experimental utilizando los enfoques: ejemplo resuelto, programación en pares y codificación en vivo.     | Ejemplo resuelto, Modalidad, Desvanecimiento de guía, Memoria colectiva de trabajo |
| (Zhi et al., 2019)                                  | Se muestran al estudiante ejemplos resueltos junto con mensajes de auto-explicación en un entorno de programación basado en bloques.                                                         | No reportan diferencias significativas en el aprendizaje.                                                                | Prototipo "The Peer Code Helper".                                                                                 | Ejemplo resuelto, Auto-explicación                                                 |
| (Winter, Friend, Matthews, Love, & Vasireddy, 2019) | Desarrollar una comprensión de la descomposición de problemas basada en funciones dentro del contexto de un dominio visual.                                                                  | Reportan resultados positivos en su aplicación.                                                                          | Herramienta de visualización (Briclayer)                                                                          | No mencionan el uso de un efecto específico, solo reducción de carga extraña.      |
| (Price et al., 2020)                                | El estudiante compara su solución, y además presenta una explicación de la solución con la proporcionada por el "Instructor's solution".                                                     | Los resultados están enfocados a involucrar al estudiante más en la reflexión, de lo cual reportan resultados positivos. | Prototipo llamado "Instructor's solution"                                                                         | Auto-explicación                                                                   |
| (Shin, 2020)                                        | Utilización de mapas conceptuales para comparar soluciones con el instructor.                                                                                                                | La aplicación del enfoque propuesto facilita la comprensión de SQL mejor que la instrucción convencional.                | Estudio experimental utilizando enfoque de mapas conceptuales.                                                    | Fundamentos de TCC sin especificar principio único.                                |



| Artículo             | Descripción                                                                                                                      | Resultados                                                                                                         | Artefacto o método                                                                               | Efecto TCC                     |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------|
| (Beege et al., 2021) | Mediante ejercicios de ejemplos con errores en el código ya sea de sintaxis o de semántica, que el estudiante debía identificar. | Los ejemplos resueltos con errores de sintaxis reflejan resultados positivos en el aprendizaje, los semánticos no. | Estudio experimental utilizando enfoque de ejemplo resuelto con errores en sintaxis y semántica. | Ejemplo resuelto (con errores) |

Tabla 3.6: Revisión de literatura sobre estrategias basadas en la TCC.

De acuerdo con lo presentado en la revisión de literatura, la utilización de la TCC en el proceso de enseñanza – aprendizaje de la programación en la mayoría de los casos presentan resultados empíricos positivos, por lo cual es una de las teorías de diseño instruccional más citadas, entre 10000 y 20000 ocasiones (Sweller, 2016), por ello la aplicación de la misma cuenta con suficientes fundamentos para futuras investigaciones.

La literatura científica ha documentado ampliamente los efectos positivos de aplicar teorías cognitivas de aprendizaje en diversos campos educativos, incluyendo la programación, matemáticas y medicina. Este documento sostiene que es posible desarrollar un modelo de diseño instruccional novedoso para mejorar la enseñanza-aprendizaje de programación basándose en la teoría de carga cognitiva, complementada con elementos de otras teorías cognitivas compatibles como la codificación dual, el aprendizaje multimedia y el diseño instruccional 4C/ID. La ausencia de un modelo que integre específicamente estos enfoques representa una oportunidad significativa para la investigación y aplicación práctica de las teorías cognitivas en el contexto del aprendizaje de programación.

### 3.2 Metacognición

Zhao, Thille, Gattani, and Zimmaro (2021) plantean que en cualquier proceso de aprendizaje intervienen procesos cognitivos y metacognitivos, como el pensamiento lógico, abstracción, reflexión, motivación, entre muchos otros. Si estos procesos cognitivos y metacognitivos no son bien aplicados, el aprendizaje se torna complicado. Las teorías cognitivas de aprendizaje siguen el enfoque de desarrollar y potencializar estos procesos cognitivos y metacognitivos en los estudiantes para facilitar el aprendizaje.

De acuerdo con Arevalo-Mercado, Muñoz-Andrade, Cardona Reyes, and Romero-Juárez (2023) la metacognición surge de estos procesos cognitivos y puede definirse como la comprensión de los propios procesos cognitivos. Esto incluye la conciencia del alumnado sobre sus capacidades y limitaciones cognitivas, así como sobre los factores externos que pueden influir en el aprendizaje. También abarca el conocimiento de la disciplina que se está aprendiendo, lo que permite al alumnado autogestionar su proceso de aprendizaje (Ubaidullah, Mohamed, Hamid, Sulaiman, & Yussof, 2021) e identificar y abordar las deficiencias de aprendizaje (Bosch et al., 2021). Se puede definir la metacognición como la cognición que regula el control cognitivo, es decir, permite al estudiante ser consciente de sus fortalezas y debilidades en un proceso de aprendizaje, y así aplicar las estrategias (Bosch et al., 2021). Es decir, el

estudiante deliberadamente prepara, monitorea y evalúa los procesos cognitivos que aplica en proceso de aprendizaje, administrando, supervisando, interpretando y comprendiendo el conocimiento (Ubaidullah et al., 2021). Habilidades cognitivas y metacognitivas son necesarias en el proceso de programación, pero estas habilidades no se adquieren en los primeros cursos introductorios de programación (Ullah, Lajis, Jamjoom, Altalhi, & Saleem, 2020), sino que se tienen que ir desarrollando conforme al estudio y practica (Eteng et al., 2022). Por ello el desarrollo de un eficiente material instruccional es necesario, así como también la forma de impartir de los conocimientos y la motivación del estudiante hacia el proceso (Berssanette & de Francisco, 2021; Y. P. Cheng et al., 2022). Por lo tanto, el conocimiento y control de procesos cognitivos aplicando estrategias o habilidades metacognitivas como planeación, orientación, supervisión, verificación, selección, evaluación, autoevaluación, entre otras, es fundamental para un aprendizaje significativo (Ubaidullah et al., 2021), sumado a la participación del profesor en el desarrollo de estas habilidades en el estudiante a través de material instruccional bien diseñado y la utilización de estrategias de enseñanza – aprendizaje adecuadas (Prather et al., 2022).

### 3.3 Teorías de motivación del aprendizaje

Bandhu et al. (2024); Morris et al. (2022) afirman que la motivación es un componente integral de la experiencia humana que representa el deseo, entusiasmo y determinación para explorar, aprender o realizar alguna actividad, de ello tenemos la motivación intrínseca que se origina en internamente en la persona por sus propios intereses y objetivos, y la motivación extrínseca originada por factores externos como incentivos .

La tabla 3.7 presenta un listado de las principales teorías de motivación con una breve descripción de las mismas, utilizadas en el contexto educativo en diferentes áreas de estudio, tomada a partir de (Bandhu et al., 2024).

| Teoría                         | Descripción                                                                                                                                                                                                  | Literatura                                                                              |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| Teoría extrínseca              | Se centra en factores externos que influyen en la motivación, como incentivos, castigos y reconocimiento social. Como, por ejemplo, dinero, calificaciones o reconocimientos.                                | (Deci, 1972; Ryan & Deci, 2020)                                                         |
| Teoría de la atribución        | Sugiere que las personas tienden a atribuir la causa de un evento a factores internos o externos y que estas atribuciones pueden influir en su motivación.                                                   | (Graham, 1991, 2020; Moring, 2022; Muschetto & Siegel, 2021; Spitzberg & Manusov, 2021) |
| Teoría de la expectativa       | Un individuo estará motivado si cree que: su esfuerzo conducirá a un alto rendimiento, un alto rendimiento conducirá a los resultados deseados y los resultados deseados son valiosos o importantes para él. | (Filipova, 2016; Shang, Moss, & Chen, 2023; Wigfield & Cambria, 2010)                   |
| Teoría intrínseca              | Las personas son motivadas por factores internos (sentimientos) como interés, disfrute y satisfacción (impulso innato).                                                                                      | (Deci & Ryan, 2008; Ryan & Deci, 2020)                                                  |
| Teoría de la expectativa-valor | La motivación está influenciada por las expectativas de logro y valor percibido.                                                                                                                             | (Filipova, 2016; Rachmatullah et al., 2021; Shang et al., 2023)                         |

| Teoría                               | Descripción                                                                                                                                                                                                                                                                                                                                                                              | Literatura                                                                                                       |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Modelo de motivación ARCS            | Se centra en el diseño de estrategias y materiales instruccionales para mejorar la motivación de los estudiantes. Mediante el desarrollo de un ambiente de aprendizaje que capte la atención de los alumnos, demuestre la relevancia del material para sus objetivos e intereses, genere confianza en su capacidad para tener éxito y proporcione una sensación de satisfacción o logro. | (N.-C. Chang & Chen, 2015; Keller, 2010a; K. Li & Keller, 2018)                                                  |
| Teoría de la autodeterminación (SDT) | Abarca motivaciones tanto intrínseca como extrínseca contextualizadas por tres factores:<br>1. La motivación intrínseca para saciar curiosidad o impulso de mejorar.<br>2. Valores sociales que impulsan la motivación extrínseca e internalización de valores.<br>3. Sentimientos de competencia, autonomía y relación.                                                                 | (Deci & Ryan, 2008; Ryan & Deci, 2020)                                                                           |
| Teoría social cognitiva (SCT)        | Las características personales, las tendencias conductuales y el contexto externo interactúan para influir en la capacidad de un individuo para aprender y desempeñarse. La autoeficacia es la principal fuente de motivación.                                                                                                                                                           | (Alahmad, 2020; Ilmiani, Wahdah, & Mubarak, 2021; Stavropoulou, Stamovlasis, & Gonida, 2023; Toomey & Heo, 2022) |
| Teoría de orientación a objetivos    | Los estudiantes analizan actividades con tres objetivos: comprender el contenido (dominio), superar a compañeros (desempeño) y evitar el fracaso. Las metas de dominio promueven la curiosidad y estudio profundo.                                                                                                                                                                       | (Domurath, Coviello, Patzelt, & Ganai, 2020; Lam, King, Yeung, & Zhoc, 2023; Urdan & Kaplan, 2020)               |

Tabla 3.7: Teorías de la motivación y literatura relacionada

### 3.3.1 Modelo de motivación ARCS

El modelo de motivación ARCS (Atención, Relevancia, Confianza y Satisfacción) se centra en el diseño de estrategias para mejorar la motivación. Mediante el desarrollo de un ambiente de aprendizaje que capte la atención de los alumnos, demuestre la relevancia del material para sus objetivos e intereses, genere confianza en su capacidad para tener éxito y proporcione una sensación de satisfacción o logro (Bandhu et al., 2024)(ver Figura 3.3).



Figura 3.3: Modelo de motivación ARCS de Keller (Bandhu et al., 2024)

El modelo de motivación ARCS está formado por cuatro componentes (Bandhu et al., 2024).

- **Atención:** el material instruccional debe captar la atención del estudiante utilizando estrategias como:
  - Utilizar ejemplos basados en el mundo real.
  - Incluir un poco de humor o actividades divertidas (no demasiado).
  - Utilizar sorpresas, situaciones novedosas o inciertas.
  - Estimular la curiosidad a través de cuestiones y preguntas desafiantes y problemáticas.
  - Actividades con participación activa.
  - Sesiones de preguntas y respuestas.
  - Variabilidad, utilizar diferentes métodos y enfoques (videos, blogs, proyectos, trabajo colaborativo, juegos, etc) para mantener el interés.
  - Incluir muchas actividades prácticas.
- **Relevancia:** el material debe ser relevante a los intereses y metas de los estudiantes, y presentado de forma que se conecte con el conocimiento previo de los estudiantes.
  - Explicar como el conocimiento que van a adquirir les ayudará en el presente.
  - Explicar como el conocimiento que van a adquirir les ayudará en el futuro.
  - Usar ejemplos concretos y establecer analogías.
  - Evaluar a los estudiantes para comprender mejor si aprenden gracias a logros, poder o afiliación.
  - Permitir que los estudiantes elijan su propio método y estrategias de instrucción.
  - Permitir a los estudiantes elegir ejemplos y/o proyectos relevantes para ellos.
  - Dar a los estudiantes una sensación de continuidad permitiéndoles establecer conexiones entre la información nueva y lo que ya saben.
  - Muestre a los estudiantes modelos a seguir utilizando el contenido que presenta para mejorar sus vidas.
- **Confianza:** el material debe proporcionar a los estudiantes una sensación de confianza y autoeficacia proporcionándoles instrucciones claras, el nivel apropiado de dificultad y retroalimentación.
  - Proporcionar a los estudiantes estándares de aprendizaje y criterios de evaluación para que puedan establecer expectativas positivas y lograr el éxito.
  - Brindar a los estudiantes la oportunidad de tener éxito brindándoles experiencias múltiples y variadas.
  - Proporcionar retroalimentación sobre sus mejoras y deficiencias durante el proceso para que puedan ajustar su desempeño.
  - Los estudiantes necesitan tener el control de su proceso de aprendizaje para poder sentir que su éxito no depende totalmente de factores externos, sino que existen factores internos que influyen en su éxito.
  - El profesor lance retos, estos no deben ser demasiado difíciles, sino que tengan un grado de dificultad ajustado.

- El profesor debe proporcionar retroalimentación permanente sobre sus fracasos y logros, pero también dando espacio para que los aprendices tengan cierto grado de control de su aprendizaje y aprendan a autorregularse.
- **Satisfacción:** el material debe proporcionar a los estudiantes una sensación de satisfacción y logro al brindarles oportunidades de éxito, resolución de problemas y creatividad.
  - Fomentar el disfrute intrínseco de la experiencia de aprendizaje para que los estudiantes se diviertan y continúen el proceso de aprendizaje sin esperar recompensas u otro tipo de elementos motivacionales externos.
  - Ofrezca a los alumnos retroalimentación positiva, recompensas y refuerzos. Tenga cuidado con la programación de los refuerzos. Es más eficaz si proporciona refuerzos en intervalos no predecibles.
  - Mantenga estándares y consecuencias consistentes para el éxito. Utilice rúbricas de evaluación consistentes y compártalas con los alumnos.

Cada componente del modelo ARCS tiene sub-categorías, la tabla 3.8 muestra algunas de las más aplicadas en la implementación del modelo.

| Componente   | Subcomponente             | Descripción                                                                                                                                                                 |
|--------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Atención     | Entusiasmo perceptivo     | Captar la atención de los estudiantes al comienzo del tema.                                                                                                                 |
|              | Entusiasmo por indagación | Mantener la atención de los estudiantes durante la clase, en lugar de proporcionar el conocimiento, se les permite concluirlo planteando preguntas y creando una discusión. |
|              | Variabilidad              | Utilización de diferentes métodos/herramientas/estrategias instruccionales.                                                                                                 |
| Relevancia   | Orientación a objetivos   | Ayudar a los estudiantes a ver cómo la programación les ayudará a alcanzar sus objetivos.                                                                                   |
|              | Concordancia de motivos   | Mostrar el valor y la utilidad de la programación en el mundo real.                                                                                                         |
|              | Familiaridad              | Conectar la programación con las experiencias previas de los estudiantes.                                                                                                   |
| Confianza    | Requisitos de aprendizaje | Dividir las tareas en pasos pequeños y proporcionar retroalimentación positiva.                                                                                             |
|              | Oportunidades de éxito    | Establecer metas realistas y comunicar claramente los criterios de éxito.                                                                                                   |
|              | Control personal          | Fomentar la creencia en la capacidad de los estudiantes para aprender.                                                                                                      |
| Satisfacción | Refuerzo intrínseco       | Diseñar actividades que sean intrínsecamente motivadoras y gratificantes.                                                                                                   |
|              | Recompensa extrínseca     | Utilizar recompensas y reconocimiento para motivar a los estudiantes.                                                                                                       |
|              | Equidad                   | Asegurar que todos los estudiantes tengan oportunidades de éxito.                                                                                                           |

Tabla 3.8: Componentes del modelo de motivación ARCS aplicado a la programación (Keller, 2010b)



### 3.3.2 Medición de la motivación basada en el modelo ARCS

Keller (2010b) presenta dos herramientas de medición que se pueden utilizar junto con el modelo ARCS.

- Course Interest Survey (CIS), diseñada para medir las reacciones de los estudiantes a la instrucción dirigida por el instructor.
- Instructional Materials Motivation Scale (IMMS), diseñada para medir las reacciones a los materiales de instrucción auto-dirigidos.

Los dos instrumentos mencionados ofrecen flexibilidad para adaptarse a contextos educativos particulares. Las expresiones genéricas como (este curso) o (esta lección) que aparecen en la formulación estándar de los ítems pueden modificarse para reflejar el escenario específico que se evalúa, ya sea (esta conferencia), (esta instrucción mediada por computadora) o (este taller). Asimismo, es factible alterar el tiempo verbal de los ítems para emplearlos como evaluación preliminar. No obstante, es importante mantener intacto el contenido esencial de los ítems, puesto que estos están diseñados para medir características motivacionales específicas.

### 3.3.3 Medición de la motivación

El conocimiento e identificación consciente del grado de motivación de los estudiantes, así como sus variaciones, en el proceso de enseñanza – aprendizaje permite la implementación de acciones sobre la motivación, llamadas estrategias de autorregulación de la motivación (Pintrich, O., & A., 1991).

De ello la importancia de conocer el grado de motivación de los estudiantes. A continuación, se muestran algunos de los instrumentos usados en el contexto educativo para medir el grado de motivación:

- Pintrich et al. (1991) presentan un instrumento, MSQL (Motivated Strategies for Learning Questionnaire) para evaluar la motivación, así como las estrategias de aprendizaje que utilizan los estudiantes universitarios en un curso, basado en una visión cognitiva de la motivación y las estrategias de aprendizaje. El MSQL consta de dos secciones, la sección de motivación que evalúa las metas, creencias sobre el valor del curso, confianza y ansiedad del estudiante, y la sección de estrategia de aprendizaje, que evalúa el uso de estrategias de aprendizaje cognitivas y metacognitivas del estudiante.
- Learning Motivation in Computer Programming courses Scale (LMCPS) desarrollada por Law, Lee, and Yu (2010) consta de 19 elementos, agrupados en seis sub-factores: actitud y expectativa individual, objetivos desafiantes, dirección clara, recompensa y reconocimiento, castigo, presión social, y competencia.
- Academic Motivation Scale (AMC) se basa en los principios de la teoría de auto-determinación (Self-determination theory), se compone de 28 ítems subdivididos en siete sub-escalas que evalúan tres tipos de motivación intrínseca (motivación intrínseca para saber, lograr cosas y experimentar estimulación), tres tipos de motivación extrínseca motivación (regulación externa, introyectada e identificada) y desmotivación (Vallerand et al., 1992).

- Cuestionario de evaluación motivacional del proceso de aprendizaje (EMPA) desarrollado por (Quevedo-Blasco, Quevedo-Blasco, & Téllez-Trani, 2016), 33 ítems con el fin de proporcionar información para medir la motivación intrínseca, extrínseca y global.

### 3.4 Protocolos verbales

Propuesto inicialmente por Ericsson and Simon (1980), el término de los protocolos verbales, Green (1998) los define como una forma específica para referirse a una fuente de datos generada bajo condiciones individuales especiales, donde el individuo piensa en voz alta, todo el proceso cognitivo que está realizando para resolver un problema en particular o realizar una tarea específica, es decir, pensar en voz alta sobre el procedimiento que está realizando (C. Arévalo & Solano, 2013).

El análisis de protocolos verbales (VPA, Verbal Protocol Analysis) es una metodología o técnica para el registro exacto de información del proceso cognitivo mediante verbalizaciones individuales en voz alta de una persona al resolver un problema o realizar una tarea, a diferencia de otras técnicas que trabajan con datos verbales, VPA se distingue de ellas debido a que los datos verbales generados son a partir de procesos cognitivos (Green, 1998).

Los protocolos verbales registran los procesos cognitivos que una persona lleva a cabo para resolver un problema, solicitándole que piense en voz alta, y registrando estos datos ya sea en grabaciones de audio o video, que posteriormente se transcriben, obteniendo así material que se puede usar para estudiar el comportamiento cognitivo de la persona ante la resolución de problemas o tareas específicas (M. C. Arévalo, 2010).

Partiendo de lo anterior, en este sentido los protocolos verbales pueden aportar información de los procesos cognitivos de un programador experto al estar programando, en base a ello se plantea que es posible emplear esta información para la generación de material instruccional como apoyo en el proceso de enseñanza – aprendizaje de la programación (C. Arévalo & Solano, 2013).

Programación es una habilidad necesaria para la industria moderna, pero a la vez es una actividad cognitiva compleja (Yilmaz & Karaoglan Yilmaz, 2023), que implica la eficiente resolución de problemas y trasladar la solución en expresiones (programas) que la computadora pueda ejecutar de una manera lo más eficientemente posible, por ello de acuerdo con la teoría de la arquitectura cognitiva lo anterior implica aplicar conocimiento declarativo y conocimiento procedimental a través de una participación activa de los estudiantes (Y. T. Lin, Yeh, & Tan, 2022; Souza & Batista, 2019). Sin embargo, el método tradicional más utilizado en la enseñanza es donde los estudiantes reciben la toda la información del profesor (conocimiento declarativo), convirtiéndose en una aprendizaje pasivo y no activo, en el cual el estudiante solo es receptor de la información (Hsu, Horng, & See, 2021). En los cursos introductorios de programación, la mayoría están más enfocados en la enseñanza de la sintaxis y estructura del lenguaje específico, es decir, conocimiento declarativo, dejando un poco de lado el proceso de resolución de problemas, o bien conocimiento procedimental, de ello al finalizar el curso muchos solo son capaces de entender la sintaxis y estructura de un programa, pero no pueden aplicar el conocimiento a la resolución de problemas reales (Y. C. Lin et al., 2022).

Mediante la aplicación de los protocolos verbales, donde el experto o el instructor resuelven un problema pensando en voz alta, los estudiantes pueden observar el proceso cognitivo

que realiza e identificar estrategias, técnicas o métodos que el experto utiliza para resolver un problema específico en un proceso de programación, complementando así el proceso de enseñanza – aprendizaje con el conocimiento procedimental que el estudiante debe aprender (Y. T. Lin et al., 2022).

### 3.5 Evaluación del aprendizaje en programación

En cualquier proceso de enseñanza – aprendizaje la evaluación del aprendizaje es parte fundamental del mismo, ya que permite tanto a profesores y estudiantes saber si los objetivos de aprendizaje se están cumpliendo o en su caso trabajar en mejorar el proceso de enseñanza – aprendizaje (Solarte Pabón & Machuca Villegas, 2019).

Es importante que el estudiante conozca de forma detalla y explicita la forma en que será evaluado, esto puede ayudar a mejorar su desempeño en el proceso de aprendizaje. Por ello, la utilización de rubricas bien detalladas de que, y como se evaluará el conocimiento, esto explicado de la forma más clara posible a los estudiantes con anterioridad al proceso de enseñanza – aprendizaje favorecerían el proceso (Solarte Pabón & Machuca Villegas, 2019).

Un factor importante en la evaluación es el instrumento utilizado para ello. La alineación de la evaluación con los resultados y objetivos de aprendizaje depende del análisis de las características de los instrumentos de evaluación (tipos de preguntas/problemas, dificultad/complejidad), así como el enfoque utilizado para su diseño y aplicación como son, exámenes de laboratorio, exámenes tradicionales en papel, evaluación grupal, exámenes colaborativos, proyectos, portafolio de trabajo, preguntas abiertas y de opción múltiple, entre otros, utilizados y pueden proporcionar una comprensión detallada de cómo se evalúa a los estudiantes (Luxton-Reilly et al., 2018).

Si bien el instrumento de evaluación, así como el diseño de una rubrica detallada de la evaluación son muy importantes en el proceso de enseñanza – aprendizaje, como se describió en la sección anterior, programación implica la adquisición de una habilidad compleja compuesta de conocimiento declarativo y conocimiento procedimental, por ello es necesario en la fase de evaluación de los objetivos de aprendizaje implementar instrumentos para evaluar ambos conocimientos (Y. C. Lin et al., 2022).

Otro factor importante en el proceso de enseñanza – aprendizaje concerniente a la evaluación es la utilización de herramientas para medir habilidades del área de programación como son, creatividad, pensamiento algorítmico, pensamiento crítico, resolución de problemas, entre otras (Yilmaz & Karaoglan Yilmaz, 2023). Estas herramientas pueden ser utilizadas al inicio del curso para identificar el nivel de estas habilidades en los estudiantes, así como al final del mismo para identificar si hubo un efecto positivo en dichas habilidades.

Algunas de estas herramientas mencionadas anteriormente son:

- Computational Thinking Scale (CTS) – Escala de pensamiento computacional: utilizada para conocer el nivel de las habilidades de pensamiento computacional de los estudiantes como son, pensamiento algorítmico, creatividad, cooperatividad, pensamiento crítico y resolución de problemas. Fue desarrollada por Korkmaz, Çakir, and Ozden (2017), utiliza una estructura de calificación tipo Likert de cinco puntos, en la que las puntuaciones más altas indican un mayor desarrollo de las habilidades de pensamiento computacional.

- Computer Programming Self-Efficacy Scale (CPSES) – Escala de autoeficacia en programación: M. J. Tsai, Wang, and Hsu (2019) emplean una estructura de calificación tipo Likert de cinco puntos y consta de 16 ítems categorizados en cinco subfactores: pensamiento lógico, cooperación, algoritmos, control y depuración. Una puntuación más alta en la escala indica una autoeficacia avanzada en programación informática por parte de los estudiantes.

### 3.6 Instrucción diferenciada

Como se mencionó anteriormente, la enseñanza – aprendizaje de la programación conlleva muchos retos que tanto el profesor como el estudiante deben enfrentar. Uno de estos retos del lado de la enseñanza frecuentemente es la diversidad de conocimientos previos de los estudiantes en un mismo grupo, donde el rendimiento de los estudiantes está fuertemente relacionado con la capacidad del profesor de detectar esta diversidad y adaptar su enseñanza a tiempo, de ello el soporte del instructor en tiempo y forma en un entorno diverso de estudiantes con niveles diferentes de conocimientos es esencial en el proceso de aprendizaje (Fonseca, Macedo, & Mendes, 2018).

La instrucción diferenciada (ID) puede definirse como un modelo pedagógico constructivo y proactivo para el desarrollo de material instruccional de acuerdo con las necesidades de los estudiantes atendiendo a diferentes factores como conocimientos previos, estilos de aprendizaje, etc. (Gheysens, Coubergs, Griful-Freixenet, Engels, & Struyven, 2022; Gibbs & McKay, 2021).

La ID está centrada en el alumno, teniendo en cuenta sus necesidades individuales, recopilando datos continuos sobre el progreso de los estudiantes, se puede adaptar su instrucción para que coincida con las necesidades de aprendizaje, no es una estrategia de enseñanza única, sino “una práctica de enseñanza de una variedad de estrategias respaldadas empíricamente en las áreas de planificación, evaluación y seguimiento del currículo, instrucción y organización del aula” (Gibbs & McKay, 2021).

En la literatura se pueden encontrar diversos modelos de instrucción diferenciada (ID) publicados en los últimos años, de los cuales uno de los más utilizados es el modelo ID de Tomlinson (2014), donde afirma que el instructor puede aplicar estrategias para diferenciar la instrucción en términos de contenido (qué es lo que se va a aprender), de proceso (¿Cómo los estudiantes aprenden?), producto (como el aprendizaje de los estudiantes observado y evaluado) y entorno (como el entorno de aprendizaje afecta el proceso).

## CAPÍTULO 4

### MODELO CONCEPTUAL DE DESARROLLO DE LA PROPUESTA

#### 4.1 Modelo conceptual y etapas de desarrollo

Propuesta por Sweller et al. (1998), la teoría de la carga cognitiva destaca por haber inspirado numerosos diseños instruccionales fundamentados en el funcionamiento de la arquitectura cognitiva humana. Esta teoría ha permitido identificar diversos efectos instruccionales que, en su mayoría, han demostrado empíricamente su eficacia, según lo documenta (Sweller, 2016). Su contribución al desarrollo de metodologías educativas basadas en evidencia la posiciona como un referente significativo en el campo del diseño instruccional.

De igual forma se han reportado algunas limitaciones en su aplicación, es por ello que se decide incorporar otras teorías cognitivas de aprendizaje como el diseño instruccional de cuatro componentes 4C/ID, modelo Diez pasos para aprendizaje complejo, protocolos verbales y teorías de motivación, ya que ninguna de esta es exclusiva, mas al contrario se complementan, robusteciendo el desarrollo del modelo propuesto. La Figura 4.1 muestra la idea general de la selección e integración de los fundamentos teóricos propuestos.

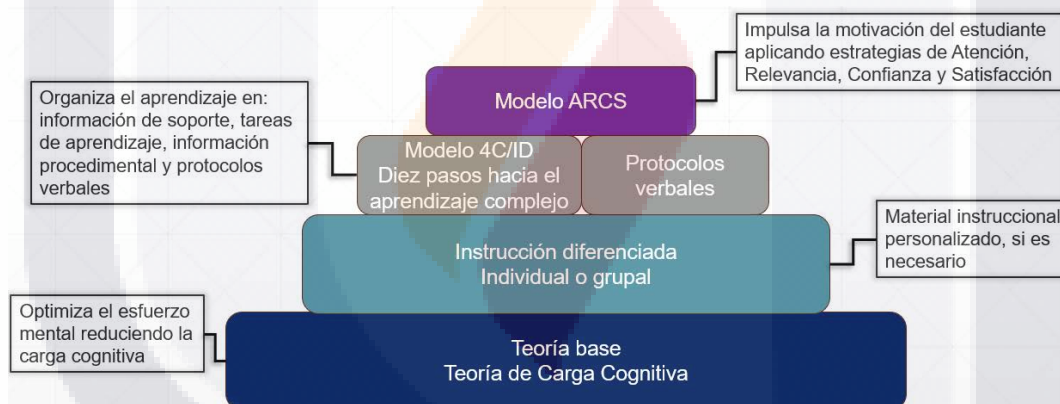


Figura 4.1: Integración de los fundamentos teóricos en el modelo.

Para el desarrollo del modelo propuesto se plantean las siguientes etapas (de forma iterativa), como se puede observar en la Figura 4.2:

1. Definición y alcance: en esta etapa se definen las teorías cognitivas de aprendizaje base para el desarrollo del modelo, los métodos y herramientas de investigación y diseño, así como enfoque de la misma.
2. Investigación, análisis y diseño:
  - a. **Teoría de carga cognitiva:**
    - i. Identificación de los efectos instruccionales a utilizar.
    - ii. Revisión y análisis de estudios relacionados con dichos efectos y sus resultados.
  - b. **Diseño instruccional de cuatro componentes 4C/ID:**
    - i. Identificación de los conceptos a integrar.



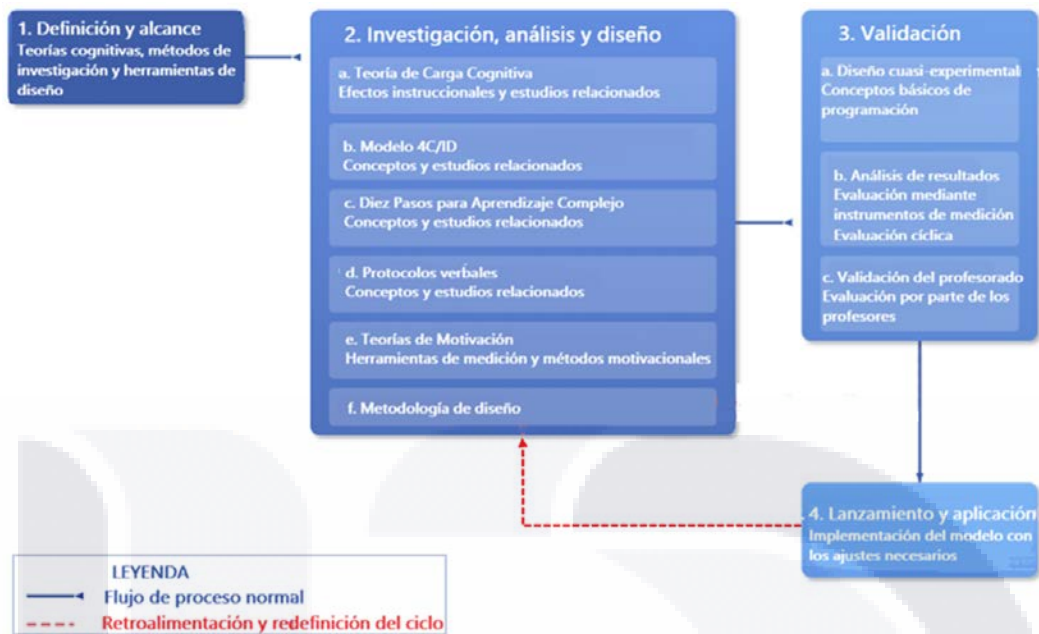


Figura 4.2: Etapas para el desarrollo del modelo propuesto

- ii. Revisión y análisis de estudios relacionados.
  - c. **Diez pasos para aprendizaje complejo:**
    - i. Identificación de los conceptos a integrar.
    - ii. Revisión y análisis de estudios relacionados.
  - d. **Protocolos verbales:**
    - i. Identificación de los conceptos a integrar.
    - ii. Revisión y análisis de estudios relacionados.
  - e. **Teorías de motivación:**
    - i. Herramientas de medición de la motivación.
    - ii. Métodos y herramientas motivacionales.
  - f. **Diseño de la metodología**
  - g. **Protocolos verbales**
3. Validación
- a. Diseño cuasi-experimental para la aplicación del modelo en un curso de programación básica.
  - b. Análisis de resultados, métodos, estrategias y herramientas aplicadas, por medio de los diferentes instrumentos de medición para cada caso, de acuerdo con este análisis, se pasa a la siguiente fase o regresa al inicio del ciclo para la re-definición del modelo.
  - c. Validación por parte de profesores del lado de la enseñanza.
4. Liberación y aplicación
- a. Una vez terminada la etapa de validación con los ajustes requeridos y resultados esperados, se libera el modelo para su aplicación.

## 4.2 Definición de los efectos de la teoría de carga cognitiva a aplicar

Atendiendo los resultados obtenidos en la revisión de literatura respecto a la aplicación de los efectos de la TCC en el proceso de enseñanza - aprendizaje de la programación, se decide la utilización de los siguientes efectos de la teoría para el desarrollo del presente trabajo como se muestra en la tabla 4.1.

| Efecto TCC                                  | Descripción                                                                                                                                           | Justificación                                                                                                                           |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Solución libre (Goal-free)                  | Utilización de ejercicios con diferentes soluciones, liberando la carga cognitiva que se puede generar por el enfoque a una sola solución.            | Debido a su naturaleza, puede aportar a la parte motivacional.                                                                          |
| Problema resuelto (Worked Example)          | Ejercicios tradicionales explicados paso a paso.                                                                                                      | Es el efecto más utilizado y reportado en los estudios.                                                                                 |
| Problema por completar (Completion Problem) | Ejercicios con partes parcialmente resueltas, donde el estudiante debe completar el resto.                                                            | Es uno de los más citados, cubre las limitaciones del ejemplo resuelto.                                                                 |
| Atención dividida (Split Attention)         | Fuentes de información relacionadas dispersas física o temporalmente en el proceso de enseñanza.                                                      | Integrar múltiples fuentes de información, como imágenes y texto, en una sola para no dividir la atención y reducir la carga cognitiva. |
| Modalidad (Modality)                        | Fuentes de información relacionadas presentadas en forma visual y auditiva. Se presupone que el procesamiento de audio se hace por un canal separado. | La utilización de recursos visuales y auditivos puede aportar al tema de motivación.                                                    |
| Auto-explicación (Self-Explanation)         | Incitar la reflexión sobre la información solicitando al estudiante explicaciones de la misma.                                                        | Es uno de los efectos reportados para el reforzamiento del conocimiento y creación de esquemas.                                         |
| Redundancia (Redundancy)                    | Misma información presentada en dos o más formas, es decir de forma repetitiva.                                                                       | Reducir múltiples fuentes de información a una sola contribuye a reducir la carga cognitiva.                                            |
| Interactividad (Interactivity)              | La complejidad del material, que tiene muchos elementos interactuando entre sí, puede sobrecargar la memoria de trabajo dificultando el aprendizaje.  | Modificar el material disminuyendo la interactividad de elementos procesando los elementos de manera individual o aislada.              |
| Variabilidad (Variability)                  | Variar los métodos, tareas, actividades, ejercicios, etc.                                                                                             | La utilización de recursos variados puede aportar al tema de motivación.                                                                |
| Imaginación (Imagination)                   | Incitar al alumno a generar vínculos de información a partir de lo que ya aprendió con temas nuevos.                                                  | Es uno de los efectos para el reforzamiento del conocimiento y creación de esquemas.                                                    |
| Elementos Aislados (Isolated Elements)      | Descomponer actividades o información compleja en más pequeñas de menor complejidad.                                                                  | Enfocado totalmente a la reducción de la carga cognitiva en información con alto nivel de complejidad.                                  |
| Reversión Experiencia (Expertise Reversal)  | No utilizar efectos de la teoría exclusivos para novatos en estudiantes más avanzados.                                                                | Debido a su naturaleza puede aportar al tema del entorno del grupo.                                                                     |

| Efecto TCC                                               | Descripción                                                                                                                                                        | Justificación                                                                                                                                                                   |
|----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Desvanecimiento de Guía (Guidance-Fading)                | Conforme la experiencia del estudiante crece, el soporte del instructor disminuye.                                                                                 | Debido a su naturaleza ayuda al alumno a aprender a aprender.                                                                                                                   |
| Memoria Colectiva de Trabajo (Collective Working Memory) | Tareas colaborativas o en equipo, aumentando los recursos cognitivos.                                                                                              | Debido a su naturaleza puede aportar al tema de motivación y entorno del grupo.                                                                                                 |
| Información Transitoria (Transient Information)          | La información desaparece antes de que el alumno tenga tiempo de procesarla, como presentaciones orales o mediante el uso de videos o animaciones instruccionales. | Información extensa, compleja y con alto nivel de interactividad no debe ser presentada en forma de texto hablado, videos o animaciones.                                        |
| Movimiento Humano (Human Movement)                       | Reemplazar visualizaciones estáticas o poco realistas con visualizaciones que muestren movimientos humanos.                                                        | El efecto de movimiento humano plantea que es mejor la utilización de animaciones en lugar de información estática cuando las tareas cognitivas a enseñar implican movimientos. |
| Auto gestión (Self-Management)                           | Instruir al estudiante para que aplique los efectos de la teoría a material educativo mal diseñado y así gestione su propio aprendizaje.                           | Los estudiantes pueden ser instruidos para utilizar activamente estrategias para gestionar la carga en su memoria de trabajo durante el aprendizaje.                            |

Tabla 4.1: Efectos de la Teoría de la Carga Cognitiva (TCC)

### 4.3 Modelo de diseño instruccional mapa de ruta CLT4BP

Con lo que se refiere anteriormente se propone el siguiente modelo de diseño instruccional “mapa de ruta CLT4BP (Cognitive Load Theory For Basic Programming)” representado en la Figura 4.2, basado en la teoría de carga cognitiva como apoyo en la enseñanza – aprendizaje de la programación, en el cual se plantean 3 fases:

1. Análisis
2. Diseño y Desarrollo
3. Aplicación y Evaluación

Para la selección y aplicación de los efectos de la TCC se sugiere la siguiente clasificación para la aplicación del modelo como se muestra a continuación:

1. **Para nuevos conocimientos:**
  - Ejemplo resuelto
  - Problema por completar
  - Modalidad
  - Variabilidad
  - Elementos aislados
  - Redundancia
  - Atención dividida

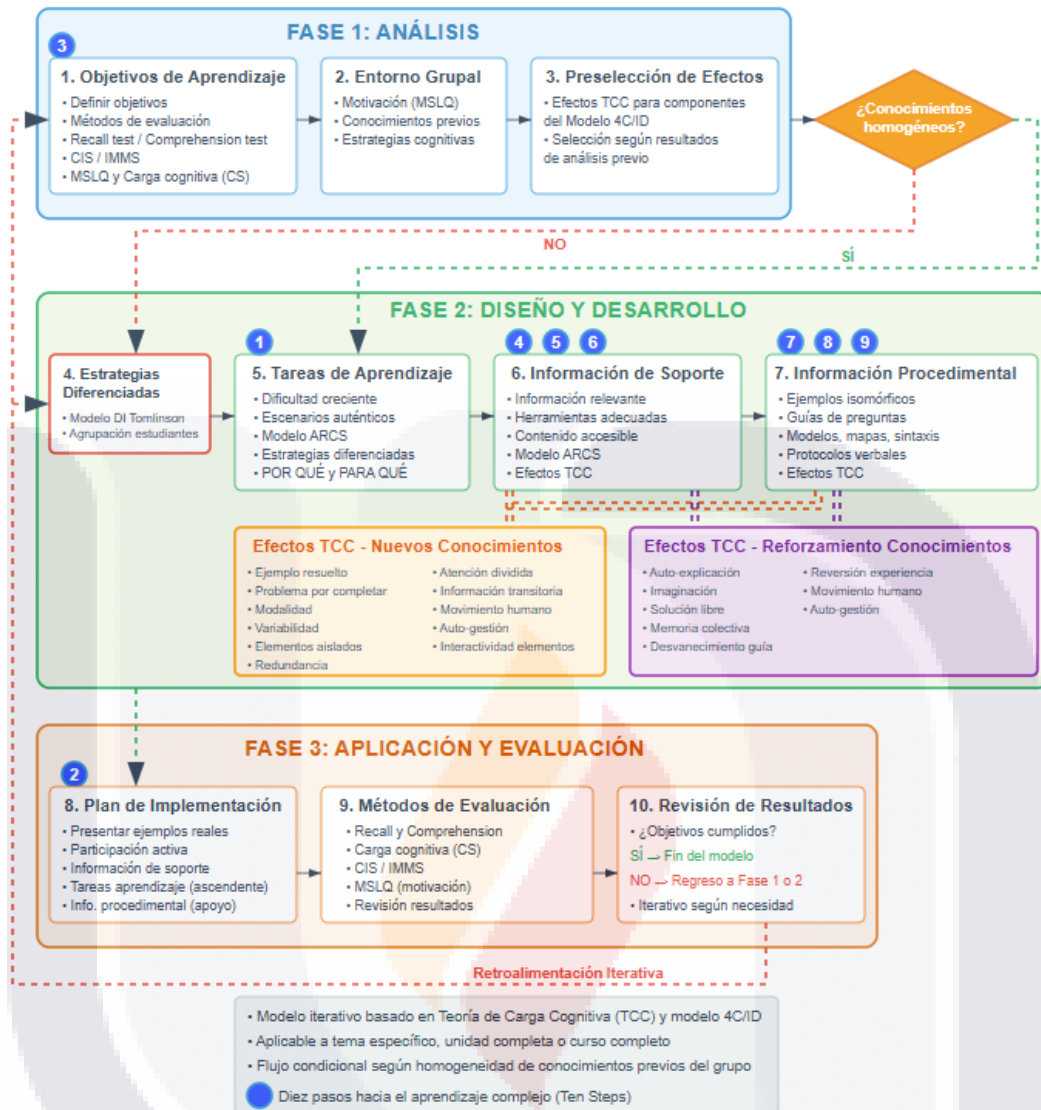


Figura 4.3: Propuesta modelo de diseño instruccional “mapa de ruta CLT4BP”.

- Información transitoria
  - Movimiento humano
  - Autogestión
  - Interactividad de los elementos
2. **Reforzamiento de conocimientos previos:**
- Auto-explicación
  - Imaginación
  - Solución libre
  - Memoria colectiva de trabajo
  - Desvanecimiento de guía
  - Reversión de la experiencia
  - Movimiento humano

- Autogestión

De igual forma tomando como base el modelo propuesto y la clasificación de los efectos de la TCC, se realiza la adhesión de los componentes del modelo instruccional 4C/ID, ver tabla 4.2:

| <b>Componentes: Información de soporte e Información procedimental</b> | <b>Componente: Tareas de aprendizaje</b> |
|------------------------------------------------------------------------|------------------------------------------|
| Ejemplo resuelto                                                       | Problema por completar                   |
| Problema por completar                                                 | Auto-explicación                         |
| Atención dividida                                                      | Imaginación                              |
| Modalidad                                                              | Solución libre                           |
| Redundancia                                                            | Memoria colectiva de trabajo             |
| Elementos aislados                                                     | Variabilidad                             |
| Reversión de la experiencia                                            | Desvanecimiento de guía                  |
| Información transitoria                                                | Reversión de la experiencia              |
| Movimiento humano                                                      | Autogestión                              |

Tabla 4.2: Clasificación efectos de la TCC en el modelo 4C/ID.



## CAPÍTULO 5

### IMPLEMENTACIÓN DEL MODELO

La implementación del modelo propuesto puede realizarse ya sea a un tema específico, como por ejemplo estructuras de datos, varios temas dentro de una unidad o bien al curso completo. El modelo como su nombre lo especifica “mapa de ruta”, se refiere a que se debe seguir una secuencia establecida similar a los diagramas de flujo. El modelo como ya se mencionó anteriormente consta de 3 fases, y está diseñado para ser iterativo según se requiera, es decir, al terminar la tercera fase si no se obtienen los resultados esperados podemos regresar a la primera o segunda fase del modelo para volver a implementar con los ajustes necesarios. A pesar de que el modelo fue diseñado y desarrollado enfocándose específicamente como apoyo en la enseñanza – aprendizaje de la programación, este se podría adaptar a casi cualquier área de estudio diferente. En este caso lo descrito en la presente sección de implementación del modelo se centra en programación.

#### 5.1 Fase de análisis

La primera fase del modelo, análisis, comprende la definición de los conocimientos a enseñar, es decir, los objetivos de aprendizaje, así como la forma que se evaluarán los estudiantes para medir estos objetivos planteados. Además, conocer a los estudiantes en aspectos como conocimientos previos, esto para plantear el utilizar o no estrategias de enseñanza – aprendizaje diferenciadas. También identificar estrategias cognitivas y metacognitivas, y grado de motivación hacia el curso. Finalmente, en base a los resultados del análisis obtenido en lo mencionado anteriormente realizar una preselección de efectos de la teoría de carga cognitiva a aplicar en el proceso de enseñanza – aprendizaje.

Todo lo descrito anteriormente lo registramos utilizando formatos libres que decida el instructor o bien aquí se presentan propuestas para ello:

1. **Formato objetivos de aprendizaje:** capturar cada objetivo de aprendizaje, tema a completar para dicho objetivo, tiempo asignado para el proceso de enseñanza – aprendizaje y fuentes de consulta adicionales de cada tema (ver tabla 5.1).

| Objetivos de aprendizaje | Temas a completar | Duración | Fuentes adicionales de consulta |
|--------------------------|-------------------|----------|---------------------------------|
|                          |                   |          |                                 |
|                          |                   |          |                                 |

Tabla 5.1: Planificación de contenidos y objetivos

2. **Formato rubrica de evaluación:** por cada estudiante se captura los resultados de los test aplicados (recall y comprehension), se captura la calificación obtenida de ambos test para poder identificar o no diferencias estadísticas significativas con los test de conocimientos previos aplicados así como con los test del grupo de control (ver tabla 5.2).

| ID | Recall test - final | Comprensión test - final |
|----|---------------------|--------------------------|
|    | Calificación        | Calificación             |
|    |                     |                          |

Tabla 5.2: Registro de resultados de las pruebas finales de desempeño

### 3. Formatos entorno de grupo:

- a. Motivación: en este formato se capturan los resultados de cada componente motivacional descrito en el cuestionario MSLQ de acuerdo a la escala descrita en el manual, así como el promedio total. Lo anterior además de informar al instructor el nivel de motivación, también con la finalidad de proporcionar la retroalimentación a los estudiantes (ver tabla 5.3).

| ID | Orientación Intrínseca | Orientación Extrínseca | Valor de la tarea | Autoeficacia | Ansiedad | Promedio |
|----|------------------------|------------------------|-------------------|--------------|----------|----------|
|    |                        |                        |                   |              |          |          |

Tabla 5.3: Registro de indicadores de motivación y autoeficacia

- b. Conocimientos previos: por cada estudiante se captura los resultados de los test aplicados (recall y comprensión), se captura la calificación de ambos test por cada estudiante para identificar si existen diferencias significativas entre los conocimientos previos de los estudiantes y así aplicar o no estrategias diferenciadas de enseñanza – aprendizaje.
- c. Estrategias cognitivas/metacognitivas: en este formato se capturan los resultados de cada componente cognitivo y metacognitivo del cuestionario MSLQ de acuerdo a la escala descrita en el manual, por cada estudiante.

| ID | Cognitivas |       |      | Metacognitivas | Manejo de recursos |         |         |         |
|----|------------|-------|------|----------------|--------------------|---------|---------|---------|
|    | Rep.       | Elab. | Org. | Cont_Comp      | Ges_TE             | Reg_Esf | Bús_Ayu | Ent_Apr |
|    |            |       |      |                |                    |         |         |         |

Tabla 5.4: Evaluación de estrategias cognitivas, metacognitivas y manejo de recursos

- d. Formato resumen general entorno de grupo: este formato capturamos los datos por grupo, es decir el promedio en motivación, la cantidad de estudiantes con conocimiento previo o no, de igual forma la cantidad de estudiantes en cada componente de las estrategias cognitivas/metacognitivas.

|        | Motivación | Previos |    | Cognitivas |   |   | Metacognitivas |    | Recursos |   |   |   |
|--------|------------|---------|----|------------|---|---|----------------|----|----------|---|---|---|
|        |            | Si      | No | 1          | 2 | 3 | Si             | No | 1        | 2 | 3 | 4 |
| Conteo |            |         |    |            |   |   |                |    |          |   |   |   |

Tabla 5.5: Resumen estadístico de variables de aprendizaje y motivación

4. **Formato Pre-selección de efectos a aplicar:** registramos los temas de aprendizaje a impartir, así como el efecto(s) a aplicar a cada tema y las herramientas/estrategias adicionales para su aplicación, por ejemplo, presentaciones, video-tutoriales, podcasts, etc.

| Tema | Efecto(s) a aplicar | Herramientas/Estrategias de aplicación |
|------|---------------------|----------------------------------------|
|      |                     |                                        |

Tabla 5.6: Plan de aplicación de efectos de la carga cognitiva por tema.

| MARCO DE TRABAJO: FASE DE ANÁLISIS                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. OBJETIVOS DE APRENDIZAJE: DEFINICIÓN Y EVALUACIÓN |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| a. Objetivos de Aprendizaje                          | Definir y establecer los temas y objetivos de aprendizaje a alcanzar.                                                                                                                                                                                                                                                                                                                                                                                          |
| b. Métodos de Evaluación                             | <p>Establecer métodos o instrumentos para evaluar los objetivos de aprendizaje establecidos:</p> <p><b>Recall test</b> - Evalúa el conocimiento teórico básico. El instructor define los conocimientos teóricos a evaluar según los objetivos de aprendizaje.</p> <p><b>Comprehension test</b> - Mide la transferencia de conocimientos adquiridos para resolver problemas reales. El instructor define problemas prácticos.</p>                               |
| c. Evaluación de la percepción de los estudiantes    | <p>Instrumentos de evaluación para medir la percepción de los estudiantes sobre los métodos de enseñanza y los materiales de instrucción:</p> <p><b>Course Interest Survey (CIS)</b> - Diseñado para medir la percepción de los estudiantes sobre cómo el instructor imparte el conocimiento (Anexo 4).</p> <p><b>Instructional Materials Motivation Scale (IMMS)</b> - Se centra en medir la reacción de los estudiantes al material didáctico (Anexo 5).</p> |
|                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 2. ENTORNO GRUPAL: MOTIVACIÓN, PREVIOS Y ESTRATEGIAS |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| a. Motivación                                        | <p>Al inicio de la aplicación del modelo, aplicar el instrumento MSLQ para medir la motivación de los estudiantes:</p> <p><b>MSLQ (Motivated Strategies for Learning Questionnaire)</b> - Utilice la sección de motivación. Se recomienda retroalimentación para fomentar el interés y la honestidad en sus respuestas (Anexo 1).</p>                                                                                                                          |
| b. Conocimiento previo                               | <p>Conocer el nivel de conocimientos previos de los estudiantes y aplicar estrategias de enseñanza diferenciadas si es necesario (siguiendo solo dos caminos: Sí/No) y diseñar materiales apropiados:</p> <p><b>Recall test</b> – evaluación inicial</p> <p><b>Comprehension test</b> – evaluación final</p>                                                                                                                                                   |

Tabla 5.7: Fase de análisis del marco de trabajo.

## 5.2 Fase diseño y desarrollo

La fase de diseño y desarrollo es donde el instructor crea el material instruccional a utilizar en el proceso de enseñanza – aprendizaje a partir de la información obtenida en la fase de análisis. De acuerdo con lo anterior, los conocimientos previos del grupo de estudiantes pueden ser homogéneos, es decir conocimientos previos similares, en este caso el flujo del modelo va directamente al desarrollo del material instruccional comenzando con la actividad “Tareas de aprendizaje” y continuamos el flujo del modelo. NOTA: Sin embargo, si el instructor desea aplicar estrategias diferenciadas a pesar de tener un grupo homogéneo en conocimientos previos puede hacerlo, esto no afecta el proceso. Si el grupo presenta una heterogeneidad significativa en los conocimientos previos, es necesario realizar una agrupación/separación de acuerdo con estas diferencias en dos grupos. Lo anterior para la aplicación de estrategias de enseñanza diferenciadas alineadas con los conocimientos previos de los estudiantes, creando dos caminos posibles a seguir en el modelo, el primero para el grupo de estudiantes sin conocimientos previos, el segundo para el grupo de estudiantes con conocimientos previos. De ello se comienza el flujo del modelo mapa de ruta en la actividad 1 “Estrategias diferenciadas”. La tabla 5.9 muestra el marco de trabajo para la fase de Diseño y desarrollo.

Se realiza el registro de las estrategias a aplicar en el siguiente formato propuesto, o bien, el instructor es libre de utilizar otro formato para el registro. La información a capturar en el formato comprende, primero el tema de aprendizaje, colocando en la línea del componente del modelo ID a aplicar (puede ser solo uno, dos, tres o los cuatro elementos), después a que componente del modelo 4C/ID se va aplicar, de igual forma que el anterior puede ser a uno, dos o los tres componentes que se utilizan en el modelo, por ultimo se detalla la estrategia diferenciada a utilizar capturandola en la columna correspondiente para el grupo de estudiantes de acuerdo al nivel de conocimientos previos.

| Elemento modelo ID | Tema(s) | Componente 4C/ID | Conocimientos previos |    |
|--------------------|---------|------------------|-----------------------|----|
|                    |         |                  | SI                    | NO |
| Contenido          |         |                  |                       |    |
| Proceso            |         |                  |                       |    |
| Producto           |         |                  |                       |    |
| Entorno            |         |                  |                       |    |

Tabla 5.8: Matriz de planificación: Instrucción Diferenciada y Modelo 4C/ID

| FASE DISEÑO Y DESARROLLO: MARCO DE TRABAJO                            |                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1. ESTRATEGIAS DIFERENCIADAS</b>                                   |                                                                                                                                                                                                                                                                                                                  |
| • Tomlinson's DI model                                                | Selección e implementación de estrategias para los elementos descritos por el modelo de Instrucción Diferenciada (DI) de Tomlinson (Anexo 6).                                                                                                                                                                    |
| <b>2. TAREAS DE APRENDIZAJE</b>                                       |                                                                                                                                                                                                                                                                                                                  |
| a. Tareas de aprendizaje                                              | <ul style="list-style-type: none"> <li>• Diseñe tareas de aprendizaje con dificultad creciente.</li> <li>• Crear escenarios auténticos de resolución de problemas.</li> <li>• Establecer el POR QUÉ y PARA QUÉ.</li> </ul>                                                                                       |
| b. Motivación<br>• ARCS Model                                         | Aplicar el modelo ARCS a las tareas de aprendizaje. La tabla 5.11 muestra ejemplos de aplicación del modelo.                                                                                                                                                                                                     |
| c. Estrategias diferenciadas                                          | Aplicar estrategias diferencias según sea el caso (Anexo 6).                                                                                                                                                                                                                                                     |
| <b>3. INFORMACIÓN DE SOPORTE</b>                                      |                                                                                                                                                                                                                                                                                                                  |
| a. Información de soporte                                             | <ul style="list-style-type: none"> <li>• Seleccione la información relevante.</li> <li>• Elija las herramientas de entrega adecuadas.</li> <li>• Diseñar contenido accesible.</li> </ul>                                                                                                                         |
| b. Motivación<br>• ARCS Model                                         | Aplicar el modelo ARCS a la información de soporte. La tabla 5.12 muestra ejemplos de aplicación del modelo.                                                                                                                                                                                                     |
| c. Estrategias diferenciadas                                          | Aplicar estrategias diferenciadas según sea el caso (Anexo 6).                                                                                                                                                                                                                                                   |
| d. Implementar efectos de la TCC                                      | Aplicar los efectos de la Teoría de la Carga Cognitiva (TCC) al material instruccional, atendiendo las recomendaciones del capítulo 6.                                                                                                                                                                           |
| <b>4. INFORMACIÓN PROCEDIMENTAL</b>                                   |                                                                                                                                                                                                                                                                                                                  |
| a. Ejemplos isomórficos                                               | <ul style="list-style-type: none"> <li>• Diseñar materiales de orientación adicionales.</li> <li>• Crear guías de preguntas de rutina.</li> <li>• Desarrollar modelos, mapas, tablas de sintaxis.</li> </ul>                                                                                                     |
| b. Implementar efectos de la TCC                                      | Aplicar los efectos de la TCC al material instruccional, atendiendo las recomendaciones del capítulo 6.                                                                                                                                                                                                          |
| c. Estrategias diferenciadas                                          | Aplicar estrategias diferenciadas según sea el caso (Anexo 6).                                                                                                                                                                                                                                                   |
| d. Protocolos verbales<br>• Instructor<br>• Desarrollador profesional | <ul style="list-style-type: none"> <li>• Desarrollar soluciones de ejemplo isomórficos.</li> <li>• Grabar las explicaciones del instructor (CÓMO y PORQUÉ).</li> <li>• Centrarse en el proceso y el razonamiento vía video o audio.</li> <li>• Garantizar la claridad y la alineación con las tareas.</li> </ul> |

Tabla 5.9: Fase de diseño y desarrollo del marco de trabajo basado en 4C/ID y TCC



### 5.3 Aplicación y evaluación

La tabla 5.10 muestra el marco de trabajo para la fase de Aplicación y Evaluación.

| FASE APLICACIÓN Y EVALUACIÓN: MARCO DE TRABAJO |                                                                                                                                                                                                                                                                               |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1. PLAN DE IMPLEMENTACIÓN</b>               |                                                                                                                                                                                                                                                                               |
| a. Información de soporte                      | Se presenta al inicio, antes de las tareas de aprendizaje. Debe permanecer disponible durante todo el proceso.<br><br><b>Estrategia:</b> Antes de presentarla, muestre un ejemplo real y fomente la participación con preguntas reflexivas (ej. "¿Cómo crees que funciona?"). |
| b. Tareas de aprendizaje                       | Presentación en orden ascendente (de menor a mayor complejidad). El apoyo del instructor disminuye progresivamente (desvanecimiento de guía).                                                                                                                                 |
| c. Información procedimental                   | Se proporciona de forma simultánea a las tareas de aprendizaje como apoyo "justo a tiempo" para la resolución de problemas.                                                                                                                                                   |
| <b>2. MÉTODOS DE EVALUACIÓN</b>                |                                                                                                                                                                                                                                                                               |
| a. Medición de objetivos de desempeño          | Evaluación final mediante los instrumentos diseñados en la fase de análisis ( <b>Recall</b> y <b>Comprehension tests</b> ).                                                                                                                                                   |
| b. Medición de Carga Cognitiva                 | Aplicación de la <i>Encuesta sobre el Componente de Carga Cognitiva de CS</i> (Morrison et al., 2014) para validar la eficiencia mental del material.                                                                                                                         |
| c. Evaluación adicional                        | <ul style="list-style-type: none"> <li>• <b>CIS y IMMS:</b> Para evaluar la percepción del método y el material didáctico.</li> <li>• <b>MSLQ (Motivación):</b> Para contrastar la motivación final/inicial de los estudiantes.</li> </ul>                                    |
| d. Revisión de resultados                      | <b>Toma de decisiones:</b> Si se cumplen los objetivos, el ciclo finaliza. De lo contrario, se debe iterar regresando a la fase de <b>Análisis o Diseño</b> según los hallazgos.                                                                                              |

Tabla 5.10: Fase de aplicación y evaluación final del marco de trabajo

| Componente          | Subcomponente             | Ejemplo 1                                                                                                     | Ejemplo 2                                                                                   |
|---------------------|---------------------------|---------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <b>Atención</b>     | Entusiasmo perceptivo     | <b>Variables y funciones:</b> Crea un programa de tienda con descuentos usando colores, emojis o animaciones. | <b>If-else:</b> Programa que determine si un número es par/impar mostrando una "P.º una I". |
|                     | Entusiasmo por indagación | Pregunta: ¿Cómo harías un sistema de descuentos por volumen? (Resolverlo como un plus).                       | Pregunta: ¿Cómo podrías resolver la paridad de un número sin usar la sentencia if-else?     |
|                     | Variabilidad              | Permitir que elijan su propio contexto (tienda de videojuegos, cafetería, etc.).                              | Incluir diferentes formas de resolución (uso de operador %, operadores ternarios, etc.).    |
| <b>Relevancia</b>   | Orientación a objetivos   | Explicar que estos conceptos son la base de las tiendas en línea actuales.                                    | Explicar que este concepto es fundamental en sistemas de validación de datos.               |
|                     | Concordancia de motivos   | Mostrar fragmentos de código real usado en plataformas de e-commerce.                                         | Mostrar cómo esta lógica se usa en videojuegos para alternar turnos de jugadores.           |
|                     | Familiaridad              | Relacionar el ejercicio con las compras cotidianas que realizan los alumnos.                                  | Relacionar con semáforos (verde para par, rojo para impar).                                 |
| <b>Confianza</b>    | Requisitos de aprendizaje | Presentar un ejemplo muy base y luego incrementar la dificultad gradualmente.                                 | Proveer ejercicios guiados (paso a paso) antes de la práctica autónoma.                     |
|                     | Oportunidades de éxito    | Brindar retroalimentación automática inmediata tras cada ejecución del código.                                | Incluir pequeños retos o checkpoints para validar la comprensión progresiva.                |
|                     | Control personal          | Permitir que ajusten parámetros iniciales para ver cómo reacciona el programa.                                | Incentivar la modificación del código para experimentar con nuevas condiciones.             |
| <b>Satisfacción</b> | Refuerzo intrínseco       | Probar el programa con valores extremos y ver resultados funcionales en tiempo real.                          | Permitir que los estudiantes exploren múltiples valores de entrada libremente.              |
|                     | Recompensa extrínseca     | Mostrar mensajes de éxito ("¡Excelente!") o insignias digitales al finalizar.                                 | Implementar un contador de aciertos para visualizar el progreso personal.                   |
|                     | Equidad                   | Pruebas automáticas que validen la lógica sin penalizar los intentos fallidos iniciales.                      | Mensajes de error explicativos que transformen el fallo en una oportunidad de mejora.       |

Tabla 5.11: Ejemplos de aplicación modelo ARCS al componente Tareas de Aprendizaje 4C/ID en programación básica.

| Componente          | Subcomponente             | Ejemplo 1                                                                                                           | Ejemplo 2                                                                                    |
|---------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| <b>Atención</b>     | Entusiasmo perceptivo     | <b>Concepto de "bucles"</b> : Muestra un video de un programa que dibuja formas repetidas.                          | <b>If-else</b> : Usar diagramas de flujo dinámicos para explicar la lógica condicional.      |
|                     | Entusiasmo por indagación | Pregunta: ¿Cómo harían esto sin escribir el mismo código una y otra vez? Debate sobre automatización y animaciones. | Permitir a los estudiantes hacer preguntas y recibir respuestas automatizadas en un chatbot. |
|                     | Variabilidad              | Juego en línea donde los estudiantes usan bucles para resolver puzzles.                                             | Incluir explicaciones en múltiples formatos: video, texto y código ejecutable.               |
| <b>Relevancia</b>   | Orientación a objetivos   | Mostrar cómo los bucles son la base para crear juegos y sitios web interactivos.                                    | Explicar que los condicionales son esenciales en IA y desarrollo de videojuegos.             |
|                     | Concordancia de motivos   | Explicar cómo los bucles ahorran tiempo y esfuerzo al escribir código profesional.                                  | Mostrar cómo estos conceptos ayudan a resolver problemas lógicos en la vida real.            |
|                     | Familiaridad              | Relacionar con acciones de la vida real que se repiten constantemente.                                              | Relacionar con decisiones cotidianas (ej: "Si llueve, llevo paraguas").                      |
| <b>Confianza</b>    | Requisitos de aprendizaje | Dividir la explicación en pasos pequeños con ejemplos en diferentes lenguajes.                                      | Ofrecer niveles de ayuda escalonados: pistas, ejemplos y luego la solución.                  |
|                     | Oportunidades de éxito    | Metas realistas: crear un programa que imprima una lista de números.                                                | Agregar mini-retos que incluyan pistas opcionales para no frustrar al alumno.                |
|                     | Control personal          | Animar a experimentar con diferentes tipos de bucles y pedir ayuda sin miedo.                                       | Permitir a los estudiantes elegir qué nivel de ayuda desean usar en cada paso.               |
| <b>Satisfacción</b> | Refuerzo intrínseco       | Permitir que creen sus propios programas para resolver problemas o crear animaciones.                               | Dejar que ejecuten ejemplos en vivo para ver resultados inmediatos de su código.             |
|                     | Recompensa extrínseca     | Ofrecer puntos extra o reconocimientos por completar proyectos creativos.                                           | Mostrar mensajes de progreso y "checkmarks" tras cada ejercicio resuelto.                    |
|                     | Equidad                   | Asegurar acceso equitativo a recursos y apoyo para todos los estudiantes.                                           | Ofrecer múltiples intentos sin penalización para fomentar la exploración libre.              |

Tabla 5.12: Ejemplos de aplicación modelo ARCS al componente Información de Soporte 4C/ID en programación básica.

## 5.4 Estudio piloto

La implementación del modelo instruccional “mapa de ruta CLT4BP” se llevó a cabo mediante un estudio cuasi-experimental (ver Figura 5.1) con un grupo de control y un grupo experimental de estudiantes universitarios de la Universidad Autónoma de Aguascalientes en un curso introductorio de programación estructurada en lenguaje C++. Los temas impartidos fueron estructuras anidadas, punteros, y memoria dinámica.

|                                                                                                                                 | Fecha/periodo de aplicación                                                                       | Grupo Experimental                                                                                                                                                                                                                                                                                                                           | Grupo Control                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Fase 1 :<br/>Entorno de grupo:<br/>Motivación, estrategias cognitivas y metacognitivas, y nivel de conocimientos previos</b> | Antes del primer parcial (viernes 7 de marzo)                                                     | <ul style="list-style-type: none"> <li>• Aplicar el cuestionario MSLQ (sección motivación).</li> <li>• Aplicar el cuestionario MSLQ (sección estrategias cognitivas/metacognitivas).</li> </ul> Informando a los estudiantes que se les dará retroalimentación del mismo.                                                                    | <ul style="list-style-type: none"> <li>• Aplicar el cuestionario MSLQ (sección motivación).</li> <li>• Aplicar el cuestionario MSLQ (sección estrategias cognitivas/metacognitivas).</li> </ul> Informando a los estudiantes que se les dará retroalimentación del mismo.                             |
|                                                                                                                                 | Resultados examen diagnóstico aplicado al inicio del semestre                                     | Medir el nivel de conocimientos previos de los estudiantes, aplicando los siguientes instrumentos: <ul style="list-style-type: none"> <li>• Recall test (conocimientos declarativos) – evaluación inicial</li> <li>• Comprensión test (conocimientos procedimentales) – evaluación inicial</li> </ul>                                        | Medir el nivel de conocimientos previos de los estudiantes, aplicando los siguientes instrumentos: <ul style="list-style-type: none"> <li>• Recall test (conocimientos declarativos) – evaluación inicial</li> <li>• Comprensión test (conocimientos procedimentales) – evaluación inicial</li> </ul> |
| <b>Fase 2 :<br/>Diseño y desarrollo del material instruccional</b>                                                              | A partir de los resultados de la fase anterior se comienza con el diseño (17 marzo a 16 de abril) | Con base en los resultados obtenidos de los cuestionarios aplicados en la fase anterior, se diseña y desarrolla el material instruccional siguiendo los lineamientos descritos en el capítulo 5: Implementación del modelo sección 5.2.                                                                                                      | Se diseña el material instruccional de forma tradicional.                                                                                                                                                                                                                                             |
| <b>Fase 3:<br/>Tratamiento y aprendizaje</b>                                                                                    | Periodo de tratamiento después del 2do parcial (12-23 mayo)                                       | <b>Presentar un ejemplo real</b> , por ejemplo, una aplicación que se relacione con los conceptos a ver, y mostrar el funcionamiento específico en esos temas. Sobre ello fomentar la participación preguntando ¿Cómo creen que funcione la aplicación?, etc.                                                                                | Clases tradicionales sobre el tema.                                                                                                                                                                                                                                                                   |
|                                                                                                                                 |                                                                                                   | <b>Presentar la Información de soporte</b> para que el alumno estudie los conceptos no rutinarios de los temas. El instructor decide el tiempo necesario para esta actividad.                                                                                                                                                                |                                                                                                                                                                                                                                                                                                       |
|                                                                                                                                 |                                                                                                   | <b>Presentar las Tareas de aprendizaje</b> , <ul style="list-style-type: none"> <li>• Orden ascendente, comenzando por la tarea de menor dificultad hasta la más compleja,</li> <li>• Disminución del soporte por parte del instructor conforme avanzan las tareas.</li> </ul> El instructor decide el tiempo necesario para esta actividad. |                                                                                                                                                                                                                                                                                                       |
|                                                                                                                                 |                                                                                                   | <b>Información procedimental</b> , se proporciona al estudiante al mismo tiempo que las tareas de aprendizaje, la cual puede ser consultada por el estudiante en caso de ser necesario, como apoyo para resolver las tareas de aprendizaje.                                                                                                  |                                                                                                                                                                                                                                                                                                       |
| <b>Fase 4:<br/>Medición de carga cognitiva</b>                                                                                  | Al terminar la fase de tratamiento (26 de mayo)                                                   | Aplicar el cuestionario CS Cognitive Load Component Survey.                                                                                                                                                                                                                                                                                  | Aplicar el cuestionario CS Cognitive Load Component Survey.                                                                                                                                                                                                                                           |
| <b>Fase 5:<br/>Evaluación de los objetivos de aprendizaje</b>                                                                   | Examen 3er parcial                                                                                | Aplicar recall test de los conceptos básicos e importantes del tema visto.                                                                                                                                                                                                                                                                   | Aplicar recall test de los conceptos básicos e importantes del tema visto.                                                                                                                                                                                                                            |
|                                                                                                                                 |                                                                                                   | Aplicar comprensión test: problemas prácticos para resolver sobre el tema visto.                                                                                                                                                                                                                                                             | Aplicar comprensión test: problemas prácticos para resolver sobre el tema visto.                                                                                                                                                                                                                      |
| <b>Fase 6:<br/>Medición de la motivación</b>                                                                                    | Semana del 26 al 30 de mayo                                                                       | Aplicar el cuestionario MSLQ (sección motivación). Informando a los estudiantes que se les dará retroalimentación del mismo.                                                                                                                                                                                                                 | Aplicar el cuestionario MSLQ (sección motivación). Informando a los estudiantes que se les dará retroalimentación del mismo.                                                                                                                                                                          |
| <b>Fase 7:<br/>Evaluación del material y la forma de enseñanza</b>                                                              | Semana del 26 al 30 de mayo                                                                       | Aplicar los cuestionarios Course Interest Survey (CIS) e Instructional Materials Motivation Scale (IMMS)                                                                                                                                                                                                                                     | Aplicar los cuestionarios Course Interest Survey (CIS) e Instructional Materials Motivation Scale (IMMS)                                                                                                                                                                                              |

Figura 5.1: Diseño experimental para estudio piloto.

El grupo de control fue instruido de forma tradicional, es decir, las estrategias de enseñanza se desarrollaron mediante explicaciones de temas y resolución de ejemplos por parte del

profesor. Para el grupo experimental se aplicó el modelo instruccional para crear el material instruccional y generar un plan de aplicación de este como se describe a continuación. Los cuestionarios e instrumentos de evaluación fueron aplicados a ambos grupos.

#### 5.4.1 Implementación fase análisis

Se aplicó el marco de trabajo correspondiente a la fase análisis del modelo instruccional “mapa de ruta CLT4BP”, obteniendo la información de objetivos de aprendizaje, entorno del grupo y efectos de la teoría de carga cognitiva a aplicar, esto para cada uno de los temas implementados en el tratamiento experimental. A continuación, se muestran la información correspondiente a la fase de análisis de los temas punteros. La primera actividad en la fase de análisis fue determinar los objetivos de aprendizaje, y dentro de esta misma actividad diseñar una rubrica e instrumentos de evaluación (ver tabla 5.14). Seguido del cuestionario de motivación, estrategias cognitivas y conocimientos previos (ver tabla 5.13), y por ultimo la selección de los efectos a aplicar (ver tabla 5.15).

| Dimensión de Análisis             | Resultados del Grupo Experimental                                                                                                   |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>Motivación</b>                 | Los resultados del cuestionario arrojan un nivel <b>medio – alto</b> de motivación promedio en el grupo.                            |
| <b>Estrategias Cognitivas</b>     | La mayoría de los estudiantes aplica estrategias cognitivas de forma habitual.                                                      |
| <b>Estrategias Metacognitivas</b> | Se observa que casi ninguno de los estudiantes aplica estrategias metacognitivas en su proceso.                                     |
| <b>Conocimientos previos</b>      | El grupo se divide en dos perfiles definidos: alumnos con <b>conocimientos básicos</b> y alumnos <b>sin conocimientos básicos</b> . |

Tabla 5.13: Resumen de diagnóstico del entorno grupal (Fase de Análisis)

| Nivel              | Objetivos de aprendizaje                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Evaluación                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Novice</b>      | <ul style="list-style-type: none"> <li>• Comprender el concepto fundamental de punteros en C++.</li> <li>• Identificar la sintaxis correcta para declarar y utilizar punteros.</li> <li>• Explicar la diferencia entre un puntero y una variable regular.</li> <li>• Aplicar operadores de dirección (&amp;) y des-referencia (*) correctamente.</li> <li>• Implementar operaciones básicas (asignación, aritmética).</li> <li>• Relacionar los punteros con arrays y funciones.</li> </ul> | <ul style="list-style-type: none"> <li>• <b>Recall test:</b> Opción múltiple sobre conceptos, identificación de sintaxis, asociación de definiciones y cuestionario de operaciones.</li> <li>• <b>Comprehension test:</b> Resolver problemas simples de implementación, corregir errores, analizar salida de código y desarrollar un programa pequeño de manipulación de datos.</li> </ul>                            |
| <b>Intermedios</b> | <ul style="list-style-type: none"> <li>• Comprender el concepto y funcionamiento avanzado de punteros.</li> <li>• Utilizar correctamente la sintaxis de declaración y asignación.</li> <li>• Implementar aritmética de punteros compleja.</li> <li>• Aplicar punteros en la creación y manipulación de estructuras de datos dinámicas.</li> <li>• Implementar funciones que modifiquen variables externas.</li> </ul>                                                                       | <ul style="list-style-type: none"> <li>• <b>Recall test:</b> Definición de propósitos, sintaxis de uso, diferencia entre operadores &amp; y *, concepto de desreferenciación y relación puntero-arreglo.</li> <li>• <b>Comprehension test:</b> Programa de estructuras dinámicas, funciones que modifiquen valores, problemas de aritmética de punteros y optimización vía asignación dinámica de memoria.</li> </ul> |

Tabla 5.14: Matriz de Objetivos de Aprendizaje y Métodos de Evaluación por Nivel



| Componente                | Efectos a aplicar (TCC)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Información de soporte    | <ul style="list-style-type: none"> <li>• <b>Efecto de Modalidad:</b> Combinación de texto e imágenes con narración en video.</li> <li>• <b>Efecto de Atención Dividida:</b> Integración de texto explicativo con diagramas de memoria.</li> <li>• <b>Efecto de Ejemplo Resuelto:</b> Ejemplos paso a paso sobre declaración y uso de punteros.</li> </ul>                                                                                                                                                                                                                                              |
| Tareas de aprendizaje     | <ul style="list-style-type: none"> <li>• <b>Efecto de Problema por Completar:</b> Código parcialmente desarrollado para que los estudiantes completen.</li> <li>• <b>Efecto de Elementos Aislados:</b> Introducción progresiva de conceptos (primero declaración, luego manipulación).</li> <li>• <b>Efecto de Desvanecimiento de Guía:</b> Reducción gradual del apoyo en las tareas.</li> <li>• <b>Efecto de Variabilidad:</b> Para presentar problemas con diferentes contextos.</li> <li>• <b>Efecto de Imaginación:</b> Aplicable a estudiantes intermedios con conocimientos previos.</li> </ul> |
| Información procedimental | <ul style="list-style-type: none"> <li>• <b>Efecto de Variabilidad:</b> Múltiples ejemplos con diferentes contextos.</li> <li>• <b>Efecto de Auto-explicación:</b> Fomentar que los estudiantes expliquen su comprensión.</li> <li>• <b>Efecto de Autogestión:</b> Para que los estudiantes optimicen sus propios materiales.</li> <li>• <b>Efecto de Desvanecimiento de Guía:</b> Para reducir gradualmente el nivel de ayuda.</li> </ul>                                                                                                                                                             |

Tabla 5.15: Aplicación de los efectos de la Teoría de la Carga Cognitiva por componente

#### 5.4.2 Implementación fase diseño y desarrollo

El material se generó para dos tipos de estudiantes (principiantes e intermedios) dentro del grupo experimental, aplicando estrategias diferenciadas basadas en el modelo de Tomlinson. Se aplicaron efectos seleccionados de la teoría de la carga cognitiva. Las directrices del modelo ARCS se implementaron en el material didáctico. En cuanto a los protocolos verbales, se generaron ejercicios isomorfos a las tareas de aprendizaje y, utilizando inteligencia artificial, se creó una transcripción que se convirtió a formato de audio y video.

Cada ejercicio se explica paso a paso como lo haría el instructor. Las Figuras 5.2 y 5.3 son un extracto del material didáctico desarrollado para estudiantes principiantes, que explica la definición, declaración, sintaxis, acceso y modificación de punteros y estructuras anidadas mediante la aplicación de diversos efectos de la teoría, como ejemplos resueltos.

**Analogía:** Imagina las variables como casas, cada una con una dirección única. Un puntero es como una nota donde anotas la dirección de una casa específica. No es la casa en sí, sino simplemente la información sobre dónde encontrarla.

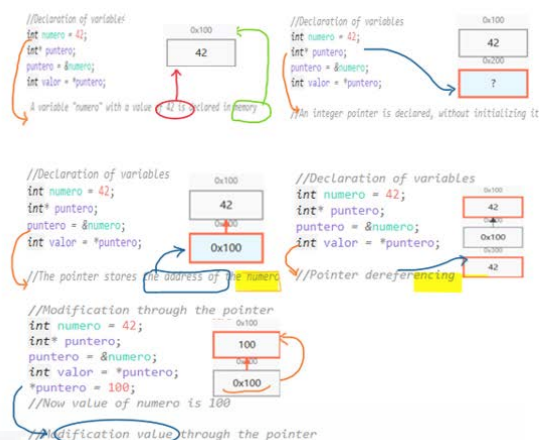


Figura 5.2: Extracto de material instruccional desarrollado para estudiantes novatos.

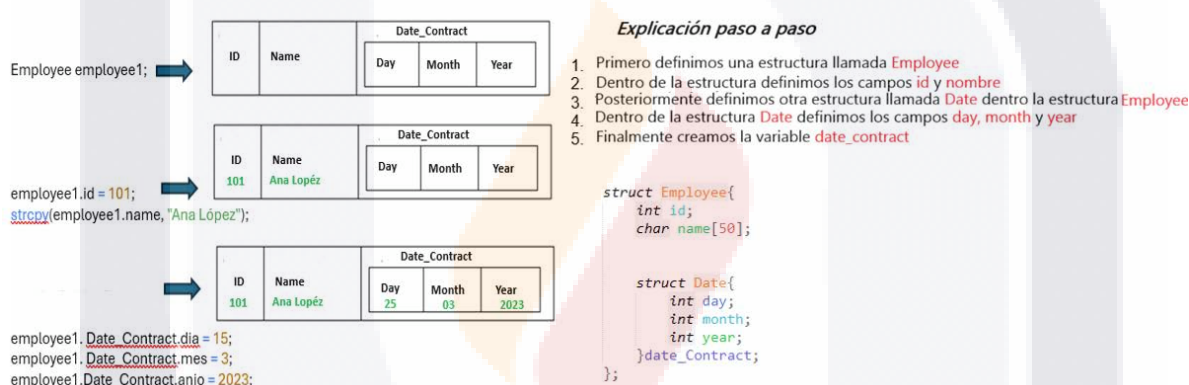


Figura 5.3: Extracto de material instruccional desarrollado para estudiantes novatos.

### 5.4.3 Implementación fase aplicación y evaluación

Una vez creado el material, el tratamiento se aplicó al grupo experimental siguiendo el marco descrito en la sección de la fase de aplicación y evaluación. La información se presenta en formato animado, integrando así todos los conceptos al final. El modelo ARCS se aplicó presentando primero una analogía de la vida real del tema "punteros" (atención), como se muestra a continuación. También se incluyeron ejemplos de aplicación en la vida real (relevancia). Progresión gradual del conocimiento (confianza). El tratamiento se implementó utilizando la plataforma oficial del aula virtual del curso, en un entorno dirigido por el instructor con apoyo decreciente para los estudiantes a medida que avanzaban en las actividades. Una vez finalizado el tratamiento, se administraron pruebas para evaluar el conocimiento declarativo y procedimental. También se administraron cuestionarios para medir la motivación, la carga cognitiva, las percepciones sobre los métodos de enseñanza utilizados por el instructor y las percepciones sobre los materiales didácticos.

#### 5.4.4 Resultados

Para evaluar la eficacia del modelo instruccional CLT4BP, se empleó un enfoque integrando análisis estadísticos descriptivos y cuantitativos. El estudio comparó el grupo experimental, que recibió instrucción mediante el modelo CLT4BP, con el grupo de control, que recibió instrucción tradicional. El análisis estadístico se basó en una metodología que combinó técnicas descriptivas e inferenciales. Para cada variable del estudio, se calcularon estadísticas descriptivas como la media, la desviación estándar y las distribuciones de frecuencia.

Se utilizó ANCOVA para controlar las diferencias iniciales en el conocimiento previo, lo que permitió una comparación más precisa del efecto del tratamiento en los resultados de comprensión. Además de calcular la  $d$  de Cohen para determinar la significancia práctica de las diferencias observadas, se utilizaron pruebas  $t$  de muestras independientes para evaluar las diferencias entre grupos. Se calculó el alfa de Cronbach para cada escala de múltiples ítems con el fin de confirmar la fiabilidad de los instrumentos. Para comprender plenamente los patrones y las relaciones que se encuentran en los datos, se utilizó el análisis de correlación de Pearson para investigar las relaciones entre las variables.

Las variables analizadas en el presente estudio piloto, correspondientes a las mediciones de carga cognitiva, motivación, estrategias cognitivas, percepción y resultados de aprendizaje, se describen a continuación:

##### 1. Variables de carga cognitiva

- Carga intrínseca (CI): Carga cognitiva relacionada con la complejidad del contenido.
- Carga extrínseca (CE): Carga cognitiva derivada del diseño instruccional.
- Carga relevante (CR): Carga cognitiva relacionada con los procesos de aprendizaje.

##### 2. Variables de motivación

- Motivación inicial: Interés (MI), Expectativa (ME), Ansiedad (MA).
- Motivación final: Interés (MI), Expectativa (ME), Ansiedad (MA).

##### 3. Variables de estrategias cognitivas

- Iniciales: Ensayo (ER), Elaboración (EE), Organización (EO), Metacognición (EM), Gestión del tiempo (ET), Regulación del esfuerzo (EE2).
- Finales: Ensayo (ER F), Elaboración (EE F), Organización (EO F), Metacognición (EM F), Gestión del tiempo (ET F), Regulación del esfuerzo (EE2 F).

##### 4. Variables de percepción

- Encuesta de interés del curso (CIS): Atención (A C), Relevancia (R C), Confianza (C C), Satisfacción (S C).
- Escala de motivación de los materiales didácticos (IMMS): Atención (A I), Relevancia (R I), Confianza (C I), Satisfacción (S I).

##### 5. Resultados de aprendizaje

- Prueba de recuerdo: Evaluación del conocimiento declarativo.
- Prueba de comprensión: Evaluación del conocimiento procedimental.

Se realizó un estudio cuasi-experimental con algunas medidas PRE-POST con dos grupos: Grupo experimental (Exp)  $n = 56$ , y Grupo de control (Cont)  $n = 43$ , con el mismo instructor, compuesto por estudiantes de segundo año de Ingeniería de Sistemas Informáticos. En este estudio, el tratamiento para Exp implicó la aplicación del material instruccional diseñado a partir del modelo.

El grupo experimental siguió el modelo instruccional CLT4BP, mientras que el grupo control recibió instrucción tradicional. Los temas tratados incluyeron punteros y estructuras anidadas. Los hallazgos sobre las características iniciales mostraron que ambos grupos tenían un nivel de conocimiento relativamente similar (heterogéneo), con estudiantes con conceptos muy básicos y otros con menos o ningún conocimiento, estrategias cognitivas y metacognitivas similares y niveles de motivación medios-altos al inicio del experimento.

En los resultados estadísticos descriptivos de la medición de la carga cognitiva para los grupos experimental y de control, no se encontraron diferencias significativas en la carga intrínseca entre los grupos ( $p = 0,475$ ), lo que sugiere que ambos grupos percibieron la complejidad inherente del contenido de manera similar. Tampoco se encontraron diferencias significativas en la carga extrínseca entre los grupos ( $p = 0,095$ ), lo que sugiere que ambos grupos percibieron el material didáctico de manera similar. Finalmente, ambos grupos mostraron niveles similares de procesamiento cognitivo relacionado con el aprendizaje, ya que no se encontraron diferencias significativas en la carga relevante entre los grupos ( $p = 0,707$ ).

Para confirmar los resultados anteriores, se realiza una prueba ANOVA multivariante (MANOVA) (véase la Tabla 5.16). IL muestra un efecto significativo ( $F = 0,514$ ,  $p = 0,475$ ), aunque el valor  $p$  sugiere una tendencia no significativa. EL muestra un efecto significativo ( $F = 2,843$ ,  $p = 0,095$ ), lo que indica que no hay diferencias significativas entre los grupos. GL muestra un efecto no significativo ( $F = 0,142$ ,  $p = 0,707$ ), por lo que no hay diferencias entre los grupos. IL, EL y GL no muestran diferencias significativas entre los grupos estudiados. El tamaño del efecto es pequeño para todas las variables ( $R^2$  entre 0,001 y 0,028).

| Source          | Variable | Type III SS         | df | Mean Square | F        | Sig.   |
|-----------------|----------|---------------------|----|-------------|----------|--------|
| Corrected model | IL       | 1.577 <sup>a</sup>  | 1  | 1.577       | .514     | .475   |
|                 | EL       | 15.425 <sup>b</sup> | 1  | 15.425      | 2.843    | .095   |
|                 | GL       | 0.541 <sup>c</sup>  | 1  | .541        | .142     | .707   |
| Intercept       | IL       | 3416.341            | 1  | 3416.341    | 1112.671 | < .001 |
|                 | EL       | 827.859             | 1  | 827.859     | 152.609  | < .001 |
|                 | GL       | 4993.116            | 1  | 4993.116    | 1311.052 | < .001 |
| Group           | IL       | 1.577               | 1  | 1.577       | .514     | .475   |
|                 | EL       | 15.425              | 1  | 15.425      | 2.843    | .095   |
|                 | GL       | .541                | 1  | .541        | .142     | .707   |
| Error           | IL       | 297.828             | 97 | 3.070       |          |        |
|                 | EL       | 526.196             | 97 | 5.425       |          |        |
|                 | GL       | 369.423             | 97 | 3.808       |          |        |
| Total           | IL       | 3795.334            | 99 |             |          |        |
|                 | EL       | 1414.474            | 99 |             |          |        |
|                 | GL       | 5436.813            | 99 |             |          |        |
| Corrected total | IL       | 299.406             | 98 |             |          |        |
|                 | EL       | 541.621             | 98 |             |          |        |
|                 | GL       | 369.963             | 98 |             |          |        |

Tabla 5.16: ANOVA multivariante (MANOVA)

La medición de las estrategias cognitivas no muestra diferencias estadísticas entre el grupo experimental y el grupo de control en todos sus componentes (véase la tabla 5.17).

| Componente | Grupo | Promedio | Desv. Est. | t-value | p-value | Cohen's d |
|------------|-------|----------|------------|---------|---------|-----------|
| CER        | Exp   | 4.60     | 1.231      | -0,636  | .527    | -0,151    |
|            | Cont  | 4.78     | 1.165      |         |         |           |
| CEE        | Exp   | 5.05     | 1.239      | -1,424  | .159    | -0,338    |
|            | Cont  | 5.45     | 1.106      |         |         |           |
| CEO        | Exp   | 3.73     | 1.314      | -1,042  | .301    | -0,247    |
|            | Cont  | 4.06     | 1.370      |         |         |           |
| CEM        | Exp   | 4.74     | 1.032      | -1,246  | .217    | -0,296    |
|            | Cont  | 5.05     | 1.016      |         |         |           |
| CET        | Exp   | 4.70     | .989       | -1,001  | .320    | -0,237    |
|            | Cont  | 4.93     | .945       |         |         |           |
| CEER       | Exp   | 4.68     | 1.397      | -0,365  | .527    | -0,151    |
|            | Cont  | 4.88     | 1.258      |         |         |           |

Tabla 5.17: Estadísticos descriptivos de estrategias cognitivas por grupos

La medición de los resultados del aprendizaje no muestra diferencias estadísticas entre el grupo experimental y el grupo de control en las pruebas de recuerdo. Sin embargo, el componente de la prueba de comprensión (CPOST) muestra una diferencia estadística significativamente mayor para el grupo experimental ( $p = 0,005$ ) con un tamaño del efecto medio ( $d = 0,557$ ). Al comparar los resultados de RPOST y CPOST con RPRE y CPRE para ambos grupos, no se encontraron diferencias estadísticas entre las mediciones de RPOST y RPRE. Sin embargo, en CPOST y CPRE, el grupo experimental mostró una diferencia estadística ( $p = 0,012$ ) con un tamaño del efecto pequeño ( $d = 0,363$ ).

El estudio piloto cuasi-experimental proporciona resultados positivos con respecto a la efectividad del modelo y sus implicaciones para la práctica educativa. El resultado más notable del estudio es la diferencia estadística obtenida por el grupo experimental en los objetivos de aprendizaje de la Prueba de Comprensión (CPOST), como se muestra en la Figura 5.4, después de ajustar por las diferencias iniciales, con un tamaño del efecto medio ( $d$  de Cohen = 0,41). Esto indica una mejora del 10,1 % en el rendimiento de comprensión. Sin embargo, la Prueba de Recuerdo no mostró diferencias estadísticas entre las mediciones PRE y POST, lo que indica que el modelo ayuda al desarrollo de la comprensión más que a la memorización de conceptos.

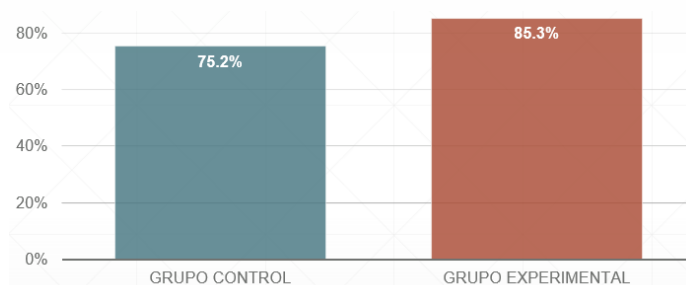


Figura 5.4: Comparación promedios en prueba de comprensión.



## CAPÍTULO 6

### EFFECTOS DE LA TEORÍA DE CARGA COGNITIVA

#### 6.1 Diseño instruccional y aplicación en programación básica

Desde sus inicios (Sweller et al., 1998) la Teoría de Carga Cognitiva (TCC), una teoría de diseño instruccional cuyo principal objetivo es mejorar el aprendizaje de habilidades cognitivas complejas (Berssanette & de Francisco, 2021; Paas & van Merriënboer, 2020; Sweller, 2020; Sweller et al., 2011, 2019) como son la medicina (Andersen, Mikkelsen, Konge, Cayé-Thomasen, & Sørensen, 2016; Frederiksen et al., 2020) matemáticas (Likourezos & Kalyuga, 2019; Maj, 2019; Rittle-Johnson, Loehr, & Durkin, 2017) ingeniería (Fey & Drechsler, 2020) ciencias de la computación y programación (Berssanette & de Francisco, 2021; Darejeh, Mashayekh, & Marcus, 2022; Duran et al., 2022; Mangaroska et al., 2022; Tiedekunta, 2023), lo anterior por medio de sugerencias, procedimientos o guías para el diseño de material pedagógico llamados “efectos instruccionales” (Duran et al., 2022), donde la base de estos efectos instruccionales es el conocimiento de cómo el proceso de cognición humana se lleva a cabo en el cerebro (Sweller, 2020; Sweller et al., 2011, 2019), el material instruccional desarrollado bajo los lineamientos de los efectos instruccionales es probado experimentalmente con métodos de enseñanza tradicional utilizando estudios controlados aleatorios, si el efecto instruccional da como resultado un mejor rendimiento el efecto ha sido comprobado (Sweller, 2020; Sweller et al., 2011, 2019).

##### 6.1.1 Solución libre (Goal-Free effect)

La solución de problemas es una de las prácticas más utilizadas en el aprendizaje de temas complejos (Purnama & Retnowati, 2020), pero genera una carga cognitiva en la memoria de trabajo durante el proceso debido a la complejidad propia del problema, así como la forma en que se presenta al estudiante, sumado a ello el solicitar una solución específica al problema incrementa la carga cognitiva (Sweller et al., 2019). De ello la finalidad del efecto de solución libre es reducir la carga cognitiva impuesta, replanteando el problema sin solicitar una solución específica, conocidos como “problemas de solución libre” (Sweller et al., 2011), minimizando la carga cognitiva extraña en el estudiante al evitar que este enfoque recursos cognitivos al concentrarse en una solución específica (Purnama & Retnowati, 2020; Sweller et al., 2019), recomendada su aplicación en estudiantes novatos. Por ejemplo, dada la figura geométrica “calcule la longitud del lado x”, pero para ello el estudiante debe calcular la longitud de otro lado creando así una sub-solución, aumentado aún más la carga cognitiva en el estudiante. Modificando este problema a “problema de solución libre”: Dada la figura geométrica “calcula la longitud de todos los lados desconocidos”.

#### Aplicación del efecto en programación

El efecto de solución libre es el primer efecto de la TCC estudiado (Sweller et al., 2019) por lo cual la evidencia empírica de su aplicación encontrada en la literatura en su mayoría data de los años 90 (Sweller et al., 2011), en donde su principal área de aplicación son las matemáticas reportando en su mayoría resultados positivos en su aplicación (Sweller et al.,



2011). En lo que respecta a los últimos 5 años se tiene poca evidencia de su aplicación (Maulidya, Hasanah, & Retnowati, 2017; Purnama & Retnowati, 2020; Sugiman, Retnowati, Ayres, & Murdanu, 2019; Sweller et al., 2019), en el cual se aplica el efecto de solución libre en el área de matemáticas específicamente en el tema de geometría en grupos de trabajo colectivo e individual. En lo que respecta al área de programación no se encontró evidencia empírica de su aplicación.

### **Recomendaciones para la aplicación en programación básica**

En base a la descripción del efecto de solución libre descrita anteriormente, se proponen a considerar algunos puntos a la hora de aplicar el efecto en el diseño de material instruccional como lo son:

- Los problemas deben poder plantearse en formato de “problemas de solución libre”, es decir, deben tener más de una solución, problemas de solución única no es posible aplicar el efecto.
- Tener bien definida la meta de aprendizaje que se pretende alcanzar con aplicación de dichos problemas en formato de solución libre.
- Planteamiento del problema de forma clara y precisa de tal forma que se alcance la meta deseada.
- Conocer el grado de conocimientos previos de los alumnos
- Definir el problema de forma que estudiante solo se enfoque en entender la naturaleza de este.
- Conocer todas las soluciones del problema planteado.
- Se recomienda su aplicación a estudiantes novatos.

El objetivo principal en la programación es la resolución de problemas complejos que requieren un análisis exhaustivo para generar la solución mediante procedimientos o acciones bien estructuradas. De ello la mayoría de estos problemas en términos de programación pueden tener más de una solución posible, lo cual hace posible la aplicación del efecto de solución libre en casi todos los temas de programación básica. A continuación, se plantean algunas propuestas de aplicación del efecto de solución libre en temas de programación.

### **Ejemplo propuesto de aplicación 1**

Tomando el tema de arreglos de conceptos de programación básica, se tiene el siguiente problema de forma tradicional:

*“Dado un arreglo llamado enteros, cuyo contenido es: 5, 6, 2, 3, 8, 1, 7, 4. Utilizando el algoritmo de “Burbuja” ordene el arreglo y muestre el resultado en pantalla”.*

Como se puede observar el problema está planteado en forma de solución específica o solución única puesto que se pide explícitamente que se ordene el arreglo utilizando el algoritmo de burbuja para ello. Replantando el problema anterior a forma de solución libre algunas propuestas serían como siguen:

*“Dado un arreglo llamado enteros, cuyo contenido es: 5, 6, 2, 3, 8, 1, 7, 4. Utilizando cualquier algoritmo de ordenación, ordene el arreglo y muestre el resultado en pantalla”.*

*“Dado un arreglo llamado enteros, cuyo contenido es: 5, 6, 2, 3, 8, 1, 7, 4. Genera todas las soluciones como te sea posible de los algoritmos de ordenación, para ordenar el arreglo y mostrar el resultado en pantalla”.*

## **Ejemplo propuesto de aplicación 2**

De los conceptos básicos dentro de la programación se encuentra el tópico de algoritmos, donde se introduce a los alumnos a la resolución de problemas mediante la creación de algoritmos utilizando pseudocódigo y una de las herramientas muy utilizadas para la interpretación de pseudocódigo es PSeInt, en el cual se utiliza un pseudolenguaje en español mediante un editor. De lo anterior se presenta un ejemplo de propuesta de aplicación:

*“Cree un algoritmo en PSeInt para sumar la serie 5, 10, 15, 20... hasta 100 utilizando la estructura repetitiva Mientras”.*

El problema anterior está planteado en formato de solución específica puesto que solicita explícitamente la utilización de una acción en particular para su resolución, cambiando el planteamiento del problema a solución libre se tiene:

*“Cree un algoritmo en PSeInt para sumar la serie 5, 10, 15, 20... hasta 100 utilizando cualquier estructura repetitiva”.*

*“Cree todos los algoritmos posibles en PSeInt para sumar la serie 5, 10, 15, 20... hasta 100 utilizando todas las estructuras repetitivas como te sea posible”.*

Replanteando los problemas a formato de solución libre se deja abierta la solución a que el estudiante resuelva con cualquier opción posible de su elección, o también se solicita generar todas las soluciones que le sean posible que conozca, liberando así el enfoque a una solución específica.

### **6.1.2 Ejemplo resuelto (Worked example effect)**

Es uno de los principios más común y utilizado para la transferencia de nuevos conocimientos, en la mayoría de áreas de estudio pero principalmente en áreas de ciencias y matemáticas (Sweller et al., 2011), cuyo objetivo, atendiendo los principios de la TCC, es disminuir la carga cognitiva del estudiante proporcionando de un problema resuelto paso a paso que el alumno debe estudiar a profundidad, liberándolo así de la carga cognitiva que se impondría al resolverlo (Berssanette & de Francisco, 2021; Sweller et al., 2019). El estudio a base de ejemplos de problemas resueltos permite al estudiante llegar a comprender dicho problema en conjunto con su solución, y en base a ello generar sus propias estrategias para la resolución de problemas similares (Cevahir, Özdemir, & Baturay, 2022). La aplicación del efecto de ejemplo resuelto se recomienda en estudiantes novatos, debido a su naturaleza descriptiva paso a paso los ejemplos resueltos son menos eficientes en estudiantes expertos al generar en ellos un efecto de redundancia de la información (Sweller et al., 2019), de igual forma si los

estudiantes novatos no son capaces de identificar la información importante en los ejemplos resueltos no se obtendrá un resultado eficiente (Bofferding, Kocabas, Aqazade, Haiduc, & Chen, 2022).

### **Aplicación del efecto en programación**

En el área de la programación la utilización del efecto de ejemplo resuelto es el más utilizado enfocado en estudiantes novatos en cursos introductorios a la programación reportando en su mayoría resultados positivos (Berssanette & de Francisco, 2021). La literatura respecto a la aplicación del efecto en programación es muy extensa al ser el efecto más utilizado no solo en esta área sino en la gran mayoría de las áreas de estudio (Sweller et al., 2011), por mencionar algunas de las más recientes (Bofferding et al., 2022; Cevahir et al., 2022; Garces et al., 2023; Heitzmann, Fischer, & Fischer, 2018; Toukiloglou & Xinogalos, 2023).

### **Recomendaciones para la aplicación en programación básica**

Al ser uno de los efectos más utilizados para la transferencia de conocimientos se deben tomar en cuenta varios puntos a la hora de su diseño y aplicación para el logro de los objetivos planteados. A continuación, se proponen algunas recomendaciones en su diseño y aplicación:

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos.
- Diseño de los ejemplos con el enfoque a reducción de la carga cognitiva extraña como lo marca la TCC.
- Los ejemplos diseñados no deben requerir que el estudiante realice un proceso mental adicional de integración de diferentes fuentes de información.
- Los ejemplos deben contener una explicación explícita paso a paso de su solución.
- Acompañar inmediatamente después del estudio de los ejemplos resueltos con problemas similares para que el estudiante resuelva y aplique el conocimiento adquirido a través del estudio de los ejemplos resueltos.
- Su aplicación se recomienda en estudiantes novatos.

### **Ejemplo propuesto de aplicación**

Del tema correspondiente a arreglos, se presenta un ejemplo de un programa en lenguaje C++ para la creación y llenado de un arreglo de números enteros identificado mediante la variable vector, donde el tamaño de dicho arreglo es definido por el usuario mediante la variable TAM.

- A. Primero planteando la solución mediante un algoritmo se tiene:
  1. Solicitar tamaño del arreglo al usuario
  2. Guardar el valor en una variable
  3. Definir el vector del tamaño introducido por el usuario
  4. Capturar el contenido del arreglo uno a uno hasta el tamaño solicitado

- B. El algoritmo de solución planteado al problema se pasa a pseudocódigo
- C. Finalmente se codifica el pseudocódigo al lenguaje de programación C++, con la explicación a base de comentarios en cada línea de código utilizada

```
Algoritmo arreglo
  Limpiar Pantalla
  definir TAM como entero
  escribir "Ingrese el tamaño del vector: " sin saltar
  leer TAM
  definir M como entero
  dimension avector[TAM]
  Para M←1 hasta TAM con paso 1 Hacer
  |   escribir "Ingrese el dato del arreglo en la posición ",M," " Sin Saltar
  |   leer avector[M]
  FinPara
FinAlgoritmo

#include <stdio.h> // se declara la librería stdio.h,
//para poder utilizar funciones como el printf, scanf

int main() { //declaración función principal main tipo int
  int TAM; //se declara la variable TAM de tipo entero
  //donde se guardará el valor del tamaño del arreglo

  printf("Ingrese el tamaño del vector: "); //se solicita
  //al usuario ingresar en pantalla el tamaño del vector

  scanf("%d", &TAM); //se guarda el valor introducido por
  //el usuario en la variable TAM declarada anteriormente

  int vector[TAM]; //se declara la variable vector con un
  //tamaño TAM introducido anteriormente por el usuario

  for (int i=0; i<TAM; i++){ //declaración del ciclo for
  //para el llenado del arreglo, con la variable de control i
  //comenzando en 0 condicionado a repetirse mientras i sea
  //menor que TAM incrementando la variable de control i en 1
  //en cada iteración
```

### 6.1.3 Problema por completar (Completion problem effect)

En el proceso de enseñanza-aprendizaje la utilización de solo ejemplos resueltos como método de estudio puede conducir al estudiante a una zona de aprendizaje pasivo y no activo (Sweller et al., 2011), donde no se anima al estudiante a ser participe del aprendizaje y probablemente no estudien a profundidad los conceptos en los ejemplos resueltos, lo que se traduce una transferencia de conocimientos deficiente o nula (Sweller et al., 2019). Los problemas por completar son un paso intermedio entre los ejemplos resueltos y los problemas convencionales, interviniendo aquí también el desarrollo de la pericia en los estudiantes novatos, a medida que

```

printf("Ingrese el elemento %d del vector: \n", i);
//se solicita al usuario ingresar un valor para el arreglo

scanf("%d", &vector[i]); //se guarda el valor ingresado en
//el arreglo en la posición i

} //cierre ciclo for
return 0; //declaración función return con 0 debido a la
//declaración tipo int del main

} //cierre función principal main

```

se va disminuyendo la ayuda por parte del profesor al pasar de poco de un ejemplo completo resuelto a uno parcialmente resuelto y finalmente a un problema por resolver (Andrade-Lotero, 2012; Sweller et al., 2019), de ello la aplicación de problemas por completar incita al estudiante a un aprendizaje activo y estudien los ejemplos de manera más profunda e identificar de mejor manera la información clave contenida en el material (Berssanette & de Francisco, 2021), mediante ejemplos parcialmente resueltos (problema por completar) que el estudiante debe terminar de resolver.

### Aplicación del efecto en programación

El problema por completar al ser un efecto identificado como enlace entre un ejemplo resuelto y problema convencional, la evidencia de su aplicación reporta su utilización en combinación en su mayoría con el ejemplo resuelto o algún otro efecto de la TCC (Berssanette & de Francisco, 2021), como se describe en (Boldbaatar & Şendurur, 2019) mediante la aplicación del ejemplo resuelto y problema por completar a estudiantes universitarios no novatos en un estudio de transferencia de información en programación, así como en (Guzman et al., 2019) donde aplican el efecto de ejemplo resuelto, problema por completar y atención dividida a estudiantes de biología para el aprendizaje del lenguaje de programación R, de igual forma en (Abdul-Rahman & Du Boulay, 2014; Harms et al., 2016; Morrison et al., 2016) se aplica el efecto de problema por completar en conjunto con el ejemplo resuelto.

### Recomendaciones para la aplicación en programación básica

El problema por completar al ser un efecto con una relación muy conectada con el efecto de ejemplo resuelto y servir de vínculo para pasar a la resolución de problemas convencionales las recomendaciones para su aplicación propuestas sumadas a las de del ejemplo resuelto se mencionan a continuación:

- Diseño de las soluciones parciales con el enfoque a los objetivos de aprendizaje, es decir enfocar las tareas por completar en los conceptos importantes que se pretenden abordar en los problemas.
- El diseño de los problemas incite al estudiante al aprendizaje activo.



## Ejemplo propuesto de aplicación

Como se ha mencionado la aplicación del efecto de problema por completar se recomienda en conjunto con el efecto de ejemplo resuelto, de ello tomando el ejemplo de propuesta de aplicación presentado en sección anterior, seguido a ello se replantea el ejemplo llevado a formato de problema por completar:

*Del tema correspondiente a arreglos, se presenta un ejemplo de un programa en lenguaje C++ para la creación y llenado de un arreglo de números enteros identificado mediante la variable vector, donde el tamaño de dicho arreglo es definido por el usuario mediante la variable TAM, donde el ejemplo está parcialmente resuelto, por favor complete todo el procedimiento en los espacios faltantes.*

Primero planteando la solución mediante un algoritmo se tiene:

1. Solicitar...
2. Guardar...
3. Definir...
4. Capturar...

El algoritmo de solución planteado al problema se pasa a pseudocódigo:

```
Algoritmo arreglo
  Limpiar Pantalla
  definir
    "Ingrese el tamaño del vector: "
  leer
  definir
  dimension
  Para
    "Ingrese el dato del arreglo en la posición ",M," "
    avector[M]
  Hacer
  Fin
FinAlgoritmo
```

Finalmente se codifica el pseudocódigo al lenguaje C++:

```
#include <stdio.h>

int main() {
    --;
    printf("--");
    scanf("--", --);
    int --;
    for (--){
        printf("Ingrese el elemento %d del vector: \n", --);
        scanf("--", --);
    }
    return 0;
}
```



#### 6.1.4 Atención dividida (Split-attention effect)

El efecto de atención dividida surge a partir de material instruccional de ejemplos resueltos mal diseñados (Sweller et al., 2011, 2019) donde dos o más fuentes de información necesarias para la comprensión del ejemplo resuelto son separadas física o temporalmente al presentar la información al estudiante (Arevalo-Mercado et al., 2023; Berssanette & de Francisco, 2021; Paas & van Merriënboer, 2020; Sweller et al., 2019), entonces surge el efecto de atención dividida forzando al estudiante a dividir la atención hacia múltiples fuentes de información dispersas las cuales debe integrar mentalmente (C. Arévalo & Solano, 2013) aumentando la carga cognitiva extraña en el proceso de aprendizaje dificultado la transferencia de conocimiento (Sweller et al., 2011). Para prevenir el efecto de atención dividida y evitar el aumento de la carga cognitiva producida por el mismo es necesario presentar las fuentes de información relacionadas entre sí de formada integrada (Arevalo-Mercado et al., 2023; Paas & van Merriënboer, 2020).

#### Aplicación del efecto en programación

Si bien la aplicación del efecto de atención dividida no tiene el mismo registro de uso que los efectos de ejemplo resuelto y problema por completar, se encuentra varia evidencia empírica de su aplicación principalmente en área de matemáticas en ciencia (Sweller et al., 2011), sin embargo en el área de programación resalta su uso y aplicación con algunos trabajos (Berssanette & de Francisco, 2021), por ejemplo Altadmri, Kölling, and Brown (2016) argumentan que la utilización de editores gráficos de código a base de bloques combinados con texto requieren menos esfuerzo cognitivo para los novatos que un editor de código de texto plano, si bien no mencionan explícitamente la utilización del efecto de atención dividida se deduce el enfoque hacia este en su desarrollo. En Guzman, Pennell, Nikelski, and Srivastava (2019) aplican en efecto de atención dividida para la enseñanza del lenguaje de programación R a graduados del área de biología, además de este efecto, utilizan en efecto de ejemplo resuelto y problema por completar. Bridson (2020) presenta una tesis de maestría de la Universidad de Memphis, mediante la creación de sistema tutorial web para la comprensión de código de conceptos básicos en Java, dicho sistema desarrollado bajo los conceptos del efecto de atención dividida y reversión de la experiencia de la TCC. Arevalo-Mercado et al. (2023) aplican el efecto de atención dividida mediante una herramienta de visualización para el aprendizaje del tema de estructuras de datos en lenguaje C++, mediante una interfaz gráfica e interactiva, incluyendo explicación de los ejemplos de código, así como ayuda contextual de la sintaxis.

#### Recomendaciones para la aplicación en programación básica

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos.
- Diseño del material con el enfoque a reducción de la carga cognitiva extraña como lo marca la TCC.
- Los ejemplos diseñados no deben requerir que el estudiante realice un proceso mental adicional de integración de diferentes fuentes de información.

- Presentar las fuentes de información relacionadas entre sí de forma integrada.
- Considerar la complejidad y tamaño de las fuentes de información a integrar, para no generar un efecto opuesto.

### Ejemplo propuesto de aplicación

El objetivo del efecto de atención dividida es presentar diferentes fuentes de información necesarias para la comprensión de algún concepto relacionadas entre sí de forma integrada, evitando así que el estudiante realice un esfuerzo cognitivo adicional de integración si la información se presenta de forma dispersa. De lo anterior se presenta una propuesta tomando como ejemplo el algoritmo de ordenamiento burbuja del tema de arreglos.

El ejemplo propuesto se muestra a través de una tabla integrada, donde se tiene en primer lugar la explicación teórica del algoritmo de ordenamiento, seguido en la siguiente columna de una explicación gráfica de cómo se van realizando las iteraciones a través del arreglo y cómo van cambiando las variables de valores, y en la tercera columna el código correspondiente al algoritmo. Para el caso de la explicación gráfica si utilizamos recursos tecnológicos para su presentación se puede presentar mediante una animación de todas las iteraciones en el arreglo, presentando la información más completa y a detalle.

| Explicación teórica                                                                                                                                                                                                                                                                                               | Explicación gráfica                                                                                                                                                                                  | Código                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Ordenamiento Burbuja:</b><br>Cada iteración se realizan comparaciones de un elemento (elemento "j") con el elemento anterior (elemento "j-1"), y si el elemento "j" es menor que el elemento "j-1" se intercambian. De esta forma, se consigue ir subiendo el elemento mayor del array hasta el final de este. | <div>6 5 3 1 8 7 2 4</div> 1era iteración:<br>for<br>i = 0;<br>TAM = 8;<br>for<br>j = 0;<br>vector[j] = 6<br>vector[j+1] = 5<br>if<br>vector[j] = 5<br>vector[j+1] = 6<br><div>5 6 3 1 8 7 2 4</div> | <pre>for (int i = 0; i &lt; TAM; i++){     for (int j = 0; j &lt; TAM-1; j++){         if(vector[j] &gt; vector[j+1]){             temporal = vector[j];             vector[j] = vector[j+1];             vector[j+1] = temporal;         }     } }</pre> |

#### 6.1.5 Modalidad (Modality effect)

En los inicios de la TCC (Sweller et al., 1998) asume que la capacidad de la memoria de trabajo para cada individuo es fija dada por la cantidad de información que puede procesar a la vez, pero el efecto de modalidad se basa en que la el procesamiento de la información en la memoria de trabajo es procesada de forma independiente (visual y auditiva) mediante dos canales sensoriales diferentes (Berssanette & de Francisco, 2021), de esta forma se reduce la carga cognitiva mediante el aumento de la eficiencia de la memoria de trabajo bajo la utilización de ambos canales sensoriales para el procesamiento de información visual y auditiva (Sweller et al., 2019). El efecto de modalidad está sumamente relacionado con el efecto de atención dividida como una alternativa al problema de este, por ejemplo, reemplazar información escrita de un ejemplo por audio puede ser igual o más efectiva que su integración física (Berssanette & de Francisco, 2021).

## Aplicación del efecto en programación

Después del efecto de ejemplo resuelto y problema por completar, el efecto de modalidad es de los que más evidencia empírica se tiene (Berssanette & de Francisco, 2021), al igual que los efectos de ejemplo resuelto y problema por completar, el efecto de modalidad se encuentra evidencia de su aplicación en otras áreas como son matemáticas y ciencias (Sweller et al., 2011). En su aplicación en el área de programación Çakiroglu and Aksoy (2017) presentan un caso de estudio sobre un sistema web de instrucción en línea para un curso introductorio de programación, realizan un análisis de los elementos que influyen en la carga cognitiva extraña de los estudiantes, como son la modalidad, redundancia e integración de elementos. Morrison (2017) realizan un estudio aplicando el efecto de modalidad mediante videos explicativos de fragmentos de código, reemplazando así explicaciones de texto por audio, similar a lo anterior Raj et al. (2020) realizan un experimento utilizando Live-coding (escribir el código y al mismo tiempo ir explicando cada instrucción en frente de los estudiantes) comparando con ejemplos de código ya resueltos. Zavgorodniaia, Hellas, Seppälä, and Sorva (2020) presentan un estudio aplicando el efecto de modalidad mediante videos explicativos del tema de variables en el lenguaje Python mediante ejemplos resueltos en tres diferentes formatos ejemplos+audio, ejemplos+texto y ejemplos+audio+texto.

## Recomendaciones para la aplicación en programación básica

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos.
- Diseño del material con el enfoque a reducción de la carga cognitiva extraña como lo marca la TCC.
- El reemplazo de audio por texto debe ser corto.
- Textos largos deben presentarse de forma escrita.
- La interactividad de los elementos debe ser alta (carga cognitiva intrínseca) para su aplicación.
- Si la complejidad es alta, resaltar partes de la información visual a las que se enfoca la información auditiva.

## Ejemplo propuesto de aplicación

De el ejemplo anterior del efecto de atención dividida, algoritmo de ordenamiento de burbuja, se pasa a formato de modalidad, sustituyendo el texto de la explicación teórica por audio, de igual forma se agrega audio a la explicación gráfica y al código.

### 6.1.6 Autoexplicación (Self-explanation effect)

El uso de ejemplos resueltos como método de estudio o repaso en el proceso educativo constituye una estrategia frecuente para fomentar la construcción de conocimiento, sin embargo, Sweller et al. (2019) señalan que esta técnica solo resulta efectiva cuando los estudiantes analizan estos ejemplos con profundidad y minuciosidad. La auto-explicación emerge como

| Explicación teórica                               | Explicación gráfica                                                                                                                                                                                                        | Código                                                                                                                                                                                                                                                                                            |
|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Audio explicación algoritmo</p> <p>con del</p> | <p>6 5 3 1 8 7 2 4</p> <p>1era iteración:</p> <pre>for i = 0; TAM = 8; for j = 0; vector[j] = 6 vector[j+1] = 5 if vector[j] = 5 vector[j+1] = 6</pre> <p>5 6 3 1 8 7 2 4</p> <p>Audio con explicación de la iteración</p> | <pre>for (int i = 0; i &lt; TAM; i++){     for (int j = 0; j &lt; TAM-1; j++){         if(vector[j] &gt; vector[j+1]){             temporal = vector[j];             vector[j] = vector[j+1];             vector[j+1] = temporal;         }     } }</pre> <p>Audio con explicación del código</p> |

un efecto facilitador para esta comprensión profunda, permitiendo que los estudiantes desarrollen y automaticen esquemas cognitivos (Sweller et al., 2011). Este efecto puede conceptualizarse como una actividad de aprendizaje activo que involucra directamente al estudiante con el contenido, facilitando la supervisión de su nivel de comprensión (Sweller et al., 2019). Se trata de un proceso cognitivo constructivo orientado a la asimilación de nueva información o destrezas mediante la generación de explicaciones propias basadas en conocimientos previos, funcionando como mecanismo de consolidación. Sweller et al. (2019) destaca que, en determinadas circunstancias, la intervención explicativa del instructor resulta necesaria cuando el estudiante no logra formular explicaciones adecuadas o completas por sí mismo.

### Aplicación del efecto en programación

El efecto de auto-explicación ha demostrado resultados favorables en diversos estudios abarcando múltiples disciplinas (Sweller et al., 2011), lo que sugiere su potencial aplicación en prácticamente cualquier campo educativo, tanto para conocimientos conceptuales como procedimentales. Souza and Batista (2019) señalan que incentivar a los estudiantes a generar explicaciones propias resulta más eficaz que presentarles explicaciones ya elaboradas, efecto que se intensifica cuando el estudiante posteriormente revisa y compara su explicación con nueva información.

En el ámbito específico de la programación, Fabic et al. (2017) reportaron resultados sobre el uso de PyKinect, una aplicación móvil para aprender Python que incorpora un sistema de indicaciones de auto-explicación basadas en menús”. Yen and Wang (2017) desarrollaron un entorno de programación para C++ fundamentado en estrategias de auto-explicación con retroalimentación adaptada. Complementariamente, Price et al. (2020) crearon el prototipo Instructor’s solution como herramienta de apoyo para tareas de programación en línea, combinando elementos de comparación y auto-explicación).

### Recomendaciones para la aplicación en programación básica

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos.
- Diseño del material con el enfoque a reducción de la carga cognitiva extraña como lo marca la TCC.
- Se recomienda su aplicación a estudiantes con conocimiento previo.

- La interactividad de los elementos debe ser alta (carga cognitiva intrínseca) para su aplicación.
- Aplicación en combinación con otros efectos como ejemplo resuelto, problema por completar, modalidad, atención dividida.

### Ejemplo propuesto de aplicación 1

La maestra envió un programa de ejemplo sobre el método de la burbuja para la ordenación en arreglos, pero algunos de tus compañeros no entendieron el tema visto en clase y por lo tanto no entienden el ejemplo presentado. ¿Puedes ayudar a tus compañeros? Revisa por favor el código y en el espacio de abajo, escribe y explica con tus palabras a tus compañeros los bloques de código del programa.

```
#include <stdio.h>

int main(int argc, const char * argv[])
{
    int TAM;
    printf("Cantidad de elementos del arreglo: ");
    scanf("%d", &TAM);
    int vector[TAM], temporal;
    for (int i=0; i<TAM; i++){
        printf("Ingrese el elemento %d del vector: \n", i);
        scanf("%d", &vector[i]);
    }
    printf("El contenido del arreglo es: \n");
    for (int i=0; i<TAM; i++){
        printf("%d \t", vector[i]);
    }
    for (int i = 0; i < TAM; i++){
        for (int j = 0; j < TAM-1; j++){
            if (vector[j] > vector[j+1]){
                temporal = vector[j];
                vector[j] = vector[j+1];
                vector[j+1] = temporal;
            }
        }
    }
    printf("\n El contenido ordenado del arreglo es: \n");
    for (int i=0; i<TAM; i++){
        printf("%d \t", vector[i]);
    }
    return 0;
}
```

### Ejemplo propuesto aplicación 2

Otra forma para aplicar el efecto de auto-explicación es la utilización de ejemplos con errores, donde el alumno tiene que identificar los errores, explicar el ¿por qué? de los errores y proponer una solución correcta con su respectiva explicación. Un ejemplo de ello se muestra a continuación:



La maestra dejó un ejercicio del tema de cadenas visto en la clase, el objetivo es realizar un programa que, dadas dos cadenas de caracteres, se realice la comparación de ambas cadenas y nos diga si las cadenas son iguales o diferentes (NOTA: sin utilizar las funciones `strcmp`, `strcpy` y `strlen`). Uno de tus compañeros ya programó su solución, pero no funciona adecuadamente. ¿Puedes apoyar en identificar cual es el error(es), y proporcionar una explicación el por qué una no funciona, así como una solución y explicación de esta?

```
Valor de la cadena 1:
Guadalajara
Valor de la cadena 2:
Aguascalientes
Las cadenas 1 y 2 son iguales...
```

```
#include <stdio.h>
int tam(char cadena[]);
int main() {
    char cad1[100], cad2[100];
    int tam1=0, tam2=0, iguales=0;
    printf("Valor de la cadena 1: \n");
    scanf("%s",&cad1);
    printf("Valor de la cadena 2: \n");
    scanf("%s",&cad2);

    tam1=tam(cad1);
    tam2=tam(cad2);
    if (tam1 != tam2){
        printf("Las cadenas 1 y 2 son distintas... \n");
    }else{
        for(int i=0; i<tam1; i++){
            if (cad1[tam1] == cad2[tam2]){
                iguales++;
            }
        }
        if (iguales == tam1){
            printf("Las cadenas 1 y 2 son iguales... \n");
        }
    }
    return 0;
}
int tam(char cadena[]){
    int tam=0;
    for(int i=0; cadena[i]!='\0';i++){
        tam++;
    }
    return tam;
}
```

#### 6.1.7 Redundancia (Redundancy effect)

El efecto de redundancia surge a partir del efecto de atención dividida y modalidad, cuando se le presenta al estudiante la misma información de diferente forma, por ejemplo texto y audio a la vez (Sweller et al., 2019), de ello es importante identificar fuentes de información independientes redundantes y evaluar para seleccionar la más efectiva para su aplicación,



eliminando así la redundancia , la cual podría provocar elevar la carga cognitiva requiriendo que el estudiante realice un esfuerzo cognitivo mayor lo que produciría un aprendizaje más pobre (Andrade-Lotero, 2012; Sweller et al., 2019).

### Aplicación del efecto en programación

La evidencia empírica sobre la aplicación del efecto de redundancia se encuentra más en las áreas de matemáticas, ciencias e ingeniería (Sweller et al., 2011), así como en la enseñanza – aprendizaje de un segundo idioma. La aplicación en programación se encuentra poca evidencia de su aplicación, Çakiroglu and Aksoy (2017) realizan un estudio de como influyen elementos de carga cognitiva como lo son modalidad, redundancia, atención dividida, en el proceso de aprendizaje de un sistema de instrucción web en línea en un curso introductorio de programación. Y Cheah and Leong (2019) analizan la presencia o no del efecto de redundancia al utilizar screencasting en un curso de programación en C++, al presentar al estudiante material con información redundante en diferente modalidad.

### Recomendaciones para la aplicación en programación básica

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos.
- Diseño del material con el enfoque a reducción de la carga cognitiva extraña como lo marca la TCC.
- Los ejemplos diseñados no deben requerir que el estudiante realice un proceso mental adicional para descartar información redundante.
- No presentar fuentes de información iguales/similares en diferentes modalidades, ejemplo audio y texto de forma simultánea.
- En fuentes de información similares (audio, texto, imágenes, video) seleccionar la más adecuada para su aplicación de acuerdo con el objetivo de aprendizaje requerido.

### Ejemplo propuesto de aplicación

Al igual que el efecto de atención dividida el efecto de redundancia es un efecto que se debe evitar a la hora de diseñar material instruccional, de ello se presenta un ejemplo de material con la presencia del efecto de redundancia, seguido de ello el mismo ejemplo eliminando información redundante eliminando así el efecto en el material instruccional.

```
//ordenación de menor a mayor (Burbuja).  
//Para ordenar de mayor a menor cambiar el operador del condicional if por un "<"  
for (int i = 0; i < TAM; i++){  
    for (int j = 0; j < TAM-1; j++){  
        if (vector[j] > vector[j+1]){  
            temporal = vector[j];  
            vector[j] = vector[j+1];  
            vector[j+1] = temporal;  
        }  
    }  
}
```

Ordenamiento Burbuja:

*Cada iteración se realizan comparaciones de un elemento (elemento "j") con el elemento anterior (elemento "j-1"), y si el elemento "j.º" menor que el elemento "j-1" se intercambian. De esta forma, se consigue ir subiendo el elemento mayor del array hasta el final de este. + AUDIO SIMULTANEO DEL TEXTO.*

En el ejemplo anterior se presenta el código para ordenar un vector numérico mediante el algoritmo de burbuja, a la par que se presenta el código se muestra la descripción del algoritmo mediante texto y a su vez un audio del mismo texto, lo cual provoca la presencia del efecto de redundancia, para eliminar el efecto, se selecciona una modalidad de las fuentes de información redundantes y se elimina la otra quedando:

```
//ordenación de menor a mayor (Burbuja).
//Para ordenar de mayor a menor cambiar el operador del condicional if por un "<"
for (int i = 0; i < TAM; i++){
    for (int j = 0; j < TAM-1; j++){
        if (vector[j] > vector[j+1]){
            temporal = vector[j];
            vector[j] = vector[j+1];
            vector[j+1] = temporal;
        }
    }
}
```

### ***Ordenamiento Burbuja: + audio descripción del algoritmo***

#### **6.1.8 Interactividad (Element interactivity effect)**

En el aprendizaje de conocimientos o tareas complejas donde interactúan muchos elementos de alta complejidad y si además se agrega más interactividad de elementos al material instruccional, esto puede sobrecargar la memoria de trabajo impidiendo el aprendizaje (Sweller et al., 2011). El efecto de interactividad de los elementos ocurre cuando los resultados que pueden ser obtenidos utilizando material con alta interactividad de elementos desaparecen o se revierten al modificar el material disminuyendo la interactividad de elementos (Berssantette & de Francisco, 2021), procesando los elementos de manera individual o aislada (Chen, Kalyuga, & Sweller, 2017; Sweller et al., 2019), o elevando el nivel de experiencia de los estudiantes (Berssantette & de Francisco, 2021; Sweller et al., 2019), sin embargo esto aplicable a estudiantes novatos, ya que un mismo material puede ser de alto nivel de interactividad de elementos para un novato pero de bajo nivel para un experto (Sweller, 2020), de ello aplicando técnicas de carga cognitiva apropiadas se reduce la interactividad de elementos asociada con la carga cognitiva extraña como reducir el número de elementos o información, eliminar información redundante e integrar fuentes de información relacionadas puede reducir la carga en la memoria de trabajo (Chen, Kalyuga, & Sweller, 2017). En resumen, el efecto de interactividad de elementos indica que cualquier efecto de la teoría de carga cognitiva puede no obtenerse si la carga intrínseca es muy baja (Chen, Woolcott, & Sweller, 2017).

#### **Aplicación del efecto en programación**

En la literatura no se encuentra evidencia empírica de la aplicación del efecto de interactividad de elementos en el área de programación, sin embargo, se tiene registro de su utilización en el área de matemáticas y otras áreas de las ciencias (Chen, Kalyuga, & Sweller, 2015, 2017).

## Recomendaciones para la aplicación en programación básica

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos.
- Diseño del material con el enfoque a reducción de la carga cognitiva extraña e intrínseca.
- Identificar el nivel de interactividad de elementos asociados tanto con la carga cognitiva intrínseca como con la carga cognitiva extraña.
- Un mismo material instruccional puede representar alto nivel de interactividad para estudiantes novatos y bajo nivel de interactividad para estudiantes con conocimientos previos.
- Plantear subobjetivos de aprendizaje para poder presentar por separado o de forma gradual.
- Reducir la interactividad asociada con la carga cognitiva extraña e intrínseca aplicando técnicas apropiadas de carga cognitiva, como el efecto de elementos aislados, redundancia, atención dividida, modalidad y variabilidad. Y mediante la segmentación de los elementos de la naturaleza de los contenidos para la reducción de la carga cognitiva intrínseca.
- Identificar en base a los subobjetivos elementos para procesar de forma individual a aislada que en conjunto.

## Ejemplo propuesto de aplicación

Como se describe en la definición del efecto, la finalidad de este efecto es disminuir la interactividad de los elementos, y ello se puede realizar combinando la aplicación algunos de los efectos ya descritos por la TCC en el presente documento como son redundancia, atención dividida, elementos aislados. También podemos aplicarlo descomponiendo el contenido en fragmento e ir explicando cada fragmento por separado, por ejemplo:

Si tenemos un ejemplo de un programa básico como el que sigue:

```
#include <iostream>
using namespace std;
int main() {
    int edad;
    cout << "Ingresa tu edad: ";
    cin >> edad;
    cout << "Tu edad es: " << edad << endl;
    return 0;
}
```

A pesar de que es un programa sencillo, para los novatos puede parecer complejo puesto que implica comprender `#include`, `using namespace std`, `cin`, `cout`, variables y `return 0`; de una sola vez. Entonces podemos separar presentando primero ejemplo básico de `cout`, como el hola mundo, explicando su función, y después integrar `cin`, explicando su función. Otras opciones de aplicación se listan:

- Enseñanza de conceptos uno a la vez mediante pseudocódigo o diagramas de flujo antes del código.
- Dividir el código en partes pequeñas y comprensibles.
- Introducir conceptos en orden progresivo (primero lo más simple, luego lo más complejo).
- Mostrar la necesidad de un concepto antes de explicarlo (ejemplo: bucles en lugar de código repetitivo).

#### **6.1.9 Variabilidad (Variability effect)**

El efecto de variabilidad ocurre cuando material instruccional con alta variabilidad se traduce en un mejor aprendizaje que material con poca variabilidad (Likourezos & Kalyuga, 2019; Likourezos, Kalyuga, & Sweller, 2019), ya que un material instruccional con variabilidad motiva al estudiante a construir un conocimiento más general, ya que aumenta la probabilidad de identificar características similares y distinguir las características relevantes de las irrelevantes (Sweller et al., 2019). La aplicación de variabilidad aumenta la interactividad de elementos y carga cognitiva intrínseca, pero también incrementa la transferencia de conocimientos por medio del incremento de la carga cognitiva pertinente (X. Li, 2016), lo anterior si la carga cognitiva total no se excede la capacidad de la memoria de trabajo (Likourezos et al., 2019), esto se puede constatar mediante la evidencia empírica de la aplicación de ejemplos basados en instrucción con alta variabilidad dan mejores resultados en la transferencia de conocimientos (X. Li, 2016; Likourezos & Kalyuga, 2019; Likourezos et al., 2019).

#### **Aplicación del efecto en programación**

En programación la aplicación del efecto de variabilidad se tiene muy poca evidencia empírica de su uso, X. Li (2016) mediante la utilización de efecto de ejemplo resuelto y el efecto de variabilidad para el aprendizaje de programación de aplicaciones en Android mediante el lenguaje de programación Java. La aplicación del efecto de variabilidad como la mayoría de los efectos de la teoría de carga cognitiva la evidencia empírica muestra mayor aplicación en el área de matemáticas o áreas afines a las ciencias (Likourezos & Kalyuga, 2019; Likourezos et al., 2019).

#### **Recomendaciones para la aplicación en programación básica**

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos.
- Diseño del material con el enfoque a reducción de la carga cognitiva extraña como lo marca la TCC.
- Se pueden lograr niveles óptimos de interactividad aumentando la variabilidad de los ejemplos/ejercicios.
- Utilización de material instruccional isomórfico como alternativa de variabilidad.

## Ejemplo propuesto de aplicación

Como se menciona en la descripción del efecto, este puede ser aplicado al modificar las situaciones de los problemas, así como las dimensiones de las tareas a realizar. De ello, para el tema de ciclos se plantean algunos ejercicios isomórficos, para la aplicación del efecto de variabilidad, modificando la situación del problema, así como las tareas a desarrollar.

1. *Dado el siguiente pseudocódigo en PSeInt, relativo al cálculo de la factorial de un número, generar el código en el lenguaje C++.*

```
Algoritmo factorial
    Definir n,x Como Entero
    Definir f Como Real
    Escribir "Ingresa un número"
    leer n
    Si n < 0 Entonces
        Escribir "El número ",n, "no se puede calcular"
    SiNo
        f = 1
        Para x = 1 Hasta n Con Paso 1 Hacer
            f = f * x
        FinPara
        Escribir "El factorial del número ",n," es: ",f
    FinSi
FinAlgoritmo
```

2. *Dado el siguiente código en C++ para el cálculo de la serie Fibonacci de un número dado, generar el pseudocódigo en PSeInt.*

```
#include <stdio.h>

int main()
{
    int x,y,z,cont;

    x=0;
    y=1;

    printf("0\n1\n",z);
    for (cont=1;cont<=5;cont++)
    {
        z=x+y;
        printf("%d\n",z);
        x=y;
        y=z;
    }
    return 0;
}
```



3. Generar el pseudocódigo en PSeInt y luego pasar a código en lenguaje C++, del algoritmo para crear un vector de tamaño definido por el usuario, llenar el arreglo con números y al final mostrar su contenido.
4. A partir de las siguientes salidas en pantalla de un programa en C++, generar el pseudocódigo en PSeInt y su correspondiente código en C++, que produzca el mismo resultado que se muestra, a partir de la captura de un número dado.
- a. Por ejemplo, dado el número 5 genere como salida la figura de un cuadrado dibujado con \* de tamaño 5x5.

```
Numero: 5
*****
*****
*****
*****
*****
*****
```

```
Numero: 5
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

```
Numero: 5
1
22
333
4444
55555
```

```
Numero: 5
12345
1234
123
12
1
```

#### 6.1.10 Imaginación (Imagination effect)

Cuando los estudiantes tienen suficiente experiencia o conocimientos previos, aplicando el efecto de imaginación, esto es imaginando el procedimiento de solución de un problema aprenderán mejor que estudiando el mismo problema ya resuelto (Sweller et al., 2019), pero la aplicación de este efecto es solo para estudiantes con conocimientos previos (Sweller et al., 2011), ya que requiere que el estudiante realice un proceso mental de los procedimientos y conceptos que ya ha trabajado (Sweller et al., 2019).

#### Aplicación del efecto en programación

La evidencia empírica de la aplicación del efecto de imaginación se tiene más en el área de ciencias (L. Cheng & Beal, 2020; H. Wang, Xu, Xie, Dong, & Wang, 2021; Y. Wang, Lin, & Chen, 2021) y matemáticas, en lo que respecta a programación se tiene registro de un trabajo en el 2003 enfocado al aprendizaje del lenguaje HTML (Sweller et al., 2011).

#### Recomendaciones para la aplicación en programación básica

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos.
- Es necesario un nivel básico de experiencia o conocimientos previos por parte del alumno para su aplicación.
- La interactividad de los elementos debe ser alta (carga cognitiva intrínseca) para su aplicación.
- Aplicación en combinación con otros efectos como ejemplo resuelto, problema por completar, modalidad, atención dividida.

## Ejemplo propuesto de aplicación

El algoritmo de ordenamiento por selección consiste en buscar el elemento más pequeño del arreglo e intercambiarlo por el que está en la primera posición y luego entre los restantes se busca de nuevo el más pequeño y se intercambia por el que está en la segunda posición y así sucesivamente con los números restantes.

*A partir de la descripción anterior del algoritmo de ordenación por selección dado el siguiente arreglo:*

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 6 | 5 | 3 | 1 | 8 | 7 | 2 | 4 |
|---|---|---|---|---|---|---|---|

*Imagina el proceso que sigue el algoritmo de ordenamiento y representa de forma manual (dibujos y texto) el proceso de iteraciones hasta ordenar el arreglo. Por último, genera el pseudocódigo en PSeInt y código en C++ de tu representación de la gráfica del algoritmo.*

### 6.1.11 Elementos aislados (Isolated elements effect)

El efecto de elementos aislados surge en el proceso de aprendizaje de información nueva, compleja y con alto nivel de interactividad en sus elementos que generalmente requiere que los alumnos procesen más elementos interactivos de los que la capacidad de la memoria de trabajo puede procesar (Berssanette & de Francisco, 2021). Se ha demostrado que es eficaz reducir la carga cognitiva intrínseca mediante una progresión escalonada del aprendizaje, desde el aprendizaje de elementos o partes más simples y de forma aislada, hasta elementos más complejos y totalmente interactivos (Lu, Kalyuga, & Sweller, 2020).

### Aplicación del efecto en programación

En la literatura la evidencia de la aplicación del efecto de elementos aislados es limitada, de la cual las áreas de aplicación del mismo son en matemáticas e ingeniería (Sweller et al., 2011) y en algunas otras áreas como lo son el aprendizaje de algún idioma (Lu et al., 2020), pero en el área concerniente a la programación no se tiene evidencia de su aplicación.

### Recomendaciones para su aplicación en programación básica

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos.
- La interactividad de los elementos debe ser alta (carga cognitiva intrínseca) para su aplicación.
- Identificar el nivel de interactividad de elementos asociados tanto con la carga cognitiva intrínseca como con la carga cognitiva extraña.
- Identificar si es posible procesar elementos de forma individual a aislada que en conjunto.
- Reducir la interactividad asociada con la carga cognitiva extraña empezando el aprendizaje con elementos o partes simples con poca interactividad e ir aumentando progresivamente a elementos complejos con alto nivel de interactividad.

## **Ejemplo propuesto de aplicación**

La idea de la aplicación del efecto de elementos aislados es una progresión escalonada del aprendizaje, comenzando por elementos o parte más simples hasta elementos más complejos. En programación la aplicación de este enfoque es algo complicado debido a la naturaleza compleja de todos sus conceptos desde los básicos hasta los más avanzados. Sin embargo, se propone escalar el aprendizaje en algunos conceptos graduales, como por ejemplo ciclos y ciclos anidados.

Partiendo de lo anterior para el aprendizaje de ciclos anidados el estudiante deberá primero tener una fase de aprendizaje de ciclos básicos, el material para esta fase se debe diseñar aplicando los efectos mencionados en el presente documento dependiendo del grado de pericia del alumno, y en una fase posterior aprendizaje de ciclos anidados aplicando material de igual forma aplicando los efectos de la TCC, pero ahora los alumnos ya tienen un nivel de pericia mayor en el tema.

### **6.1.12 Reversión de experiencia (Expertise reversal effect)**

Inicialmente predicho por la teoría de carga cognitiva como una forma del efecto de redundancia (Sweller et al., 2011), es una variante del efecto de interactividad de elementos, cuando la experiencia del estudiante crece el nivel de interactividad de los elementos decrece, esto es, los resultados que se pueden obtener con la aplicación de efectos a información con alta interactividad de elementos para estudiantes novatos pueden provocar un efecto negativo en estudiantes con conocimientos o experiencia previa, por ejemplo estudiar a base ejemplos resueltos benefician a los novatos pero no a estudiantes con experiencia (Sweller et al., 2019), entonces conforme la experiencia del estudiante crece es más efecto la resolución de problemas que continuar con el estudio de ejemplos resueltos (Sweller, 2020), de ello los procedimientos de instrucción que facilitan el aprendizaje de los estudiantes pueden obstaculizar a los estudiantes más avanzados (Bounajim et al., 2021).

## **Aplicación del efecto en programación**

Más que la aplicación del efecto mediante material instruccional, la evidencia empírica encontrada en la literatura hace referencia a la presencia o no del efecto de acuerdo al grado de pericia del estudiante, Sullivan, Swanson, and Edwards (2021) en su estudio aplican material instruccional mediante ejercicios básicos de sintaxis en el área de programación, en sus resultados reportan que estudiantes con conocimientos previos encontraron los ejercicios poco útiles y molestos, así como en Bounajim et al. (2021) aplicaron un estudio a estudiantes STEM graduados para el aprendizaje de programación básica mediante un sistema en línea, con heterogeneidad en conocimientos previos en los participantes, donde estudiantes sin previos conocimientos expresan el material era complejo y no estaba bien diseñado para principiantes. Boldbaatar and Şendurur (2019) también hacen referencia a la presencia del efecto de reversión de experiencia en su estudio aplicado es estudiantes con conocimientos previos y novatos, mediante la utilización del ejemplo resuelto, problema por completar y solución de problemas. De igual forma, en el área de matemáticas Chen, Retnowati, and Kalyuga (2019) confirman el efecto de reversión de la experiencia en conjunto con la interactividad de elementos mediante la aplicación del efecto del ejemplo resuelto y solución

de problemas pasando gradualmente de ejemplos resueltos completos a solución de problemas, aumentando así el nivel de experiencia y disminuyendo el nivel de interactividad de elementos.

### **Recomendaciones para su aplicación en programación básica**

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos.
- Identificar el nivel de interactividad de elementos asociados tanto con la carga cognitiva intrínseca como con la carga cognitiva extraña.
- Diseñar el material instruccional con enfoque a la experiencia de los alumnos para evitar la presencia del efecto de reversión de la experiencia. Utilizar efectos para estudiantes con conocimientos previos, como, auto-explicación, imaginación, problema por completar, autogestión, variabilidad, desvanecimiento de guía, solución de problemas, así como evitar la redundancia y ejemplos resueltos completos.

### **Ejemplo propuesto de aplicación**

Como se menciona anteriormente, el efecto de reversión de experiencia se da cuando el estudiante adquiere pericia sobre algún conocimiento particular, entonces la aplicación de efectos como el ejemplo resuelto completos, problema por completar básicos, atención dividida o modalidad, que resultan útiles para los novatos, pueden provocar efectos opuestos, interfiriendo con el aprendizaje. De ello, es recomendable para estudiantes con conocimientos básicos o un nivel de pericia mayor a novatos, implementar efectos como auto-explicación, problemas por completar avanzados o variantes de estos, imaginación, autogestión, variabilidad, desvanecimiento de la instrucción, así como la resolución completa de problemas y sus variantes de aplicación. De tal suerte, como propuesta de aplicación sería la utilización de algunos de los ejemplos de los efectos mencionados para estudiantes con conocimientos previos, y descritos en el presente documento, o la combinación de dos o más de ellos.

#### **6.1.13 Desvanecimiento de guía (Guidance-fading effect)**

El efecto de desvanecimiento de guía está estrechamente relacionado con los efectos de reversión de la experiencia e interactividad de los elementos para los cuales el efecto de redundancia es importante también (Sweller et al., 2019). La aplicación de este efecto es importante para programas educativos de larga duración, donde el estudiante conforme avanza el curso adquiere conocimientos y experiencia acerca del tema (Paas & van Merriënboer, 2020), ya que los métodos instruccionales que necesita un estudiante novato son diferentes a los que necesita un estudiante más avanzado (Paas & van Merriënboer, 2020; Sweller et al., 2019). De ello el efecto de desvanecimiento de guía proporciona una forma de adaptar adecuadamente un entorno de aprendizaje con base en el incremento de la pericia de los estudiantes (Chen, Kalyuga, & Sweller, 2017), comenzando con la aplicación de ejemplos resueltos, seguidos de problemas por completar (incrementando gradualmente las partes a completar) y al final con problemas convencionales (Sweller et al., 2019).

## Aplicación del efecto en programación

Evidencia de su aplicación se encuentra más en áreas como las matemáticas y ciencias ((Chen et al., 2019; Miller-Cotto & Auxter, 2021), encontrando también en la literatura su aplicación en el diseño de sistemas de aprendizaje en línea aplicando los lineamientos de este y otros efectos de la TCC (Chen, Kalyuga, & Sweller, 2017; Darejeh et al., 2022), y en el área de programación con algunos trabajos reportados (Velazquez-Iturbide, 2017).

## Recomendaciones para su aplicación en programación básica

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos al inicio del curso e ir monitoreando el avance para la modificación del material instruccional.
- Identificar el nivel de interactividad de elementos asociados tanto con la carga cognitiva intrínseca como con la carga cognitiva extraña.
- Diseñar el material instruccional con enfoque a la experiencia de los alumnos para evitar la presencia del efecto de reversión de la experiencia conforme avanza el curso.
- Desvanecer la instrucción conforme aumenta la pericia del estudiante, pasando de ejemplos totalmente resueltos con la mayor explicación posible a problemas por completar incrementando gradualmente la cantidad de tareas a completar (se puede combinar con el efecto de auto-explicación), y finalmente pasar a problemas para solucionar por completo, combinando de igual manera con el efecto de auto-explicación o imaginación para un mejor resultado.

## Ejemplo propuesto de aplicación

Tomando el tema de conceptos básicos de ciclos en lenguaje de programación C++, se plantea la siguiente propuesta para la aplicación del efecto de desvanecimiento de la instrucción.

Para se proponen 3 fases del proceso, en la primer fase la aplicación del ejemplo resuelto con el mayor detalle posible de explicación, segunda fase cuando el estudiante ya cuenta con el conocimiento presentado en la primer fase, mediante la utilización de problemas por completar (esta fase se puede dividir en varias fases, dependiendo del aumento o no de la pericia del estudiante, presentando los ejercicios por completar incrementando de forma gradual el número de tareas a completar o resolver por parte del estudiante y disminuyendo de igual forma las opciones de ayuda a los estudiantes), y por último la tercera fase pasando a la resolución completa de problemas y sus variantes de aplicación sin ninguna opción de ayuda para el estudiante.

### *Fase 1. Ejemplo resuelto*

*Se desea crear dos algoritmos en PSeInt que muestre en pantalla los números pares comprendidos entre el número 50 y 100, utilizando las instrucciones Mientras y Para. Después pasar estos algoritmos a lenguaje C++, con While y For. Solución:*

*Fase 2. Problema por completar (1ra iteración, problema isomórfico al ejemplo resuelto con partes a completar, pero con ayuda en los comentarios)*



```
//Algoritmo para imprimir los números pares de un rango de 50 a 100
//Planes de programación: ciclos
Proceso ParesM
    definir contador Como Entero; //se define contador como entero
    contador = 50; //se asigna el valor de 50, que es el inicio del rango planteado en el problema
    Mientras contador <= 100 Hacer // el ciclo se ejecuta mientras la expresión contador menor o igual a 100
        Escribir "el número ",contador, " es par"; //se escribe el número par en pantalla
        contador=contador+2; //para continuar con el siguiente número, se incrementa su valor en 2
    FinMientras
FinProceso

//Algoritmo para imprimir los números pares de un rango de 50 a 100
//Planes de programación: ciclos
Proceso ParesP
    definir contador Como Entero; //se define contador como entero
    Para contador=50 Hasta 100 Con Paso 2 Hacer // el ciclo comienza en contador=50 hasta 100 incrementando de 2 el contador
        Escribir "el número ",contador, " es par"; //se escribe el número par en pantalla
    FinPara
FinProceso

#include <stdio.h> //llamado de la librería stdio.h para utilizar
//la función de impresión en pantalla
int main() {
    int contador; //declaración contador como entero
    contador = 50; //asignamos el valor de 50 a la variable contador

    while (contador<=100){ //inicio ciclo, se ejecuta mientras contador<=100
        printf("El numero: %d es par\n",contador); //imprime en pantalla el valor de contador
        contador+=2; //incrementa el valor de contador en 2
    }

    return 0;
}

#include <stdio.h> //llamado de la librería stdio.h para utilizar
//la función de impresión en pantalla
int main() {
    int contador; //declaración contador como entero

    for(contador=50;contador<=100;contador+=2){ //inicio ciclo, contador=50 hasta <=100, incrementa en 2
        printf("El numero: %d es par\n",contador); //imprime en pantalla el valor de contador
        //de cada iteración
    }

    return 0;
}
```

Se desea crear dos algoritmos en PSeInt que muestre en pantalla el factorial de un número dado por el usuario, utilizando las instrucciones Mientras y Para. Después pasar estos algoritmos a lenguaje C++, con While y For. A continuación, se muestra una solución incompleta, por favor completa la solución:

```
//Algoritmo para calcular el factorial de un número dado
//Planes de programación: ciclos
Proceso FactorialM
    definir num, fac Como Entero; //declaración de las variables
    fac=1; //Asignar el valor de 1 a la variable fac
    Escribir "Teclee el número para obtener su factorial: "; //solicitar al usuario un número
    Leer num; //Guardar el número en la variable num
    Mientras [ ] Hacer // el ciclo se ejecuta mientras la variable de control cumpla la condición
        fac=[ ]; // se hace el cálculo de cada iteración y se guarda en la variable fac
        num=[ ]; //se modifica la variable de control del ciclo
    FinMientras
    Escribir "El factorial es: [ ]"; //se imprime el factorial en pantalla
FinProceso

//Algoritmo para calcular el factorial de un número dado
//Planes de programación: ciclos
Proceso FactorialP
    definir num, fac Como Entero; //declaración de las variables
    fac=1; //Asignar el valor de 1 a la variable fac
    Escribir "Teclee el número para obtener su factorial: "; //solicitar al usuario un número
    Leer num; //Guardar el número en la variable num
    Para [ ] Hacer // el ciclo se ejecuta desde hasta con paso de [ ]
        [ ]; // se hace el cálculo de cada iteración y se guarda en la variable fac
    FinPara
    Escribir [ ]; //se imprime el factorial en pantalla
FinProceso
```

Fase 2: Problema por completar (2da iteración, problema isomórfico al ejemplo resuelto con más partes a completar que el anterior, sin comentarios de ayuda )

```
#include <iostream>

using namespace std;

int main() {
    int fac=1, num, i; //declaración de las variables, asignar el valor de 1 a la variable fac
    cout << "Dame un numero: "; //solicitar al usuario un número
    cin >> num; //Guardar el numero en la variable num

    for (i=1; i<=num; i++) { // el ciclo se ejecuta desde hasta con paso de 1
        fac = fac * i; // se hace el calculo de cada iteración y se guarda en la variable fac
    }
    cout << "El factorial de " << num << " es: " << fac << endl; //se imprime el factorial en pantalla
    return 0;
}

#include <iostream>

using namespace std;

int main() {
    int fac=1, num; //declaración de las variables, asignar el valor de 1 a la variable fac
    cout << "Dame un numero: "; //solicitar al usuario un número
    cin >> num; //Guardar el numero en la variable num

    while (num > 0) { // el ciclo se ejecuta mientras la variable control cumpla la condicion
        fac = fac * num; // se hace el calculo de cada iteración y se guarda en la variable fac
        num--; //se modifica la variable de control del ciclo
    }
    cout << "El factorial de " << num << " es: " << fac << endl; //se imprime el factorial en pantalla
    return 0;
}
```

Se desea crear dos algoritmos en PSeInt que muestre en pantalla la serie Fibonacci de acuerdo con un número dado por el usuario, utilizando las instrucciones Mientras y Para. Después pasar estos algoritmos a lenguaje C++, con While y For. A continuación, se muestra una solución incompleta, por favor completa la solución:

```
//Algoritmo para imprimir la serie Fibonacci de acuerdo a un número dado
//Planes de programación: ciclos
Proceso FibonacciM
    definir a, b, c, cont Como Entero;
    a=0;
    b=1;
    Escribir "Teclee la cantidad de numeros para la serie: ";
    Leer cont;
    Mientras (cont > 0) Hacer
        c=a+b;
        a=b;
        b=c;
        Escribir c;
    FinMientras
FinProceso
```

```
//Algoritmo para imprimir la serie Fibonacci de acuerdo a un número dado
//Planes de programación: ciclos
Proceso FibonacciP
    definir a, b, c, cont Como Entero;
    a=0;
    b=1;
    Escribir "Teclee la cantidad de numeros para la serie: ";
    Leer cont;
    Para (i=1) Hasta (cont) Hacer
        c=a+b;
        a=b;
        b=c;
        Escribir c;
    FinPara
FinProceso
```

```

#include <iostream>
using namespace std;
int main()
{
    int x,y,z,cont,i;
    x=0;
    y=1;
    cout << "Teclee la cantidad de numeros para la serie: ";
    cin >> cont;
    for ( )
    {
        ;
        ;
        ;
        ;
    }
    return 0;
}

```

```

#include <iostream>
using namespace std;
int main()
{
    int x,y,z,cont;
    x=0;
    y=1;
    cout << "Teclee la cantidad de numeros para la serie: ";
    cin >> cont;
    while ( )
    {
        ;
        ;
        ;
        ;
    }
    return 0;
}

```

### *Fase 3: Resolución problema completo*

*Genera los algoritmos en PSeInt para imprimir la tabla de multiplicar de un número dado por el usuario, utilizando las instrucciones Mientras y Para. Después pasar estos algoritmos a lenguaje C++, con While y For. Nota: Se puede combinar con el efecto de auto-explicación solicitando al estudiante explique paso a paso sus soluciones.*

#### **6.1.14 Memoria colectiva de trabajo (Collective working memory effect)**

La memoria colectiva de trabajo descrita (Sweller et al., 2019) hace referencia a que un grupo de estudiantes trabajando colaborativamente puede considerarse como un sistema único de procesamiento de información con múltiples memorias de trabajo y diferentes niveles de conocimientos previos, los cuales pueden crear un espacio colectivo más efectivo que un solo estudiante y tener mejores resultados de aprendizaje (Sweller et al., 2011), ya que no todos los integrantes del grupo de trabajo necesitan procesar toda la información al mismo tiempo dividiéndose ésta entre el grupo (Janssen & Kirschner, 2020), así cuando algún estudiante tiene un déficit en un tema particular, puede recibir ayuda de sus compañeros, adquiriendo conocimientos, creando dependencia unos de otros para generar y adquirir conocimientos (Janssen & Kirschner, 2020; Paas & van Merriënboer, 2020; Sweller et al., 2019).

La carga cognitiva intrínseca generada por la complejidad de las tareas o información a aprender en un proceso de transferencia de conocimientos se puede dividir entre los integrantes de un grupo de aprendizaje colaborativo, aumentando así la capacidad cognitiva, sin embargo se requiere comunicación y coordinación entre los integrantes en el proceso, lo que deriva en un esfuerzo cognitivo adicional (Sweller et al., 2019), pero buen equilibrio entre la división del procesamiento de la información y el esfuerzo mental de las actividades transactivas derivan

en el efecto de la memoria colectiva de trabajo (Janssen & Kirschner, 2020).

### **Aplicación del efecto en programación**

Janssen and Kirschner (2020) proponen la aplicación de la teoría de carga cognitiva mediante el efecto de memoria colectiva de trabajo en el aprendizaje colaborativo asistido por computadora mediante pautas propuestas de acuerdo con la teoría, que puede ser aplicada a cualquier área del conocimiento. De igual forma Kirschner et al. (2018) presentan una investigación de la aplicación del efecto de memoria colectiva de trabajo aplicado al aprendizaje colaborativo asistido por computadora, proponiendo principios a seguir en su aplicación. Por otro lado Zambrano R., Kirschner, Sweller, and Kirschner (2019) presentan un estudio aplicando el efecto de memoria colectiva de trabajo en el área de matemáticas, reportando mejores resultados en el aprendizaje que en el aprendizaje individual. En lo que respecta al área de programación no se tiene mucha evidencia empírica de su aplicación en la literatura, un ejemplo de su aplicación L. Zhang, Kalyuga, Lee, and Lei (2016) mediante una plataforma de aprendizaje web, aplicaron el efecto de memoria colectiva de trabajo en comparación con el aprendizaje individual, en dos diferentes formas, con grupos de estudiantes con conocimientos homogéneos y otros de conocimientos heterogéneos, para aprendizaje de temas complejos en programación, reportando mejores resultados en los grupos con conocimientos heterogéneos.

### **Recomendaciones para su aplicación en programación básica**

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos para la formación de grupos de trabajo y el diseño del material instruccional.
- Identificar el nivel de interactividad de elementos asociados tanto con la carga cognitiva intrínseca como con la carga cognitiva extraña.
- Diseñar el material instruccional con enfoque a la experiencia de los alumnos para evitar la presencia del efecto de reversión de la experiencia conforme avanza el curso.
- De preferencia formar grupos de trabajo con conocimientos previos heterogéneos.
- Las tareas de aprendizaje deben implicar altos niveles de interactividad para poder aplicar un aprendizaje colaborativo eficaz.
- Para tareas con bajos niveles de interactividad el aprendizaje individual es más efectivo.
- Un factor muy importante para tomar en cuenta es la carga cognitiva generada por el costo de comunicación y coordinación de los integrantes del grupo, si esta carga es excede la reducción de la carga cognitiva asociada con la colaboración, es decir, si el beneficio de dividir las tareas/información no es proporcional o mayor que el esfuerzo que representar el poder coordinar y comunicar entre sí, su aplicación no será efectiva.

### **Ejemplo propuesto de aplicación**

La aplicación del efecto de memoria colectiva de trabajo es muy importante la formación de los grupos de trabajo, para ello se debe identificar el conocimiento previo que tiene cada estudiante y pueda contribuir a la generación de nuevos conocimientos en los demás compañeros,

además controlar temas de coordinación y comunicación entre ellos para evitar esto influya de forma negativa generando una carga cognitiva adicional en el proceso.

Las tareas de aprendizaje deben implicar niveles altos de complejidad, esto es que de preferencia planteamiento de problemas de tamaño considerable que impliquen la utilización de dos o más conceptos de programación para su resolución. Entonces con base en lo planteado se proponen los siguientes ejemplos de problemas:

*Escriba un programa que cuente el número de operadores no igual (!=) en un archivo que contiene un programa en C++. Se puede hacer contando sólo el número de veces que aparece un signo de cierre de admiración (!) seguido de un signo igual (=) en la entrada. Una manera de hacer esto es leer en el archivo de entrada un carácter a la vez, siguiendo la pista de los dos caracteres más recientes, el valor actual y el valor previo. En cada iteración del ciclo, se lee un nuevo valor actual y el valor anterior se convierte en el valor previo. Cuando se llega al final del archivo, el ciclo termina. Ejemplo tomado de (Weems & Naps, 2007)*

El problema planteado implica temas de variables, condicionales, lectura de archivos y ciclos, de ello es importante formar los grupos de trabajo con base a los conocimientos previos de estos temas en forma heterogénea, es decir, que no todos tengan conocimientos previos sobre todos los temas, o en su defecto que no tengan conocimientos sólidos sobre todos, para que se complementen unos con otros.

Se proponen plantear equipos con problemas diferentes y que varíen un poco en los conceptos de programación, por ejemplo:

*Escribir un programa que visualice un cuadrado mágico de orden impar  $n$  comprendido entre 3 y 11; el usuario debe elegir el valor de  $n$ . Un cuadrado mágico se compone de números enteros comprendidos entre 1 y  $n$ . La suma de los números que figuran en cada fila, columna y diagonal son iguales. Ejemplo tomado de (Weiss Allen, 2013).*

8 1 6  
3 5 7  
4 9 2

En este segundo problema implica temas de variables, condicionales, arreglos bidimensionales, ciclos. La designación de los grupos sería de igual forma basándose en el nivel de conocimientos de los estudiantes procurando en todo momento que se complementen unos a otros para generar conocimiento.

Posterior a la resolución de los problemas plantear una explicación a los demás equipos sobre su problema en particular explicando a detalle su solución.

#### **6.1.15 Información transitoria (Transient information effect)**

Se define el efecto de información transitoria como una pérdida de aprendizaje debido a que la información desaparece antes de que el alumno tenga tiempo de procesarla (Sweller et al., 2011), por ejemplo en presentaciones orales o mediante el uso de videos o animaciones instruccionales (Sweller et al., 2019), el texto hablado así como videos o animaciones instruccionales son transitorios, es decir que desaparecen después de unos pocos segundos en un proceso de enseñanza (Chen, Kalyuga, & Sweller, 2017), pero si la información es extensa



TESIS TESIS TESIS TESIS TESIS

y compleja, requiere que los alumnos cuenten con más tiempo para poder revisarla varias veces y poder asimilarla (Sweller, 2020), por lo cual información extensa, compleja y con alto nivel de interactividad de elementos no debe ser presentada en forma de texto hablado, videos o animaciones, ya que puede reflejarse en un decremento del aprendizaje comparado con la utilización de información no transitoria (Chen, Woolcott, & Sweller, 2017). Se pueden utilizar estrategias para el diseño y aplicación de información transitoria que eliminen los efectos negativos descritos, como son el auto ritmo o la segmentación (Sweller et al., 2019). Este efecto es particularmente relevante en sistemas de aprendizaje asistidos por computadora. La tecnología instruccional moderna nos permite fácilmente convertir información permanente en información transitoria como son audios, videos, animaciones, pero esta facilidad no justifica por si sola su uso, por ello se debe considerar estrategias para su diseño y aplicación (Chen, Kalyuga, & Sweller, 2017; Chen et al., 2019; Chen, Woolcott, & Sweller, 2017; Sweller, 2020).

### **Aplicación del efecto en programación**

La revisión de literatura no arroja resultados de su aplicación en el área de programación, sin embargo, evidencia de su estudio y aplicación se encuentra en diferentes áreas como son matemáticas y física (Castro-Alonso, de Koning, Fiorella, & Paas, 2021; Y. C. Lin et al., 2022; Y. T. Lin et al., 2022; Liu, Lin, Hsu, Hsu, & Paas, 2021), que presentan estudios sobre su aplicación mediante la aplicación de diferentes estrategias para el diseño del material instruccional con enfoque a la información transitoria.

### **Recomendaciones para su aplicación en programación básica**

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos para el diseño del material instruccional.
- Identificar el nivel de interactividad de elementos asociados tanto con la carga cognitiva intrínseca como con la carga cognitiva extraña.
- La información transitoria:
  - Texto hablado (audio) debe ser corto y con bajo nivel de complejidad.
  - Video, proporcionar acceso al estudiante para que pueda revisar el material cuantas veces lo requiera, utilizando la estrategia de énfasis o señalización en la información importante.
  - Animaciones, utilizar la estrategia de énfasis o señalización en la información importante, configurar las animaciones con suficiente tiempo para que el estudiante pueda asimilar la información.
  - Incorporar las estrategias de Self-pacing (autoritmo), que estudiante tenga control sobre la presentación de la información y Segmentation (segmentación), si la información transitoria es muy extensa dividir en segmentos.
  - Se puede mejorar atendiendo los principios de diseño de los efectos de atención dividida, modalidad y redundancia.

## Ejemplo propuesto de aplicación

Como se menciona en la sección de recomendaciones, el enfoque del efecto de la información transitoria va hacia el diseño de material que incluye información no permanente o fija, es decir explicaciones orales, videos, animaciones, las cuales si están cargadas con alto nivel de interactividad y el tiempo de presentación al alumno es muy poco, la transferencia de conocimiento será muy deficiente. Se presenta un ejemplo de ello mediante el algoritmo de ordenamiento de la burbuja, la idea es presentar el material con una descripción teórica mediante audio, una descripción gráfica mediante animación, y el código explicándolo igualmente mediante audio. Tomando las recomendaciones de diseño propuestas en el efecto:

- Se plantea la definición del algoritmo mediante audio, el cual debe ser corto, con información precisa y concreta.
- Explicación gráfica mediante una animación, esta debe ser con suficiente tiempo y utilizando la técnica de señalización para identificar puntos importantes, en cada iteración para que el estudiante sea capaz de procesar cada secuencia. Se añade un audio explicando las iteraciones de la animación, este debe ser corto, con información precisa y concreta.
- Al final se presenta el código en el lenguaje C++, acompañado de audio con su explicación correspondiente. Se puede agregar señalización al código para resaltar puntos importantes.

| Algoritmo<br>ordenamiento de<br>la burbuja                                          | Explicación gráfica (Animación)                                                                                                                                                                                                                                                        | Código                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Audio corto con información precisa y concreta con explicación del algoritmo</p> | <p>6 5 3 1 8 7 2 4</p> <p>1era iteración:</p> <pre>for i = 0; TAM = 8; for j = 0; vector[j] = 6 vector[j+1] = 5 if vector[j] &gt; 5 vector[j] = 5 vector[j+1] = 6</pre> <p>5 6 3 1 8 7 2 4</p> <p>Audio corto con información precisa y concreta con explicación de cada iteración</p> | <pre>for (int i = 0; i &lt; TAM; i++){     for (int j = 0; j &lt; TAM-1; j++){         if(vector[j] &gt; vector[j+1]){             temporal = vector[j];             vector[j] = vector[j+1];             vector[j+1] = temporal;         }     } }</pre> <p>Audio corto con información precisa y concreta con explicación del código</p> |

### 6.1.16 Movimiento humano (Human movement effect)

A partir de lo planteado por efecto de información transitoria el cual se presenta cuando la información transitoria es muy extensa y compleja en la mayoría de los casos provoca menor eficiencia en el aprendizaje que con la utilización de información estática (Sweller et al., 2019), la razón por la cual la inconsistencia en la efectividad de la información transitoria es debido a su propia naturaleza (Kátaí & Osztaián, 2022). Sin embargo, el efecto de movimiento humano plantea que es mejor la utilización de animaciones en lugar de información estática cuando las tareas cognitivas a enseñar implican el movimiento humano (Sweller et al., 2019).

### Aplicación del efecto en programación

Como se indica en su definición implica movimiento humano en su diseño de aplicación, por lo cual evidencia de su aplicación es más común encontrar en tareas o habilidades motoras

humanas (Sweller et al., 2019). En programación se encuentran algunas evidencias empíricas de su aplicación (Káтай & Osztián, 2021, 2022,?, 2023), para el aprendizaje de algoritmos de ordenamiento representando mediante el movimiento humano a través de la danza el comportamiento de dichos algoritmos. Recomendaciones para su aplicación en programación básica

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Planteamiento del material de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos para el diseño del material instruccional.
- Identificar el nivel de interactividad de elementos asociados tanto con la carga cognitiva intrínseca como con la carga cognitiva extraña.
- Diseñar animaciones realistas a través de movimientos, relacionando los conceptos a aprender con movimientos del cuerpo humano, de forma que exista una coherencia entre los movimientos y los conceptos a aprender.

### **Ejemplo propuesto de aplicación**

La aplicación del efecto de movimiento indica que se debe relacionar los conceptos con movimientos humanos, de ello la dificultad de aplicarlo en áreas que no implican el aprendizaje de habilidades motoras humanas. De lo anterior surge una idea de aplicación en un tema específico en programación como son las estructuras de datos como sigue.

*El profesor puede explicar los conceptos de estructuras de datos utilizando elementos u objetos tangibles simulando su comportamiento mediante el movimiento de esos objetos con sus manos, por ejemplo, la estructura de datos pila, aquí puede explicar su definición y comportamiento apilando objetos e indicando sus operaciones básicas y demás características.*

#### **6.1.17 Autogestión (Self-management effect)**

El efecto de autogestión sugiere que los estudiantes pueden ser instruidos para utilizar activamente estrategias para gestionar la carga en su memoria de trabajo durante el aprendizaje (S. Zhang, de Koning, & Paas, 2022) aplicando estas estrategias mediante los efectos de la teoría de carga cognitiva ellos mismos (Sweller et al., 2019). Este efecto surge a partir de la globalización del conocimiento donde los estudiantes no solo tienen acceso a los materiales diseñados por el instructor sino también a todos los disponibles en internet (Sweller et al., 2019), pero la mayoría de estos materiales no están diseñados de forma óptima (S. Zhang et al., 2022), de ello se les debe enseñar a los estudiantes a cómo mejorar estos mal diseños ellos mismos para un mejor rendimiento en el aprendizaje (Eitel, Endres, & Renkl, 2020).

### **Aplicación del efecto en programación**

De acuerdo con la literatura el efecto de autogestión solo ha sido estudiado en combinación el efecto de atención dividida, donde la mayoría de estos estudios se basan en tres condiciones experimentales, en dos fases. En la primera fase, primero materiales instruccionales con en formato de atención dividida, segundo, autogestión, los alumnos son instruidos de como optimizar el material, por ejemplo, reorganizando texto y diagramas, y tercero, se le proporciona

el material integrado por el instructor para estudiar y comparar con su integración. En la segunda fase se le presentan materiales instruccionales en formato de atención dividida para que aplique las estrategias aprendidas para optimizar el material (Sweller, 2020). Reportes de algunos estudios han demostrado los beneficios de diferentes estrategias de autogestión para el aprendizaje en diferentes áreas como matemáticas, ciencias y contabilidad (Castro-Alonso et al., 2021; de Koning, Rop, & Paas, 2020a, 2020b; Roodenrys, Agostinho, Roodenrys, & Chandler, 2012; Sithole, Chandler, Abeysekera, & Paas, 2017; S. Zhang et al., 2022), todos ellos en combinación con el efecto de atención dividida, pero la literatura nos dice que además de la combinación con efecto de atención dividida se puede combinar también con el efecto de redundancia e información transitoria, aunque se reportan pocos estudios, todos ellos muestran resultados positivos de su aplicación (Eitel et al., 2020)(Eitel et al., 2020). En lo que respecta al área de programación no se tiene evidencia en la literatura de su aplicación.

### **Recomendaciones para su aplicación en programación básica**

- Tener bien definida la meta de aprendizaje que se pretende alcanzar.
- Descripción de los principios/estrategias para optimizar materiales instruccionales de forma clara y precisa.
- Conocer el grado de conocimientos previos de los alumnos para la formación de grupos de trabajo y el diseño del material instruccional.
- Identificar los problemas en los materiales instruccionales mal diseñados: demasiado texto, presenta texto y visualizaciones por separado, contiene información redundante o no esencial, no enfatiza la información importante, muestra demasiada información transitoria.
- Utilizar estrategias (Castro-Alonso et al., 2021) para mitigar los problemas encontrados en los materiales instruccionales, como lo son:
  - Complementar texto con visualizaciones.
  - Creación de mapas, dibujos, graficas o demás visualizaciones que acompañen al texto.
  - Integrar la información escrita y visual.
  - Integrar textos dentro las visualizaciones.
  - Eliminar información no esencial, proporcionar textos y visualizaciones concretas y comprensibles, proporcionar narraciones sin texto en pantalla.
  - Generar resúmenes textuales o visuales.
  - Señalar información esencial (con o sin elementos añadidos).
  - Subrayar o resaltar textos o visualizaciones.
  - Segmentar visualizaciones dinámicas.
  - Controle el ritmo de las visualizaciones dinámicas.

### **Ejemplo propuesta aplicación**

Con base en la descripción del efecto y su aplicación revisado en la literatura, se propone aplicar dos fases en el proceso:

1. Fase de entrenamiento

- Proporcionar material instruccional con problemas de atención dividida, redundancia o demasiada información transitoria.
- Identificar los problemas dentro del material instruccional.
- Plantear y explicar al estudiante las estrategias propuestas para atacar los problemas en el material.
- El estudiante aplica las estrategias al material para generar un nuevo material.
- Se otorga al estudiante el material optimizado por el instructor para estudio y comparación con su aplicación.

## 2. Fase de transferencia de conocimientos

- Se proporciona nuevo material instruccional con problemas utilizando un tema diferente al anterior.
- El estudiante aplica las estrategias aprendidas.
- Evaluación tanto de la transferencia de conocimientos en el tema del material instruccional, así como de la transferencia de conocimientos de las estrategias de optimización del material.

Partiendo de lo anterior se propone un ejemplo:

*Fase de entrenamiento:*

*a. Material instruccional con problemas*

*Ordenamiento de arreglos de tipo entero utilizando método de la burbuja.*

*El Ordenamiento de burbuja (Bubble Sort en inglés) es un sencillo algoritmo de ordenamiento. Funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado. Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista esta ordenada. También es conocido como el método del intercambio directo. Dado que solo usa comparaciones para operar elementos, se lo considera un algoritmo de comparación, siendo el más sencillo de implementar.*

*El método de ordenación por burbuja se basa en comparaciones sucesivas de dos elementos consecutivos y realizar un intercambio entre los elementos hasta que queden ordenados. Por ejemplo, tenemos el vector:  $\text{int } v[] = \{3, 34, 1, 53, 15, 6\}$ ; Para realizar la ordenación seguimos estos pasos:*

1. *Se comparan los dos primeros elementos, si el segundo es superior al primero, se dejan tal como están, pero si el primero es el más grande, se intercambian los elementos.*

|                | Posición |    |   |    |    |   |
|----------------|----------|----|---|----|----|---|
|                | 0        | 1  | 2 | 3  | 4  | 5 |
| Datos v        | 3        | 34 | 1 | 53 | 15 | 6 |
| 1ª Comparación | 3        | 34 | 1 | 53 | 15 | 6 |

2. *A continuación, se compara el segundo elemento con el tercero aplicando los mismos criterios del paso anterior.*



|                | Posición |    |    |    |    |   |
|----------------|----------|----|----|----|----|---|
|                | 0        | 1  | 2  | 3  | 4  | 5 |
| Datos v        | 3        | 34 | 1  | 53 | 15 | 6 |
| 1ª Comparación | 3        | 34 | 1  | 53 | 15 | 6 |
| 2ª Comparación | 3        | 1  | 34 | 53 | 15 | 6 |

3. De esta forma se repite la operación de comparación con todos los elementos que forman el vector, y se obtiene un vector con la primera ordenación, quedando el elemento mayor al final del vector.

|                | Posición |    |    |    |    |    |
|----------------|----------|----|----|----|----|----|
|                | 0        | 1  | 2  | 3  | 4  | 5  |
| Datos v        | 3        | 34 | 1  | 53 | 15 | 6  |
| 1ª Comparación | 3        | 34 | 1  | 53 | 15 | 6  |
| 2ª Comparación | 3        | 1  | 34 | 53 | 15 | 6  |
| 3ª Comparación | 3        | 1  | 34 | 53 | 15 | 6  |
| 4ª Comparación | 3        | 1  | 34 | 15 | 53 | 6  |
| 5ª Comparación | 3        | 1  | 34 | 15 | 6  | 53 |

4. Se repite la operación de ordenación (comparación) de todos los elementos del vector excepto con el último.

|                | Posición |   |    |    |    |    |
|----------------|----------|---|----|----|----|----|
|                | 0        | 1 | 2  | 3  | 4  | 5  |
| Datos v        | 3        | 1 | 34 | 15 | 6  | 53 |
| 1ª Comparación | 1        | 3 | 34 | 15 | 6  | 53 |
| 2ª Comparación | 1        | 3 | 34 | 15 | 6  | 53 |
| 3ª Comparación | 1        | 3 | 15 | 34 | 6  | 53 |
| 4ª Comparación | 1        | 3 | 15 | 6  | 34 | 53 |

5. 5. Reiterando el proceso descrito, el vector quedará ordenado cuando solo se comparen los dos primeros elementos.

|                | Posición |   |    |    |    |    |
|----------------|----------|---|----|----|----|----|
|                | 0        | 1 | 2  | 3  | 4  | 5  |
| Datos v        | 1        | 3 | 15 | 6  | 34 | 53 |
| 1ª Comparación | 1        | 3 | 15 | 6  | 34 | 53 |
| 2ª Comparación | 1        | 3 | 15 | 6  | 34 | 53 |
| 3ª Comparación | 1        | 3 | 6  | 15 | 34 | 53 |

|                | Posición |   |   |    |    |    |
|----------------|----------|---|---|----|----|----|
|                | 0        | 1 | 2 | 3  | 4  | 5  |
| Datos v        | 1        | 3 | 6 | 15 | 34 | 53 |
| 1ª Comparación | 1        | 3 | 6 | 15 | 34 | 53 |
| 2ª Comparación | 1        | 3 | 6 | 15 | 34 | 53 |

|                | Posición |   |   |    |    |    |
|----------------|----------|---|---|----|----|----|
|                | 0        | 1 | 2 | 3  | 4  | 5  |
| Datos v        | 1        | 3 | 6 | 15 | 34 | 53 |
| 1ª Comparación | 1        | 3 | 6 | 15 | 34 | 53 |

```
for (int i = 0; i < TAM; i++){
    for (int j = 0; j < TAM-1; j++){
        if(vector[j] > vector[j+1]){
            temporal = vector[j];
            vector[j] = vector[j+1];
            vector[j+1] = temporal;
        }
    }
}
```

Código en C++, para aplicar el algoritmo de burbuja mediante ciclos anidados For.  
Problemas identificados

- Redundancia en el texto
- Imágenes y texto separados
- Atención dividida en las imágenes de las iteraciones y código
- No se tiene una explicación del funcionamiento del código
- Plantear y explicar al estudiante las estrategias propuestas en la sección de recomendaciones
- El estudiante aplica las estrategias al material para generar un nuevo material
- Se otorga al estudiante el material optimizado por el instructor para estudio y comparación con su aplicación

| Descripción                                                                                                                                                                                                                                                                                                                                                                                | Descripción gráfica                                                                                                                              | Código en C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| El Ordenamiento de burbuja (Bubble Sort en inglés) es un sencillo algoritmo de ordenamiento. Funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado. Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada. | <p>Se continúa el proceso de comparación con cada par siguiente de números. Se realizan más iteraciones hasta que el arreglo quede ordenado.</p> | <pre>for (int i = 0; i &lt; TAM; i++){     for (int j = 0; j &lt; TAM-1; j++){         if(vector[j] &gt; vector[j+1]){             temporal = vector[j];             vector[j] = vector[j+1];             vector[j+1] = temporal;         }     } }</pre> <p>El primer ciclo corresponde a las iteraciones del algoritmo el cual indica que se aplicará el número de iteraciones de acuerdo con el tamaño del arreglo. El segundo ciclo corresponde a las comparaciones, donde en cada una de las iteraciones del primer ciclo realizará la comparación de cada par consecutivo de elementos del arreglo. Si el primero es mayor los intercambia de posición en el arreglo.</p> |

Fase de transferencia de conocimientos

- Se proporciona nuevo material instruccional con problemas de redundancia, atención dividida o información transitoria, utilizando un tema diferente al anterior.
- El estudiante aplica las estrategias aprendidas.
- Evaluación tanto de la transferencia de conocimientos en el tema del material instruccional, así como de la transferencia de conocimientos de las estrategias de optimización del material.

## CAPÍTULO 7

### DISEÑO Y DESARROLLO APLICACIÓN WEB CLT4BP (Producto Mínimo Viable)

#### 7.1 IA Generativa en la enseñanza de programación

El uso de la Inteligencia Artificial Generativa (IA Gen) para crear recursos pedagógicos en el campo de la educación está creciendo de forma exponencial. De acuerdo Nuryadin, Karlimah, and Nugraha (2026) con la mayoría de los materiales desarrollados utilizando IA Gen son textuales, como planes de clase y evaluaciones, utilizando herramientas como ChatGPT y Copilot, reportando además una optimización en eficiencia de tiempo y costos, así como personalización del aprendizaje de acuerdo con los requerimientos de cada estudiante. Sin embargo, los desafíos reportados hacen referencia a la calidad del contenido, requiriendo revisión y corrección por parte de los docentes.

En programación la literatura reporta diferentes enfoques de aplicación de la IA como apoyo en el proceso de enseñanza – aprendizaje, si bien la mayoría de los estudios reportan beneficios como la mejora productividad y personalización de material instruccional, genera riesgos como dependencia excesiva y deterioro de pensamiento lógico de los estudiantes (Nathaniel, Oyelere, Suhonen, & Tedre, 2025).

Yilmaz and Karaoglan Yilmaz (2023), realizan una investigación sobre el efecto de la IA Gen en el pensamiento computacional y la autoeficacia, aplicando un enfoque de aula invertida (flipped classroom) combinado con actividades prácticas de codificación utilizando ChatGPT. Sus reportes referencian un aumento en las habilidades de pensamiento computacional, motivación y autoeficacia de los estudiantes.

Por otro lado, Kohen-Vacs, Usher, and Jansen (2025) presentan un estudio de análisis de como los estudiantes novatos utilizan generadores de código basados en modelos LLM. Aplicando entornos de aprendizaje asistidos por IA con sugerencias en tiempo real y corrección de sintaxis, complementado con el análisis de los registros de chat de los estudiantes. Sus resultados reportan variaciones en el rendimiento que atribuyen al compromiso de los estudiantes, concluyen que estas herramientas mejoran las habilidades prácticas, pero deben combinarse con la instrucción tradicional.

También se reporta el desarrollo de herramientas de explicabilidad y tutoría personalizada, como el estudio realizado por M.-L. Tsai, Ong, and Chen (2023), donde desarrollan y evalúan una herramienta basada en ChatGPT para la explicación de código. La herramienta integra módulos de generación de pistas y explicaciones personalizadas para fomentar el aprendizaje. Reportan resultados positivos sobre la comprensión de código de los estudiantes, disminuyendo la intervención del instructor.

Otro enfoque de aplicación de IA Gen en la enseñanza de programación es el aplicado por MacNeil et al. (2024), su estudio consiste en la comparación de las capacidades de detección de errores lógicos en el lenguaje de programación C y proporcionar explicaciones amigables para principiantes, entre humanos y modelos de lenguaje. Sus resultados reportan que los modelos IA son capaces de generar explicaciones útiles que ayudan a novatos a entender errores de código.

Si bien la mayoría de los estudios reportan resultados positivos en la utilización de IA Gen

en el proceso de enseñanza – aprendizaje de programación, también coinciden en los riesgos de caer en dependencia excesiva y decremento del desarrollo del pensamiento lógico de los estudiantes si se abusa de su uso, por ello la importancia de incorporarlo como herramienta de apoyo en la enseñanza tradicional, no como la base del proceso de enseñanza – aprendizaje.

Otro punto para tomar en cuenta es que solo un estudio reporta la integración de IA Gen utilizando alguna teoría de aprendizaje. Por ello el desarrollo del prototipo de software en el presente trabajo representa un aporte significativo en la integración de IA Gen en la educación, en este caso en el área de programación. Lo anterior basado en que el desarrollo está sustentado en un modelo de diseño instruccional, desarrollado a partir de la integración de varios elementos de teorías cognitivas de aprendizaje como son la teoría de carga cognitiva, teorías de motivación, así como modelo de instrucción personalizada.

## 7.2 Plan de desarrollo e implementación

Objetivo: Desarrollar un producto mínimo viable funcional del sistema CLT4BP que permita, gestión básica de usuarios (estudiantes, instructores y administrador), evaluación MSLQ y diagnóstico de conocimientos, generación de material instruccional con IA, seguimiento básico de progreso, retroalimentación y un editor de código integrado para tareas. La Figura 7.12 presenta el cronograma de las fases desarrollo e implementación, y seguido de ello se describen las actividades de cada fase.



Figura 7.1: Plan de desarrollo aplicando la metodología ágil Scrum.

Esta arquitectura proporciona una base sólida para el MVP de CLT4BP, cumpliendo con todos los requisitos funcionales especificados en el modelo instruccional CLT4BP. El sistema está diseñado para ser escalable, mantenible y extensible. La integración con Claude API

permite generar material instruccional altamente personalizado, mientras que la arquitectura de microservicios híbrida facilita el mantenimiento independiente de cada componente.

### 7.3 Aplicación web CLT4BP: Visión general

La aplicación web CLT4BP es una herramienta de apoyo especializada en la enseñanza de programación que implementa el modelo instruccional "mapa de ruta CLT4BP". Este modelo se basa en la Teoría de Carga Cognitiva (TCC) y el Diseño Instruccional de Cuatro Componentes (4C/ID), incorporando estrategias motivacionales y cognitivas para optimizar el proceso de enseñanza-aprendizaje.

#### Características principales

- **Adaptabilidad:** Implementa estrategias diferenciadas según los conocimientos previos de los estudiantes.
- **Carga cognitiva optimizada:** Aplica 17 efectos de la Teoría de Carga Cognitiva.
- **Estructura 4C/ID:** Organiza el contenido siguiendo el modelo de cuatro componentes.
- **Monitoreo motivacional:** Integra instrumentos de evaluación y optimización motivacional (MSLQ, ARCS).
- **Evaluación integral:** Mide conocimiento declarativo, procedimental y carga cognitiva.
- **Proceso iterativo:** Permite ajustes basados en resultados de evaluación.

El desarrollo de un producto mínimo viable de la aplicación web "CLT4BP" como apoyo en la creación de material instruccional basado en el modelo instruccional CLT4BP y la integración de una herramienta de IA mediante conexión API para generar material instruccional proporcionando el modelo CLT4BP como base de entrenamiento para la IA. La Figura 7.1 muestra la arquitectura general y la Figura 7.2 muestra el diseño arquitectónico del producto mínimo viable de la aplicación web "CLT4BP".

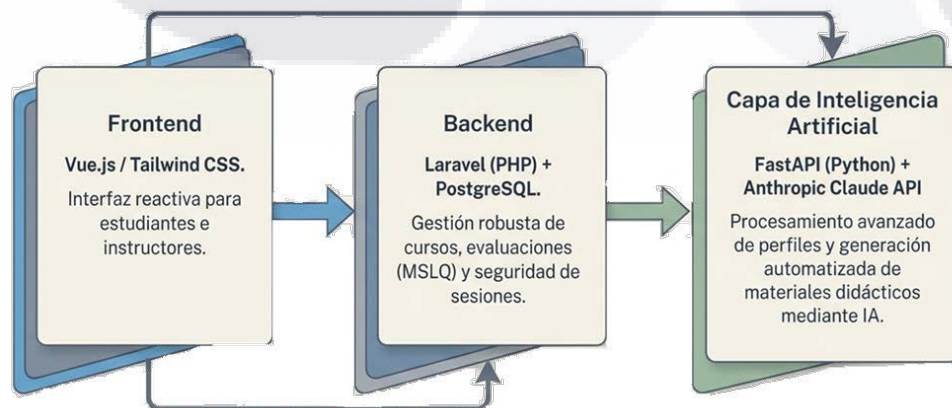


Figura 7.2: Arquitectura general del sistema.



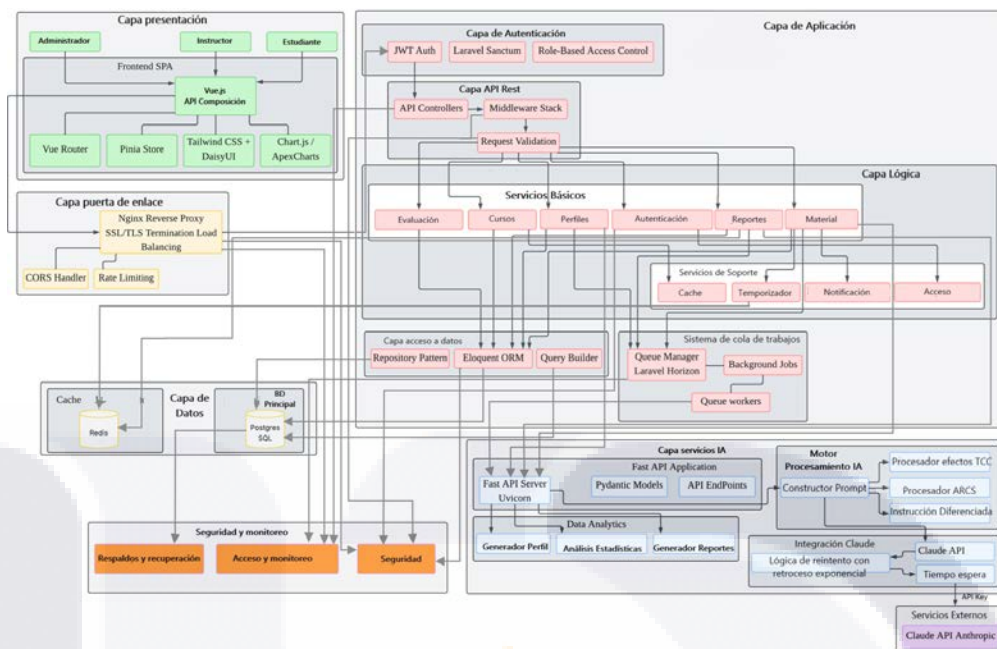


Figura 7.3: Diagrama arquitectónico producto mínimo viable.

El sistema implementa una arquitectura de microservicios híbrida con los siguientes componentes:

#### Aplicación PHP/Laravel:

- Gestión de usuarios, autenticación y autorización.
- Gestión de cursos, evaluaciones y material instruccional.
- API REST para el frontend.
- Gestión de sesiones y flujos de trabajo

#### Servicio Python/FastAPI:

- Procesamiento de perfiles estudiantiles
- Integración con Claude API.
- Generación de prompts personalizados.
- Análisis estadístico y generación de reportes.

Para el desarrollo del prototipo se propone el siguiente stack tecnológico que se muestra en las tablas 7.1 y 7.2:

El sistema se compone de los siguientes módulos principales:

1. Capa de Presentación: Vue.js SPA (Single Page Application).
2. API Gateway: Laravel API REST con middleware de autenticación.
3. Capa de Negocio: Controllers, Services, Repositories (Laravel).
4. Servicios IA: FastAPI con integración Claude API.
5. Capa de Datos: MySQL/PostgreSQL + Redis.

| Componente          | Tecnología       | Propósito                                      |
|---------------------|------------------|------------------------------------------------|
| Framework principal | PHP + Laravel    | API Rest, autenticación y gestión de sesiones  |
| Servicios IA        | Python + FastAPI | Integración Claude API, procesamiento de datos |
| Base de datos       | PostgreSQL       | Almacenamiento relacional principal            |
| Cache               | Redis            | Sesiones, cache de consultas, colas            |

Tabla 7.1: Arquitectura tecnológica backend producto mínimo viable CLT4BP.

| Componente   | Tecnología                 | Propósito                              |
|--------------|----------------------------|----------------------------------------|
| Framework    | Vue.js con Composition API | Interfaz reactiva y componentes        |
| UI Framework | Tailwind CSS + DaisyUI     | Diseño responsivo y componentes UI     |
| Gráficos     | Chart.js / ApexCharts      | Visualización de reportes y dashboards |

Tabla 7.2: Arquitectura tecnológica frontend producto mínimo viable CLT4BP.

## 7.4 Componentes principales del sistema

### 7.4.1 Autenticación y usuarios

#### Funcionalidades

- Registro de estudiantes (auto-registro).
- Creación de instructores por administrador.
- Login/Logout con JWT tokens.
- Control de acceso basado en roles (RBAC).
- Gestión de perfiles de usuario.

#### Roles del sistema

- **Administrador:** Crear instructores, gestión completa del sistema.
- **Instructor:** Crear cursos, asignar estudiantes, configurar evaluaciones, generar material con IA, activar/desactivar contenidos, consultar reportes.
- **Estudiante:** Responder cuestionarios, realizar evaluaciones, estudiar material instruccional, consultar sus propios reportes.

### 7.4.2 Componente gestión de cursos

#### Funcionalidades

- Creación y edición de cursos.
- Definición de objetivos de aprendizaje.
- Asignación de estudiantes al curso.
- Gestión del estado del curso (activo/inactivo).

### 7.4.3 Componente de evaluaciones

#### Tipos de evaluaciones

- Tests iniciales: Recall test (teorico), comprehension test (practico).
- Cuestionarios de perfil: MSLQ (motivación y estrategias cognitivas).
- Evaluaciones finales: recall test, comprehension test, MSLQ, CIS (Course Interest Survey), IMMS (Instructional Materials Motivation Survey) y carga cognitiva (Morrison, 2014).

### 7.4.4 Componente Generación de Material con IA

#### Servicio Python - Claude API Integration

Este servicio es responsable de la comunicación con Claude API y la generación de material instruccional personalizado.

#### Estructura del prompt builder

El sistema construye 2 prompts estructurados que posteriormente une en un solo, prompt usuario que incluye:

1. Información del perfil (individual o grupal)
2. Objetivos de aprendizaje del curso
3. Efectos CLT preseleccionados por el instructor
4. Componente específico a generar
5. Aplicación del modelo ARCS de motivación
6. Estrategias diferenciadas (solo para perfil individual)
7. Formato de salida esperado

#### Tipos de material generado

Tipo de material generado con las especificaciones de la tabla 7.3.

#### Efectos CLT a implementar

El instructor puede preseleccionar efectos de la Teoría de Carga Cognitiva según el perfil del grupo:

Efectos para Nuevo Conocimiento

- Solución Libre (Goal-Free Effect)
- Ejemplo Resuelto (Worked Example Effect)
- Problema por Completar (Completion Problem)
- Atención Dividida (Split Attention)
- Modalidad (Modality Effect)
- Redundancia (Redundancy Effect)
- Variabilidad (Variability Effect)
- Elementos Aislados (Isolated Elements)

| Componente                       | Descripción                                                                                                                                                                                                                                                                             |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Tareas de Aprendizaje</b>     | <ul style="list-style-type: none"> <li>• Diseño con dificultad creciente</li> <li>• Escenarios auténticos</li> <li>• Explicación del POR QUÉ y PARA QUÉ</li> <li>• Aplicación modelo ARCS</li> <li>• Estrategias diferenciadas (individual)</li> <li>• Efectos CLT aplicados</li> </ul> |
| <b>Información de Soporte</b>    | <ul style="list-style-type: none"> <li>• Teoría y conceptos fundamentales</li> <li>• Aplicación modelo ARCS</li> <li>• Estrategias diferenciadas (individual)</li> <li>• Efectos CLT aplicados</li> </ul>                                                                               |
| <b>Información Procedimental</b> | <ul style="list-style-type: none"> <li>• Ejemplos isomórficos</li> <li>• Guías de preguntas</li> <li>• Modelos y mapas conceptuales</li> <li>• Estrategias diferenciadas (individual)</li> <li>• Efectos CLT aplicados</li> </ul>                                                       |
| <b>Protocolos Verbales</b>       | <ul style="list-style-type: none"> <li>• Audio y/o transcripción</li> <li>• Enfoque en CÓMO y PORQUÉ</li> <li>• Perspectiva profesional</li> </ul>                                                                                                                                      |

Tabla 7.3: Especificaciones para el tipo de material a crear.

- Interactividad (Element Interactivity)

Efectos para Conocimiento Previo

- Auto-explicación (Self-Explanation)
- Imaginación (Imagination Effect)
- Reversión de Experiencia (Expertise Reversal)
- Desvanecimiento de Guía (Guidance-Fading)
- Memoria Colectiva (Collective Memory)
- Autogestión (Self-Management)
- Movimiento Humano (Human Movement)
- Información Transitoria (Transient Information)

Y el prompt de sistema, que establece el contexto inicial y el rol que debe asumir la IA, el siguiente es el prompt utilizado

Eres un diseñador instruccional experto especializado en:

- Teoría de la Carga Cognitiva (CLT)
- Diseño Instruccional de Cuatro Componentes (4C/ID)
- El modelo ARCS de motivación (Atención, Relevancia, Confianza, Satisfacción)
- Estrategias de aprendizaje diferenciado

- Programación y desarrollo de software

Tu tarea es generar materiales instruccionales de alta calidad, adaptados al perfil del estudiante o grupo, aplicando los principios pedagógicos solicitados.

IMPORTANTE:

- Generar contenido en español claro y profesional
- Seguir estrictamente las instrucciones sobre los efectos de la CLT
- Adaptar el contenido al nivel de conocimientos previos del estudiante
- Utilizar ejemplos de programación concretos y relevantes
- Proporcionar el contenido en el formato JSON estructurado solicitado.

#### 7.4.5 Módulo de análisis de datos

El Módulo de Análisis de Datos, una parte esencial del sistema CLT4BP, genera perfiles de aprendizaje individualizados y grupales sin intervención manual. El módulo evalúa y sintetiza los datos adquiridos a partir de varios exámenes, entre ellos el MSLQ (Cuestionario de Estrategias Motivadas para el Aprendizaje), las pruebas de pre-evaluación sobre la práctica de comprensión y la teoría de memoria. Con esta información, se elabora un perfil individual integral que muestra las preferencias de aprendizaje, las características motivacionales y la experiencia de cada alumno.

Además, elabora un perfil grupal que representa la media de todos los alumnos inscritos en el curso para brindarle al instructor una visión integral del grupo. Estos perfiles, al estar guardados en formato JSON organizado, pueden ser incluidos de manera directa en las solicitudes a la API de Claude para la creación de recursos personalizados para el aprendizaje. Así, el módulo sirve de conexión entre la evaluación diagnóstica y los materiales personalizados para el aprendizaje, optimizando la carga cognitiva en función de lo que cada estudiante o el grupo entero necesiten.

La generación de perfiles individuales sigue un proceso sistemático aplicando diferentes reglas.

1. Condiciones de datos imprescindibles: El sistema necesita que el alumno haya terminado las evaluaciones en la secuencia indicada para crear un perfil individual: Cuestionario MSLQ - Parte de Motivación (inicial), Prueba de Recuerdo (conocimientos anteriores), Prueba de Comprensión y Cuestionario MSLQ - Parte de Estrategias Metacognitivas y Cognitivas.
2. Tratamiento del cuestionario MSLQ: Las respuestas del MSLQ se procesan agrupándolas por dimensión. Las 81 preguntas que conforman el MSLQ se distribuyen en 15 dimensiones predefinidas, a las cuales se les asignan ítems concretos. Cálculo de la media: Se computa la media aritmética de las respuestas a los ítems de cada dimensión (escala Likert de 1 a 7). Clasificación por niveles: Cada dimensión es categorizada en tres niveles basándose en su promedio.

$$\text{Alto} : \text{promedio} \geq 5,5 \quad (7.1)$$

$$\text{Medio} : 3,5 > \text{promedio} \leq 5,5 \quad (7.2)$$

$$\text{Bajo} : \text{promedio} \leq 3,5 \quad (7.3)$$



3. Categorización de las dimensiones del MSLQ. Las dimensiones se organizan en dos categorías principales: Dimensiones de motivación (Orientación intrínseca a la meta, Orientación extrínseca a la meta, Valor de la tarea, Creencias de control, Autoeficacia y Ansiedad ante los exámenes). Dimensiones de estrategia: Cognitivas (Ensayo, Elaboración, Organización, Pensamiento crítico) y Metacognitivas (Autorregulación metacognitiva, Gestión de recursos: Gestión del tiempo, Regulación del esfuerzo, Aprendizaje entre iguales, Búsqueda de ayuda).
4. Análisis de las pruebas de conocimiento. Para las pruebas de memoria y comprensión, el porcentaje se calcula de la siguiente manera: (respuestas correctas / total de preguntas) 100, clasificando el nivel de conocimiento de la siguiente manera:

$$\text{Alto : porcentaje} \geq 75 \% \quad (7.4)$$

$$\text{Medio : } 50 \% > \text{porcentaje} \leq 75 \% \quad (7.5)$$

$$\text{Bajo : porcentaje} \leq 50 \% \quad (7.6)$$

5. Generación del Resumen del Perfil. El sistema genera automáticamente un resumen que identifica: motivación, calculada mediante el promedio de la Orientación Intrínseca a la Meta, el Valor de la Tarea y la Autoeficacia; y estrategias, calculadas mediante el promedio de la Elaboración, la Organización y la Autorregulación Metacognitiva. Se clasifica en Fortalezas Clave: todas las dimensiones calificadas como "altas" más conocimientos previos si son altas, y Áreas de Apoyo: todas las dimensiones calificadas como "bajas" más conocimientos previos si son bajas.
6. Generación de Recomendaciones Personalizadas. Se generan recomendaciones específicas basadas en reglas condicionales.
- (a) Si la motivación general es baja, aplique las estrategias ARCS con énfasis en la Atención y la Relevancia. Incluya ejemplos del mundo real y aplicaciones prácticas.
  - (b) Si las estrategias generales son débiles, proporcione guías de estudio explícitas. Modele los procesos metacognitivos mediante protocolos verbales.
  - (c) Si el conocimiento previo es bajo, utilice los efectos de la CLT para principiantes: ejemplos resueltos, problemas a resolver. Minimice la carga cognitiva extrínseca.
  - (d) Si el conocimiento previo es alto, aplique los efectos de la CLT para expertos: auto explicación, visualización. Considere invertir la experiencia para algunos materiales.

La generación del perfil grupal genera promedios grupales y debe haber al menos dos perfiles individuales. Se realiza un análisis de la distribución grupal, determinando las características predominantes y generando una categorización grupal. Si  $> 60 \%$  tiene alta motivación y  $< 30 \%$  tiene bajo conocimiento: «Grupo altamente motivado con conocimientos previos». Si  $< 30 \%$  tiene alta motivación y  $> 50 \%$  tiene bajo conocimiento: «Grupo requiere apoyo motivacional y contenido fundamental». Si  $> 50 \%$  tiene alta motivación: «Grupo motivado con conocimientos previos variables». En caso contrario: «Grupo heterogéneo requiere diferenciación instruccional».

#### 7.4.6 Motor de procesamiento IA

El núcleo innovador de la aplicación CLT4BP es el motor de procesamiento de IA, que se implementó con un servicio independiente creado en Python utilizando FastAPI y conectado con la API Claude de Anthropic. Este módulo crea materiales didácticos personalizados y fundamentados pedagógicamente, empleando perfiles preestablecidos como base para adecuar el contenido.

El módulo genera cuatro tipos de material didáctico alineados con el modelo 4C/ID: Tareas de Aprendizaje, problemas de dificultad creciente que incluyen una explicación del propósito y la relevancia de cada actividad; Información de Apoyo, la teoría y los conceptos necesarios para comprender el dominio de aprendizaje; Información Procedimental, ejemplos isomórficos, guías de preguntas y modelos conceptuales para desarrollar la automatización de procedimientos; y Protocolos Verbales, que pueden presentarse en formato de audio o transcripción y modelan el pensamiento experto, centrándose en el cómo y el "por qué" de la resolución de problemas desde una perspectiva profesional.

Implementado a través de una arquitectura por capas en el backend de Laravel de un servicio autónomo creado en Python con FastAPI. Esta estructura asegura que la totalidad de las interacciones con la API Claude se realicen únicamente por medio del servicio Python, enviando solicitudes HTTP autenticadas a través de tokens de API guardados en variables de entorno. La Figura 7.3 muestra el proceso que se ejecuta para la creación del prompt para la generación del material instruccional.

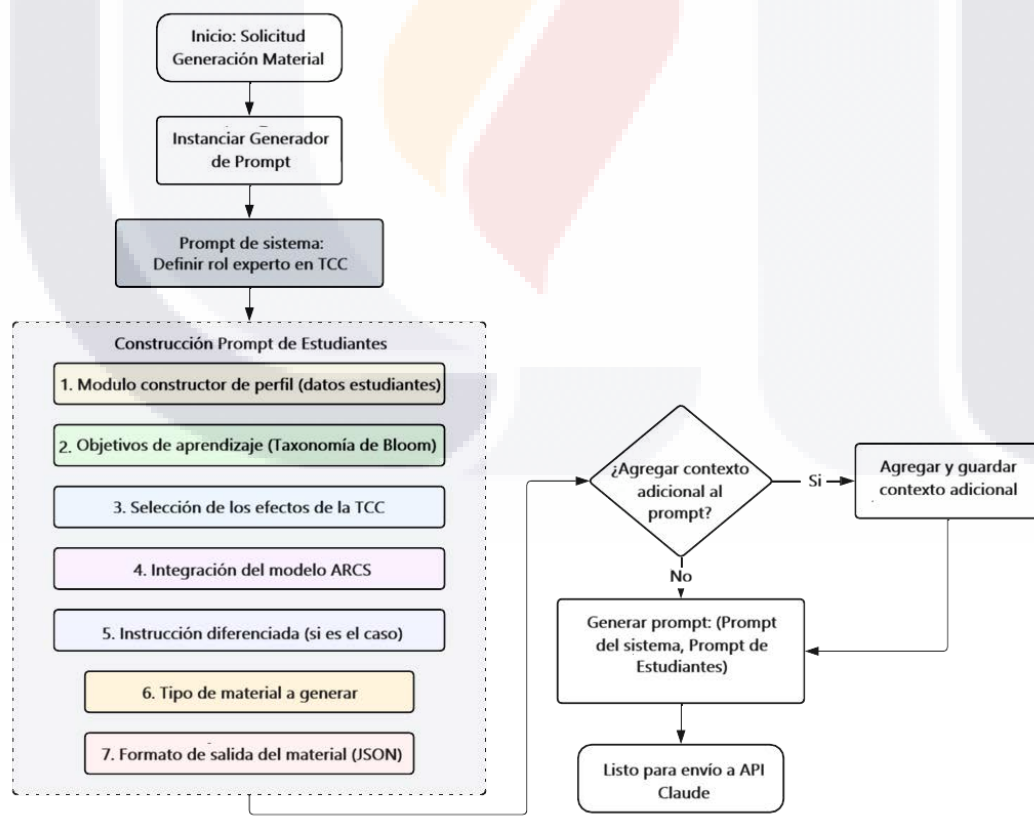


Figura 7.4: Creación de prompt para solicitud de material instruccional.

#### 7.4.7 Modelado de datos

La Figura 7.4 muestra el diseño entidad – relación de la base de datos principal. Las tablas utilizadas en el diseño de la base datos son: users, courses, instructional\_material, course\_enrollments, student\_profiles, group\_profiles, clt\_effects\_selection, student\_responses, assessments, reports y material\_access\_logs.

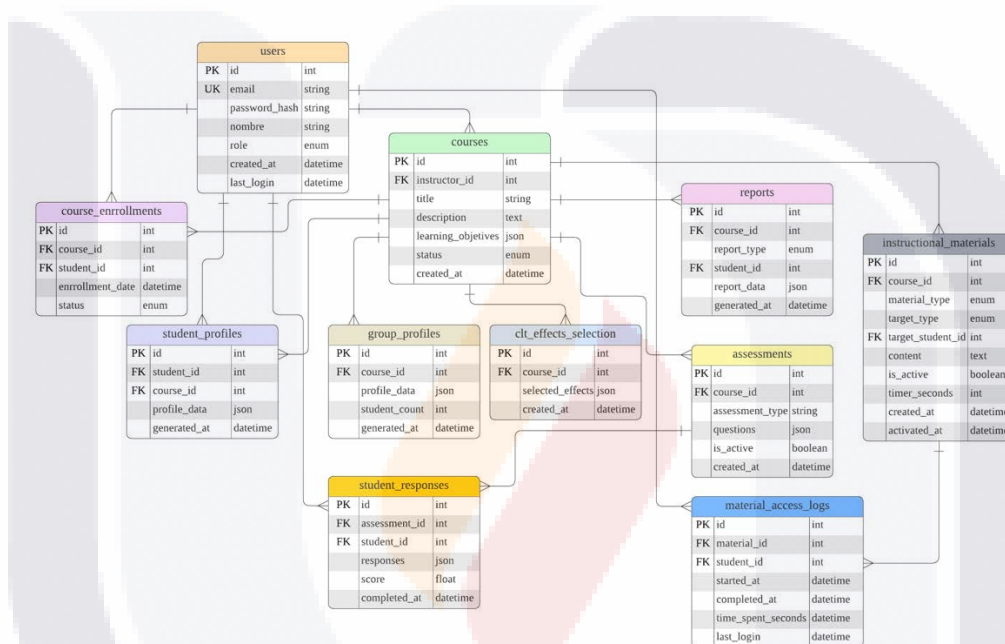


Figura 7.5: Diagrama ER diseño base datos CLT4BP.

#### 7.4.8 Diseño de interfaces

Las Figuras 7.5 a 7.10 muestran la interfaz de la aplicación web CLT4BP. Figura 7.5 muestra la interfaz de inicio de sesión (login) donde el usuario captura sus credenciales para acceder a la aplicación. Si aún no está registrado el estudiante puede realizarlo en la opción Regístrate aquí y posteriormente regresar al login para iniciar sesión. La Figura 7.6 es la pantalla principal del perfil de instructor, donde se muestran los datos de su perfil, las opciones de ver o crear curso.

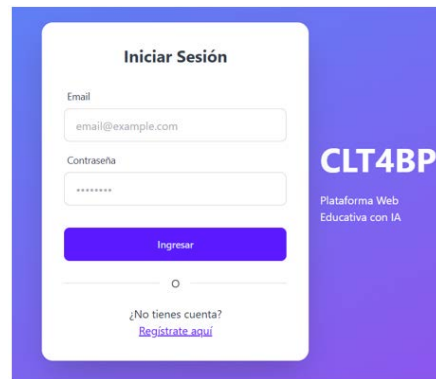


Figura 7.6: Interfaz de inicio de sesión o registro (login).

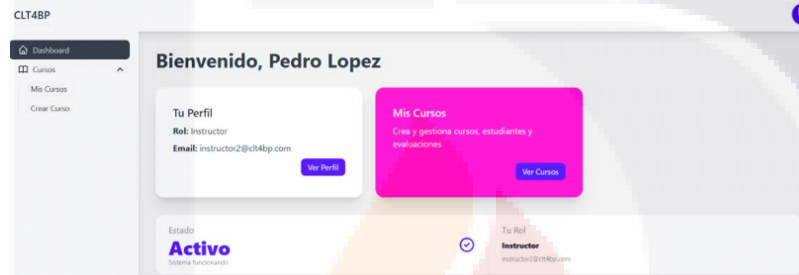


Figura 7.7: Interfaz de inicio acceso perfil instructor.

La Figura 7.7 muestra los cursos que tienen los diferentes perfiles (instructor o estudiante). Aquí las opciones de acceso varían un poco dependiente del perfil, en este caso la imagen corresponde al perfil de instructor donde tiene opciones de ver, editar o eliminar curso. Para el perfil de estudiante solo puede ver (evaluaciones o material).

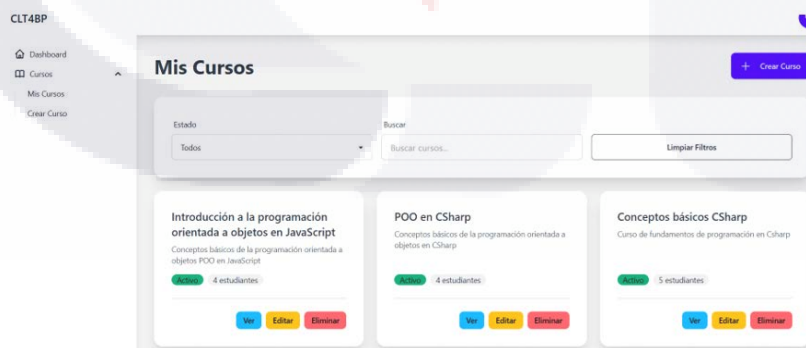


Figura 7.8: Interfaz cursos perfil instructor.

La Figura 7.8 corresponde a la interfaz de edición de cursos, aquí es donde el instructor asigna estudiantes al curso, crea evaluaciones, revisa perfiles de estudiantes, genera material instruccional, selecciona efectos aplicar, revisa reportes, califica evaluaciones manuales, entre otras actividades.

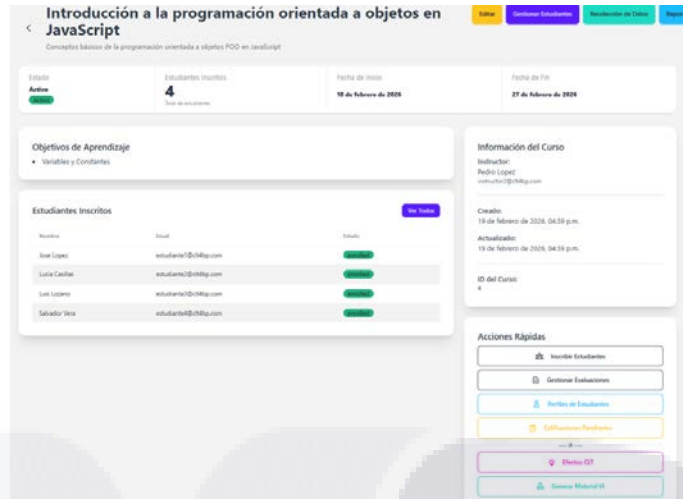


Figura 7.9: Interfaz de administración de cursos.

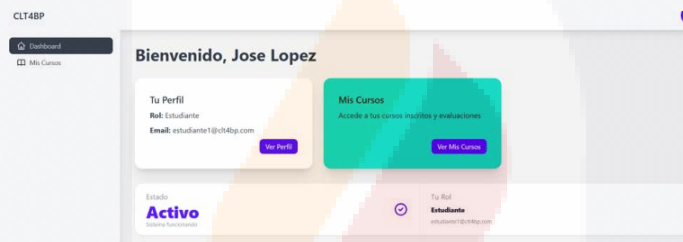


Figura 7.10: Interfaz pantalla de inicio perfil estudiante.

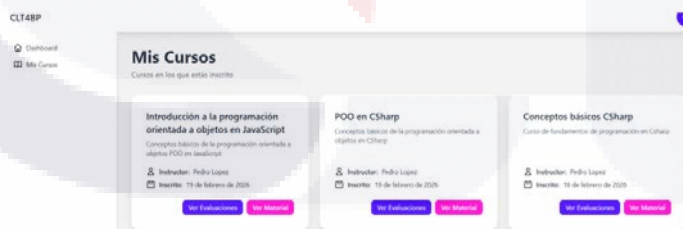


Figura 7.11: Interfaz de cursos estudiantes.

Las Figuras 7.9 y 7.10 corresponde al perfil de estudiante, donde accede a los cursos en los que está registrado, así como al material y evaluaciones correspondientes.

#### 7.4.9 Flujo de actividades detallado

El flujo de actividades de la aplicación se describe en la Figura 7.11.

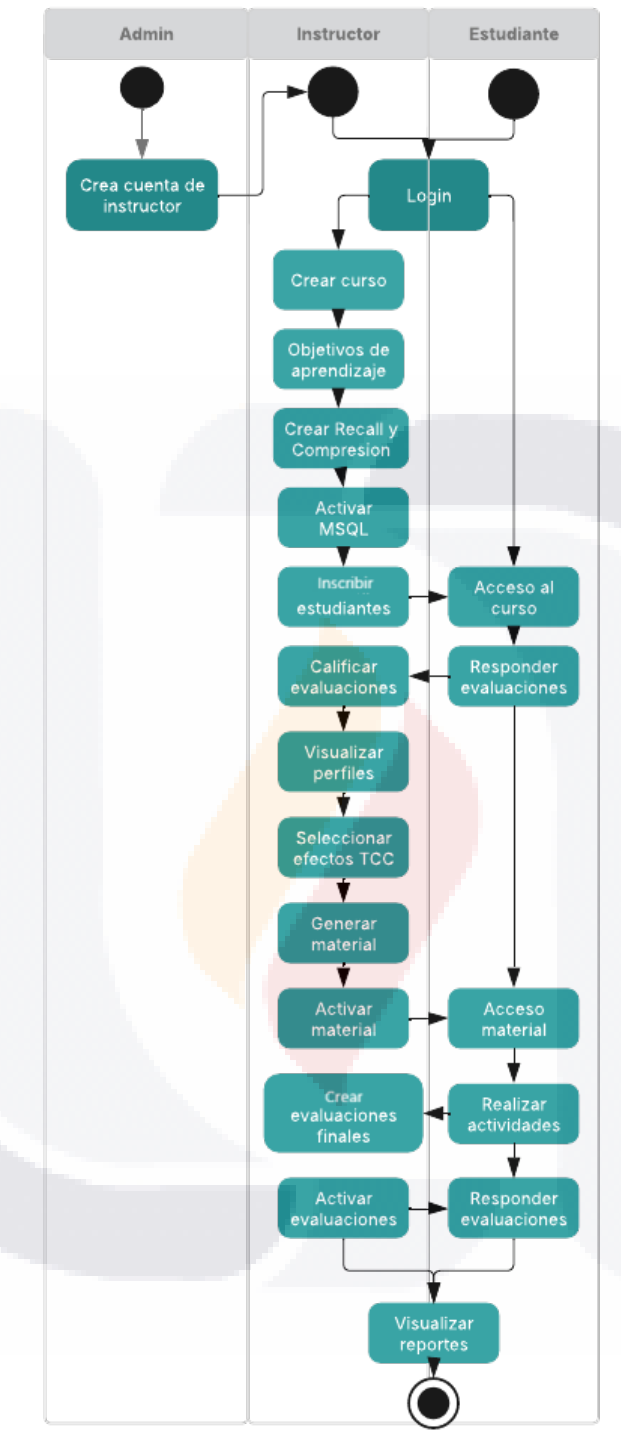


Figura 7.12: Diagrama de actividades.



## 7.5 Estudio piloto: Aplicación web CLT4BP

La implementación de la herramienta web CLT4BP se realizó a un solo grupo experimental con una muestra total de 43 estudiantes, de los cuales solo un total de 33 completaron el estudio. Aplicando mediciones PRE - POST (motivación y conocimientos), así como los cuestionarios de Carga Cognitiva, CIS e IMMS al final del estudio. El tema utilizado para la implementación fue manejo de archivos de texto en lenguaje C. La aplicación web fue desplegada en un servicio de hosting (railway) para el acceso de estudiantes e instructor.

Siguiendo el mapa de ruta del modelo instruccional, una vez que los estudiantes se registraron en la plataforma, el instructor creó el curso con el objetivo de aprendizaje mencionado, inscribiendo a los estudiantes registrados en el curso, el flujo de actividades se describe a continuación:

1. El instructor creó las evaluaciones iniciales (motivación, estrategias cognitivas y conocimientos previos).
2. Los estudiantes completan evaluaciones iniciales.
3. Instructor activa la opción de generación automática de perfiles individual y grupal.
4. Instructor activa la opción de generación automática de los materiales instruccionales.
5. Estudiantes acceder al material.
6. Instructor genera evaluaciones finales.
7. Estudiantes completan evaluaciones finales.

Los materiales generados para el estudio piloto comprendieron: información de soporte (fundamentos teóricos, definiciones, ejemplos), tareas de aprendizaje (ejercicios prácticos), información procedimental (guías, sintaxis) y protocolos verbales (transcripciones de protocolos verbales). La Figura 7.13 es un ejemplo de material de tareas de aprendizaje generado y la Figura 7.14 de información procedimental.

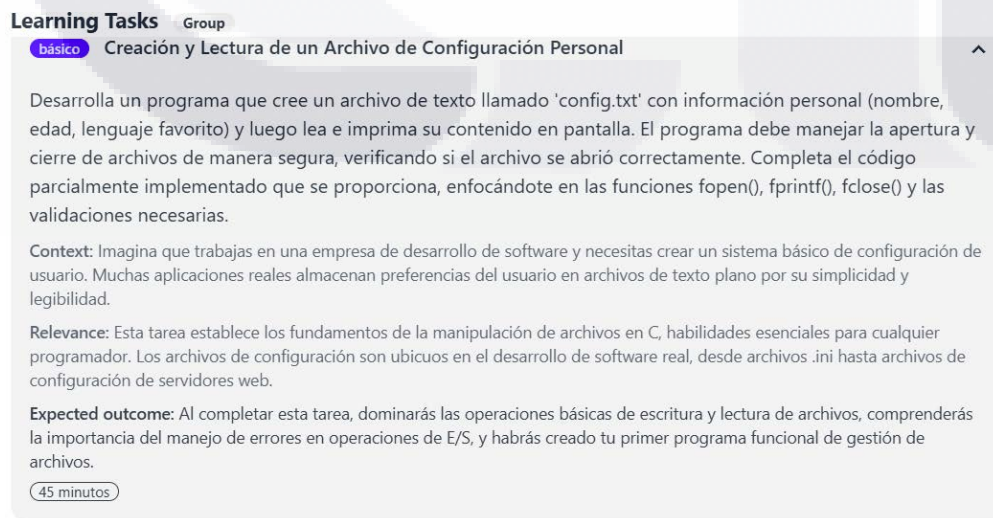


Figura 7.13: Ejemplo material generado del componente tareas de aprendizaje.



Figura 7.14: Ejemplo material generado del componente información procedimental.

### 7.5.1 Resultados

Para el análisis de los resultados en primer lugar se realizó un prueba de normalidad (Shapiro-Wilk) a todas las variables del estudio. Los resultados obtenidos muestran una distribución normal en las variables de motivación, carga intrínseca y carga pertinente para poder aplicar la prueba t de Student. Sin embargo, calificación inicial, calificación final, y carga extrínseca no muestran una distribución normal por lo que se opta por aplicar el coeficiente de Spearman y la prueba de Friedman para la comparación de las cargas cognitivas, los resultados se muestran en la Tabla 7.4.

| Variable             | Estadística | valor p | Resultado              |
|----------------------|-------------|---------|------------------------|
| Motivación Inicial   | 0.952       | 0.152   | Distribución normal    |
| Motivación Final     | 0.980       | 0.794   | Distribución normal    |
| Calificación inicial | 0.827       | 0.000   | Distribución NO normal |
| Calificación final   | 0.889       | 0.003   | Distribución NO normal |
| Carga Intrínseca     | 0.966       | 0.373   | Distribución normal    |
| Carga Extrínseca     | 0.921       | 0.019   | Distribución NO normal |
| Carga Pertinente     | 0.953       | 0.158   | Distribución normal    |

Tabla 7.4: Resultados de la prueba de normalidad por variable.

Se aplicó entonces la prueba t Student para las variables de motivación, los resultados arrojan que no existe diferencia estadística significativa (valor  $p = 0.444$ ) entre las mediciones de motivación PRE - POST, como se muestra en la Figura 7.15. También se calculó el tamaño del efecto mediante la  $d$  de Cohen. El tamaño del efecto se reporta: pequeño ( $d < 0,2$ ), mediano ( $d = 0,5$ ), grande ( $d \geq 0,8$ ). De ello el valor obtenido fue 0.133 (tamaño pequeño).

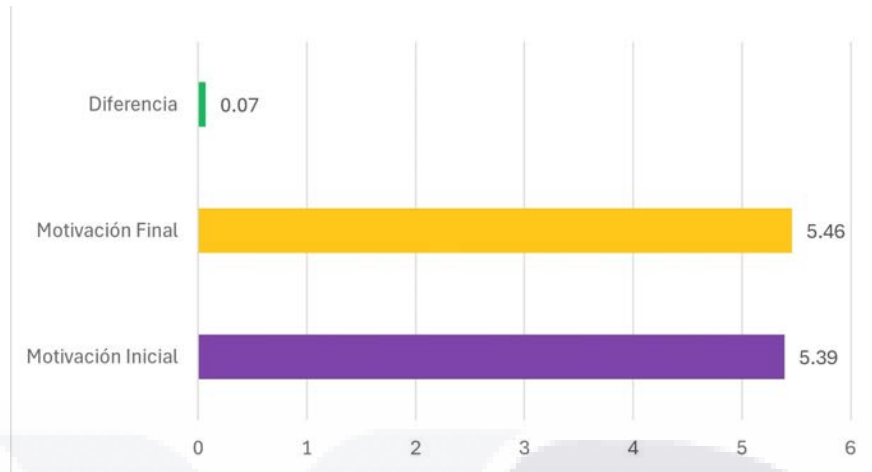


Figura 7.15: Resultados promedio grupal motivación Pre-Post. Escala Likert (1 a 7, donde 1 = Totalmente en desacuerdo y 7 = Totalmente de acuerdo).

Dedibo a que la Carga Extrínseca no tiene una distribución normal, se utilizó la prueba no paramétrica de Friedman para comparar los tres tipos de carga cognitiva entre sí. Se aplicaron comparaciones post-hoc mediante la prueba de Wilcoxon con corrección de Bonferroni ( $\alpha = ,017$  para tres comparaciones). La Tabla 7.5 muestra las estadísticas descriptivas, y la Tabla 7.6 los resultados de la prueba Friedman y comparaciones post-hoc (Wilcoxon-Bonferroni).

| Tipo de carga                | Media       | DE          | Mín         | Máx         |
|------------------------------|-------------|-------------|-------------|-------------|
| Carga Intrínseca (CI)        | 4.58        | 1.25        | 2.67        | 7.00        |
| Carga Extrínseca (CE)        | 2.88        | 1.38        | 1.00        | 7.00        |
| <b>Carga Pertinente (CP)</b> | <b>5.55</b> | <b>0.88</b> | <b>3.00</b> | <b>7.00</b> |

Tabla 7.5: Estadísticos descriptivos según el tipo de carga.

| Comparación                        | Estadístico         | p                  | Resultado  |
|------------------------------------|---------------------|--------------------|------------|
| <b>Prueba de Friedman (global)</b> | $\chi^2(2) = 40,18$ | <b>p &lt; ,001</b> | <b>***</b> |
| CI vs. CE (post-hoc)               | $W = 30,00$         | $p < ,001$         | * sig.     |
| CI vs. CP (post-hoc)               | $W = 70,50$         | $p = ,001$         | * sig.     |
| CE vs. CP (post-hoc)               | $W = 7,00$          | $p < ,001$         | * sig.     |

*Nota.* \* Significativo con corrección de Bonferroni ( $\alpha = ,017$ ). \*\*\* Significativo al nivel .001. CI = Carga Intrínseca; CE = Carga Extrínseca; CP = Carga Pertinente.

Tabla 7.6: Comparación por pares mediante la prueba de Friedman.

Respecto a los resultados de carga cognitiva, el grupo mostró nivel bajo en el carga cognitiva extrínseca o extraña, que de acuerdo con la TCC corresponde al material instruccional y como es presentando a los estudiantes, lo cual sugiere que el material proporcionado a través de la plataforma web no generó carga cognitiva alta en los estudiantes. Por otro lado,

la carga cognitiva intrínseca, correspondiente a la generada por la complejidad del tema, arrojó resultados de un nivel medio, lo que sugiere que la percepción de los estudiantes hacia la complejidad del tema no fue alta. Y por último, la carga cognitiva pertinente, que de acuerdo con los resultados mostró un nivel medio-alto, lo que sugiere con base en la TCC, los estudiantes no tuvieron sobrecarga cognitiva. Lo anterior representa el patrón de cargas cognitivas ( $CE < CI < CP$ ) ideal según la Teoría de la Carga Cognitiva (Sweller, 1988) como se muestra en la Figura 7.16.



Figura 7.16: Resultados carga cognitiva grupal.

Se calculó el coeficiente de correlación de Spearman por ser una medida no paramétrica adecuada para el tamaño de la muestra y la distribución anormal de las variables, con la finalidad de revisar la correlación entre los cambios en motivación y calificación, así como entre los tipos de carga cognitiva y los resultados del aprendizaje, los resultados se muestran en la Tabla 7.7.

| Par de variables                                                      | $\rho$ de Spearman | p           | Significancia |
|-----------------------------------------------------------------------|--------------------|-------------|---------------|
| Carga Intrínseca $\leftrightarrow$ Motivación final                   | 0,154              | ,393        | —             |
| Carga Extrínseca $\leftrightarrow$ Motivación final                   | -0,324             | ,066        | tendencia †   |
| <b>Carga Pertinente <math>\leftrightarrow</math> Motivación final</b> | <b>0.546</b>       | <b>.001</b> | <b>**</b>     |
| Carga Intrínseca $\leftrightarrow$ Calificación final                 | 0,078              | ,665        | —             |
| Carga Extrínseca $\leftrightarrow$ Calificación final                 | -0,063             | ,727        | —             |
| Carga Pertinente $\leftrightarrow$ Calificación final                 | -0,050             | ,784        | —             |

Nota. \*\*  $p < ,01$ . † Tendencia no significativa ( $p < ,10$ ).  $\Delta$  = diferencia entre medición final e inicial.

Tabla 7.7: Análisis de correlación de Spearman entre las variables de estudio.

De acuerdo con los resultados, la única correlación estadísticamente significativa es la observada entre Carga Pertinente y Motivación final ( $\rho = 0,546$ ,  $p = ,001$ ), lo que indica una asociación positiva moderada a alta. Esto sugiere que los estudiantes que percibieron una mayor carga pertinente al finalizar el uso de la aplicación también reportaron niveles más

altos de motivación, lo cual se alinea con los fundamentos de la Teoría de la Carga Cognitiva (TCC) de Sweller y el modelo ARCS de Keller. Adicionalmente, la tendencia negativa entre Carga Extrínseca y Motivación final ( $\rho = -0,324$ ,  $p = ,066$ ) también guarda coherencia con los fundamentos teóricos y podría alcanzar significancia estadística con una muestra mayor.

Los cuestionarios CIS (Course Interest Survey) e IMMS (Instructional Materials Motivation Survey) miden las cuatro dimensiones del modelo ARCS de Keller (Atención, Relevancia, Confianza y Satisfacción) en escalas de respuesta de 1 a 7. Los ítems marcados como invertidos en las guías de puntuación fueron recodificados ( $\text{valor\_invertido} = 8 \text{ menos respuesta\_original}$ ) antes de calcular las medias por dimensión. Las Tablas 7.8 y 7.9 muestran los resultados de los estadísticos descriptivos de ambos cuestionarios.

| Dimensión         | Media       | DE          | Mín         | Máx         | Nivel         |
|-------------------|-------------|-------------|-------------|-------------|---------------|
| Atención          | 4.88        | 0.71        | 3.75        | 6.25        | Moderada      |
| <b>Relevancia</b> | <b>5.67</b> | <b>0.81</b> | <b>4.00</b> | <b>7.00</b> | <b>Alta</b>   |
| Confianza         | 3.81        | 0.33        | 3.00        | 4.25        | Moderada-baja |
| Satisfacción      | 3.99        | 0.36        | 3.44        | 5.11        | Moderada-baja |

*Nota.* CIS = Course Interest Survey. Los ítems de la categoría Atención corresponden a Q1, Q4(i), Q10, Q15, Q21, Q24, Q26(i), Q29; Relevancia: Q2, Q5, Q8(i), Q13, Q20, Q22, Q23, Q25(i), Q28; Confianza: Q3, Q6(i), Q9, Q11(i), Q17(i), Q27, Q30, Q34; Satisfacción: Q7(i), Q12, Q14, Q16, Q18, Q19, Q31(i), Q32, Q33. (i) = ítem invertido.

Tabla 7.8: Resultados descriptivos por dimensión del Course Interest Survey (CIS).

| Dimensión           | Media       | DE          | Mín         | Máx         | Nivel         |
|---------------------|-------------|-------------|-------------|-------------|---------------|
| Atención            | 5.10        | 0.81        | 3.45        | 6.82        | Moderada-alta |
| <b>Relevancia</b>   | <b>5.36</b> | <b>0.78</b> | <b>4.11</b> | <b>6.89</b> | <b>Alta</b>   |
| Confianza           | 4.56        | 0.70        | 3.11        | 5.78        | Moderada      |
| <b>Satisfacción</b> | <b>5.36</b> | <b>0.92</b> | <b>3.67</b> | <b>7.00</b> | <b>Alta</b>   |

*Nota.* IMMS = Instructional Materials Motivation Survey. Los ítems de la categoría Atención corresponden a Q2, Q8, Q11, Q12(i), Q15(i), Q17, Q22(i), Q24, Q28, Q29(i), Q31(i); Relevancia: Q6, Q9, Q10, Q16, Q18, Q23, Q26(i), Q30, Q33; Confianza: Q1, Q3(i), Q4, Q7(i), Q13, Q19(i), Q25, Q34(i), Q35; Satisfacción: Q5, Q14, Q21, Q27, Q32, Q36. (i) = ítem invertido.

Tabla 7.9: Resultados descriptivos por dimensión del Instructional Materials Motivation Survey (IMMS).

Los resultados de los cuestionarios CIS e IMMS, asociados a la aplicación del modelo de motivación ARCS muestran nivel alto en la percepción de los estudiantes en algunos de los componentes (Relevancia y Satisfacción). Lo anterior se puede observar también en la Figura 7.17.

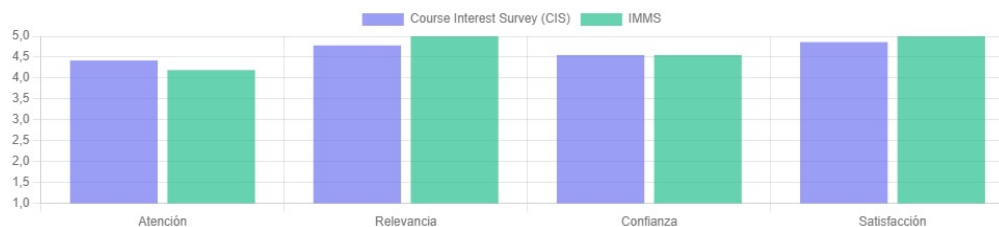


Figura 7.17: Resultados CIS e IMMS.

Por lo anterior, se calculó el coeficiente alpha de Cronbach ( $\alpha$ ) para evaluar la consistencia interna de las subescalas del CIS y el IMMS en esta muestra específica. Se consideran los siguientes criterios de interpretación:  $\alpha \geq ,70 = \text{aceptable}$ ;  $,60 \leq \alpha < ,70 = \text{cuestionable}$ ;  $\alpha < ,60 = \text{insuficiente}$  (George & Mallery, 2024). La Tabla 7.10 muestra los resultados.

| Instrumento / Dimensión    | N ítems   | $\alpha$    | Valoración          | Recomendación                       |
|----------------------------|-----------|-------------|---------------------|-------------------------------------|
| <b>CIS – Atención</b>      | <b>8</b>  | <b>,491</b> | <b>Insuficiente</b> | <b>Interpretar con cautela</b>      |
| CIS – Relevancia           | 9         | ,670        | Cuestionable        | Usar con reservas                   |
| <b>CIS – Confianza</b>     | <b>8</b>  | <b>,480</b> | <b>Insuficiente</b> | <b>Interpretar con cautela</b>      |
| <b>CIS – Satisfacción</b>  | <b>9</b>  | <b>,595</b> | <b>Insuficiente</b> | <b>Interpretar con cautela</b>      |
| <b>CIS Total</b>           | <b>34</b> | <b>,862</b> | <b>Bueno</b>        | <b>Confiable ✓</b>                  |
| <b>IMMS – Atención</b>     | <b>11</b> | <b>,466</b> | <b>Insuficiente</b> | <b>Interpretar con cautela</b>      |
| IMMS – Relevancia          | 9         | ,760        | Aceptable           | Confiable ✓                         |
| <b>IMMS – Confianza</b>    | <b>9</b>  | <b>,318</b> | <b>Insuficiente</b> | <b>No recomendado por subescala</b> |
| <b>IMMS – Satisfacción</b> | <b>6</b>  | <b>,838</b> | <b>Bueno</b>        | <b>Confiable ✓</b>                  |
| <b>IMMS Total</b>          | <b>36</b> | <b>,838</b> | <b>Bueno</b>        | <b>Confiable ✓</b>                  |

*Nota.* Los *scores* totales de CIS ( $\alpha = ,862$ ) e IMMS ( $\alpha = ,838$ ) presentan buena consistencia interna. Los *scores* por subescala deben interpretarse con cautela y la baja confiabilidad de algunas subescala representan limitación del estudio piloto.

Tabla 7.10: Análisis de confiabilidad (Alfa de Cronbach) para los instrumentos CIS e IMMS.

En resumen, a pesar de que la muestra fue pequeña, los resultados fueron en su mayoría positivos, si bien no se presentaron diferencias estadísticas significativas, generan valor para futuros trabajos. De igual forma el hallazgo más importante del estudio fue el patrón de la carga cognitiva que se alinea con los fundamentos teóricos de la Teoría de Carga Cognitiva. Otro punto interesante es la correlación de la carga pertinente y la motivación final. Si bien los resultados de los cuestionarios CIS e IMMS presentan resultados altos, debido a la naturaleza del estudio no se puede afirmar que sea suficientemente confiables.



## CAPÍTULO 8

### DISCUSIÓN

El modelo instruccional propuesto es una herramienta de apoyo para el proceso de enseñanza-aprendizaje de la programación. Presenta un diseño integral basado en la Teoría de la Carga Cognitiva, complementado por el modelo 4C/ID, los Diez Pasos para el Aprendizaje Complejo y protocolos verbales. Para abordar la heterogeneidad en el conocimiento previo, el modelo integra el uso de la instrucción diferenciada. En cuanto a la motivación, el modelo ARCS se incorpora a la estructura del modelo instruccional. El estudio piloto cuasi-experimental arrojó resultados positivos respecto a la efectividad del modelo y sus implicaciones para la práctica educativa.

Los resultados de la prueba de conocimiento declarativo (teórico) no mostraron diferencias estadísticamente significativas entre las mediciones pre y post-prueba. Sin embargo, el hallazgo más notable del estudio es la diferencia estadísticamente significativa alcanzada por el grupo experimental en los objetivos de aprendizaje de la prueba de conocimiento práctico (CPOST), tras ajustar por las diferencias iniciales, con un tamaño del efecto medio ( $d$  de Cohen = 0,41). Esto indica una mejora del 10,1 % en el rendimiento de comprensión, lo que sugiere que el modelo favorece el desarrollo de la comprensión en lugar de la memorización de conceptos.

Reducir la carga cognitiva en el proceso de aprendizaje es el objetivo principal o premisa del CLT; sin embargo, no se observaron diferencias estadísticamente significativas en las mediciones (IL, EL o GL) entre los grupos. En base a esto, planteamos la hipótesis de que es probable que (1) ambas estrategias de enseñanza funcionaran dentro de límites razonables de carga cognitiva, (2) la naturaleza subjetiva de la herramienta de evaluación de la carga cognitiva dependa de la honestidad de los estudiantes al responder el cuestionario, o (3) las ventajas del modelo CLT4BP se demuestren mediante una mejora en el procesamiento cognitivo en lugar de una disminución de la carga cognitiva.

Al igual que la carga cognitiva, la motivación no mostró diferencias estadísticamente significativas entre sus componentes. Sin embargo, el grupo experimental presentó inicialmente niveles más altos de ansiedad (MA I). El instrumento de medición también depende de la honestidad de los estudiantes, lo que indica un área para futuras investigaciones.

La heterogeneidad de conocimientos previos de los estudiantes en un curso introductorio de programación genera una de las necesidades más importantes en las aulas de programación: la capacidad de integrar a estudiantes con diferentes niveles de conocimiento (principiante e intermedio) en un mismo marco de enseñanza. El modelo propuesto ayuda a los docentes a abordar este desafío mediante el diseño de materiales didácticos adaptados a los niveles de conocimiento de los estudiantes.

El diseño del modelo a tres fases (Análisis, Diseño y Desarrollo, Implementación y Evaluación) ofrece un enfoque organizado, adaptable e iterativo para el diseño instruccional. Los educadores pueden aplicar el modelo en la práctica utilizando los marcos integrales que se muestran en el documento. Para los protocolos verbales, se puede integrar la tecnología de IA para generar dicho material. Esta conexión tecnológica abre la posibilidad de un uso más amplio en entornos de aprendizaje semipresencial y en línea.

El diseño y la creación del Producto Mínimo Viable (PMV) de la aplicación web CLT4BP

constituyen una prolongación lógica del modelo instruccional hacia un ambiente tecnológico operativo. Su visión como instrumento de mediación responde a una de las restricciones metodológicas detectadas: la necesidad del instructor de implementar manualmente los principios del modelo en materiales instructivos específicos, un proceso que consume mucho tiempo.

Desde el punto de vista de la Teoría de la Carga Cognitiva, la herramienta se centra en disminuir la carga intrínseca del alumno a través de la elaboración de materiales diferenciados que se basan en los niveles de conocimiento previo detectados por medio del módulo de análisis de datos. Además, disminuye tanto el tiempo como el trabajo del profesor al generar material instruccional diferenciado. Lo anterior diferencia a CLT4BP de otras propuestas de integración de inteligencia artificial que, basándose en la revisión bibliográfica, tienden a ser meramente personalizadas en términos de contenido sin una base teórica sobre aprendizaje.

No obstante, es importante tener en cuenta que el PMV mostrado es un prototipo funcional, que por definición requiere validación empírica. La exactitud del perfil creado, la idoneidad de la solicitud del sistema y las habilidades del modelo subyacente son determinantes para la calidad del material producido por la IA. La bibliografía revisada señala peligros, como el declive del razonamiento lógico y la dependencia excesiva del alumno en las herramientas de IA cuando estas reemplazan, en vez de complementar, la resolución de problemas (Nathaniel et al., 2025).

Por último, el hecho de combinar la herramienta con la API de Claude como motor generativo supone un aporte innovador al conjunto de herramientas para enseñar programación. CLT4BP, a diferencia de las investigaciones que emplean ChatGPT o Copilot de modo genérico, establece un modelo de IA con un prompt del sistema explícito que encierra el saber teórico del modelo instruccional, convirtiendo la IA en un asistente pedagógico especializado. Esta perspectiva abre un camino importante para la investigación: determinar en qué grado la especificación pedagógica del prompt mejora la calidad y pertinencia del material producido, si se lo compara con las solicitudes genéricas.

## 8.1 Limitaciones y consideraciones metodológicas

El alcance del estudio piloto se limitó a temas de programación específicos (estructuras anidadas y punteros) dentro de una sola institución, por lo que los resultados no son generalizables, pero sí válidos como estudio de campo. El uso de métodos de medición subjetivos, donde los resultados dependen de la honestidad de las respuestas de los estudiantes, puede generar datos erróneos. Dado que se trataba de una actividad voluntaria, la participación estudiantil no fue la esperada y algunos estudiantes no completaron el proceso.

En cuanto a la carga cognitiva, los resultados similares de ambos grupos al final del cuasi-experimento generan controversia, ya que la reducción de la carga es la premisa de la teoría, o bien la herramienta de medición de la carga fue inadecuada. Futuros estudios deberían investigar técnicas alternativas para evaluar la carga cognitiva, como mediciones fisiológicas o indicadores de carga cognitiva en tiempo real, para mejorar el proceso de medición.

Un aspecto importante que requiere mayor investigación es el indicador de ansiedad inicial en el grupo experimental. Para el refinamiento del modelo, es esencial determinar si esta ansiedad es una respuesta normal en procesos de aprendizaje complejos como la programación o un aspecto que podría mejorarse e incluirse en el modelo propuesto.

## CAPÍTULO 9

### CONCLUSIONES

El modelo instruccional mapa de ruta CLT4BP, representa una contribución significativa a la investigación y la práctica en la enseñanza de la programación, centrándose en factores como el diseño instruccional, la motivación y el nivel de conocimiento del estudiante. Integra diversas teorías del aprendizaje cognitivo, teorías de la motivación, instrucción diferenciada y protocolos verbales. Los resultados del estudio piloto demuestran la eficacia del modelo para mejorar la comprensión, al tiempo que resaltan consideraciones importantes sobre la motivación y la carga cognitiva.

La principal contribución de este estudio radica en los resultados obtenidos en las pruebas de comprensión. La mejora del 10,1 % en el rendimiento de comprensión representa un avance sustancial, dados los desafíos persistentes en la enseñanza de la programación y las altas tasas de fracaso documentadas en la literatura. Como se propuso inicialmente, un diseño instruccional integral con fundamentos teóricos probados puede mejorar significativamente el aprendizaje del estudiante y la práctica docente en programación.

El uso de la instrucción diferenciada, junto con las estrategias motivacionales y los protocolos verbales, complementa las teorías cognitivas y aborda las críticas que estas plantean. Ofrecen un enfoque holístico que reconoce los factores cognitivos, motivacionales e individuales que influyen en el aprendizaje de la programación.

Sin embargo, el estudio piloto también revela la complejidad de implementar modelos instruccionales integrales y la necesidad de investigación continua para que el proceso de implementación sea más sencillo y ágil. Los hallazgos inesperados sobre la carga cognitiva y la ansiedad inicial resaltan la importancia de la evaluación continua y el perfeccionamiento del modelo.

En conclusión, el modelo instruccional mapa de ruta CLT4BP representa un enfoque prometedor para la enseñanza de la programación que merece mayor investigación y una implementación cuidadosa. Su fundamento teórico, su aplicabilidad práctica y su eficacia comprobada para mejorar la comprensión la posicionan como una valiosa contribución a la investigación y la práctica en la enseñanza de la programación.

El diseño y desarrollo de la aplicación web CLT4BP como Producto Mínimo Viable constituye una contribución tecnológica y pedagógica complementaria al modelo instruccional. Su desarrollo responde a la necesidad práctica de operacionalizar los principios del modelo en un entorno accesible, sistemático y escalable, reduciendo la brecha entre el diseño instruccional teórico y su implementación en el aula. La herramienta se enfoca en trasladar los principios del modelo CLT4BP al entorno digital, esta fidelidad al modelo teórico distingue a CLT4BP de las herramientas de IA para educación que carecen de fundamento pedagógico explícito.

El desarrollo del PMV abre múltiples líneas de investigación futura, entre las que destacan: la validación empírica de la calidad del material generado frente al diseñado manualmente por instructores expertos; el análisis del impacto de la especificación pedagógica del prompt en los resultados de aprendizaje; y el estudio del efecto de la herramienta sobre la carga de trabajo del instructor y la personalización efectiva del proceso de enseñanza-aprendizaje.

TESIS TESIS TESIS TESIS TESIS

En suma, la aplicación web CLT4BP demuestra la viabilidad de integrar la inteligencia artificial generativa en el diseño instruccional de manera fundamentada, responsable y alineada con los principios de la Teoría de la Carga Cognitiva. Su desarrollo no solo complementa el modelo instruccional propuesto, sino que establece un precedente metodológico para la construcción de herramientas educativas que incorporen IA con sustento teórico sólido.



## REFERENCIAS

- Alahmad, M. (2020). Strengths and weaknesses of cognitive theory. *Budapest International Research and Critics Institute (BIRCI-Journal): Humanities and Social Sciences*, 3(3), 1584–1593.
- Alomyan, H., & Green, D. (2019). Learning theories: Implications for online learning design. In *Acm international conference proceeding series* (pp. 126–130).
- Altadmri, A., Kölling, M., & Brown, N. C. C. (2016). The cost of syntax and how to avoid it: Text versus frame-based editing. In *Proceedings - international computer software and applications conference* (Vol. 1, pp. 748–753).
- Andersen, S. A. W., Mikkelsen, P. T., Konge, L., Cayé-Thomasen, P., & Sørensen, M. S. (2016). The effect of implementing cognitive load theory-based design principles in virtual reality simulation training of surgical skills: a randomized controlled trial. *Advances in Simulation*, 1(1), 1–8.
- Andrade-Lotero, L. A. (2012). Cognitive load theory, design and multimedia learning: A state of the art. *Magis: Revista Internacional de Investigación En Educación*, 5(10), 75–92.
- Andreessen, B. M. (2011). Why software is eating the world. *The Wall Street Journal*, 1–9.
- Ardiana, D. P. Y., & Loekito, L. H. (2020). Gamification design to improve student motivation on learning object-oriented programming. In *Journal of physics: Conference series* (Vol. 1516).
- Arevalo-Mercado, C., Muñoz-Andrade, E., Cardona Reyes, H., & Romero-Juárez, M. (2023). Applying cognitive load theory and the split attention effect to learning data structures. *IEEE Revista Iberoamericana de Tecnologías Del Aprendizaje*, 18(1), 107–113.
- Argelio, C., Mercado, A., Lizbeth, E., Andrade, M., Manuel, J., & Reynoso, G. (2018). El efecto de la autoeficacia y el trabajo colaborativo en estudiantes novatos de programación. *Investigación y Ciencia de la Universidad Autónoma de Aguascalientes*.
- Arévalo, C., & Solano, L. (2013). Patrones de comportamiento de estudiantes de programación al utilizar una herramienta de visualización de protocolos verbales. In *Laclo 2013*.
- Arévalo, M. C. (2010). *Teoría de diseño de sistemas de información para sistemas visualizadores de protocolo verbales para el aprendizaje de la programación básica* (Unpublished doctoral dissertation).
- Aureliano, V. C. O., Tedesco, P. C. D. A. R., & Caspersen, M. E. (2016). Learning programming through stepwise self-explanations. In *Iberian conference on information systems and technologies, cisti*.
- Bandhu, D., Mohan, M. M., Nittala, N. A. P., Jadhav, P., Bhadauria, A., & Saxena, K. K. (2024). Theories of motivation: A comprehensive analysis of human behavior drivers. *Acta Psychologica*, 244.
- Beege, M., Schneider, S., Nebel, S., Zimm, J., Windisch, S., & Rey, G. D. (2021). Learning programming from erroneous worked-examples. which type of error is beneficial for learning? *Learning and Instruction*, 75, 101497.
- Beltrán, J., Sánchez, H., & Rico, M. (2015). Análisis cuantitativo y cualitativo del aprendizaje de programación i en la universidad central del ecuador. *Revista Tecnológica - ESPOL*, 28(5), 194–210.
- Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, 39(2), 32–36.
- Bennedsen, J., & Caspersen, M. E. (2019). Failure rates in introductory programming - 12 years later. *ACM Inroads*, 10(2), 30–35.
- Berssanette, J. H., & de Francisco, A. C. (2021). Cognitive load theory in the context of teaching and learning computer programming: A systematic literature review. *IEEE Transactions on Education*, 65(3), 440–449.
- Bofferding, L., Kocabas, S., Aqazade, M., Haiduc, A.-M., & Chen, L. (2022). The effect of play and worked examples on first and third graders' creating and debugging of programming algorithms. In *Computational thinking in prek-5* (pp. 19–29).
- Boldbaatar, N., & Şendurur, E. (2019). Developing educational 3d games with starlogo: The role of backwards fading in the transfer of programming experience. *Journal of Educational Computing Research*, 57(6), 1468–1494.
- Bosch, N., Zhang, Y., Paquette, L., Baker, R. S., Ocumpaugh, J., & Biswas, G. (2021, May). Students' verbalized metacognition during computerized learning. In *Conference on human factors in computing systems - proceedings*.
- Bounajim, D., Rachmatullah, A., Hinckle, M., Mott, B., Lester, J., Smith, A., ... Wiebe, E. (2021). Applying



- cognitive load theory to examine stem undergraduate students' experiences in an adaptive learning environment: A mixed-methods study. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 65, pp. 556–560).
- Bridson, K. A. (2020). *Implications for the design of tutorial systems for teaching program comprehension: An empirical user study* (Unpublished doctoral dissertation). University of Memphis.
- Bustingorry, S. O., & Mora, S. J. (2008). Metacognition: Un camino para aprender a aprender. *Estudios Pedagógicos*, 34(1), 187–197.
- Castro-Alonso, J. C., de Koning, B. B., Fiorella, L., & Paas, F. (2021). Five strategies for optimizing instructional materials: Instructor- and learner-managed cognitive load. *Educational Psychology Review*, 33(4), 1379–1407.
- Centeno Bragado, L. A. (2018). *Virtualización de un juego serio para el aprendizaje de la programación* (Unpublished doctoral dissertation). Universidad de Valladolid.
- Cerón Garnica, C., Archundia Sierra, E., Cervantes Márquez, A. P., & Beltrán Martínez, B. (2021). Experiencias de las e-actividades de evaluación de las competencias del nivel básico del área de programación. *Tecnología Educativa Revista CONAIC*, 5(3), 65–74.
- Cevahir, H., Özdemir, M., & Baturay, M. H. (2022). The effect of animation-based worked examples supported with augmented reality on the academic achievement, attitude and motivation of students towards learning programming. *Participatory Educational Research*, 9(3), 226–247.
- Chang, N.-C., & Chen, H.-H. (2015). A motivational analysis of the arcs model for information literacy courses in a blended learning environment. *Libri*, 65(2).
- Chang, Y. H., Song, A. C., & Fang, R. J. (2018). Integrating arcs model of motivation and pbl in flipped classroom: A case study on a programming language. *Eurasia Journal of Mathematics, Science and Technology Education*, 14(12).
- Cheah, C. S., & Leong, L. M. (2019). Investigating the redundancy effect in the learning of c++ computer programming using screencasting. *International Journal of Modern Education and Computer Science*, 11(6), 19–25.
- Chen, O., Kalyuga, S., & Sweller, J. (2015). The worked example effect, the generation effect, and element interactivity. *Journal of Educational Psychology*, 107(3), 689–704.
- Chen, O., Kalyuga, S., & Sweller, J. (2017). The expertise reversal effect is a variant of the more general element interactivity effect. *Educational Psychology Review*, 29(2), 393–405.
- Chen, O., Retnowati, E., & Kalyuga, S. (2019). Effects of worked examples on step performance in solving complex problems. *Educational Psychology*, 39(2), 188–202.
- Chen, O., Woolcott, G., & Sweller, J. (2017). Using cognitive load theory to structure computer-based learning including moocs. *Journal of Computer Assisted Learning*, 33(4), 293–305.
- Cheng, L., & Beal, C. R. (2020). Effects of student-generated drawing and imagination on science text reading in a computer-based learning environment. *Educational Technology Research and Development*, 68(1), 225–247.
- Cheng, Y. P., Shen, P. D., Hung, M. L., Tsai, C. W., Lin, C. H., & Hsu, L. C. (2022). Applying online content-based knowledge awareness and team learning to develop students' programming skills, reduce their anxiety, and regulate cognitive load in a cloud classroom. *Universal Access in the Information Society*, 21(2), 557–572.
- Compañ-Rosique, P., Satorre-Cuerda, R., Llorens-Largo, F., & Molina-Carmona, R. (2015). Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *Revista de Educación a Distancia (RED)*(46).
- Cáceres, Z., & Munévar, O. (2017). Evolucion de las teorías cognitivas y sus aportes a la educación. *Actividad Física Y Desarrollo Humano*, 7(2).
- Darejeh, A., Mashayekh, S., & Marcus, N. (2022). Cognitive-based methods to facilitate learning of software applications via e-learning systems. *Cogent Education*, 9(1).
- Dawar, D. (2020). Towards improving student expectations in introductory programming course with incrementally scaffolded approach. In *Edsig conference* (Vol. 19, pp. 1–16).
- Deci, E. L. (1972). Intrinsic motivation, extrinsic reinforcement, and inequity. *Journal of Personality and Social Psychology*, 22(1), 113–120.
- Deci, E. L., & Ryan, R. M. (2008). Self-determination theory: A macrotheory of human motivation, development, and health. *Canadian Psychology / Psychologie Canadienne*, 49(3), 182–185.



- de Koning, B. B., Rop, G., & Paas, F. (2020a). Effects of spatial distance on the effectiveness of mental and physical integration strategies in learning from split-attention examples. *Computers in Human Behavior*, 110.
- de Koning, B. B., Rop, G., & Paas, F. (2020b). Learning from split-attention materials: Effects of teaching physical and mental learning strategies. *Contemporary Educational Psychology*, 61, 101873.
- Domurath, A., Coviello, N., Patzelt, H., & Ganai, B. (2020). New venture adaptation in international markets: A goal orientation theory perspective. *Journal of World Business*, 55(1), 101019.
- Duran, R., Zavgorodniaia, A., & Sorva, J. (2022). Cognitive load theory in computing education research: A review. *ACM Transactions on Computing Education*, 1–26.
- Eitel, A., Endres, T., & Renkl, A. (2020). Self-management as a bridge between cognitive load and self-regulated learning: the illustrative case of seductive details. *Educational Psychology Review*, 32(4), 1073–1087.
- Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological review*, 87(3), 215.
- Eteng, I., Akpotuzor, S., Akinola, S. O., & Agbonlahor, I. (2022). A review on effective approach to teaching computer programming to undergraduates in developing countries. *Scientific African*, 16, e01240.
- Fabic, G. V. F., Mitrovic, A., & Neshatian, K. (2017). Investigating the effectiveness of menu-based self-explanation prompts in a mobile python tutor. In *Aied 2017* (Vol. 1, pp. 498–501).
- Fey, G., & Drechsler, R. (2020). Self-explaining digital systems: Technical view, implementation aspects, and completeness. In *Advanced boolean techniques* (pp. 1–20).
- Filipova, A. A. (2016). Expectancy theory in organizations. In *Global encyclopedia of public administration, public policy, and governance* (pp. 1–9). Springer International Publishing.
- Fonseca, N. G., Macedo, L., & Mendes, A. J. (2018). Supporting differentiated instruction in programming courses through permanent progress monitoring. In *Sigcse 2018 - proceedings of the 49th acm technical symposium on computer science education* (pp. 209–214).
- Fraser, S., Bak, L., DeLine, R., Feamster, N., Kuper, L., Lopes, C., & Wu, P. (2015). The future of programming languages and programmers (panel). In *Splash companion 2015 - companion proceedings of the 2015 acm sigplan international conference on systems, programming, languages and applications: Software for humanity* (pp. 63–66).
- Frederiksen, J. G., Sørensen, S. M. D., Konge, L., Svendsen, M. B. S., Nobel-Jørgensen, M., Bjerrum, F., & Andersen, S. A. W. (2020). Cognitive load and performance in immersive virtual reality versus conventional virtual reality simulation training of laparoscopic surgery: a randomized trial. *Surgical Endoscopy*, 34(3), 1244–1252.
- Frerejean, J., van Geel, M., Keuning, T., Dolmans, D., van Merriënboer, J. J. G., & Visscher, A. J. (2021). Ten steps to 4c/id: training differentiation skills in a professional development program for teachers. *Instructional Science*, 49(3), 395–418.
- Frerejean, J., van Merriënboer, J. J. G., Kirschner, P. A., Roex, A., Aertgeerts, B., & Marcellis, M. (2019). Designing instruction for complex learning: 4c/id in higher education. *European Journal of Education*, 54(4), 513–524.
- Garces, S., Vieira, C., Ravai, G., & Magana, A. J. (2023). Engaging students in active exploration of programming worked examples. *Education and Information Technologies*, 28(3), 2869–2886.
- George, D., & Mallery, P. (2024). *Ibm spss statistics 29 step by step: A simple guide and reference*. routledge.
- Gheyssens, E., Coubergs, C., Griful-Freixenet, J., Engels, N., & Struyven, K. (2022). Differentiated instruction: the diversity of teachers' philosophy and praxis to adapt teaching to students' interests, readiness and learning profiles. *International Journal of Inclusive Education*, 26(14), 1383–1400.
- Gibbs, K., & McKay, L. (2021). Differentiated teaching practices of australian mainstream classroom teachers: A systematic review and thematic analysis. *International Journal of Educational Research*, 109.
- Goswami, L., Senges, A., Estier, T., & Cherubini, M. (2023). Supporting co-regulation and motivation in learning programming in online classrooms. *Proceedings of the ACM on Human-Computer Interaction*, 7(CSCW2).
- Graham, S. (1991). A review of attribution theory in achievement contexts. *Educational Psychology Review*, 3(1), 5–39.
- Graham, S. (2020). An attributional theory of motivation. *Contemporary Educational Psychology*, 61, 101861.

- Green, A. (1998). Verbal protocol analysis in language testing research.
- Guzman, L. M., Pennell, M. W., Nikelski, E., & Srivastava, D. S. (2019). Successful integration of data science in undergraduate biostatistics courses using cognitive load theory. *CBE Life Sciences Education*, 18(4), 1–10.
- Heitzmann, N., Fischer, F., & Fischer, M. R. (2018). Worked examples with errors: when self-explanation prompts hinder learning of teachers diagnostic competences on problem-based learning. *Instructional Science*, 46(2), 245–271.
- Hsu, T. H., Horng, G. J., & See, A. R. (2021). Change in learning motivation observed through the introduction of design thinking in a mobile application programming course. *Sustainability*, 13(13).
- Huang, H., & Li, Y. (2021). Exploring the motivation of livestreamed users in learning computer programming and coding. *The Electronic Journal of E-Learning*, 19(5), 363–375.
- Ilmiani, A. M., Wahdah, N., & Mubarak, M. R. (2021). The application of albert bandura’s social cognitive theory: A process in learning speaking skill. *Ta’lim al-’Arabiyyah: Jurnal Pendidikan Bahasa Arab & Kebahasaaraban*, 5(2).
- Insuasti, J. (2016). Problemas de enseñanza y aprendizaje de los fundamentos de programación. *Revista Educación y Desarrollo Social*, 10(2), 12011–15318.
- Janssen, J., & Kirschner, P. A. (2020). Applying collaborative cognitive load theory to computer-supported collaborative learning: towards a research agenda. *Educational Technology Research and Development*, 68(2), 783–805.
- Justo, A. C., García, L. C., Elizabeth, W., Salinas, A., & Lara, F. (2021). Estrategias educativas digitales como apoyo a cursos de ciencias básicas de ingeniería. , 52–67.
- Juárez, J., López, M., & Villareal, Y. (2016). Estrategias para reducir el índice de reprobación en fundamentos de programación de sistemas computacionales del i. t. mexicali. *Revista de Gestión Empresarial y Sustentabilidad*, 2, 25–41.
- Kato, J., & Shimakage, K. (2020). Rethinking programming ‘environment’: Technical and social environment design toward convivial computing. In *Acm international conference proceeding series* (pp. 149–157).
- Kazimoglu, C. (2020). Enhancing confidence in using computational thinking skills via playing a serious game: A case study to increase motivation in learning computer programming. *IEEE Access*, 8, 221831–221851.
- Keller, J. M. (2010a). The arcs model of motivational design. In *Motivational design for learning and performance* (pp. 43–74). Springer US.
- Keller, J. M. (2010b). Tools to support motivational design. In *Motivational design for learning and performance* (pp. 267–295). Springer US.
- Kim, J., Agarwal, S., Marotta, K., Li, S., Leo, J., & Chau, D. H. (2019). Mixed reality for learning programming. In *Proceedings of the 18th acm international conference on interaction design and children, idc 2019* (pp. 574–579).
- Kirschner, P. A., Sweller, J., Kirschner, F., & Zambrano, J. R. (2018). From cognitive load theory to collaborative cognitive load theory. *International Journal of Computer-Supported Collaborative Learning*, 13(2), 213–233.
- Kohen-Vacs, D., Usher, M., & Jansen, M. (2025). Integrating generative ai into programming education: Student perceptions and the challenge of correcting ai errors. *International Journal of Artificial Intelligence in Education*, 1–19.
- Korkmaz, O., Çakir, R., & Ozden, M. Y. (2017). A validity and reliability study of the computational thinking scales (cts). *Computers in Human Behavior*, 72, 558–569.
- Kátai, Z., & Osztián, E. (2021). Improving algorithimics teaching-learning environment by asking questions. *International Journal of Instruction*, 14(2), 27–44.
- Kátai, Z., & Osztián, E. (2022). Visualizing algorithms: Schematic computer animations versus realistic dance choreography illustrations. *Acta Polytechnica Hungarica*, 19(1), 193–210.
- Kátai, Z., & Osztián, E. (2023). Learning computer algorithms through dynamic visualizations: Benefits of ‘algorithimics’ videos. *Journal of Computer Assisted Learning*.
- Lam, W. C., King, R. B., Yeung, S. S.-S., & Zhoc, C. H. (2023). Mind-sets in early childhood: The relations among growth mindset, engagement and well-being among first grade students. *Early Education and Development*, 34(6), 1325–1340.
- Latoza, T. D. (2016). *Programmers are users too: Human-centered methods for improving programming tools*

- (Unpublished doctoral dissertation).
- Law, K. M. Y., Lee, V. C. S., & Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. *Computers and Education*, 55(1), 218–228.
- Li, K., & Keller, J. M. (2018). Use of the arcs model in education: A literature review. *Computers & Education*, 122, 54–62.
- Li, X. (2016). Application of cognitive load theory in programming teaching. *Journal of Higher Education Theory and Practice*, 16(6), 57–65.
- Likourezos, V., & Kalyuga, S. (2019). The variability effect in high and low guidance instruction: A cognitive load perspective. In *Proceedings of the 42nd annual conference of the mathematics education research group of australasia* (pp. 444–451).
- Likourezos, V., Kalyuga, S., & Sweller, J. (2019). The variability effect: When instructional variability is advantageous. *Educational Psychology Review*, 31(2), 479–497.
- Lin, Y. C., Liu, T. C., & Kalyuga, S. (2022). Strategies for facilitating processing of transient information in instructional videos by using learner control mechanisms. *Instructional Science*, 50(6), 863–877.
- Lin, Y. T., Yeh, M. K. C., & Tan, S. R. (2022). Teaching programming by revealing thinking process: Watching experts' live coding videos with reflection annotations. *IEEE Transactions on Education*, 65(4), 617–627.
- Liu, T. C., Lin, Y. C., Hsu, C. Y., Hsu, C. Y., & Paas, F. (2021). Learning from animations and computer simulations: Modality and reverse modality effects. *British Journal of Educational Technology*, 52(1), 304–317.
- Lu, J., Kalyuga, S., & Sweller, J. (2020). Altering element interactivity and variability in example-practice sequences to enhance learning to write chinese characters. *Applied Cognitive Psychology*, 34(4), 837–843.
- Luik, P., Lepp, M., Feklistova, L., Säde, M., Rõõm, M., Palts, T., ... Tõnisson, E. (2020). Programming moocs—different learners and different motivation. *International Journal of Lifelong Education*, 39(3), 305–318.
- Luxton-Reilly, A., Simon, Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., ... Szabo, C. (2018). Introductory programming: A systematic literature review. In *Annual conference on innovation and technology in computer science education, iticse*.
- MacNeil, S., Rogalska, M., Leinonen, J., Denny, P., Hellas, A., & Crosland, X. (2024). Synthetic students: A comparative study of bug distribution between large language models and computing students. In *Proceedings of the 2024 on acm virtual global computing education conference v. 1* (pp. 137–143).
- Maj, S. P. (2019). Cognitive load optimisation - a new, practical, easy-to-use teaching method for enhancing stem educational outcomes based on the science of learning. In *Proceedings of 2018 ieee international conference on teaching, assessment, and learning for engineering, tale 2018* (pp. 142–147).
- Mangaroska, K., Sharma, K., Gašević, D., & Giannakos, M. (2022). Exploring students' cognitive and affective states during problem solving through multimodal data: Lessons learned from a programming activity. *Journal of Computer Assisted Learning*, 38(1), 40–59.
- Margulieux, L. E., Morrison, B. B., & Decker, A. (2020). Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. *International Journal of STEM Education*, 7(1).
- Maulidya, S. R., Hasanah, R. U., & Retnowati, E. (2017). Can goal-free problems facilitating students' flexible thinking? In *Aip conference proceedings* (Vol. 1868).
- Mayer, R. E. (1998). Cognitive, metacognitive, and motivational aspects of problem solving. *Instructional Science*, 26(1–2), 49–63.
- Mayer, R. E. (2014). Cognitive theory of multimedia learning. In *The cambridge handbook of multimedia learning, second edition* (pp. 43–71).
- Mayer, R. E., & Moreno, R. (1996). A cognitive theory of multimedia learning: Implications for design principles. , 1–10.
- Medeiros, R. P., Ramalho, G. L., & Falcao, T. P. (2019). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, 62(2), 77–90.
- Miller-Cotto, D., & Auxter, A. E. (2021). Testing the ecological validity of faded worked examples in algebra. *Educational Psychology*, 41(2), 191–205.
- Mohanarajah, S. (2018). Increasing intrinsic motivation of programming students: Towards fix and play

- educational games. *Issues in Informing Science and Information Technology*, 15, 069–077.
- Morales, J., Rusu, C., & Quinones, D. (2020). Programmer experience: A systematic mapping. *IEEE Latin America Transactions*, 18(6), 1111–1118.
- Moring, A. (2022). Basic principles of motivation. In *Ai on the job* (pp. 49–65). Springer Berlin Heidelberg.
- Morris, L. S., Grehl, M. M., Rutter, S. B., Mehta, M., & Westwater, M. L. (2022). On what motivates us: A detailed review of intrinsic v. extrinsic motivation. *Psychological Medicine*, 52(10), 1801–1816.
- Morrison, B. B. (2017). Dual modality code explanations for novices: Unexpected results. In *Icer 2017 - proceedings of the 2017 acm conference on international computing education research* (pp. 226–235).
- Muschetto, T., & Siegel, J. T. (2021). Bibliometric review of attribution theory: Document cocitation analysis. *Motivation Science*, 7(4), 439–450.
- Nakagawa, E. Y., Antonino, P. O., Schnicke, F., Kuhn, T., & Liggesmeyer, P. (2021). Continuous systems and software engineering for industry 4.0: A disruptive view. *Information and Software Technology*, 135, 106562.
- Nathaniel, J., Oyelere, S. S., Suhonen, J., & Tedre, M. (2025). Literature review on the integration of generative ai in programming education. *International Journal of Artificial Intelligence in Education*, 1–32.
- Nuryadin, A., Karlimah, K., & Nugraha, M. R. (2026). Generative artificial intelligence in education: A systematic literature review on instructional materials development. *Participatory Educational Research*, 13(2), 226–247.
- Oberhauser, R., & Lecon, C. (2017). Virtual reality flythrough of program code structures. In *Acm international conference proceeding series*.
- Paas, F., & van Merriënboer, J. J. G. (2020). Cognitive-load theory: Methods to manage working memory load in the learning of complex tasks. *Current Directions in Psychological Science*, 29(4), 394–398.
- Paivio, A. (2014). *Bilingual dual coding theory and memory*.
- Pintrich, P. R., O., A., & A. (1991). Motivated strategies for learning questionnaire (mslq). *Mediterranean Journal of Social Sciences*, 6(1), 156–164.
- Prather, J., Margulieux, L., Whalley, J., Denny, P., Reeves, B. N., Becker, B. A., ... Bosch, N. (2022). Getting by with help from my friends: Group study in introductory programming understood as socially shared regulation. In *Icer 2022 - proceedings of the 2022 acm conference on international computing education research* (Vol. 1, pp. 164–176).
- Price, T. W., Williams, J. J., Solyst, J., & Marwan, S. (2020). Engaging students with instructor solutions in online programming homework. In (pp. 1–7).
- Purnama, P. W., & Retnowati, E. (2020). The effectiveness of goal-free problems for studying triangle similarity in collaborative groups. *JRAMathEdu (Journal of Research and Advances in Mathematics Education)*, 6(1), 32–45.
- Quevedo-Blasco, R., Quevedo-Blasco, V. J., & Téllez-Trani, M. (2016). Malp: Motivational assessment questionnaire of the learning process. *European Journal of Investigation in Health, Psychology and Education*, 6(2), 83–105.
- Rachmatullah, A., Reichsman, F., Lord, T., Dorsey, C., Mott, B., Lester, J., & Wiebe, E. (2021). Modeling secondary students' genetics learning in a game-based environment: Integrating the expectancy-value theory of achievement motivation and flow theory. *Journal of Science Education and Technology*, 30(4), 511–528.
- Radianti, J., Majchrzak, T. A., Fromm, J., & Wohlgenannt, I. (2020). A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers and Education*, 147, 103778.
- Rahman, M. M., Watanobe, Y., & Nakamura, K. (2020). Evaluation of source codes using bidirectional lstm neural network. In *Proceedings of the 3rd ieee international conference on knowledge innovation and invention 2020, ickii 2020* (pp. 140–143).
- Raj, A. G. S., Gu, P., Zhang, E., Annie, A. X. R., Williams, J., Halverson, R., & Patel, J. M. (2020). Live-coding vs static code examples: Which is better with respect to student learning and cognitive load? In *Ace 2020 - proceedings of the 22nd australasian computing education conference, held in conjunction with australasian computer science week* (pp. 152–159).
- Rittle-Johnson, B., Loehr, A. M., & Durkin, K. (2017). Promoting self-explanation to improve mathematics learning: A meta-analysis and instructional design principles. *ZDM - Mathematics Education*, 49(4),



- 599–611.
- Roodenrys, K., Agostinho, S., Roodenrys, S., & Chandler, P. (2012). Managing one's own cognitive load when evidence of split attention is present. *Applied Cognitive Psychology*, 26(6), 878–886.
- Ryan, R. M., & Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25(1), 54–67.
- Ryan, R. M., & Deci, E. L. (2020). Intrinsic and extrinsic motivation from a self-determination theory perspective: Definitions, theory, practices, and future directions. *Contemporary Educational Psychology*, 61, 101860.
- Sands, P. (2019). Addressing cognitive load in the computer science classroom. *ACM Inroads*, 10(1), 44–51.
- Shang, C., Moss, A. C., & Chen, A. (2023). The expectancy-value theory: A meta-analysis of its application in physical education. *Journal of Sport and Health Science*, 12(1), 52–64.
- Shin, S. S. (2020). Structured query language learning: Concept map-based instruction based on cognitive load theory. *IEEE Access*, 8, 100095–100110.
- Silva-Maceda, G., David Arjona-Villicana, P., & Edgar Castillo-Barrera, F. (2016). More time or better tools? a large-scale retrospective comparison of pedagogical approaches to teach programming. *IEEE Transactions on Education*, 59(4), 274–281.
- Simon, Luxton-Reilly, A., Ajanovski, V. V., Fouh, E., Gonsalvez, C., Leinonen, J., ... Thota, N. (2019). Pass rates in introductory programming and in other stem disciplines. In *Annual conference on innovation and technology in computer science education, iticse* (pp. 53–71).
- Singh, G. (2017). Using virtual reality for scaffolding computer programming learning. In *Proceedings of the acm symposium on virtual reality software and technology, vrst*.
- Sithole, S. T. M., Chandler, P., Abeysekera, I., & Paas, F. (2017). Benefits of guided self-management of attention on learning accounting. *Journal of Educational Psychology*, 109(2), 220–232.
- Solarte Pabón, O., & Machuca Villegas, L. (2019). Fostering motivation and improving student performance in an introductory programming course: An integrated teaching approach. *Revista EIA*, 16(31), 65–76.
- Souza, S. M., & Batista, R. A. (2019). Motivation and engagement with pbl in an introductory programming course.
- Spitzberg, B. H., & Manusov, V. (2021). Attribution theory. In *Engaging theories in interpersonal communication* (pp. 39–51). Routledge.
- Stavropoulou, G., Stamovlasis, D., & Gonida, S.-E. (2023). Probing the effects of perceived teacher goals and achievement-goal orientations on students' self-efficacy, cognitive and metacognitive strategies in writing: A person-centered approach. *Learning and Motivation*, 82, 101888.
- Sugiman, Retnowati, E., Ayres, P., & Murdanu. (2019). Learning goal-free problems: Collaboratively or individually? *Cakrawala Pendidikan*, 38(3), 590–600.
- Sullivan, S., Swanson, H., & Edwards, J. (2021). Student attitudes toward syntax exercises in cs1. In *Sigcse 2021 - proceedings of the 52nd acm technical symposium on computer science education* (pp. 782–788).
- Sweller, J. (2016). Cognitive load theory and computer science education. In (pp. 1–1).
- Sweller, J. (2020). Cognitive load theory and educational technology. *Educational Technology Research and Development*, 68(1), 1–16.
- Sweller, J., Ayres, P., & Kalyuga, S. (2011). *Cognitive load theory*. Springer Science+Business Media, LLC.
- Sweller, J., van Merriënboer, J., & Paas, F. (1998). Cognitive architecture and instructional design. *Educational Psychology Review*, 10(3), 251–295.
- Sweller, J., van Merriënboer, J. J. G., & Paas, F. (2019). Cognitive architecture and instructional design: 20 years later. *Educational Psychology Review*, 31(2), 261–292.
- Tanielu, T., Akau Ola, R., Varoy, E., & Giacaman, N. (2019). Combining analogies and virtual reality for active and visual object-oriented programming. In *CompEd 2019 - proceedings of the acm conference on global computing education* (pp. 92–98).
- Tiedekunta, I. (2023). Decorative images' effect on the cognitive load of computer programming learning.
- Tomlinson, C. A. (2014). *The differentiated classroom: Responding to the needs of all learners*. ASCD.
- Toomey, N., & Heo, M. (2022). The role of dispositional envy, state envy, and social comparisons in predicting learning outcomes. *Learning and Motivation*, 80, 101845.
- Toukiloglou, P., & Xinogalos, S. (2023). Adaptive support with working examples in serious games about programming. *Journal of Educational Computing Research*, 61(4), 766–789.

- Tsai, M. J., Wang, C. Y., & Hsu, P. F. (2019). Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research*, 56(8), 1345–1360.
- Tsai, M.-L., Ong, C. W., & Chen, C.-L. (2023). Exploring the use of large language models (llms) in chemical engineering education: Building core course problem models with chat-gpt. *Education for Chemical Engineers*, 44, 71–95.
- Ubaidullah, N. H., Mohamed, Z., Hamid, J., Sulaiman, S., & Yussof, R. L. (2021). Improving novice students' computational thinking skills by problem-solving and metacognitive techniques. *International Journal of Learning, Teaching and Educational Research*, 20(6), 88–108.
- Ullah, Z., Lajis, A., Jamjoom, M., Altalhi, A., & Saleem, F. (2020). Bloom's taxonomy: A beneficial tool for learning and assessing students' competency levels in computer programming using empirical analysis. *Computer Applications in Engineering Education*, 28(6), 1628–1640.
- Urdu, T., & Kaplan, A. (2020). The origins, evolution, and future directions of achievement goal theory. *Contemporary Educational Psychology*, 61, 101862.
- Vallerand, R. J., Pelletier, L. G., Blais, M. R., Briere, N. M., Senecal, C., & Vallieres, E. F. (1992). The academic motivation scale: A measure of intrinsic, extrinsic, and amotivation in education. *Educational and Psychological Measurement*, 52(4), 1003–1017.
- Van Merriënboer, J. J., & Kirschner, P. A. (2018). 4c/id in the context of instructional design and the learning sciences. In *International handbook of the learning sciences* (pp. 169–179). Routledge.
- van Merriënboer, J. J. G., & Kirschner, P. (2018). *Ten steps to complex learning: A systematic approach to four-component instructional design* (3rd ed.). Routledge.
- Velazquez-Iturbide, J. A. (2017). Exploring the joint use of educational theories and information technology to improve cs courses. In *Ieee global engineering education conference, educon* (pp. 1561–1570).
- Vieira, C., Magana, A. J., Falk, M. L., & Garcia, R. E. (2017). Writing in-code comments to self-explain in computational science and engineering education. *ACM Transactions on Computing Education*, 17(4), 1–21.
- Wang, H., Xu, M., Xie, X., Dong, Y., & Wang, W. (2021). Relationships between achievement goal orientations, learning engagement, and academic adjustment in freshmen: Variable-centered and person-centered approaches. *Frontiers in Psychology*, 12.
- Wang, Y., Lin, L., & Chen, O. (2021). The benefits of teaching on comprehension, motivation, and perceived difficulty: Empirical evidence of teaching expectancy and the interactivity of teaching. *British Journal of Educational Psychology*, 91(4), 1275–1290.
- Watson, C., & Li, F. W. B. (2014). Failure rates in introductory programming revisited. In *Iticse 2014 - proceedings of the 2014 innovation and technology in computer science education conference* (pp. 39–44).
- Weems, C., & Naps, D. (2007). *Programación y resolución de problemas con c++*.
- Weiss Allen, M. (2013). *Estructura de datos*. Universidad Tecnológica de Pereira.
- Wigfield, A., & Cambria, J. (2010). Expectancy-value theory: retrospective and prospective. In (pp. 35–70).
- Winter, V., Friend, M., Matthews, M., Love, B., & Vasireddy, S. (2019). Using visualization to reduce the cognitive load of threshold concepts in computer programming. In *Proceedings - frontiers in education conference, fie* (pp. 1–9).
- Yen, C. W., & Wang, T. I. (2017). Using self-explanation and ontology for providing proper feedbacks in a programming environment. In *Proceedings - 2017 6th iiai international congress on advanced applied informatics, iiai-aaai 2017* (pp. 585–590).
- Yilmaz, R., & Karaoglan Yilmaz, F. G. (2023). The effect of generative artificial intelligence (ai)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence*, 4.
- Zambrano R., J., Kirschner, F., Sweller, J., & Kirschner, P. A. (2019). Effects of prior knowledge on collaborative and individual learning. *Learning and Instruction*, 63, 101214.
- Zavgorodniaia, A., Hellas, A., Seppälä, O., & Sorva, J. (2020). Should explanations of program code use audio, text, or both? a replication study. In *Acm international conference proceeding series*.
- Zhang, L., Kalyuga, S., Lee, C., & Lei, C. (2016). Effectiveness of collaborative learning of computer programming under different learning group formations according to students' prior knowledge: A cognitive load perspective. *Journal of Interactive Learning Research*, 27(2), 171–192.
- Zhang, S., de Koning, B. B., & Paas, F. (2022). Finger pointing to self-manage cognitive load in learning



- from split-attention examples. *Applied Cognitive Psychology*, 36(4), 767–779.
- Zhao, J., Thille, C., Gattani, N., & Zimmaro, D. (2021). A novel framework for discovering cognitive models of learning. In *L@s 2021 - proceedings of the 8th acm conference on learning @ scale* (pp. 271–274).
- Zhi, R., Price, T. W., Marwan, S., Milliken, A., Barnes, T., & Chi, M. (2019). Exploring the impact of worked examples in a novice programming environment. In *Sigcse 2019 - proceedings of the 50th acm technical symposium on computer science education* (pp. 98–104).
- Çakiroglu, U., & Aksoy, D. A. (2017). Exploring extraneous cognitive load in an instructional process via the web conferencing system. *Behaviour and Information Technology*, 36(7), 713–725.



## CAPÍTULO A

### ANEXOS

#### A.1 Motivated Strategies for Learning Questionnaire (MSLQ).

##### Motivación (31 ítems)

Evalúa aspectos relacionados con los objetivos, el valor que los estudiantes asignan a las tareas académicas, y sus creencias sobre sus capacidades. Se divide en las siguientes subescalas:

- Orientación intrínseca hacia la meta: Grado en que el aprendizaje se realiza por interés o placer.
- Orientación extrínseca hacia la meta: Enfoque en logros externos como calificaciones o reconocimiento.
- Valor de la tarea: Creencias sobre la utilidad, importancia y disfrute de una tarea.
- Autoeficacia: Confianza en la capacidad de realizar tareas académicas.
- Ansiedad ante los exámenes: Nivel de preocupación o nerviosismo frente a pruebas.

##### Detalle de preguntas

##### **Orientación intrínseca hacia la meta:**

1. En una clase como esta, prefiero material de curso que realmente me desafíe para poder aprender cosas nuevas.
2. En una clase como ésta, prefiero material que despierte mi curiosidad, aunque sea difícil de aprender.
3. Lo más satisfactorio para mí de este curso es intentar comprender el contenido lo más a fondo posible.
4. Cuando tengo la oportunidad en esta clase, elijo tareas del curso que puedo aprender incluso si no garantizan una buena calificación.

##### **Orientación extrínseca hacia la meta:**

5. Obtener una buena nota en esta clase es lo más satisfactorio para mí en este momento.
6. Lo más importante para mí en este momento es mejorar mi promedio general de calificaciones, por lo que mi principal preocupación en esta clase es obtener una buena calificación.
7. Si puedo, quiero obtener mejores notas en esta clase que la mayoría de los otros estudiantes.
8. Quiero tener un buen desempeño en esta clase porque es importante mostrar mi capacidad a mi familia, amigos, empleador u otros.

##### **Valor de la tarea:**

9. Creo que podré utilizar lo aprendido en este curso en otros cursos.
10. Es importante para mí aprender el material del curso en esta clase.
11. Estoy muy interesado en el contenido de este curso.

12. Pienso que el material del curso en esta clase es útil para aprender.
13. Me gusta el tema de este curso.
14. Para mí es muy importante comprender el tema de este curso.

**Autoeficacia:**

15. Si estudio de manera apropiada, entonces podré aprender el material de este curso.
16. Es mi culpa si no aprendo el material de este curso.
17. Si me esfuerzo lo suficiente, entenderé el material del curso.
18. Si no entiendo el material del curso es porque no me esforcé lo suficiente.
19. Creo que recibiré una calificación excelente en esta clase.
20. Estoy seguro de que puedo comprender el material más difícil presentado en las lecturas de este curso.
21. Estoy seguro de que puedo comprender los conceptos básicos enseñados en este curso.
22. Estoy seguro de que puedo comprender el material más complejo presentado por el instructor en este curso.
23. Estoy seguro de que puedo hacer un excelente trabajo en las tareas y exámenes de este curso.
24. Espero tener un buen desempeño en esta clase.
25. Estoy seguro de que puedo dominar las habilidades que se enseñan en esta clase.
26. Teniendo en cuenta la dificultad de este curso, el profesor y mis habilidades, creo que me irá bien en esta clase.

**Ansiedad:**

27. Cuando hago un examen, pienso en lo mal que me está yendo en comparación con otros estudiantes.
28. Cuando hago un examen, pienso en elementos de otras partes del examen que no puedo responder.
29. Cuando hago exámenes pienso en las consecuencias de fallar.
30. Tengo una sensación de inquietud y malestar cuando hago un examen.
31. Siento que mi corazón late rápido cuando hago un examen.

**Estrategias de Aprendizaje (50 ítems)**

Evalúa las estrategias cognitivas y metacognitivas, así como el manejo de los recursos por parte de los estudiantes. Incluye las siguientes subescalas:

**Estrategias Cognitivas:**

- Repetición: Estrategias básicas como releer o tomar notas.
- Elaboración: Conexión entre nuevos conocimientos y conocimientos previos.
- Organización: Uso de técnicas como mapas conceptuales o resúmenes.

**Estrategias Metacognitivas:**

- Control de la comprensión: Capacidad para monitorear y ajustar el aprendizaje.

**Manejo de Recursos:**

- Gestión del tiempo y el estudio: Organización del tiempo para cumplir objetivos académicos.
- Regulación del esfuerzo: Persistencia frente a tareas desafiantes.
- Búsqueda de ayuda: Capacidad para buscar apoyo cuando es necesario (de pares o profesores).
- Entorno de aprendizaje: Uso efectivo de los recursos y condiciones ambientales.

### **Detalle de preguntas**

### **Estrategias Cognitivas**

#### **Repetición**

1. Tomo notas durante las clases para ayudarme a recordar la información importante.
2. Repito mentalmente los términos clave mientras estudio para recordarlos mejor.
3. Escribo las ideas principales varias veces para reforzarlas en mi memoria.
4. Practico ejercicios repetitivos para dominar el contenido.

#### **Elaboración**

5. Relaciono el contenido que estoy aprendiendo con cosas que ya conozco.
6. Explico el material a mí mismo para entenderlo mejor.
7. Trato de encontrar ejemplos reales de los conceptos que aprendo.
8. Conecto diferentes temas para crear una visión más completa de lo que estudio.
9. Hago preguntas sobre el material para profundizar en mi comprensión.
10. Uso asociaciones para recordar la información más fácilmente.

#### **Organización**

11. Uso esquemas o diagramas para organizar la información importante.
12. Subrayo las ideas principales al leer el material de estudio.
13. Hago resúmenes de lo que aprendo para enfocarme en los puntos clave.
14. Divido el material en secciones más pequeñas para aprenderlo mejor.

### **Estrategias Metacognitivas**

#### **Control de la comprensión**

15. Reviso regularmente si estoy entendiendo lo que estoy estudiando.
16. Hago una pausa para reflexionar sobre lo que acabo de leer o aprender.
17. Me esfuerzo por encontrar soluciones alternativas si no entiendo algo.
18. Evalúo la calidad de mi trabajo académico antes de entregarlo.
19. Identifico las partes más difíciles de un tema para enfocarme en ellas.
20. Me hago preguntas a mí mismo mientras estudio para comprobar mi conocimiento.
21. Planifico mis sesiones de estudio antes de comenzar.
22. Cambió mi estrategia de aprendizaje si noto que algo no funciona.
23. Verifico si mis notas son completas después de una clase o lectura.
24. Pienso en cómo el contenido aprendido podría ser útil en el futuro.
25. Reflexiono sobre cómo mejoré en mi aprendizaje después de una tarea difícil.

26. Me pregunto cómo lo que aprendí se relaciona con mi vida diaria.

### **Manejo de Recursos**

#### **Gestión del tiempo y el estudio**

27. Establezco horarios claros para completar mis actividades académicas.

28. Intento cumplir con las fechas límite sin retrasos.

29. Distribuyo mi tiempo entre diferentes tareas para mantener el equilibrio.

30. Dedico tiempo extra a temas que me parecen más complicados.

#### **Regulación del esfuerzo**

31. Sigo trabajando en tareas difíciles, aunque me resulten frustrantes.

32. Mantengo el esfuerzo incluso cuando el material no me interesa.

33. Busco motivación extra para terminar mis estudios si pierdo interés.

34. Me obligo a estudiar incluso cuando no tengo ganas.

#### **Búsqueda de ayuda**

35. Pido ayuda a mis profesores cuando no entiendo algo.

36. Formo grupos de estudio para aprender junto con mis compañeros.

37. Me siento cómodo solicitando aclaraciones sobre temas complicados.

38. Recorro a fuentes externas, como libros o internet, si necesito más información.

#### **Entorno de aprendizaje**

39. Me aseguro de tener un lugar tranquilo y ordenado para estudiar.

40. Elijo ambientes sin distracciones para concentrarme mejor.

41. Organizo mis materiales de estudio antes de comenzar.

42. Utilizo recursos como bibliotecas o laboratorios para mejorar mi aprendizaje.

#### **Características del MSLQ**

- Formato: Cuestionario autoadministrado con escala Likert (1 a 7, donde 1 = Totalmente en desacuerdo y 7 = Totalmente de acuerdo).
- Flexibilidad: Se puede usar completo o seleccionar subescalas según el interés del investigador.
- Aplicaciones: Es utilizado en contextos educativos para identificar fortalezas y áreas de mejora en los hábitos de aprendizaje y motivación de los estudiantes.

## **A.2 Computational Thinking Scale (CTS) – Escala de pensamiento computacional**

Escala de pensamiento computacional de 29 ítems que mide la creatividad, la cooperación, el pensamiento crítico algorítmico y la resolución de problemas. El instrumento utiliza una escala tipo Likert de cinco puntos que va desde nunca (1), rara vez (2), a veces (3), generalmente (4), siempre (5).

### **Creatividad**



1. Me gustan las personas que están seguras de la mayoría de sus decisiones.
2. Me gusta la gente que es realista y neutral.
3. Creo que puedo resolver la mayoría de los problemas a los que me enfrente si tengo suficiente tiempo y si demuestro esfuerzo.
4. Tengo la convicción de que puedo resolver los problemas que puedan surgir cuando me enfrente a una nueva situación.
5. Confío en que puedo aplicar el plan mientras lo hago para resolver un problema mío.
6. Soñar hace que mis proyectos más importantes salgan a la luz.
7. Confío en mis intuiciones y sentimientos de “verdad” e “incorrecto” cuando abordo la solución de un problema.
8. Cuando me encuentro con un problema, me detengo antes de pasar a otro tema y pienso en ese problema.

### **Pensamiento algorítmico**

9. Puedo establecer inmediatamente la objetividad que dará la solución de un problema.
10. Creo que tengo un interés especial en los procesos matemáticos.
11. Creo que aprendo mejor las instrucciones realizadas con ayuda de símbolos y conceptos matemáticos.
12. Creo que puedo captar fácilmente la relación entre las cifras.
13. Puedo expresar matemáticamente las formas de solución de los problemas que enfrente en la vida diaria.
14. Puedo digitalizar un problema matemático expresado verbalmente.

### **Cooperatividad**

15. Me gusta experimentar el aprendizaje cooperativo junto con mi grupo de amigos.
16. En el aprendizaje cooperativo, creo que logro/lograré resultados más exitosos porque estoy trabajando en grupo.
17. Me gusta resolver problemas relacionados con proyectos grupales junto con mis amigos en el aprendizaje cooperativo.
18. Mas ideas surgen en el aprendizaje colaborativo.

### **Pensamiento critico**

19. Soy bueno preparando planes regulares sobre la solución de problemas complejos.
20. Es divertido intentar resolver los problemas complejos.
21. Estoy dispuesto a aprender nuevas cosas desafiantes.
22. Estoy orgulloso de poder pensar con gran precisión.
23. Utilizo un método sistémico al comparar las opciones que tengo a mano y al tomar una decisión.

### **Resolución de problemas**

24. Tengo problemas en la demostración de la solución de un problema en mi mente.
25. Tengo problemas en el tema de dónde y cómo debo utilizar las variables como X e Y en la solución de un problema.
26. No puedo aplicar las soluciones que planeo de manera respectiva y gradual.

27. No puedo generar tantas opciones mientras pienso en las posibles formas de solución a un problema.
28. No puedo desarrollar mis propias ideas en el entorno del aprendizaje cooperativo.
29. Me cansa intentar aprender algo junto con mi grupo de amigos en el aprendizaje cooperativo.

### **A.3 Computer Programming Self-Efficacy Scale (CPSES) – Escala de autoeficacia en programación**

Utilizando la escala que aparece a continuación, indique en qué medida cada uno de los siguientes elementos corresponde. Por favor sea lo más honesto posible para poder recibir una retroalimentación correcta.

- 1 – Totalmente en desacuerdo
- 2 – En desacuerdo
- 3 – Indeciso
- 4 – De acuerdo
- 5 – Totalmente de acuerdo

#### **Pensamiento lógico**

- 1. Puedo entender la estructura lógica básica de un programa.
- 2. Puedo entender una expresión condicional como `if . . . else . . .`.
- 3. Puedo predecir el resultado final de un programa con condiciones lógicas.
- 4. Puedo predecir el resultado de un programa cuando me dan sus valores de entrada.

#### **Cooperación**

- 5. Sé que el trabajo de programación se puede dividir en subtareas para las personas.
- 6. Puedo trabajar con otros mientras escribo un programa.
- 7. Puedo utilizar divisiones de tareas para mejorar la eficiencia de la programación.

#### **Algoritmia**

- 8. Puedo entender los procedimientos del programa sin una muestra.
- 9. No necesito la ayuda de otros para construir un programa.
- 10. Puedo hacer uso de la programación para resolver un problema.

#### **Control**

- 11. Puedo abrir y guardar un programa en un editor de programas.
- 12. Puedo editar y revisar un programa en un editor de programas.
- 13. Puedo ejecutar y probar un programa en un editor de programas.

#### **Depuración**

- 14. Puedo encontrar el origen de un error mientras pruebo un programa.
- 15. Puedo corregir un error mientras pruebo un programa.
- 16. Puedo aprender más sobre programación a través del proceso de depuración.

#### A.4 Course Interest Survey (CIS)

El CIS tiene 34 ítems con cantidades aproximadamente iguales en cada una de las cuatro categorías ARCS. Los ítems se enumeran a continuación en el orden en que se administran normalmente. Sin embargo, cada una de las cuatro subescalas se puede utilizar y puntuar de forma independiente. Además, el formato de la encuesta se puede modificar para utilizar escalas de tipo Likert y métodos de puntuación electrónicos.

- 1 = Nada cierto
- 2 = Ligeramente cierto
- 3 = Moderadamente cierto
- 4 = Mayormente cierto
- 5 = Muy cierto

1. El instructor sabe cómo hacernos sentir entusiasmados con el tema de este curso.
2. Las cosas que estoy aprendiendo en este curso me serán útiles.
3. Tengo confianza en que me irá bien en este curso.
4. Esta clase tiene muy poco que capte mi atención.
5. El instructor hace que el tema de este curso parezca importante.
6. Debes tener suerte para obtener buenas notas en este curso.
7. Tengo que trabajar muy duro para tener éxito en este curso.
8. No veo cómo el contenido de este curso se relaciona con algo que ya sé.
9. Depende de mí si tengo éxito o no en este curso.
10. El instructor crea suspense al ir avanzando en el curso.
11. El tema de este curso es demasiado difícil para mí.
12. Siento que este curso me da mucha satisfacción.
13. En esta clase, trato de establecer y alcanzar altos estándares de excelencia.
14. Siento que las calificaciones u otros reconocimientos que recibo son justos en comparación con otros estudiantes.
15. Los estudiantes de esta clase parecen sentir curiosidad por el tema.
16. Disfruto trabajando en este curso.
17. Es difícil predecir qué calificación dará el instructor a mis tareas.
18. Estoy satisfecho con las evaluaciones que el instructor hizo de mi trabajo en comparación con lo bien que creo que lo he hecho.
19. Me siento satisfecho con lo que he obtenido de este curso.
20. El contenido de este curso se relaciona con mis expectativas y objetivos.
21. El instructor hace cosas inusuales o sorprendentes que son interesantes.
22. Los estudiantes participan activamente en esta clase.
23. Para lograr mis objetivos, es importante que me vaya bien en este curso.
24. El instructor utiliza una interesante variedad de técnicas de enseñanza.
25. No creo que me beneficie mucho este curso.
26. A menudo sueño despierto mientras estoy en esta clase.
27. Mientras tomo esta clase, creo que puedo tener éxito si me esfuerzo lo suficiente.
28. Los beneficios personales de este curso están claros para mí.
29. Mi curiosidad a menudo se ve estimulada por las preguntas formuladas o los problemas

propuestos sobre el tema tratado en esta clase.

30. Considero que el nivel de desafío de este curso es el adecuado: ni demasiado fácil ni demasiado difícil.

31. Me siento bastante decepcionado con este curso.

32. Siento que recibo suficiente reconocimiento por mi trabajo en este curso por medio de calificaciones, comentarios u otra retroalimentación.

33. La cantidad de trabajo que tengo que realizar es apropiada para este tipo de curso.

34. Recibo suficiente retroalimentación para saber qué tan bien lo estoy haciendo.

Guía de puntuación CIS para las cuatro categorías del modelo ARCS.

| Atención       | Relevancia     | Confianza      | Satisfacción   |
|----------------|----------------|----------------|----------------|
| 1              | 2              | 3              | 7 (invertido)  |
| 4 (invertido)  | 5              | 6 (invertido)  | 12             |
| 10             | 8 (invertido)  | 9              | 14             |
| 15             | 13             | 11 (invertido) | 16             |
| 21             | 20             | 17 (invertido) | 18             |
| 24             | 22             | 27             | 19             |
| 26 (invertido) | 23             | 30             | 31 (invertido) |
| 29             | 25 (invertido) | 34             | 32             |
|                | 28             |                | 33             |

No existen normas para la encuesta. Como se trata de una medida específica para cada situación, no se espera una distribución normal de las respuestas. Las puntuaciones se determinan sumando las respuestas de cada subescala y la escala total. Tenga en cuenta que los ítems marcados invertidos se expresan de manera negativa. Las respuestas deben invertirse antes de que se puedan agregar al total de respuestas. Es decir, para los ítems invertidos, 5=1, 4=2, 3=3, 2=4, 1=5.

## A.5 Instructional Materials Motivation Scale (IMMS)

La encuesta IMMS consta de 36 ítems. Las subescalas de Relevancia y Confianza tienen 9 ítems cada una, la subescala de Satisfacción tiene 6 y la subescala de Atención tiene 12.

1 = Nada cierto

2 = Ligeramente cierto

3 = Moderadamente cierto

4 = Mayormente cierto

5 = Muy cierto

1. Cuando miré esta lección por primera vez, tuve la impresión de que sería fácil para mí.

2. Hubo algo interesante al comienzo de esta lección que llamó mi atención.

3. Este material fue más difícil de entender de lo que me hubiera gustado.

4. Después de leer la información introductoria, me sentí seguro de saber lo que debía aprender de esta lección.

5. Completar los ejercicios de esta lección me dio una satisfactoria sensación de logro.

6. Me queda claro cómo el contenido de este material se relaciona con cosas que ya sé.

7. Muchas de las páginas tenían tanta información que era difícil seleccionar y recordar los puntos importantes.
8. Estos materiales son llamativos.
9. Había historias, imágenes o ejemplos que me mostraron cómo este material podría ser importante para algunas personas.
10. Completar esta lección con éxito fue importante para mí.
11. La calidad de la escritura ayudó a mantener mi atención.
12. Esta lección es tan abstracta que me resultó difícil mantener la atención en ella.
13. Mientras trabajaba en esta lección, estaba seguro de que podía aprender el contenido.
14. Disfruté tanto esta lección que me gustaría saber más sobre este tema.
15. Las páginas de esta lección parecen secas y poco atractivas.
16. El contenido de este material es relevante para mis intereses.
17. La forma en que la información está organizada en las páginas ayudó a mantener mi atención.
18. Hay explicaciones o ejemplos de cómo las personas usan el conocimiento en esta lección.
19. Los ejercicios de esta lección fueron demasiado difíciles.
20. Esta lección tiene cosas que estimularon mi curiosidad.
21. Disfruté mucho estudiando esta lección.
22. La cantidad de repetición en esta lección hizo que a veces me aburriera.
23. El contenido y el estilo de escritura en esta lección transmiten la impresión de que vale la pena conocer su contenido.
24. Aprendí algunas cosas que fueron sorprendentes o inesperadas.
25. Después de trabajar en esta lección por un tiempo, estaba seguro de que podría aprobar un examen sobre ella.
26. Esta lección no era relevante para mis necesidades porque ya sabía la mayor parte.
27. La redacción de los comentarios después de los ejercicios o de otros comentarios en esta lección me ayudó a sentirme recompensado por mi esfuerzo.
28. La variedad de pasajes de lectura, ejercicios, ilustraciones, etc., ayudaron a mantener mi atención en la lección.
29. El estilo de escritura es aburrido.
30. Pude relacionar el contenido de esta lección con cosas que he visto, hecho o pensado en mi propia vida.
31. Hay tantas palabras en cada página que resulta irritante.
32. Me sentí bien al completar esta lección con éxito.
33. El contenido de esta lección me será útil.
34. Realmente no pude entender gran parte del material de esta lección.
35. La buena organización del contenido me ayudó a tener confianza de que aprendería este material.
36. Fue un placer trabajar en una lección tan bien diseñada.

No existen normas para la encuesta. Como se trata de una medida específica para cada situación, no se espera una distribución normal de las respuestas. Las puntuaciones se determinan sumando las respuestas de cada subescala y la escala total. Tenga en cuenta que los ítems marcados invertidos se expresan de manera negativa. Las respuestas deben invertirse antes de que se puedan agregar al total de respuestas. Es decir, para los ítems invertidos,





## A.6 Guía de referencia estrategias diferenciadas

| Componente ID          | Estrategia                                                                                                      | Ejemplo en programación                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------|-----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Contenido diferenciado | Ofrecer materiales y actividades de aprendizaje adaptados a diferentes niveles de experiencia.                  | <ul style="list-style-type: none"> <li>Principiantes: Tutoriales paso a paso con explicaciones detalladas.</li> <li>Intermedios: Ejercicios guiados con mayor autonomía.</li> <li>Avanzados: Desafíos de programación más complejos y abiertos.</li> <li>Utilizar recursos variados como tutoriales, documentación, videos y ejercicios de codificación interactivos, chatbots, IA, etc.</li> </ul>                                                                                                                                                                                                                                                  |
|                        | Ofrecer alternativas                                                                                            | <ul style="list-style-type: none"> <li>Ofrecer alternativas de lenguajes de programación o plataformas para explorar según el interés o el nivel de habilidad de los estudiantes.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Proceso diferenciado   | Trabajo en grupo                                                                                                | <ul style="list-style-type: none"> <li>Programación en pares o proyectos grupales, agrupando a los estudiantes en función de intereses o niveles de conocimientos previos.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|                        | Actividades/proyectos escalables                                                                                | <ul style="list-style-type: none"> <li>Diseñar actividades/proyectos con distintos niveles de estructura y apoyo.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|                        | Materiales diversos                                                                                             | <ul style="list-style-type: none"> <li>Utilice diagramas, diagramas de flujo o representaciones visuales de código para ilustrar conceptos (por ejemplo, un diagrama de flujo para explicar la lógica del flujo de control de un programa).</li> <li>Proporcionar enlaces a conferencias en video o podcasts que expliquen conceptos de programación relacionados con el artículo, ofreciendo una explicación verbal de temas técnicos.</li> <li>Anime a los estudiantes a combinar su investigación en papel con ejercicios prácticos de codificación, proyectos o incluso simulaciones donde puedan aplicar lo que leen en tiempo real.</li> </ul> |
| Producto diferenciado  | Ofreciendo diversas formas para que los estudiantes demuestren sus habilidades y conocimientos de programación. | <ul style="list-style-type: none"> <li>En lugar de que cada estudiante complete el mismo proyecto final, la instrucción diferenciada en programación permite a los estudiantes demostrar su aprendizaje en un formato adecuado a sus fortalezas e intereses.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                              |
|                        | Diversos métodos de evaluación                                                                                  | <ul style="list-style-type: none"> <li>Utilice diversas técnicas de evaluación, como revisiones de código, presentaciones o demostraciones prácticas, para evaluar el aprendizaje de los estudiantes de manera integral.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Entorno diferenciado   | Integración de tecnología                                                                                       | <ul style="list-style-type: none"> <li>Utilice plataformas de codificación en línea, tutoriales interactivos y juegos educativos para respaldar la instrucción diferenciada.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|                        | Permitir a los estudiantes a explorar proyectos que se alineen con sus pasiones o metas profesionales.          | <ul style="list-style-type: none"> <li>Permitir que los estudiantes trabajen en proyectos basados en intereses, como crear un sitio web personal, diseñar un juego o escribir un programa que realice una tarea que les interese.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                         |

Ejemplos de aplicación en temas básicos de programación

## 1. Contenido diferenciado

a. En una clase en la que los estudiantes están aprendiendo programación con Python, a los estudiantes con diferentes niveles de experiencia se les pueden dar diferentes niveles de complejidad en el contenido que necesitan cubrir.

- i. Los principiantes pueden comenzar con conceptos básicos como variables, bucles y condicionales, utilizando ayudas visuales o recursos simplificados.
- ii. Los estudiantes intermedios pueden explorar funciones, listas y diccionarios, con pequeños proyectos de codificación para practicar.
- iii. A los estudiantes avanzados se les pueden asignar tareas más desafiantes, como crear API, trabajar con bases de datos o implementar algoritmos.

## 2. Proceso diferenciado

### a. Para principiantes

- i. Proporcionar ejercicios de codificación guiados donde el profesor o una plataforma en línea ofrecen sugerencias e instrucciones paso a paso.
- ii. Utilice la programación en pares para que los principiantes puedan trabajar con compañeros más experimentados.
- iii. Ofrecer desafíos de codificación con retroalimentación inmediata, donde los principiantes puedan identificar y corregir errores rápidamente (por ejemplo, trabajar en ejercicios de codificación simples en un IDE con detección de errores incorporada).

### b. Para estudiantes más avanzados

- i. Ofrecer desafíos de codificación más independientes donde los estudiantes tengan que encontrar soluciones por sí solos, como construir un proyecto completo sin instrucciones paso a paso.
- ii. Incluir aprendizaje basado en proyectos, donde los estudiantes diseñan e implementan sus propios programas de principio a fin. Estos proyectos pueden ser complejos, como desarrollar una aplicación web, crear un juego o trabajar con API.
- iii. Permitir el aprendizaje a su propio ritmo, en el que los estudiantes eligen qué temas avanzados explorar y establecen sus propios plazos para completarlos.

## 3. Producto diferenciado

### a. Para principiantes

- i. Permitir que los estudiantes presenten proyectos más simples, como construir una calculadora básica o una aplicación de lista de tareas pendientes.
- ii. Proporcionar plantillas o andamiaje (por ejemplo, funciones prescritas o código parcial) para que los estudiantes puedan centrarse en tareas específicas, como escribir la lógica detrás de una característica o depurar.

iii. Utilice cuestionarios de opción múltiple o preguntas de respuesta corta para evaluar la comprensión de conceptos de programación como bucles, condicionales y tipos de datos.

b. Para estudiantes más avanzados

i. Permitir a los estudiantes diseñar proyectos más complejos, como crear una aplicación web o desarrollar un juego utilizando Unity o Unreal Engine.

ii. Incentive a los estudiantes a mostrar sus habilidades a través de contribuciones de código abierto o mediante la creación de proyectos que resuelvan problemas del mundo real.

iii. Ofrecer la opción a los estudiantes de publicar sus proyectos en plataformas como GitHub o Heroku, permitiéndoles demostrar su trabajo a un público más amplio y recibir comentarios.

4. Entorno diferenciado

a. Usando herramientas que ayudan a los estudiantes a codificar de diferentes maneras:

i. Las plataformas de codificación en línea permiten a los estudiantes trabajar a su propio ritmo, con ejercicios de codificación interactivos que brindan retroalimentación instantánea.

ii. Aprendizaje gamificado, ofrecen a los estudiantes una forma atractiva de aprender programación al convertir la codificación en juegos, lo que puede resultar especialmente atractivo para programadores más jóvenes o principiantes.

iii. Codificación colaborativa: el uso de GitHub u otros sistemas de control de versiones permite a los estudiantes trabajar en proyectos compartidos y realizar un seguimiento del progreso a lo largo del tiempo, con comentarios y colaboración tanto de profesores como de compañeros.

## **A.7 Encuesta sobre el componente de carga cognitiva de CS**

1. Los temas tratados en la actividad fueron muy complejos.

2. La actividad abordó código de programa que percibí como muy complejo.

3. La actividad abordó conceptos y definiciones que percibí como muy complejos.

4. Las instrucciones o explicaciones durante la actividad fueron muy confusas.

5. Las instrucciones o explicaciones fueron, en términos de aprendizaje, muy ineficaces.

6. Las instrucciones o explicaciones estaban llenas de lenguaje confuso.

7. La actividad mejoró significativamente mi comprensión del/de los tema(s) tratado(s).

8. La actividad mejoró significativamente mis conocimientos y comprensión de programación.

9. La actividad mejoró significativamente mi comprensión del código de programa tratado.

10. La actividad mejoró significativamente mi comprensión de los conceptos y definiciones.

Carga Intrínseca: 1, 2 y 3

Carga Extrínseca: 4, 5 y 6

Carga pertinente: 7, 8, 9 y 10