



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**  
**Facultad de Ciencias de la Electrónica**

---

***METODOLOGÍA DE DESARROLLO DE SISTEMA TUTOR  
ENFOCADA EN HABILIDADES ALGORÍTMICAS***

---

**TESIS**

Presentada para obtener el título de

**DOCTOR EN SISTEMAS Y AMBIENTES EDUCATIVOS**

Presenta:

**Guillermina Sánchez Román**

**MIEMBROS DEL COMITÉ TUTORIAL**

Directora: Dra. Josefina Guerrero García

Co-director: Dr. Daniel Mocencahua Mora

**Evaluadores externos:**

Dra. Patricia Balcázar Nava

Dr. Javier Francisco García Orozco

**Lectores:**

Dr. Juan Manuel González Calleros

Dr. Iván Olmos Pineda

Puebla, Puebla. Mayo de 2018

## **AGRADECIMIENTOS**

A mi madre por su ejemplo y cariño, por enseñarme a alcanzar objetivos.  
Q.E. P. D.

A mis hermanos Leticia, Elizabeth, Jose Juan y Rene que son los pilares que me apoyan en cada momento.

A la Dra. Josefina Guerrero García y al Dr. Daniel Mocencahua Mora por guiarme durante este proceso de aprendizaje como asesores para culminar este gran proyecto.

Al Dr. Juan Manuel González Calleros y al Dr. Iván Olmos Pineda por su acompañamiento y aportaciones como expertos en el equipo de investigación.

Para la Dra. Patricia Balcázar Nava y al Dr. Javier Francisco García Orozco por sus valiosos aportes como asesores externos.

Al Dr. Cesar Collazos por compartir su conocimiento durante la estancia en la Universidad del Cauca, Colombia. Además, de permitir la colaboración con otros investigadores para crear redes de investigación.

A la Dra. Yadira Navarro por su acompañamiento y gestiones como Coordinadora del Doctorado en Sistemas y Ambientes Educativos.

A los miembros núcleo académico del DSAE de las tres instituciones: BUAP, ITSON y UV por contribuir en mi formación.

A la Facultad de Ciencias de la Computación por su apoyo a través de gestiones y financiamiento para permitir mi superación académica durante el doctorado.

A las instituciones que hicieron posible mi formación doctoral otorgando recursos: Benemérita Universidad de Puebla (BUAP) y el Consejo Nacional de Ciencia y Tecnología (CONACYT).

# Índice

1 Introducción .....	8
1.1 Resumen.....	8
1.2 Protocolo .....	10
1.2.1 Justificación .....	10
1.2.2 Problema de investigación .....	11
1.2.3 Objetivo general.....	13
1.2.4 Objetivos específicos.....	13
1.2.5 Alcance del proyecto .....	13
1.2.6 Plan de tesis.....	14
2 Estado del arte .....	17
2.1 Revisión de la literatura .....	18
2.1.1 Los sistemas Inteligentes.....	18
2.1.2 Avances en las teorías del conocimiento para los ST .....	19
2.1.3 Teoría ACT-R.....	20
2.1.4 Sistemas Tutores Cognitivos basados en reglas de producción.....	21
2.1.5 Sistemas Tutores tipo Example-Tracing .....	22
2.1.6 Sistemas Tutores basados en restricciones.....	23
2.1.7 Herramientas de Autor.....	24
2.1.8 Sistemas Tutores basados en Internet .....	27
2.1.9 Análisis comparativo .....	28
2.2. Articulación Teórico-Conceptual.....	32
2.2.1 Limitantes encontradas en la literatura .....	35
2.2.2 Requerimiento ontológico .....	36
2.2.3 Requerimientos metodológicos .....	36
3. Marco Teórico-Conceptual.....	41
3.1 Principios pedagógicos mediados por la tecnología en nivel educativo superior. ....	43
3.3 Teorías del Aprendizaje.....	47

3.4 Estilos de Aprendizaje .....	49
3.4.1 Modelo de Felder-Silverman.....	52
3.5 Estrategias de Aprendizaje.....	54
3.5.1 Teoría de la Carga Cognitiva.....	54
3.5.2 Resolución de problemas .....	55
3.5.3 Ejemplos y problemas por completar .....	56
3.5.4 Técnicas de Gamificación .....	57
3.6 Habilidades algorítmicas .....	58
3.7 Sistema Tutor Inteligente .....	61
3.7.1 Módulos de un Sistema Tutor Inteligente.....	63
3.8 Dissertación .....	66
3.9 Conclusiones.....	67
4. Metodología de diseño de Sistema Tutor para apoyar habilidades Algorítmicas .....	69
4.1 Detección del perfil de estudiante.....	73
4.2 Definir la estrategia didáctica.....	77
4.3 Gestión y adaptación de contenidos.....	80
4.4 Descripción del conocimiento y contenidos .....	81
4.5 Actualizar el perfil de estudiante .....	87
5. Diseño Metodológico .....	90
5.1 Método.....	90
5.2 IBD Fase 1 .....	93
5.2.1 Análisis de la situación (A).....	93
5.3 IBD Fase 2 .....	95
5.3.1 Análisis funcional (B) .....	96
5.3.2 Análisis funcional (C) .....	100
5.4 IBD Fase 3 .....	110
5.4.1 Diseño y validación (D) .....	110
5.4.2 Diseño y validación (E) .....	113
5.4.3 Diseño y validación (F).....	115
6. Resultados .....	119
6.1 Validación de expertos .....	119
6.2 Validación de estudiantes .....	122

6.3. Discusión de resultados.....	125
7. Conclusiones.....	148
7.1. Trabajo futuro .....	150
Referencias.....	152
Anexo A: Tabla de operacionalización de habilidades algorítmicas .....	161
Anexo B: Cuestionario de validación para el instrumento de Habilidades algorítmicas .....	162
Anexo C: Tabla de artículos del Estado del Arte .....	166
Anexo D: Instrumento de estilos de aprendizaje de Felder y Silverman .....	167
Anexo E: Instrumento de Habilidades Algorítmicas.....	172
Anexo F: Cuestionario de validación de la Metodología de ST .....	187

# Índice de figuras

FIGURA 1: MAPA DE LECTURA DEL TRABAJO DE INVESTIGACIÓN.....	16
FIGURA 2: MAPA DE LECTURA DEL ESTADO DEL ARTE.....	17
FIGURA 3. RELACIÓN DE CONCEPTOS TEÓRICOS SOBRE LA METODOLOGÍA DE ST.....	34
FIGURA 4. MODELO DE INSTRUCCIÓN.....	37
<i>FIGURA 5. RELACIÓN DE CONCEPTOS TEÓRICOS DE LA INVESTIGACIÓN.....</i>	41
<i>FIGURA 6. ESTRUCTURA DE LA ARQUITECTURA COGNITIVA HUMANA (ACH). .....</i>	55
<i>FIGURA 7. TAREAS PROPUESTAS EN LOS MODELOS DE UN STI. ....</i>	65
FIGURA 8: ESCHEMA DE TAREAS PARA LOS ACTORES.....	70
FIGURA 9: PASOS DE LA METODOLOGÍA DE DESARROLLO DE SISTEMA TUTOR EN APOYO DE LAS HABILIDADES ALGORÍTMICAS.....	72
FIGURA 10: ELEMENTOS DEL MODELO DE ESTUDIANTE.....	74
FIGURA 11 : MODELO DE ESTUDIANTE DE ACUERDO AL ESTILO DE APRENDIZAJE Y NIVEL DE CONOCIMIENTO..	76
FIGURA 12. MODELO GENERAL DE ESTUDIANTE. ....	77
FIGURA 13: DISEÑO DE UNA ACTIVIDAD A PARTIR DE LAS TEORÍAS. ....	80
<i>FIGURA 14: ESTRUCTURA TEMÁTICA DE ACUERDO A LA MATERIA DE METODOLOGÍA DE LA PROGRAMACIÓN..</i>	83
FIGURA 15: MODELO DE DOMINIO SOBRE LA EVALUACIÓN DE TEST DE PENSAMIENTO COMPUTACIONAL. ....	86
FIGURA 16: PROTOTIPOS DE INTERFAZ DE USUARIO PARA LAS ACTIVIDADES. ....	89
FIGURA 17. PASOS PARA LA INVESTIGACIÓN BASADA EN EL DISEÑO (IBD). ....	91
FIGURA 18. PROCESO DE LA INVESTIGACIÓN GENERAL. ....	92
FIGURA 19 . ANÁLISIS DE LA SITUACIÓN. ....	93
FIGURA 20. ANÁLISIS FUNCIONAL, SUBFASE B.....	96
FIGURA 21. ANÁLISIS FUNCIONAL, SUBFASE C.....	100
<i>FIGURA 22. MAPA GENERAL DE UNIDADES DE ANÁLISIS EN LAS CATEGORÍAS.....</i>	109
FIGURA 23. DISEÑO DE LA METODOLOGÍA, SUBFASE D. ....	110
FIGURA 24. VALIDACIÓN DE EXPERTOS SOBRE LA METODOLOGÍA, SUBFASE E. ....	113
FIGURA 25. VALIDACIÓN DE LA METODOLOGÍA CON ESTUDIANTES, SUBFASE F.....	115

# Índice de gráficas

GRÁFICA 1. PORCENTAJE DEL ÍNDICE DE REPROBACIÓN DE 2009-2013. FUENTE: DAE (BUAP).....	12
<i>GRÁFICA 2. ESTILOS DE APRENDIZAJE DE ESTUDIANTES DE NUEVO INGRESO. ....</i>	99
<i>GRÁFICA 3. NIVEL DE HABILIDADES ALGORÍTMICAS. ....</i>	100
<i>GRÁFICA 4. RESULTADOS DE LA APLICACIÓN DE LA ESTRATEGIA (PRETEST Y POSTEST). ....</i>	118
<i>GRÁFICA 5. EVALUACIÓN DE LA CATEGORÍA DE MODELO DE ESTUDIANTE. ....</i>	119
<i>GRÁFICA 6. EVALUACIÓN DE LA CATEGORÍA DE MODELO DE DOMINIO.....</i>	120
<i>GRÁFICA 7. EVALUACIÓN DE LA CATEGORÍA DE MODELO TUTOR. ....</i>	121
<i>GRÁFICA 8. EVALUACIÓN DE LA CATEGORÍA GENERAL. ....</i>	121
<i>GRÁFICA 9: RESULTADOS DEL PRETEST Y POSTEST DE LAS HABILIDADES DEL ESTUDIANTE 1 .....</i>	122
<i>GRÁFICA 10. RESULTADOS DEL PRETEST Y POSTEST DE HABILIDADES DEL ESTUDIANTE 2. ....</i>	123
<i>GRÁFICA 11 . RESULTADOS DEL PRETEST Y POSTEST DE HABILIDADES DEL ESTUDIANTE 3.....</i>	123
<i>GRÁFICA 12 . RESULTADOS DEL PRETEST Y POSTEST DE HABILIDADES DEL ESTUDIANTE 4.....</i>	124
<i>GRÁFICA 13 . RESULTADOS DEL PRETEST Y POSTEST DE HABILIDADES DEL ESTUDIANTE 5.....</i>	124

## Índice de tablas

TABLA 1: CATEGORÍAS DE ANÁLISIS EN LA LITERATURA SOBRE SISTEMAS TUTORES.	31
TABLA 2: <i>HABILIDADES ALGORÍTMICAS CLASIFICADAS DE ACUERDO A LA TAXONOMÍA DE BLOOM.</i>	39
TABLA 3: <i>COMPARACIÓN DE ESTILOS DE APRENDIZAJE</i>	51
TABLA 4: <i>CATEGORIZACIÓN DE LAS HABILIDADES ALGORÍTMICAS.</i>	60
TABLA 5: HABILIDADES Y TIPOS DE ESTUDIANTES CON BASE A LA TAXONOMÍA DE BLOOM	85
TABLA 6: <i>VALIDACIONES DE LOS INSTRUMENTOS</i>	95
TABLA 7: <i>CATEGORIZACIÓN Y CODIFICACIÓN DE LAS ENTREVISTA</i>	104
TABLA 8: <i>RESULTADOS DE EVALUACIÓN DE USABILIDAD DE PROTOTIPOS DE IU</i>	112
TABLA 9: <i>MODELO DE ESTUDIANTE</i>	126
TABLA 10: <i>MODELO DE TUTOR</i>	129
TABLA 11: <i>MODELO DE DOMINIO</i>	131
TABLA 12: <i>MODELO DE INTERFAZ DE USUARIO</i>	133
TABLA 13: <i>VALORACIÓN GENERAL</i>	135
TABLA 14: <i>EXPERIENCIA DURANTE LA ESTRATEGIA DIDÁCTICA</i>	139

# **1 Introducción**

En este apartado se abordan los aspectos relacionados con la problemática, la justificación que lleva a definir los objetivos de la investigación, y todos aquellos aspectos que se consideraron para el desarrollo de esta investigación.

## **1.1 Resumen**

Desde hace varias décadas se ha notado una deficiencia en el aprendizaje de programación (proceso de diseñar, codificar, depurar y mantener código fuentes de programas de computadora), sobre todo en el desarrollo de habilidades algorítmicas. Es importante observar la forma en que el estudiante aprende de forma autónoma para mejorar las carencias identificadas. Este trabajo de investigación propone una metodología de desarrollo de Sistema Tutor (ST) enfocada a las habilidades algorítmicas, en la cual se determina el conocimiento previo del estudiante y lo acompaña en su progreso a través de estrategias de aprendizaje lúdicas que le motiven y ayuden a desarrollar sus habilidades de pensamiento algorítmico (Aparicio, Gutierrez, González, y Isla, 2012; Orejuela, García, Hurtado, y Collazos, 2013). El estudiante es foco de atención y se necesita saber qué conocimientos tiene y cómo aprende; para esto se utiliza la clasificación de estilo de aprendizaje (Felder y Silverman, 1988) en los estudiantes de la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla. Los datos obtenidos de la caracterización de los estudiantes se integrarán en el modelo de estudiante para el ST y darán pauta al diseño de las actividades de aprendizaje acorde a la preferencia en su estilo de aprendizaje.

La importancia de la integración de la tecnología en la educación se incrementa cada vez más con las diferentes propuestas de plataformas y software educativos que se desarrollan para los diferentes niveles y áreas educativas. El área de ciencias se ha caracterizado por tener poca demanda de estudiantes y mayor dificultad en el aprendizaje, en particular en el área de Ciencias de la Computación los estudiantes tienen dificultades en los cursos iniciales de programación y esto puede dar lugar a altas tasas de reprobación. (Moroni y Señas, 2005; Rosanigo y Paur, 2006). Particularmente, en la clase de metodología de la programación que cursan estudiantes de nuevo ingreso en la Facultad de Ciencias de la Computación (FCC) de la Benemérita Universidad Autónoma de Puebla (BUAP) se identificó un alto porcentaje de reprobación mayor al 50%, lo cual genera el interés para mejorar el nivel de rendimiento académico en esta área.

En este trabajo de investigación se propone identificar las tendencias en el estilo de aprendizaje de los estudiantes de Computación de la FCC, con el fin de apoyarles de manera personalizada, ya que esto les da confianza, permite un acompañamiento y sobre todo les da los contenidos de acuerdo a su avance. Los ST permiten ajustar su funcionamiento a las metas, tareas, intereses y otras características de los usuarios. Estos sistemas posibilitan la emulación de un tutor humano para determinar qué enseñar, cómo enseñar y a quién enseñar (Zulma Cataldi y Lage, 2009). Las dos dimensiones: estilo de aprendizaje y conocimiento previo en habilidades algorítmicas, permitirán caracterizar al estudiante en un ST para dar seguimiento a través de un aprendizaje autodirigido y personalizado.

**Palabras Clave:** Estrategia de aprendizaje, Habilidades algorítmicas, Inteligencia Artificial, Sistemas Tutores, Modulo de Aprendizaje, Juegos Educativos.

**Keywords:** Learning Strategy, Alghoritm Skills, Artificial Intelligence, Intelligent Tutoring Systems, Learning Module, Educational Games.

## 1.2 Protocolo

### 1.2.1 Justificación

En las habilidades de pensamiento de orden superior se incluye la destreza para solucionar problemas; por esta razón, se requiere seleccionar estrategias efectivas para ayudar a que los estudiantes las desarrolle. Para atender esta necesidad, la programación de computadoras constituye una buena alternativa ya que posibilita no solo activar una amplia variedad de estilos de aprendizaje sino desarrollar el pensamiento algorítmico. Otra habilidad de pensamiento que también se puede ayudar a desarrollar con un curso de Algoritmos y Programación: la creatividad. Cada vez es mayor el número de empresas que fundamentan su modelo de negocio en la creatividad y la innovación.

Estudios realizados muestran que las habilidades algorítmicas para crear programas básicos o algoritmos de los estudiantes universitarios que ingresan a carreras del área de Ciencias de la Computación son todavía deficientes. Además, existe ausencia del dominio o nivel de expertise en aspectos cognitivos e instrumentales relacionados con éstas habilidades, para su enseñanza y evaluación. Por supuesto sus habilidades de pensamiento algorítmico no están exentas (Moroni y Señas, 2005; Robins, Rountree, y Rountree, 2003; Rosanigo y Paur, 2006).

La metodología propuesta en este trabajo integra elementos que definen las características de los estudiantes y sugiere la forma en que se debe de enseñar el área a fin de particularizar la enseñanza-aprendizaje del contenido de dominio. La investigación abona al área educativa puntualmente al área de instrucción en el aprendizaje de las habilidades básicas de programación. Existen diversas investigaciones que abordan el aprendizaje de la programación; sin embargo es importante abonar a desarrollar las habilidades básicas para crear un algoritmo para resolver un problema. De tal manera, que los estudiantes novatos adquieran las habilidades necesarias a través de estrategias guiadas y personalizadas a través de la tecnología, en este caso de Sistemas Tutores.

Los estudiantes de nuevo ingreso a nivel superior del área de Computación son los principales beneficiados debido a que se propone una instrucción diseñada para el área de desarrollo de habilidades algorítmicas. Al desarrollar las habilidades algorítmicas, tendremos la certeza de que podrán resolver problemas más complejo tratando de crear nuevas estrategias posteriormente cuando conozcan algún lenguaje de programación.

De acuerdo a las líneas de investigación de Doctorado en Sistemas y Ambientes Educativos (DSAE), se aporta a la línea de redes de conocimiento y aprendizaje; dada la orientación y objetivo de la investigación se propuso una estrategia didáctica en apoyo a las habilidades algorítmicas. Debido a que se analiza cómo los estudiantes van adquiriendo y generando el conocimiento sobre la solución de problemas a través de los algoritmos, de tal forma que se monitoree el conocimiento que tiene el estudiante en el área de algoritmos básicos y que se categoricen el nivel de sus habilidades algorítmicas, y finalmente se determina cómo los estudiantes gestionan su conocimiento para resolver nuevos problemas.

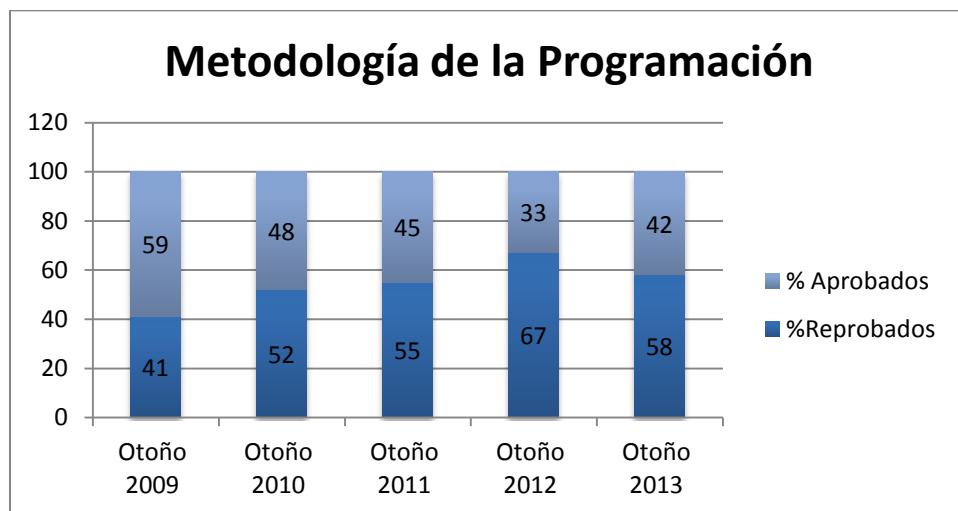
En particular la investigación abona a tener un panorama de cómo se puede generar un Sistema Tutor y las tendencias a partir de las metodologías existentes, además ver desde perspectivas psicológicas, pedagógicas y tecnológicas, qué componentes pueden estructurarlo y fundamentarlo. En esta investigación se realizó el análisis cualitativo que permite definir la necesidad de hacer este tipo de metodologías para integrar en un ST. En la literatura se han abordado desde puntos enfoque cuantitativos, si este estudio se aborda desde el enfoque cualitativo se espera encontrar hallazgos interesantes a partir de la evaluación de expertos, lo cual enriquece la investigación.

### *1.2.2 Problema de investigación*

Desde hace años cientos de estudiantes de diversos estados del país, y de diferentes subsistemas educativos ingresan a la Facultad de Ciencias de la Computación (FCC) de la Benemérita Universidad Autónoma de Puebla (BUAP). Habitualmente los grupos se conforman de 45 alumnos y se asignan a un

profesor, el asesorar a cada estudiante se hace una tarea compleja aunada al hecho de que el profesor debe impartir otros cursos. Si el estudiante se queda con dudas le puede generar un nivel de frustración y probablemente un desinterés progresivo. Además, se ha observado que los estudiantes no logran identificar estructuras básicas de selección y de repetición, en programación, incluso al analizar el problema muchas veces no identifican variables importantes del mismo.

En la Gráfica 1 se muestra el porcentaje de reprobación en la materia de Metodología de la programación, los datos se obtienen de la Dirección de Administración Escolar (DAE) gestionado por la Secretaría Académica de la FCC, donde se observa la tendencia en aumento del porcentaje de estudiantes reprobados entre los años 2009-2013. Por lo tanto, se considera un aspecto urgente a atender y entender cuáles son los elementos que se involucran para mejorar el desempeño de las habilidades algorítmicas que el estudiante de nuevo ingreso tiene para la resolución de problemas.



Gráfica 1. Porcentaje del índice de reprobación de 2009-2013. Fuente: DAE (BUAP).

Pregunta de investigación:

- ❖ ¿Cuáles son los elementos para definir la metodología de desarrollo de Sistema Tutor (ST) con base a los referentes teóricos, que apoyen al desarrollo de habilidades algorítmicas?

### *1.2.3 Objetivo general*

- ❖ Diseñar una metodología de Sistema Tutor enfocada en las habilidades algorítmicas, orientada a estudiantes de nuevo ingreso a la Facultad de Ciencias de la Computación.

### *1.2.4 Objetivos específicos*

- ❖ Definir los conocimientos y habilidades algorítmicas a desarrollar con base en la taxonomía de Bloom.
- ❖ Caracterizar a los estudiantes de nuevo ingreso a computación de la FCC, BUAP acuerdo a su nivel de: conocimiento previo en habilidades algorítmicas y estilo de aprendizaje.
- ❖ Diseñar la estrategia didáctica y de retroalimentación para desarrollar habilidades algorítmicas.
- ❖ Diseñar una metodología de Sistema tutor que cumple con las especificaciones identificadas en la literatura y valoración de expertos.

### *1.2.5 Alcance del proyecto*

Desde el punto de vista educativo, la programación de computadoras compromete a los estudiantes en la consideración de varios aspectos importantes para la solución de problemas: decidir sobre la naturaleza del problema, seleccionar una representación que ayude a resolverlo, así como monitorear sus propios pensamientos (metacognición) y estrategias de solución. En este proyecto de investigación se trata de reconocer cuáles son las habilidades necesarias para resolver problemas a través de algoritmos para que los estudiantes de Computación de nuevo ingreso puedan ser capaces de desarrollar programas de computadora en cursos posteriores. Además, se necesita identificar un tipo de instrucción que apoye el desarrollo en las habilidades de algoritmia incorporándola al diseño de un Sistema educativo apoyado por la tecnología que les dé seguimiento de manera particular.

Se identifica un déficit en el proceso de resolución de problemas en estudiantes de nuevo ingreso en Computación. Se han integrado algunos tutores

para ayudar a programar, sin embargo se considera que es importante verificar las habilidades de algoritmia que tiene el estudiante y desarrollarlas hasta que tenga la capacidad de plasmar los problemas como algoritmos. Actualmente hay entornos que ayudan a programar normalmente en algún lenguaje de programación, sin embargo es importante reconocer el estado cognitivo respecto a las habilidades algorítmicas que tienen los estudiantes.

Existen métodos para aprender a programar a través de objetos de aprendizaje, lenguajes de programación, juegos serios, entre otros. En particular se propone trabajar con Sistemas Tutores donde se integre un tipo de instrucción que favorezca al desarrollo de las habilidades algorítmicas, y que soporten el grado de seguimiento del estudiante a partir de que el sistema aprenda cómo se comporta el estudiante.

Es necesario saber cómo el estudiante genera el proceso cognitivo para la resolución del problema y proponer el diseño pedagógico instruccional para que se pueda integrar a la metodología para el desarrollo de Sistema Tutor (ST) para desarrollar habilidades algorítmicas.

En este trabajo se dará la metodología para diseñar un Sistema Tutor que apoye las habilidades algorítmicas, donde se definan las características del estudiante con base al estilo de aprendizaje y las habilidades algorítmicas. La metodología que se describe en el capítulo 4 integra las bases teóricas de la estrategia didáctica para resolver problemas algorítmicos y medir las habilidades en la taxonomía de Bloom. Existe otra dimensión que es la parte tecnológica la cual queda fuera del alcance de este trabajo como siguiente líneaFigura 1 de investigación donde se propondrán los elementos de implementación como lo recomendaron los expertos.

#### *1.2.6 Plan de tesis*

La presente tesis tiene la estructura de la Figura 1 en el Capítulo uno se integra una introducción al trabajo propuesto, objetivo, supuestos, límites y aportes de la investigación, en el Capítulo dos se describen los trabajos relacionados, las

metodologías de diseño de Sistemas Tutores, el análisis de propuestas metodológicas que se enfocan a área de ciencias y en particular de Computación, se integra una articulación teórica-conceptual sobre las metodologías, las teorías instruccionales y los Sistemas Tutores de acuerdo al objeto de estudio. En el Capítulo tres, se plantean las teorías que fundamentan y articulan la metodología de diseño de ST propuesta como las habilidades algorítmicas, ejemplos resueltos y gamificación, entre otras que apoyan la instrucción a los estudiantes de nuevo ingreso a la carrera de computación. En el Capítulo cuatro se describe la metodología propuesta como resultado del proceso de investigación. En el Capítulo cinco se incluyen la metodología de investigación de tipo cualitativo que se aplicó, el proceso de recolección de datos y las etapas de la investigación, el procesamiento de datos a través de MaxQda y el análisis de los resultados, en el Capítulo 6 se realiza la discusión de los resultados obtenidos contrastados con la teoría. En el Capítulo 7 se describen los hallazgos encontrados, los trabajos futuros y el logro de objetivos.

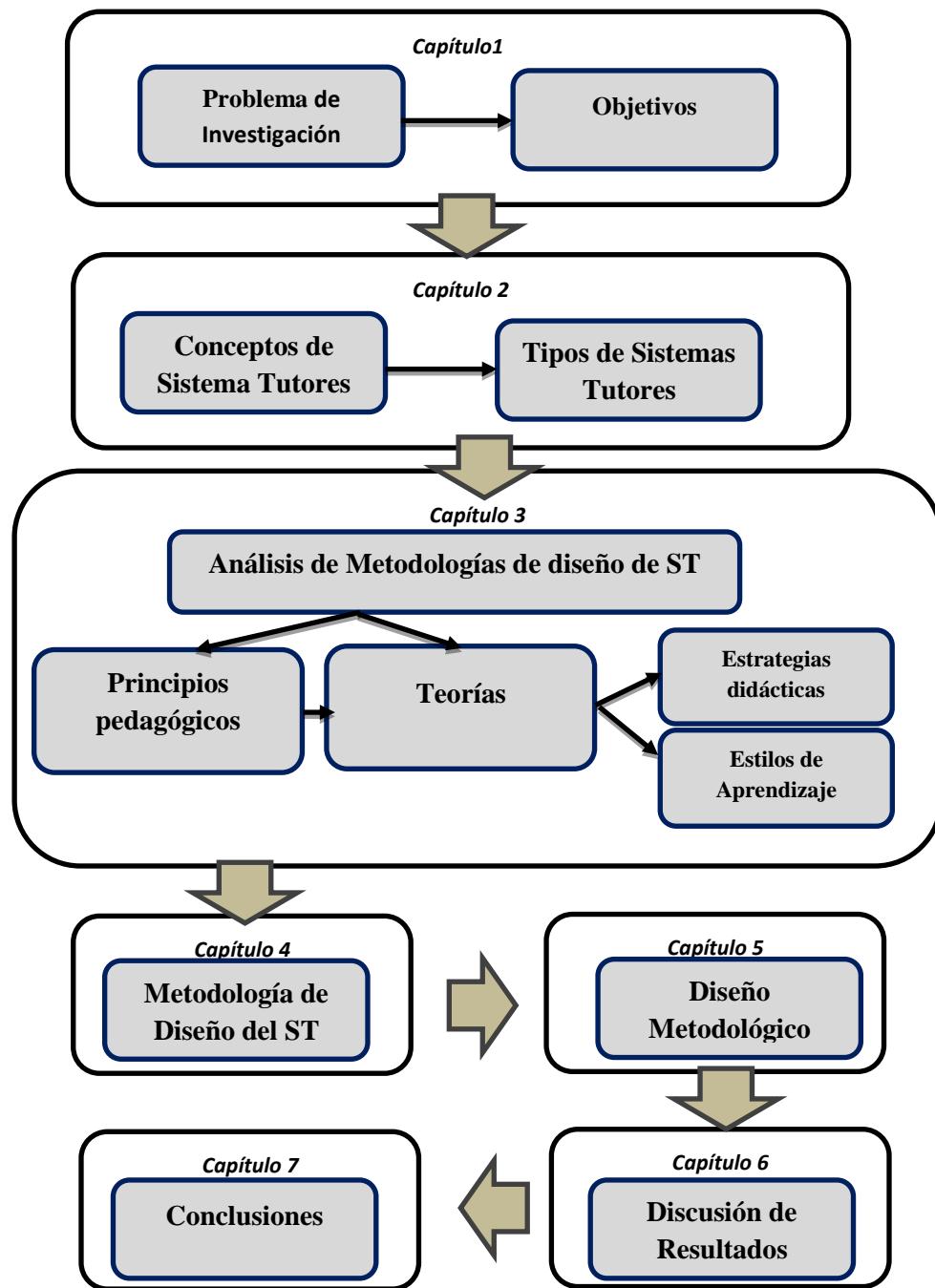


Figura 1: Mapa de lectura del trabajo de investigación.

Fuente: Elaboración propia.

## 2 Estado del arte

Debido a que las Tecnologías de la Información y Comunicación han estado acompañando al docente en su práctica educativa a través de medios electrónicos, aplicaciones multimedia, plataformas y sistemas de aprendizaje es importante que estos avances se incluyan en el acompañamiento del estudiante dentro y fuera del aula. Algunas tecnologías que se han desarrollado desde hace más de una década son los Sistemas Tutores (ST), éstos desean modelar y representar un conjunto de principios instruccionales lo suficientemente generales para ofrecer una instrucción efectiva a través de un conjunto de tareas de enseñanza. A continuación se presenta en la Figura 2 el mapa de lectura del estado del arte, con los temas relevantes que fundamentaron los vacíos y la orientación de la investigación.

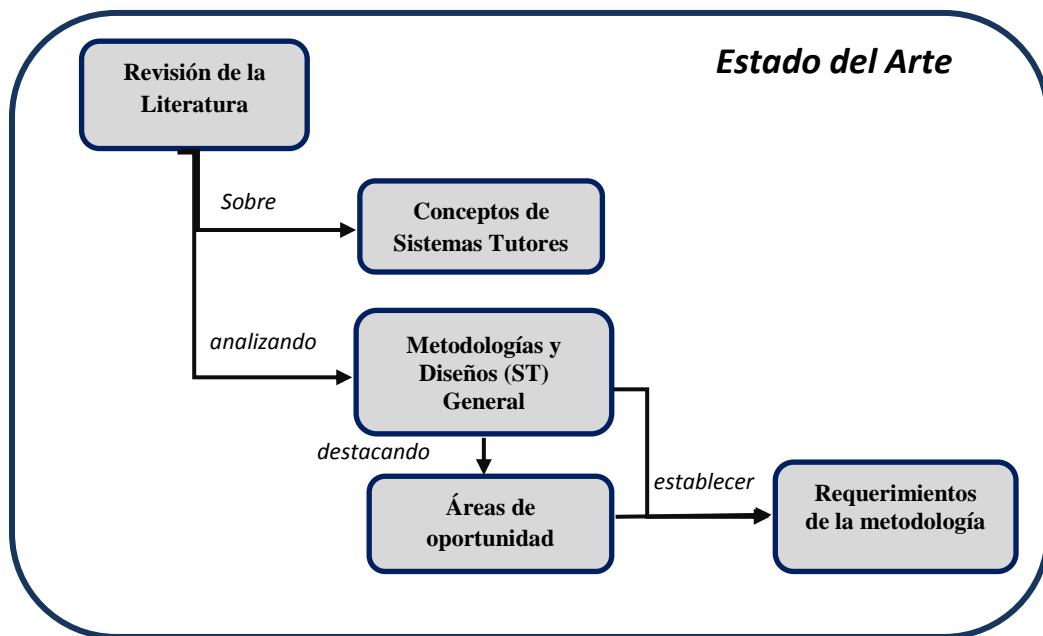


Figura 2: Mapa de lectura del estado del arte.

Fuente: Elaboración propia.

Se inicia con la revisión de la literatura mencionando investigaciones relacionadas con el tema de metodologías de diseño de sistemas tutores aplicados en el área de ciencias y en particular el área de programación en apoyo

al aprendizaje de los estudiantes. Posteriormente, se hace un análisis comparativo de los artículos más relacionados al objeto de estudio y se hace la articulación teórica-conceptual. Seguidamente, se mencionan las limitantes encontradas en la literatura y los requerimientos metodológicos.

## 2.1 Revisión de la literatura

### 2.1.1 *Los sistemas Inteligentes*

Desde hace años ha existido un interés en el ámbito educativo por tener herramientas de apoyo dentro y fuera del aula. Por lo que las ciencias cognitivas, como la educación, psicología y computación se han conjuntado para dar paso a la solución de esta necesidad la cual ha estado vigente en los últimos cincuenta años. La instrucción asistida por computadora (IAC) tiene sus inicios a finales de los años 50 en universidades americanas; un ejemplo es el proyecto *Plato* de la Universidad de Illinois en EE.UU; con la denominación de CBT (Computer Based Training) fueron conocidos los sistemas de entrenamiento en las pequeñas universidades en los años 70 y paralelamente en el área de la Inteligencia Artificial se comenzaron a construir CBTs que intentaban simular el razonamiento o lógica humana. Según Camacho (2006) fue a finales de los años 70 cuando surge una nueva rama denominada Desarrollo de Sistemas Tutoriales Inteligentes de Instrucción Asistida por Computadora, hasta convertirse en Sistema Tutorial Inteligente.

Los Sistemas Tutores (ST) también llamados Sistemas de Enseñanza Inteligentes (SEIs) denominados así por Laureano y de Arriaga (2001) son definidos a partir de cuatro componentes:

- 1) Modelo de estudiante
- 2) Modelo de tutor
- 3) Modelo de dominio
- 4) Modelo de interfaz de usuario.

Un tutor inteligente, se le denota como: “un sistema de software que utiliza técnicas de inteligencia artificial (IA) para representar el conocimiento e interactúa con los estudiantes para enseñárselo” (VanLehn, 1988, p.58). Wolf (1984) define los ST como: “sistemas que modelan la enseñanza, el aprendizaje, la comunicación y el dominio del conocimiento del especialista y el entendimiento del estudiante sobre ese dominio”. Por su parte, Giraffa, 1997 los define como “un sistema que incorpora técnicas de Inteligencia Artificial (IA) a fin de crear un ambiente que considere los diversos estilos cognitivos de los alumnos que utilizan el programa”.

Los Sistemas Tutores Inteligentes (Wenger 1987; Sleeman y Brown, 1982; Murray 1998; Murray 2003) son herramientas cognitivas; estos sistemas usan resultados de las ciencias cognitivas, teorías instruccionales, técnicas de Inteligencia Artificial y computacionales para desarrollar sistemas que se adaptan al estado de conocimiento del alumno y al estilo de enseñanza del tutor. Estos sistemas han demostrado ser efectivos en la enseñanza uno-a-uno (Murray, 1998; Chi et al., 2001). Actualmente, no son solo herramientas de laboratorio, sino también han ingresado a las aulas y lugares de trabajo. Nkambou, et al (2010) menciona que el reto principal en un Sistema Tutor Inteligente es obtener la información previa del alumno, representar el conocimiento del alumno y seleccionar un plan instruccional para el alumno con el objetivo de mejorar el proceso de aprendizaje de ese alumno.

### *2.1.2 Avances en las teorías del conocimiento para los ST*

La inclusión del tipo de conocimiento en ST se refiere a las diferentes técnicas de tratamiento existentes, de acuerdo con la clasificación del conocimiento: procedimental, declarativo o cualitativo.

En el caso del conocimiento procedimental, que está incluido en la resolución de un problema, hace falta una clase de interpretación que relacione al modelo del estudiante con el conocimiento de resolución del problema, debido a que el intérprete tiene que tomar decisiones basadas en conocimiento local.

Existen dos tipos de conocimiento procedimental: a) ordenado o jerárquico y b) no ordenado.

Cuando se habla de conocimiento ordenado, sabemos que lleva implícito subobjetivos. Esta característica va íntimamente ligada con el tipo de diagnóstico utilizado, por lo tanto, se necesita conocer las condiciones que disparan otro estado del problema, así como el conjunto total de estado y los subobjetivos.

En los años 90, los avances de la psicología cognitiva, las neurociencias y los nuevos paradigmas de programación, han permitido la evolución de los STI desde una propuesta instructiva conductista inicial hacia entornos de descubrimiento y experimentación del nuevo conocimiento (Bruner, 1991; Perkins, 1995, Pozo; 1998) desde la pedagogía de la comprensión (Perkins, 1995, Stone-Wiske, 2007, 2008). Las dificultades de representación se centran en la identificación de los diferentes estadios evolutivos del estudiante y en el reconocimiento de los preconceptos o concepciones erróneas. Así, las teorías ingenuas o intuitivas, se basan en ideas que en general no coinciden con las explicaciones científicas.

### *2.1.3 Teoría ACT-R*

La teoría Control Adaptativo de Pensamiento Racional, en inglés Adaptive Control of Thought-Rational (ACT-R) del aprendizaje y resolución de problemas (Anderson, 1983), trata la manera en que se realiza la adquisición de capacidades y habilidades cognitivas. La teoría sostiene que una habilidad cognitiva consiste, en gran parte, de unidades de conocimiento relacionadas entre sí por los objetivos. La adquisición de habilidades cognitivas implica la formulación de miles de reglas que relacionan los objetivos de una tarea y los estados de la tarea con las acciones y las consecuencias. La teoría utiliza un formalismo basado en reglas para representar el conocimiento orientado a objetivos.

En el trabajo de Sklar y Richards (2006) se indica que el Tutor Cognitivo (TC) (Koedinger y Corbett, 2006), se origina en los departamentos de sicología y ciencias de la Computación de la Universidad de Carnegie Mellon (CMU), a través

de la teoría cognitiva ACT-R (Anderson, 1983). ACT-R es una arquitectura que analiza la cognición en partes mínimas o atómicas que luego integra para modelar los procesos más complejos. El Tutor Cognitivo utilizó el enfoque de resolución de problemas como recurso principal de la enseñanza-aprendizaje de las matemáticas. Los resultados de la utilización del TC mostró un incremento en el rendimiento académico de un 85% (Ritter, John y Anderson, 2007).

#### *2.1.4 Sistemas Tutores Cognitivos basados en reglas de producción*

De acuerdo al trabajo de Sklar y Richards (2006) describen tres tipos de Sistemas Tutores Inteligentes que utilizan técnicas específicas para la creación de cada uno de los componentes que conforman un sistema tutor inteligente. Estos tipos de sistemas tutores han sido probados con suficiencia y han sido exitosos en diferentes dominios de conocimiento (Vanlehn, 2011). Los dos primeros tipos (Model-Tracing y Constraint-Based), corresponden a la categoría de sistemas tutores cognitivos pues ambos están basados en teorías que intentan modelar el entendimiento humano. Los tutores tipo Example-Tracing, corresponden al resultado obtenido tras la simplificación del proceso de construcción de los tutores Model-Tracing.

Los tutores cognitivos tipo Model-tracing son un tipo Sistemas Tutores Inteligentes que se basan en teorías de psicología cognitiva (Anderson, Corbett, Koedinger y Pelletier, 1995; Anderson, Boyle, Corbell y Lewis 1990) en particular la teoría ACT-R (Anderson, 1983). El desarrollo de un Tutor Model-tracing implica la creación de un modelo cognitivo que representa la forma cómo un alumno aborda la resolución de un problema. Este modelo cognitivo se realiza escribiendo las reglas de producción que caracterizan la variedad de estrategias y conceptos erróneos que los estudiantes suelen adquirir. Las reglas de producción se escriben en forma modular de manera que se puedan aplicar a un objetivo en un determinado contexto independiente de lo que lleve al cumplimiento de ese objetivo.

La técnica de Model Tracing es usada para monitorear el progreso del estudiante a través de la solución del problema (Anderson, Boyle, Corbell y Lewis 1990). Este seguimiento es realizando por el tutor, de manera silenciosa, tratando de emparejar las acciones del estudiante con aquellas que pueden ser generadas por el modelo. Cuando el estudiante necesita ayuda, el tutor conoce la ubicación del estudiante dentro del problema y puede proveer mensajes de ayuda, que son individuales para cada estudiante según la manera en que el estudiante se encuentre abordando el problema.

### *2.1.5 Sistemas Tutores tipo Example-Tracing*

Un tutor tipo Example Tracing, o también llamado Pseudo-Tutor, es un sistema educativo que emula comportamiento de un Tutor Inteligente, pero lo hace sin usar código de programación de técnicas de Inteligencia Artificial. (Sería más exacto, aunque más confuso, llamar a estos tutores "Pseudo Tutores Inteligentes" para enfatizar que es la falta de un motor interno de Inteligencia Artificial que los hace "pseudo" y no la falta de un comportamiento inteligente) (Koedinger, Aleven, Heffernan, McLaren, y Hockenberry, 2004).

Dos características clave existentes en los tutores cognitivos así como en otros tipos de Sistemas Tutores Inteligentes, son:

- 1) Ayudar a los estudiantes en la construcción de conocimiento a través de la realimentación y la instrucción en el contexto de la actividad que se está realizando.
- 2) Proveer a los estudiantes la flexibilidad necesaria para explorar diferentes estrategias que representen alternativas de solución a un problema, lo cual les permitirá aprender haciendo (Learning by Doing).

Los tutores cognitivos pueden proporcionar estas características, pero con algunas limitaciones, sin embargo, hay una ganancia en términos del tiempo requerido para realizar el desarrollo de estos sistemas. En particular, se han añadido una serie de técnicas que otorgan flexibilidad para comparar el

comportamiento del estudiante contra los ejemplos de referencia, de modo que un tutor puede reconocer una amplia gama de comportamientos correctos y no sólo los ejemplos demostrados como tal (Aleven, McLaren, Sewall, y Koedinger, 2009).

En última instancia, los tutores cognitivos son más flexibles y más robustos. Aun así, esta flexibilidad no siempre es necesaria para todos los entornos y para todos los contextos, por lo cual se pone en consideración los costos de procesamiento en que se incurren cuando se crea un modelo cognitivo basado en reglas, el cual es alto.

Como es típico de los Sistemas Tutores Inteligentes, los tutores tipo Example Tracing son construidos de tal forma que puedan proporcionar una interfaz que hace el pensamiento visible (Anderson et al., 1995), en el sentido en que el diseño visual de la interfaz gráfica del tutor hace visible los pasos intermedios en la solución de un problema.

#### *2.1.6 Sistemas Tutores basados en restricciones*

Los sistemas tutores inteligentes basados en restricciones, son un tipo de tutores cognitivos cuyas raíces teóricas están basadas en una teoría psicológica del aprendizaje que fue propuesta por Stellan Ohlsson en 1992. Esta teoría, de la misma forma que la teoría ACT-R, reconoce la existencia de un conocimiento procedimental y uno declarativo. Se reconoce que el conocimiento procedimental se refuerza con la práctica y que resulta de una transformación del conocimiento declarativo que ocurre cuando el estudiante se enfrenta a un problema. Esta teoría adicionalmente reconoce que el conocimiento procedimental está relacionado con la ejecución de acciones mientras que el conocimiento declarativo cumple una función evaluativa, permitiendo al estudiante evaluar las consecuencias de sus acciones.

El proceso de aprendizaje se describe en dos etapas: una de reconocimiento del error y otra de corrección del error. En la primera etapa el estudiante detecta por él mismo que una acción fue incorrecta. Este es el caso

cuando el conocimiento declarativo que posee el estudiante es correcto y le permite evaluar correctamente las consecuencias de su acción incorrecta. Cuando el conocimiento declarativo del estudiante es incorrecto, el estudiante no puede evaluar sus acciones o su evaluación es incorrecta y por lo tanto se requiere de la intervención de un tercero, el tutor. El tutor, que puede ser humano o artificial, permite que el estudiante note su error y adicionalmente provee ayuda para corregir el error.

La teoría de Ohlsson establece que el conocimiento declarativo se puede representar en forma de restricciones que modelan principios fundamentales del dominio de conocimiento que no pueden ser violadas. Esto corresponde al principio básico de los tutores basados en restricciones: Todas las soluciones correctas, independientemente del problema, comparten la característica de no violar ningún principio fundamental del dominio.

Mitrovic, Koedinger y Martin (2003) se realizaron un análisis comparativo entre las técnicas de modelamiento tipo Model-Tracing (MT) y los tutores tipo Model Constraint-Based (CMB). Los autores integran una tabla con las propiedades representativas de ambas técnicas y determinan “el aprendizaje desde la ejecución de errores”, ambos modelos son basados en la distinción entre el conocimiento declarativo y procedural, de tal forma que el conocimiento se da en dos fases, la primera en la codificación del conocimiento declarativo, y la segunda fase, es donde el conocimiento declarativo es pasado a un conocimiento más eficiente como el procedural. La diferencia entre las teorías es el esfuerzo aplicado en cada fase, y también el enfoque de instrucción de base en cada teoría, lo cual nos da una orientación del modelo que se puede utilizar de acuerdo al propósito de la investigación.

### *2.1.7 Herramientas de Autor*

Durante la última década se han planteado bastantes mecanismos acerca de cómo realizar y hacer más fácil la implementación de los Sistemas Tutores Inteligentes. Entre las técnicas más importantes se pueden mencionar:

- El uso de Patrones de Diseño (Harrer, Pinkwart, McLaren, y Scheuer, Devedzic y Harrer, 2005).
- La creación y uso de componentes reutilizables y objetos de aprendizaje (Koedinger, Suthers, y Forbus, 1999; Ritter y Koedinger, 1997; Ritter, Blessing, y Wheeler, 2003).
- La creación en grupo usando herramientas colaborativas. (Aleahmad, Aleven, y Kraut, 2008).
- El uso de herramientas de Autor (por ejemplo, Murray, 2003)).

Normalmente, cada herramienta de edición y creación de STI se centra en un tipo específico, por ejemplo, algunas herramientas se centran en prestar facilidades para el diseño de tutores basados en restricciones (Mitrovic et al., 2006) mientras que otras permiten el diseño de Tutores tipo “Model-Tracing” (Blessing, Gilbert, Ourada, y Ritter, 2007). En general el objetivo de las herramientas de autor es reducir significativamente la cantidad de tiempo necesario para crear un tutor, y disminuir la cantidad de conocimientos requeridos en temáticas no relacionadas con el dominio de interés como por ejemplo la programación de procesos soportados con técnicas de inteligencia artificial.

A continuación se describen dos herramientas de autor que se encuentran disponibles para la construcción de los tres tipos de sistemas tutores abordados en este análisis comparativo, una de estas, CTAT (Cognitive Tutor Authoring Tools), es una herramienta de autor que permite la creación de tutores tipo Model-Tracing y Example-Tracing; mientras que ASPIRE, es una herramienta de autor que permite la creación de sistemas tutores inteligentes basados en restricciones.

CTAT es una herramienta de autor que ha sido desarrollada durante los últimos 15 años por el Human-Computer Interaction Group de la Universidad Carnegie Mellon (Aleven et al., 2009; Koedinger et al., 2004). La herramienta está disponible para descarga en su sitio web <http://ctat.pact.cs.cmu.edu>. CTAT soporta la creación de dos tipos de tutores: Tutores tipo Example-Tracing, los cuales

pueden crearse sin necesidad de programación y Tutores Model-Tracing, que requieren la programación de reglas de producción usando un lenguaje de Inteligencia Artificial.

ASPIRE es una herramienta de autor y servidor de despliegue que apoya el proceso de desarrollo de los sistemas tutores inteligentes basados en restricciones, mediante la automatización de algunas tareas y el acceso a herramientas que facilitan la ejecución de tareas que no son automatizables. ASPIRE consta de ASPIRE-Autor, el servidor de autoría, que sirve como ambiente de desarrollo de los sistemas tutores, y ASPIRE-Tutor, el servidor de tutoría, que permite desplegar a los estudiantes los sistemas tutores creados con ASPIRE-Author. ASPIRE-Autor facilita tareas como la definición de los conceptos involucrados en el dominio de interés, las tareas que los estudiantes van a realizar, así como para especificar los problemas y sus soluciones. El proyecto ASPIRE es financiado por el “e-Learning Collaborative Development Fund” de Nueva Zelanda y es desarrollado y mantenido por el “Intelligent Computer Tutoring Group” de la Universidad de Canterbury en Nueva Zelanda. (Mitrovic et al., 2006, 2008, 2009, 2007).

El proceso de creación de un sistema tutor inteligente va más allá que el simple uso de las herramientas de autor. Por ejemplo, una metodología típica para la creación de sistemas tutores tipo Example-Tracing, inicia con la etapa de la definición de los objetivos de aprendizaje y la ejecución de diferentes metodologías que guían otras etapas del proceso de construcción. (Baker, Corbett, Koedinger y Wagner 2004; Koedinger y Corbett, 2006). Por lo tanto, es necesaria una metodología que guíe la construcción de STI independientemente del área de conocimiento, pero es necesaria una herramienta de autor que de la misma forma integre diversidad en modelos de enseñanza-aprendizaje debido a que cada área de conocimiento es particular en su proceso de enseñanza-aprendizaje.

## *2.1.8 Sistemas Tutores basados en Internet*

En el trabajo de Abu-Naser et al. (2011) se define un Sistema Tutor basado en Web, donde se presentan un mayor número de módulos con el objetivo de hacer reusables los componentes. Los componentes de conocimiento experto y conocimiento pedagógico lo manejan como simples bases de datos sin funcionalidad lo que les da mayor flexibilidad en su arquitectura. El JO-Tutor trata de enseñar java (lenguaje de programación) bajo el paradigma orientado a objetos; tienen la característica de crear los problemas de forma ilimitada y no necesariamente cuantitativos.

Debido a que las Tecnologías de la Información y Comunicación han estado acompañando al docente en su práctica educativa a través de medios electrónicos, aplicaciones multimedia, plataformas y sistemas de aprendizaje es importante que el tipo de enseñanza a través de nuevos sistemas acompañen el proceso de enseñanza aprendizaje del estudiante. Desde hace más de una década los Sistemas Tutores Inteligentes (STI) han avanzado, estos desean modelar y representar un conjunto de principios instruccionales lo suficientemente generales para ofrecer una instrucción efectiva a través de un conjunto de tareas de enseñanza (González, 2004). Otra definición que plantea (Ovalle y Jiménez, 2004) es que: "Los STI realizan evaluaciones, detectan errores, dan sugerencias, ejemplos, plantean simulaciones, recomendaciones al estudiante y constantemente re-planean el modelo de aprendizaje, como lo haría un tutor humano".

Los STI han avanzado en diversas áreas de conocimiento desde el apoyo a la medicina, matemáticas, entrenamientos de tareas; incluyendo desde teorías de aprendizaje cognitivista y constructivista (Cataldi, Figueroa, Méndez, Lage, Vigliecca y Kraus, 2006; Peña, Marzo, de la Rosa, y Fabregat, 2002; Zatarain, Barrón, y Reyes, 2011), apoyándose en estrategias como aprendizaje basado en problemas de forma individual hasta aprendizaje colaborativo(Bruno, 2005; Harrer, McLaren, Walker, Bollen y Sewall, 2005; Rau, Aleven, Rummel y Pardos, 2014).

Algunos STI ya publicados en una versión de prueba o con costo son:

- Tabtor que proporciona a los maestros el rendimiento de los estudiantes en función de su capacidad para personalizar la retroalimentación y la instrucción en el aprendizaje en matemática se puede encontrar en <https://www.tabtor.com/>.
- Frontrowed integra prácticas adaptativas, administra el seguimiento de cada estudiante, información para cada estudiante o para grupos pequeños, hace el reporte del seguimiento su sitio en línea es <https://www.frontrowed.com/>.
- Heraclito ayuda a los estudiantes a resolver problemas de pruebas de lógica proposicional, a través de una deducción natural una versión en línea es: <http://obaa.unisinos.br/heraclito/>.
- Khan Academy ofrece ejercicios de práctica, videos instructivos y un panel de aprendizaje personalizado que permite a los alumnos aprender a su propio ritmo, dentro y fuera del salón de clases. En particular es una herramienta para el docente integrar su contenido de su área de dominio (matemáticas, ciencia, programación, historia, economía, entre otras) su sitio en línea es <https://es.khanacademy.org>.
- Knewton se basa en sistemas de tutoría de adaptación, psicometría, y la teoría del aprendizaje cognitivo en varios dominios su versión está en <https://www.knewton.com>.

### 2.1.9 Análisis comparativo

Se revisaron diferentes tipos de fuentes documentales como artículos, tesis y proceedings sobre el desarrollo de los Sistemas Tutores con el fin de determinar cuáles son los avances, tendencias y sobre todo hallazgos y vacíos que hay hasta el momento.

El análisis que se hace es con base a la problemática y deficiencia en habilidades algorítmicas que se ha identificado en los estudiantes de nuevo ingreso a la Facultad de Ciencias de la Computación. También se analizan las

tendencias de los Sistemas Tutores en apoyo a áreas como las matemáticas básicas y programación básica; y en particular se analizan las metodologías y sus aportes.

La búsqueda se vertió en aquellas investigaciones que se enfocaran al área de matemáticas básicas y programación, ya que en ambas se resuelven problemas y es necesario desarrollar habilidades algorítmicas.

Se revisaron más de 40 documentos, de los cuales se tomaron 21, los cuales son en su mayoría artículos y journals que se han publicado en los últimos cinco años.

Algunas de las categorías de análisis que se eligieron son:

- Sujeto de estudio
- Estrategia de instrucción
- Metodología de Investigación
- Aportes o resultado

Dado que existe mayor información como las diversas categorías se generaron subcategorías como teorías pedagógicas o estrategias didácticas, las habilidades que se desarrollarán y los instrumentos que aplicaron. En la categoría de metodología de investigación se derivaron los enfoques cuantitativo, cualitativo y mixto de tal manera que se contemplan como se aplican en las investigaciones de ST y finalmente en la categoría de aportes y resultados, se definen las diferentes etapas en las que se pueden desarrollar los ST desde la metodología, diseños y arquitecturas, y finalmente como se validaron cada uno de estos. A continuación se listan las 4 principales categorías se desglosan en las subcategorías mencionadas:

- Sujeto de estudio
- Estrategia de instrucción
  - Teoría pedagógica
  - Instrumentos para medir el aprendizaje
  - Habilidades a desarrollar en el área de conocimiento
- Metodología de Investigación
- Sujeto de estudio

- Estrategia de instrucción
  - Teoría pedagógica
  - Instrumentos para medir el aprendizaje
  - Habilidades a desarrollar en el área de conocimiento

En la Tabla 1 muestra los resultados de la clasificación de acuerdo a las categorías que se analizaron y cómo éstas dan pauta al análisis descrito posteriormente.

Con base en el objetivo de estudio sobre el desarrollo de las habilidades algorítmicas en los estudiantes de nuevo ingreso, se han observado aproximadamente 14 trabajos en donde los diseños de STI tienden a centrarse en el estudiante en áreas de programación de sistemas, estructuras de datos o aprendizaje de algún lenguaje de programación (Abu Naser, 2009; Kessler, Stein, y Schunn, 2015; Lemus, 2010), sin embargo el apego a un lenguaje hace determinar contenido muy exclusivo para el aprendizaje de programación. Algunos diseños centrados en el modelo de dominio de un Sistema Tutor apoyan a la creación de materiales, con el fin de minimizar los tiempos de creación de estos, incluyendo la integración de Objetos de Aprendizajes (Pires y Cota, 2016). Considerar las habilidades algorítmicas como parte del proceso de inicio en el área de programación permitirá apoyar de forma más precisa a los estudiantes que deseen aprender y desarrollar habilidades para resolver problemas integrando su capacidad de pensamiento algorítmico.

Un 80 % de los documentos revisados realizó investigación de tipo cuantitativa, un 5% cualitativo y un 5% mixto, lo cual indica que la mayoría de los diseños y arquitecturas se prueban de acuerdo a las aplicaciones y validaciones basados en test para evaluar desarrollo de aprendizajes, niveles cognitivos (Kessler et al., 2015; Zatarain, Barrón, y Reyes, 2011). Sin embargo, es importante que las metodologías que se han desarrollado integren investigaciones de tipo cualitativa y mixta, con el fin de ver diferentes perspectivas como la estrategia pedagógica, el diseño de instrucción y el planteamiento de la metodología (Astudillo, Bast y Willging, 2016; González, Mora y Toledo, 2014; Lemus, 2010; Pires y Cota, 2016).

Tabla 1:

*Categorías de análisis en la literatura sobre Sistemas Tutores.*

No	Sujeto estudiante	Programación	Estilo Aprendizaje	Gamificación	STI	Resolución de problemas	Investigación cuantitativa	Investigación Cualitativa	Validación del Sistema	Journal
1	X				X			X		
2	X	X			X	X	X			
3	X		X		X	X	X		X	
4	X	X			X	X	X			X
5					X	X	X			X
6	X		X				X		X	X
7		X	X		X		X			X
8	X	X			X		X		X	
9		X			X		X		X	X
10	X									
11		X					X		X	
12	X	X		X	X	X	X			
13	X	X				X	X			X
14										
15	X			X			X			
16	X	X		X		X	X			
17	X				X		X			
18	X	X			x-	X		X		X
19				X	X		X			X
20		X			X					
21	X			X	X		X			

Fuente: Elaboración propia.

En la categoría de estrategias pedagógicas al menos cuatro documentos se refieren a un estudio cognitivo en base a la resolución de problemas, con el objetivo de disponer de estrategias que acompañen al estudiante en su proceso de aprendizaje (Kessler et al., 2015; Samsonovich, Kitsantas, Brien y Jong, 2015; Sundararaja y Nitta, 2015)

De acuerdo a las categorías encontradas en la subcategoría de gamificación se tienen 7 artículos que han integrado aspectos de estrategias lúdicas y técnicas de gamificación en el diseño de las actividades (Hicks, 2012; Lemus, 2010; Sundararaja y Nitta, 2015), las integraron en su arquitectura y en la interacción de la interfaz de usuario, donde se puede recuperar información de los aciertos y fallos, así como determinar una interfaz amigable y de motivación del estudiante. Aunque hay trabajos que integran la gamificación, solo son un tercio de los trabajos revisados uno de los aspectos que se resaltan es la integración gráfica de los problemas para los estudiantes novatos (González et al., 2014; Hooshyar et al., 2016; Pires y Cota, 2016)

## 2.2. Articulación Teórico-Conceptual

Este trabajo de investigación aborda los términos principales que definen la “Metodología de diseño de un Sistema Tutor enfocada en habilidades algorítmicas”. La forma en que se presentarán son: a) metodología de diseño de sistema tutor, donde los términos secundarios que lo integran son diseño de sistema tutor, modelo de estudiante, modelo de tutor, modelo de dominio y modelo de interfaz; b) estrategias de aprendizaje que integra los subconceptos: técnicas de gamificación y resolución de problemas; y c) perfil de estudiante que integra el estilo de aprendizaje, las habilidades algorítmicas.

Cabe mencionar que debido a que la investigación es aplicada, sólo se profundiza en los términos principales. Para comenzar, se consideran la pregunta y objetivo de la investigación que se describen en la sección 1.2.3, a partir de estos puntos se toman los conceptos más importantes que se muestran en la Figura 3, para definir el marco teórico conceptual donde se visualizan los conceptos principales en los niveles superiores, en este caso el principal es la

metodología, y los conceptos secundarios son las estrategias de enseñanza aprendizaje, el diseño del ST y el perfil de estudiante, quedan en niveles inferiores términos relacionado que conforman los principales.

Los conceptos se estructuran de acuerdo al orden de importancia jerárquicamente con el fin de desarrollar su conceptualización, la interpretación y postura de acuerdo al tema de investigación que se está haciendo.

- Metodología de diseño de Sistema Tutor
- Estrategia de aprendizaje
- Estilos de aprendizaje
- Resolución de problemas
- Habilidades algorítmicas (PC)
- Sistema Tutor (ST)
- Técnicas de gamificación

La metodología, del griego *metà* (más allá), *odòs* (camino) y *logos* (estudio), hace referencia al conjunto de procedimientos basados en principios lógicos, utilizados para alcanzar una gama de objetivos que rigen en una investigación científica o en una exposición doctrinal. Lo que principalmente hace la metodología es estudiar los métodos para luego determinar cuál es el más adecuado a aplicar o sistematizar en una investigación o trabajo. De acuerdo a la Real Academia Española, metodología es el conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal. La metodología está constituida por el modelo de ST, una arquitectura, tareas del tutor (explicación y retroalimentación), además de un modelo de inferencia del conocimiento (Lemus, 2010).

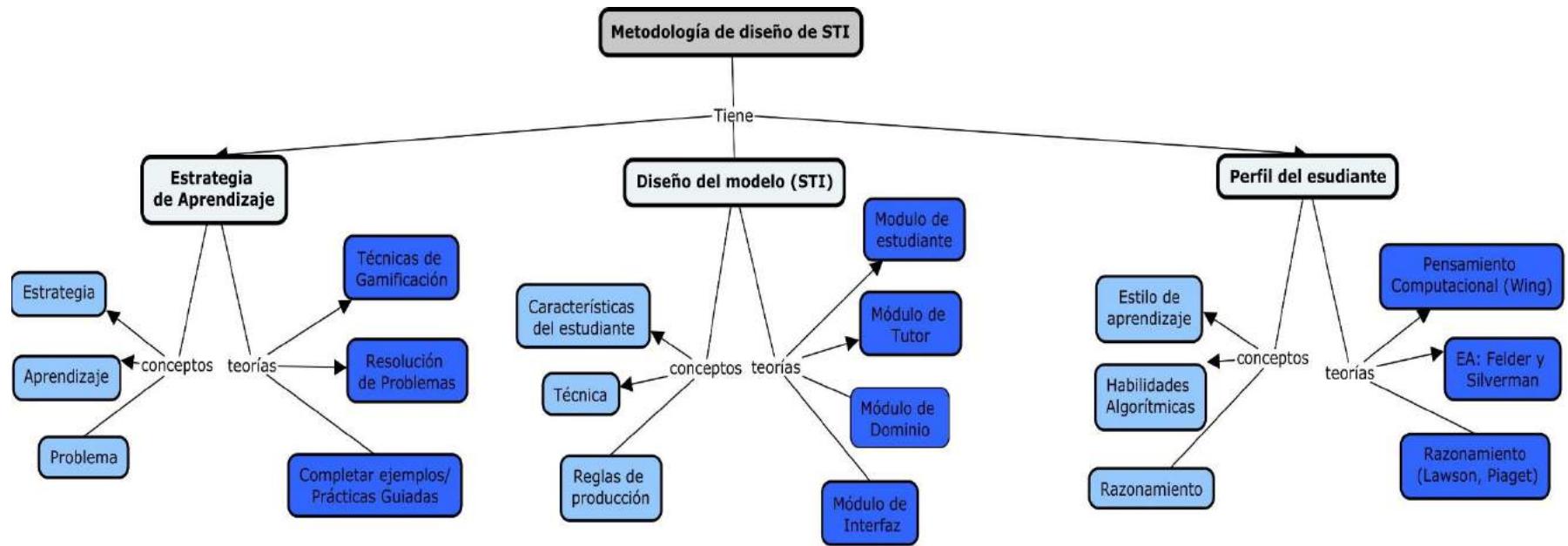


Figura 3.Relación de conceptos teóricos sobre la metodología de ST.

Fuente: Elaboración propia

Existe cierta complejidad en la construcción de ST a pesar de los avances que se han dado en las ciencias cognitivas y en las técnicas de IA, sin embargo, aún no existen metodologías flexibles que sean independientes del área de conocimiento. Otra propuesta de metodología fue dada por (Sierra, Gracia-Martinez, Cataldi y Hossian, 2004), que define que está compuesta por tres modelos que interactúan entre sí:1) el modelo de dominio o conocimiento (el mecanismo en la metodología que propone), 2) el modelo de estudiante y 3) el modelo de tutor (modelo de instrucción). En la metodología se propone definir un conjunto de reglas que definen cuándo y cómo los modelos son utilizados, las reglas constituyen una meta-modelo en el sentido de que controlan los modelos básicos y sus reglas asociadas.

### *2.2.1 Limitantes encontradas en la literatura*

Una metodología está compuesta por un plan de instrucción que define la identificación de los usuarios así como las estrategias de aprendizaje que integran técnicas de Inteligencia Artificial, la evaluación, la identificación de caso a analizar y finalmente la aplicación (Tirado, Mancilla y Cruz, 2009), además la estrategia de aprendizaje tiene que ver con los contenidos y las preferencias de aprendizaje. La metodología es un proceso que define el desarrollo de un software educativo que incorpora estrategias de instrucción, características del usuario (estudiante, profesor, desarrollador) y generar esquemas de representación del conocimiento, así como el contexto es importante, y el modelo debe ser independiente de la implementación (Trella, 2006).

La mayoría de los diseños y sistemas no son reutilizables dada la complejidad de los mismos que llevan a implementaciones que no se pueden adaptar a diversos contextos. La propuesta que se tienen es considerar a partir de una metodología que sea flexibles y sencilla, de acuerdo a las necesidades de orientación, deben de ser modulares y con estrategias de instrucción que permitan motivar el aprendizaje y desarrollo de habilidades.

Una de las limitantes que se encontró es hay escaso trabajo sobre las habilidades algorítmicas, y la propia definición se ha realizado con base al término

de Pensamiento Computacional. Las metodologías sobre ST han sido trabajadas con menor proporción a los diseños e implementación de ST; sin embargo, es importante crear una metodología que integre módulos que sean independientes de la implementación y del área de conocimiento.

Con base a las diferentes propuestas de metodología de STI, se pretende generar una metodología que permita la adaptación del contenido de conocimiento para que sea flexible de utilizar en otros contextos de aprendizaje, la arquitectura compuesta por módulos con objetivos particulares independientemente de la implementación. Además, proponer las técnicas de IA que se aplicarán de acuerdo a la estrategia de enseñanza propuesta y el tipo de instrucción definida para dar retroalimentación y seguimiento al estudiante (Sierra et al., 2004; Trella, 2006).

### *2.2.2 Requerimiento ontológico*

Se han encontrado diversidad de conceptos en el área de programación relacionados con habilidades algorítmicas, a pesar que lo asumen como una dimensión del pensamiento computacional, no se tiene una definición exacta, sin embargo se retoman varios acercamientos al concepto hasta generar una conceptualización para la investigación planteada.

### *2.2.3 Requerimientos metodológicos*

Para llevar a cabo la investigación sobre la metodología de diseño de ST enfocada en las habilidades algorítmicas algunos requerimientos son:

- Las tipos de tareas a realizar
- Las habilidades a desarrollar
- La estrategia de instrucción con base a una teoría de aprendizaje
- Un criterio taxonómico para evaluar

#### *Los tipos de tareas a realizar*

Se determina de acuerdo a la clasificación de los tipos de contenidos y materiales a integrar, como se ve en la Figura 4, puede ser de acuerdo a la instrucción y las características del contenido.

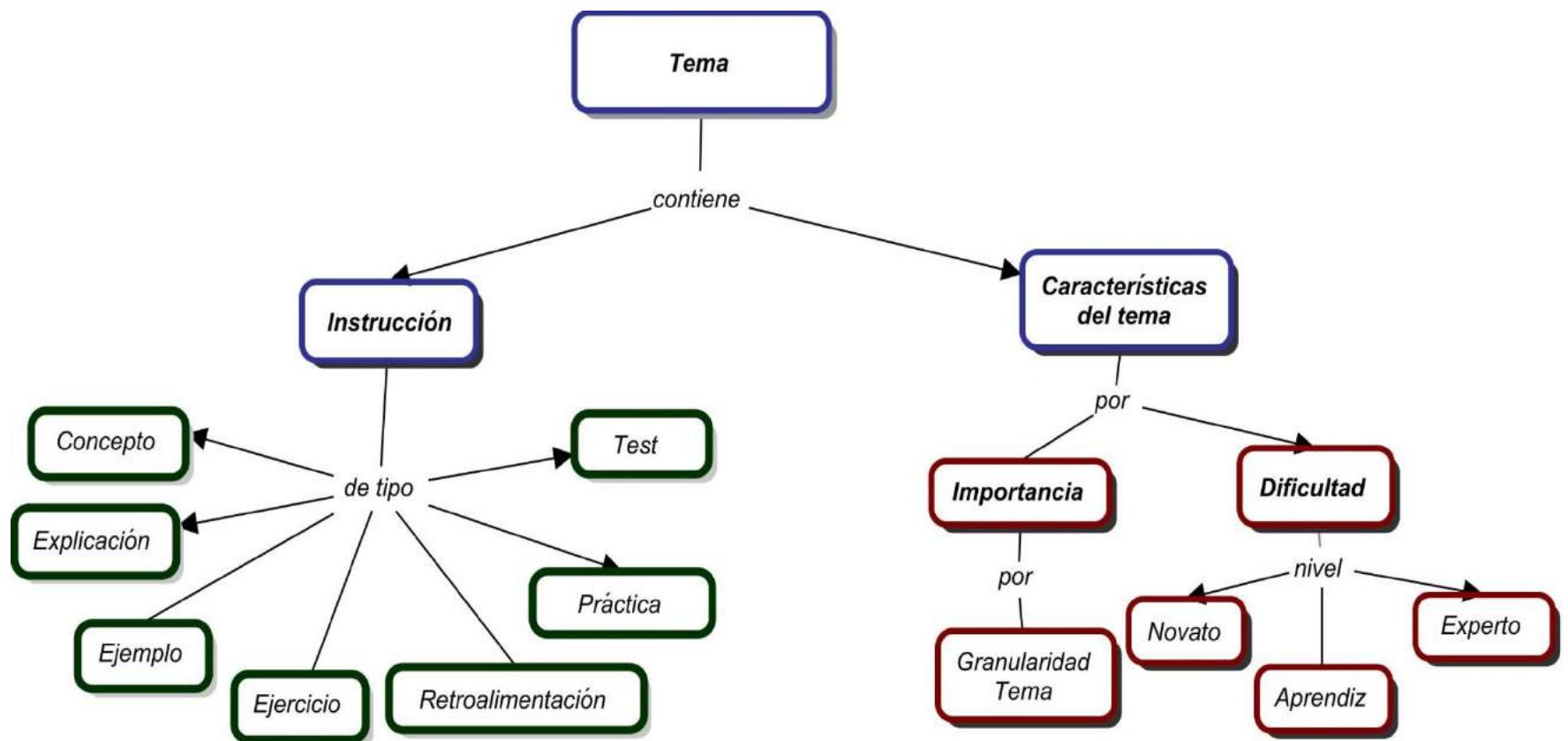


Figura 4. Modelo de instrucción.  
Fuente: Elaboración propia.

Las habilidades algorítmicas a desarrollar son:

- Interpretar los cambios que ocurren al modificar una instrucción.
- Traducir un algoritmo dado a pseudocódigo o diagrama de flujo.
- Utilizar estructuras con un fin determinado y claro (decisiones, ciclos).
- Modificar un algoritmo.
- Predecir los posibles efectos al ejecutar un algoritmo línea por línea.
- Detectar posibles errores en un algoritmo.
- Evaluar requerimientos a nivel de usuario, sistema, organización y desarrollar una solución algorítmica coherente que los tome en cuenta.
- Comparar soluciones algorítmicas que resuelven un mismo problema.
- Verificar un conjunto de instrucciones a través de pruebas de escritorio.
- Generar un algoritmo completo a partir de las especificaciones de un problema
- Generalizar un algoritmo.

Criterio taxonómico para evaluar:

Se utiliza la taxonomía de Bloom para clasificar las habilidades algorítmicas (ver Tabla 2) y se generan los niveles generales para evaluar.

Tabla 2:  
*Habilidades algorítmicas clasificadas de acuerdo a la taxonomía de Bloom.*

Taxonomía de Bloom	Habilidades
<b>Recordar</b>	<b>Reconocer la</b> sintaxis de una instrucción. <b>Reconocer la</b> semántica de una instrucción.
<b>Comprender</b>	<b>Interpretar</b> que cambios ocurren al modificar una instrucción.
<b>Aplicar</b>	<b>Traducir</b> un algoritmo dado a pseudocódigo o diagrama de flujo. <b>Utilizar</b> estructuras con un fin determinado y claro (decisiones, ciclos). <b>Modificar</b> un algoritmo.
<b>Analizar</b>	<b>Predecir</b> los posibles efectos al ejecutar un algoritmo línea por línea. <b>Detectar</b> posibles errores en un algoritmo.
<b>Probar</b>	<b>Evaluar</b> requerimientos a nivel de usuario, sistema, organización y desarrollar una solución algorítmica coherente que los tome en cuenta. <b>Comparar</b> soluciones algorítmicas que resuelven un mismo problema. <b>Verificar</b> un conjunto de instrucciones a través de pruebas de escritorio)
<b>Crear</b>	<b>Generar</b> un algoritmo completo a partir de las especificaciones de un problema Generalizar un algoritmo.

Fuente: Elaboración propia.

Se concluye que existen diversos vacíos, uno de ellos es que la mayoría de las investigaciones que se centran en el área de programación se van a cursos avanzados, por lo que se pretende trabajar con un curso de metodología de la programación para desarrollar las habilidades básicas para desarrollar algoritmos resolviendo problemas.

La mayoría de los diseños y sistemas no son reutilizables dada la complejidad de los mismos que llevan a implementaciones que no se pueden adaptar a diversos contextos. La propuesta es considerar a partir de una metodología que sea flexible y sencilla, de acuerdo a las necesidades de

orientación, deben de ser modular y con estrategias de instrucción que permitan motivar el aprendizaje y desarrollo de habilidades.

Por lo tanto, es necesaria una metodología que guíe la construcción de ST independientemente del área de conocimiento, y una herramienta de autor que de la misma forma integre diversidad en modelos de enseñanza-aprendizaje debido a que cada área de conocimiento, en especial el área de programación básica necesita un nivel de abstracción para resolver problemas, por lo que se requiere además del análisis de cómo se puede generar una estrategia didáctica que persiga un acompañamiento o retroalimentación para ir desarrollando las habilidades algorítmicas requeridas para los estudiantes de nuevo ingreso a nivel superior en la Facultad de Ciencias de la Computación.

### 3. Marco Teórico-Conceptual

En este apartado se describen los conceptos para llevar a cabo el proceso de enseñanza-aprendizaje enfocados a las habilidades algorítmicas, estos conceptos se abordan en un nivel más detallado. Paralelamente se integran aspectos esenciales sobre las metodologías de diseño de Sistemas Tutores y que respaldan la propuesta de esta investigación (Figura 5).

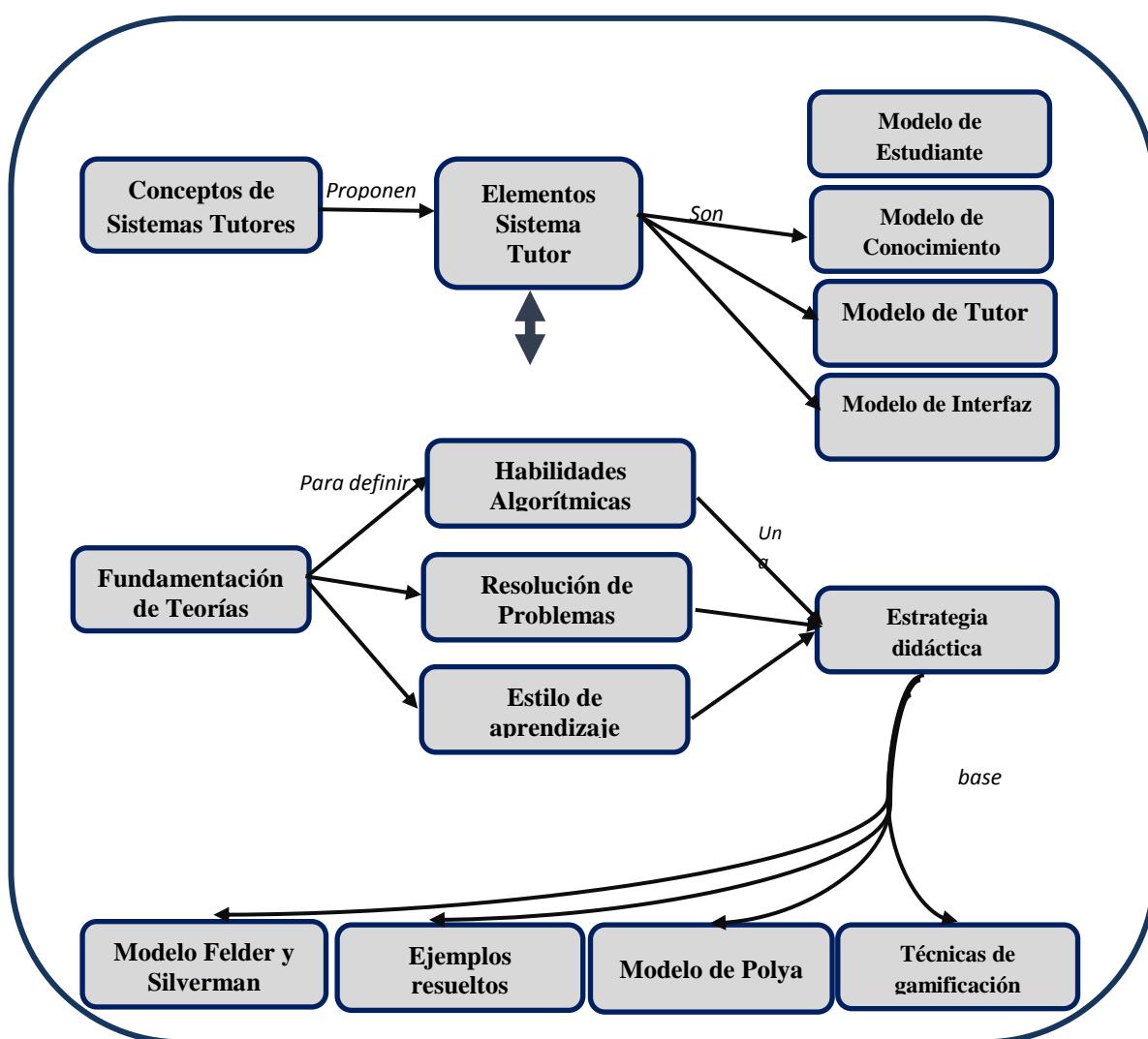


Figura 5. Relación de conceptos teóricos de la investigación.  
Fuente: Elaboración propia.

Los elementos principales que integran la metodología y generalizan las teorías de la Figura 5 son:

- Metodología de diseño de Sistema Tutor
- Estrategia didáctica
- Estilos de aprendizaje
- Resolución de problemas
- Habilidades algorítmicas
- Sistema Tutor (ST)
- Técnicas de gamificación

A continuación se hace la conceptualización, interpretación y postura de cada teoría que se integra al tema de investigación. Para iniciar *metodología* se define como parte del proceso de investigación o método científico, que sigue a la propedéutica, y permite sistematizar los métodos y las técnicas necesarias para llevarla a cabo. Los métodos elegidos por el investigador facilitan el descubrimiento de conocimientos seguros y confiables que, potencialmente, solucionarán los problemas planteados.

La metodología está constituida por el modelo de ST, una arquitectura, tareas del tutor (explicación y retroalimentación), además de un modelo de inferencia del conocimiento (Lemus, 2010). Existe cierta complejidad en la construcción de ST a pesar de los avances que se han dado en las ciencias cognitivas y en las técnicas de IA, sin embargo, aún no existen metodologías flexibles que sean independientes del área de conocimiento. En la metodología se propone definir un conjunto de reglas que definen cuándo y cómo los modelos son utilizados, las reglas constituyen un meta-modelo en el sentido de que controlan los modelos básicos y sus reglas asociadas.

De acuerdo al objetivo de la investigación sobre Sistemas Tutores se definirá una metodología compuesta por un plan de instrucción para la identificación de los estudiantes así como las estrategias de didácticas que integran técnicas o métodos, la evaluación, la identificación de caso a analizar y finalmente el diseño (Tirado, Mancilla y Cruz, 2009), además la estrategia de

aprendizaje que integre los contenidos y las preferencias de aprendizaje y desarrollo de habilidades algorítmicas.

### 3.1 Principios pedagógicos mediados por la tecnología en nivel educativo superior.

Gómez (2007) propone fusionar un sistema que integre un ejercicio (la parte educativa) y un nivel (parte lúdica) de tal manera que el estudiante aprenda y al aumentar de nivel se vea motivado a seguir aprendiendo, como menciona el autor, es una forma de integrar el modelo constructivista en cohesión con el aprendizaje por descubrimiento que se integra en varios videojuegos. Si bien la idea es similar a la que proponemos, es posible identificar los modelos que se mencionarán en la parte educativa para nuestro ST, ya que ambos, el ciclo de ejecución de un videojuego y el de una aplicación educativa basada en problemas son similares. Sin embargo, se debe definir algunos aspectos particulares dependiendo de la temática a manejar, ¿Cómo elige el programa el siguiente ejercicio a realizar? ¿Cómo puede analizar la corrección de una posible solución errónea? ¿Cómo y cuándo proporcionar ayuda al estudiante? Son algunas preguntas relevantes a contestar en el proceso de construcción de esta propuesta de diseño.

Con base a las nuevas tecnologías ahora se tienen también otras teorías como la denominada “aprendizaje mediado” ésta se basa en el análisis reflexivo de la teoría histórico-cultural o sociocultural y la actualización de las aportaciones de Vigotski (1978) sobre el estudio de tres aspectos: a) el origen de los procesos psicológicos superiores, b) el poder mediador de los signos e instrumentos para el desarrollo del pensamiento humano y c) la noción de mediación cognitiva y el papel de los instrumentos mediadores en la construcción del pensamiento a través de sus materiales.

Partiendo de la formulación de un problema por parte del profesor, a ser abierto y relacionado con la experiencia de los alumnos/as, estos formulan sus propuestas, que posteriormente se discuten y seleccionan en el aula,

estableciéndose así hipótesis o propuestas de aula que luego son corroboradas con apoyo de profesor.

En el trabajo de Alba (Alba, Elola y Luffiego, 2008) se proponen varios principios de forma integral en varios factores educativos a considerar para mejorar la enseñanza/aprendizaje y que podemos contemplar en este trabajo. Uno de estos principios es la reflexión individual y colectiva del profesorado para clarificar y planificar con realismo la programación, debe ser uno de los pilares sobre los que se apoye la mejora de la práctica docente, es decir el trabajo colegiado. Otro de los principios es referente a las capacidades y contenidos ya que frente al aprendizaje memorístico, hay que potenciar la reflexión del alumnado para que pueda establecer relaciones con sus conocimientos previos, con conocimientos de otras materias y aplicar su conocimiento a diferentes contextos, mencionan que se debe tener muy presente el desarrollo cognitivo del alumno/a porque el desarrollo de algunas de las capacidades depende de su edad y grado de madurez. Otro aspecto a considerar son los contenidos de acuerdo a las capacidades que se quieran trabajar en el área de dominio.

El último principio que se menciona en ese mismo estudio, es referente a la metodología, se deben aplicar diferentes estrategias sin lugar a dudas la investigación en el aula es vital, pero debemos de hacerla más motivadora y a su vez más acorde con un enfoque basado en competencias. El aprendizaje más relevante requiere poner en marcha procesos de reflexión, búsqueda, experimentación, comunicación e intercambio del conocimiento.

El enfoque de competencias pone énfasis en las tareas de transferencia de conocimientos a otros contextos, lo cual no es una tarea meramente reproductiva sino creativa. Por este motivo, este tipo de actividades deben ser ejercitadas con frecuencia en el aula, y por supuesto con herramientas tecnológicas (Alba et al., 2008).

Retomando el trabajo de Scott (2003) que utiliza la taxonomía de Bloom para aplicar pruebas en clases de ciencias de la computación, en el domino

cognitivo rescata seis categorías que debe desarrollar el estudiante: recuperación de datos, comprensión, aplicación, análisis, síntesis y evaluación, por lo que existen coincidencias que dan un punto de partida como características y habilidades a trabajar y modelar la forma en que el estudiante le sea más fácil de aprender y por lo tanto construir su propio conocimiento y mejorar sus habilidades en resolución de problemas.

En el trabajo de Altuna et al. (2014) los autores describen la integración de una retroalimentación dependiendo de la forma del problema ya sea estática o dinámica. Manejan el modelo del dominio con las formas de representación y razonamiento que le permiten funcionar con fuente de conocimiento y un estándar para evaluar las acciones del estudiante dentro de la temática que aprende. Su forma de enfocar el desarrollo de un modelo de dominio está directamente relacionada con la estructura de conocimientos y la idoneidad para soportar los procesos de tutoría. Se trabaja con un enfoque similar al usado para el entrenamiento de estudiantes para competencias (Urbancic y Trampus, 2012; Zhao, Wang et al., 2013). Se enuncia el problema y se especifican formatos rígidos de entrada y salida, en este trabajo el estudiante deberá implementar el programa completamente, usando las librerías básicas del lenguaje de programación. Éste trabajo es interesante en el sentido de que el tutor/profesor le da seguimiento continuo al estudiante, dando la retroalimentación.

Otro trabajo que maneja la parte de motivación y autoeficacia es el de López, Hernández, y Farran (2011) donde después de usar una plataforma de evaluación automática que va guiando al estudiante en su proceso le va midiendo su avance, lo que lo corrige y a la vez motiva en el caso de que sus respuestas sean correctas, una vez encaminado en ésta forma de trabajo la eficacia en la programación del estudiante se ve mejorada. Es decir, se demostró que la evaluación automática aumenta la eficacia en la solución de problemas de programación, de hecho los ejemplos que se integran se parecen mucho a los exámenes que algunas empresas hacen para verificar que el personal a contratar sabe programar mostrando segmentos de código a resolver.

Posiblemente uno de los primeros elementos a identificar es su nivel en las habilidades algorítmicas y análisis de datos para proponer un seguimiento en su proceso de aprendizaje, para que en base a esto se pueda dar un seguimiento del avance en la solución de problemas. Por otra parte, se considera elegir un tema o unidad para proponer los problemas, además de definir las habilidades y conocimientos que se requiere alcanzar. Éste proceso continuará cambiando de acuerdo a las propuestas e investigación que se realice, lo que se identifica en éste momento es que se basará en un lenguaje estructurado y visual que le permita tener una menor carga cognitiva y proponer soluciones viables al estudiante.

En particular se requiere ofrecer un método de entrenamiento que personalice el proceso de aprendizaje y que permitirá su entrenamiento dirigido de acuerdo a su avance. Se propone modelar el aprendizaje personalizado y el modelo de dominio (del objetivo de la materia de programación), dando un seguimiento exclusivo del alumno para conseguir el entrenamiento para el tema elegido en la materia de metodología de la programación.

En la interacción estudiante-tutor es necesario identificar cómo el estudiante está resolviendo un problema y brindarle ayuda cuando cometa errores, en ese momento debe dar un soporte de materiales, y con explicaciones tratar de resolver el problema.

De acuerdo a diversos autores se proponen algunos métodos llevados a cabo en el nivel superior con materias iniciales de programación, uno de ellos lo proponen Byrne y Lyons (2001) de la Universidad de Irlanda, como el modelo de Inventario de Estilos de Aprendizaje contemplando un ciclo de cuatro modos de aprendizaje adaptativo: experiencia concreta, observación reflexiva, conceptualización abstracta, y experimentación activa. Lo que se sugiere es identificar la forma en que el estudiante se ve inmerso en el ciclo de aprendizaje, es decir si solo tratando de resolver el problema conceptualmente o es necesario resolverlo en la práctica/ ejecución de tal manera que pase por los cuatro modos pasando de lo abstracto a los conceptos concretos y de la experimentación a la

experiencia activa. La ventaja que da este modelo es que se aplican test para identificar en cuál cuadrante se encuentra el estudiante y partir de ello pasar por el ciclo completo.

Para poder definir algunas de las habilidades que se requieren trabajar con los estudiantes es posible referenciar el trabajo de Scott (2003) que utiliza la taxonomía de Bloom para aplicar pruebas en clases de ciencias de la computación, identifica tres tipos de dominios en el aprendizaje humano: dominio cognitivo (aprendizaje/habilidades), dominio afectivo (sentimientos o emociones en el área de aprendizaje), dominio psicomotor (habilidades manuales o físicas), pero se centra en el domino cognitivo que rescata seis categorías que debe desarrollar el estudiante: recuperación de datos, comprensión, aplicación, análisis, síntesis y evaluación.

De acuerdo a la literatura se pretende integrar la taxonomía de Bloom como base de la construcción del instrumento que evaluará las habilidades algorítmicas e identificará el aprendizaje previo que tiene el estudiante en su proceso de resolución de problemas verificando cómo el estudiante pasa de lo conceptual a lo abstracto y como pueden dar solución a través de la evaluación de su propuesta de solución en un lenguaje simple. Al tratar de crear niveles de estudiantes según las categorías de la taxonomía de Bloom se dará una propuesta de contenido diverso para apoyarlos de forma precisa (Byrne y Lyons, 2001; Scott, 2003).

### 3.3 Teorías del Aprendizaje

Actualmente existen varios instrumentos de apoyo en el aprendizaje de los lenguajes de programación, como libros, herramientas o ambientes de programación como JCreator, NetBeans, PHP, .NET entre otras, estas ayudan a desarrollar los sistemas, pero ¿quién asiste a los estudiantes para entender el paradigma de la programación y comenzar a plasmar la solución de un problema? Realmente no hay ambientes que promuevan el aprendizaje de la programación, de tal manera que de acuerdo al estilo de aprendizaje del estudiante se dé un seguimiento del contenido a presentar. Por lo tanto, es importante facilitar la

comprensión y entendimiento de la programación básica para resolver problemas complejos que se presenten al estudiante.

A continuación se describirán algunas teorías que se derivan de la psicología del aprendizaje y se preocupan de los cambios permanentes en el comportamiento del individuo, estas teorías consideran diversos aspectos en el proceso de enseñanza-aprendizaje como la motivación, los intereses y las necesidades de los estudiantes.

Para la realización de este trabajo es considerado el enfoque cognitivista, el cual encarga del estudio de los procesos que intervienen durante el proceso de aprendizaje, se preocupa de los procesos de comprensión, transformación, almacenamiento y uso de información. Desde la entrada sensorial, pasando por el sistema cognitivo, hasta que la respuesta es producida. Así mismo, en el enfoque cognitivista se retoma algunas teorías de los autores como Piaget, Ausubel y Bruner, con el fin de compaginar con los test propuestos de estilo de aprendizaje y el test de pensamiento computacional para generar el aprendizaje significativo a través del andamiaje.

De acuerdo a Ausubel, Novak y Hanesian (2000), ellos consideran que es importante conocer el estado actual de nuestro conocimiento y distinguir los principales tipos de aprendizaje:

**Recepción:** El aprendizaje por recepción tiene como entrada el contenido de lo que se mostrará al alumno en su forma final. En éste, el alumno se internaliza en el contenido dado, recupera y produce a futuro, los ejemplos más claros son en la enseñanza tradicional en aula donde se enseñan las tablas de multiplicar, teoremas o aprende nombre de objetos.

**Descubrimiento:** En el aprendizaje por descubrimiento debe de ordenar la información, integrarla en una estructura cognoscitiva y reorganizar o transformar la combinación integrada de manera que produzca el producto final deseado o descubra la relación entre medios y fines que hacía falta (Ausubel, Novak y Hanesian, 2000).

**Significativo:** El aprendizaje significativo es cuando relacionamos los conocimientos previos con los nuevos, interactuando la estructura cognoscitiva previa, además de adaptarlo a un contexto, que permite al individuo realizar una metacognición (aprende a aprender) y le sirve a lo largo de la vida. El aprendizaje significativo sucede una vez que el aprendiz acepta que las cosas pueden hacerse de diferente manera a la que él cree, frente a una realidad objetiva.

De acuerdo a los aprendizajes mencionados, se considera que el aprendizaje que tendrá mayor impacto es el significado donde interviene la comprensión ya que integra los conocimientos adquiridos previamente por el estudiante, así como el contexto, sus necesidades y potencialidades, por tanto, se espera que sea a largo plazo.

### 3.4 Estilos de Aprendizaje

Debido a que cada persona piensa de forma diferente, por ejemplo, un conjunto de personas pueden dar opiniones diferentes de una cosa o problema, de acuerdo a su experiencia, conocimiento, percepción y contexto en la educación se debe tomar en consideración estos factores dentro del proceso enseñanza-aprendizaje.

Normalmente se utiliza el término “estilo de aprendizaje” para referirse al hecho de que cada persona utiliza su propio método o estrategias para aprender, aunque las estrategias varían de acuerdo a lo que se requiera aprender, por lo tanto, cada individuo debe desarrollar sus propias preferencias que identifique y defina su estilo de aprendizaje.

El concepto de “estilo” aparece por primera vez en la psicología con Lewin 1935 (citado en Hederich, 2004) además de la relación con la personalidad incluyó las características de evaluación y la disposición del estudiante para desarrollar las habilidades puramente cognitivas. Sin embargo, Lewin (1935) y Aguirre-Chaves (2007) se centran en el funcionamiento y habilidades cognitivas para definir el estilo de aprendizaje.

Existen varios modelos de estilos de aprendizaje que muestran como el humano procesa la información en alguna etapa de acuerdo a su percepción (sentidos). Los estilos de aprendizaje son rasgos cognoscitivos, afectivos y fisiológicos que sirven como indicadores relativamente estables, de cómo las personas perciben, interactúan y responden a sus ambientes de aprendizaje (Alonso, Gallegos y Honey, 1999).

En la Tabla 3, se describen las características o factores que algunos autores como Kolb, Felder y Silverman, Dunn y Honey-Murford han considerado para estudiar los estilos de aprendizaje de los estudiantes. Las diversas conceptualizaciones de estos autores son con base en las características de los estudiantes.

Para este trabajo se maneja la siguiente definición de estilo de aprendizaje: es la adaptación que realiza el estudiante al interactuar en un contexto e información, para propiciar su aprendizaje. Dado los posibles factores que intervienen: afectivos, fisiológicos, cognitivos y de interacción. Las dimensiones que ayudan a identificar su preferencia de aprendizaje son: organizar, procesar, percibir, recibir y entender nueva información (Sánchez-Román, 2017).

El modelo para determinar el estilo de aprendizaje que da mayor información en el área de ciencias de acuerdo a la literatura y ha dado mejores resultados para identificar el estilo preferente, es el propuesto por Felder y Silverman, por lo que se considera este modelo para evaluar a los estudiantes de ciencias de la computación para identificar cómo aprenden a partir de los elementos que definen el modelo y a continuación se describe.

Tabla 3

Comparación de estilos de aprendizaje.

Tests de Estilos de Aprendizaje					
Autores	Dunn y Dunn(1974)	Kolb (1984)	Honey-Murford (1989)	Felder y Silverman (1988)	Howard Gardner (1993)
Características del aprendizaje	Se enfoca en cinco variables: ambiental, sociológica, emocional, física, psicológica determinando factores que impactan en el proceso de aprendizaje del estudiante.	Se basa en la teoría del Aprendizaje Experiencial que modela el proceso de aprendizaje e incorpora la función primordial de la experiencia en este proceso.	El EA se define a partir de rasgos cognoscitivos, afectivos y fisiológicos que sirven como indicadores relativamente estables, de cómo las personas perciben, interactúan y responden a sus ambientes de aprendizaje.	El EA se define cómo los estudiantes prefieren organizar, procesar, percibir, recibir y entender nueva información.	Inteligencia: Capacidades para resolver problemas.
Número de items	118	9	80	44	80
Resultado	Preferencia alta/baja	Preferencia individual	Estilo predominante	Estilo predominante	
Nombre del test	Building Excellence Inventory (Rundle y Dunn, 2000)	Learning Style Inventory (LSI) (Kolb, 1976), CHAEA (Cuestionario de Honey y Alonso de Estilos de Aprendizaje) (1994)	Learning Stilles Questionnaire (LSQ).	Index of Learning Styles (ILS) (Felder y Solomon, 2004)	Inteligencias Multiples
Dimensiones	Ambiental Emocional Física Percepción Psicológica	Convergente, asimiladores, divergente, acomodadores,	activo, reflexivo, teórico y pragmático	inductivo/deductivo; activo/reflexivo; sensorial/intuitivo; verbal/visual:secuencial/global.	Inteligencia s:Linguistico-verbal, musical, corporal, kinestésica, lógica-matemática, espacial, interpersonal, intrapersonal, Naturalista

Fuente: Elaboración propia.

### *3.4.1 Modelo de Felder-Silverman*

En el modelo de Felder-Silverman (1988) los estudiantes están representados por su clasificación en cinco dimensiones. Estas dimensiones están basadas en las consideradas más importantes dentro del campo de los estilos de aprendizaje, y son independientes unas de otras. Muestran cómo los estudiantes prefieren organizar (inductivo/deductivo), procesar (activo/reflexivo), percibir (sensorial/intuitivo), recibir (verbal/visual), y entender (secuencial/global) nueva información.

A pesar de que estas dimensiones no son nuevas en el campo de los estilos de aprendizaje, la forma en la que Felder-Silverman las describen sí lo es. Mientras que la mayoría de los modelos de estilos de aprendizaje que incluyen dos o más dimensiones obtienen tipos de estudiantes para estas dimensiones, como en el modelo de Kolb (1984), Felder y Silverman describen los estilos de aprendizaje usando escalas que van de 11 a -11 puntos para cada dimensión (incluyendo solo valores impares). Los autores justifican la eliminación de la dimensión inductivo/deductivo desde un punto de vista estrictamente pedagógico, ya que no desean proporcionar a profesores y alumnos una herramienta que justifique unos hábitos que no consideran positivos. En concreto, no desean que esa dimensión sirva para potenciar la enseñanza y el aprendizaje deductivos, claramente preferidos tanto por profesores como por alumnos: la mayoría de los alumnos conscientemente prefiere una enseñanza deductiva, en el sentido de que sólo tienen interés en aprender aquello de lo que van a ser examinados y también para la mayoría de los profesores este estilo de enseñanza es más sencillo que su opuesto inductivo. Por lo tanto, el estilo de aprendizaje de cada estudiante está representado por cuatro valores entre 11 y -11, uno por cada dimensión. Felder y Silverman (1988) consideran las preferencias como tendencias, esto es, que incluso un estudiante con una fuerte preferencia por un estilo de aprendizaje particular puede actuar algunas veces de forma diferente.

En la dimensión sensorial/intuitivo en éste caso a los estudiantes les gusta aprender hechos y material concreto, usando sus experiencias sensoriales de

hechos particulares como la primordial fuente de información. Les gusta resolver problemas con aproximaciones estándar y tienden a ser cuidadosos con los detalles. Son considerados más realistas, sensatos y prácticos, y les gusta relacionar el material aprendido con el mundo real. Como polos opuestos a los sensoriales están los estudiantes intuitivos: éstos prefieren aprender materiales abstractos como teorías y sus significados subyacentes, con principios generales en vez de hechos concretos, siendo estos principios la principal fuente de información. Les gusta descubrir posibilidades y relaciones y tienden a ser más innovadores y creativos.

Como resultado, obtienen mejores puntuaciones en los exámenes con preguntas abiertas que en los exámenes tipo test. Esta dimensión difiere de la dimensión activo/reflexivo en un aspecto importante: la dimensión sensorial/intuitivo se refiere a la fuente de información preferida mientras que la dimensión activo/reflexivo tiene que ver con el proceso de transformación de la información percibida en conocimiento.

La dimensión visual/verbal hace referencia a la preferencia en cuanto al formato de los datos. Diferencia entre estudiantes que recuerdan mejor lo que han visto (fotos, dibujos, diagramas, gráficas, etc.) de los que prefieren representaciones textuales, independientemente de que sean escritas u orales.

En la dimensión secuencial o global, se detecta la forma en que los estudiantes entienden el material, en esta dimensión los estudiantes secuenciales eran llamados serialistas y los globales, holísticos. Los estudiantes secuenciales aprenden en pequeños pasos incrementales y, por lo tanto, tienen un progreso lineal en su aprendizaje. Tienden a seguir soluciones lógicas y estructuradas. En oposición, los estudiantes globales usan un proceso de pensamiento holístico y aprenden en grandes saltos. Tienden a absorber el material de aprendizaje de una forma casi aleatoria, sin ver las conexiones que existen, para obtener de repente una visión general. Son capaces de resolver problemas complejos y poner cosas juntas de formas novedosas; sin embargo, tienen dificultades en explicar cómo lo han hecho. Ya que la visión general es tan importante para ellos, tienden a estar

interesados en los resúmenes y en el conocimiento extenso, mientras que los estudiantes secuenciales están más interesados en los detalles. Posteriormente Felder y Solomon desarrollaron el cuestionario Index of Learning Styles (ILS) (Felder y Solomon, 2004), un cuestionario con 44 preguntas. La preferencia en cada dimensión se expresa como el resultado de las respuestas ("a" o "b") a las 11 cuestiones. El resultado es la resta de las respuestas "b" a las respuestas "a".

En este estudio se utiliza el modelo de Felder-Silverman, debido a que para resolver los problemas se necesita saber cómo procesa la información de acuerdo a las habilidades que se requieren para programar. Además el tipo de cuestionario tiene varios items que pueden dar mayor precisión en el tipo dominante del alumno y una de las propuestas que se hace es tener reservado el tipo de estilo predominante de forma jerárquica, de tal manera que se pueda mostrar el contenido más preciso y el que se tenga de forma completa para dar su seguimiento.

### 3.5 Estrategias de Aprendizaje

Para implementar una estrategia de aprendizaje es importante desarrollar los aspectos de competencia y procesos cognitivos, además es importante tener definidos los objetivos de aprendizaje (Pimienta, 2012). Por otra parte, el estudiante debe ser capaz de identificar las acciones a seguir mediante recursos y procedimientos para generar una autonomía de aprendizaje (Danilov y Skatin, 1985; Pimienta, 2012).

#### 3.5.1 Teoría de la Carga Cognitiva

La Teoría de la Carga Cognitiva (TCC) (Sweller, 1994) y la Teoría Cognitiva del Aprendizaje Multimedia (TCAM) (Najjar y Mayer, 2003), en esencia, pretenden alinear el diseño de material instruccional con la Arquitectura Cognitiva Humana (ACH). Entender la forma cómo el cerebro piensa es uno de los principales retos para explicar y mejorar el aprendizaje. Según los teóricos de la TCC, cualquier instrucción o enseñanza es efectiva solo si su diseño ha tenido en cuenta las características de la cognición humana. Así pues, la arquitectura cognitiva es la

manera como las estructuras y funciones cognitivas del ser humano están organizadas. Según la TCC, la información que entra al cerebro es procesada en tres diferentes estructuras, a saber: a) la memoria sensorial, b) la memoria de trabajo, y c) la memoria de largo plazo (Ver Figura 6).

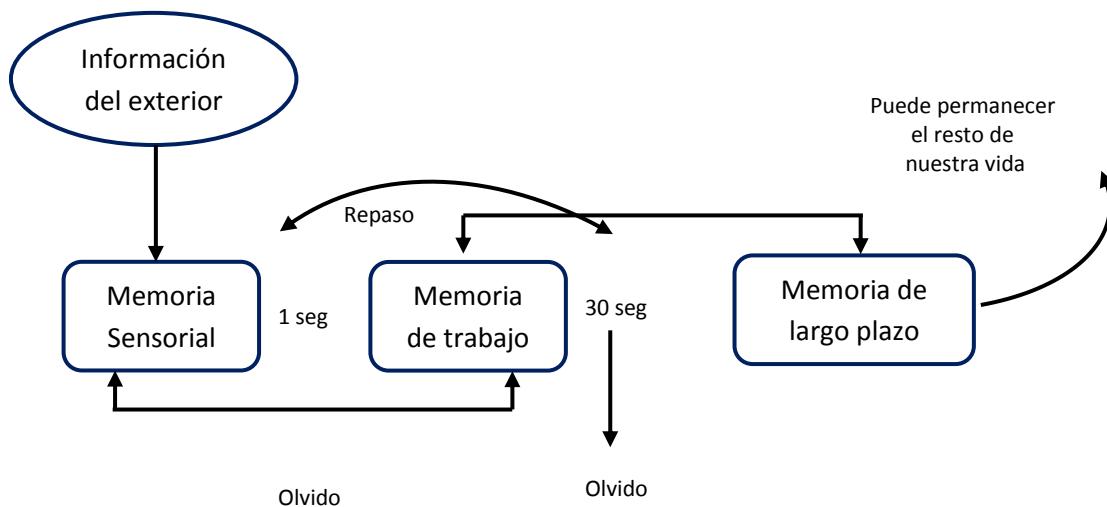


Figura 6. Estructura de la Arquitectura Cognitiva Humana (ACH).  
Fuente: Modificado de Dale Shaffer, Wendy y Juhani Touvinen, 2003.

### 3.5.2 Resolución de problemas

Es importante saber qué es un problema, y se define como “un obstáculo que separa la situación actual de una meta deseada” (Bransford y Sein, 1984). Resolver un problema consiste en pasar de una situación a otra, para ello son importantes las estrategias heurísticas que son las operaciones útiles en la solución, o las sugerencias heurísticas o indicaciones dirigidas a centrar la atención del resolutor sobre aspectos del problema. Normalmente en un modelo de resolución de problemas se analizan las fases del proceso que llevan a la solución del problema.

A partir de la desagregación del concepto desde la definición de varios autores, se puede decir que la resolución de problemas está fundamentada en activar la curiosidad, descubrimiento y exploración en el estudiante para resolver el problema (Polya, 1985; Godino, 2004), por otra parte se define como la

adquisición de modelos de pensamiento a partir de la construcción de pensamiento declarativo y procedimental. Al continuar del proceso, se debe dar confianza ante situaciones no familiares, fomentar el reto y llegar al triunfo (solución del problema) (Polya 1965; Godino, 2004; Piaget 1978).

### *3.5.3 Ejemplos y problemas por completar*

La teoría de la carga cognitiva (Sweller, 1988; Sweller, 1994) sugiere estrategias instruccionales (llamadas efectos) que toman en cuenta las limitaciones intrínsecas del cerebro humano. En el contexto de esta teoría, se ha documentado que el uso de ejemplos resueltos y de problemas por completar son estrategias instruccionales efectivas para enseñar a estudiantes principiantes (Gerjets, Scheiter y Catrambone, 2004; Renkl, Atkinson y Grobe, 2004; Renkl, Hilbert y Schworm, 2009).

La teoría de la codificación dual (Paivio, 1990; Sadoski y Paivo, 2005) señala que la mente humana es capaz de retener y recuperar información y conocimiento; si estos se presentan en una modalidad dual, por medio de lenguaje verbal e imágenes.

Como alternativa a los ejemplos resueltos, Van Merriënboer y Krammer (1987,1990) propusieron el uso de “problemas por completar”. Los problemas por completar son problemas en los que se cuenta con un estado de avance parcial y un estado objetivo deseado y se solicita a los aprendices que provean una o varias soluciones parciales e intermedias. Los ejemplos por completar son efectivos en dominios orientados al diseño de software, diseño de circuitos, procesos de planeación de producción, programación de control numérico y arquitectura (Sweller et al., 1998). Los problemas por completar proveen un puente entre los ejemplos resueltos y problemas convencionales. Es decir, que los ejemplos resueltos son problemas con soluciones completas y los problemas convencionales son problemas por completar sin una solución provista.

Por lo tanto, dentro de la estrategia instruccional propuesta se comenzará con los ejemplos resueltos para que los estudiantes novatos puedan aprender el

proceso completo y continuar con problemas por completar que provean casi toda la solución y gradualmente avanza hacia problemas por completar en los que el estudiante provea a cada caso toda la solución que se conoce como estrategia de completar (Sweller et al., 1998), de esta forma se va avanzando gradualmente y no se da todo el contenido a un estudiante aprendiz, sino sólo lo que amerita para llegar al objetivo del tema. De esta manera, la estrategia se integra como parte de la metodología en el módulo de tutor.

### *3.5.4 Técnicas de Gamificación*

En un sentido general, la Gamificación se describe como el proceso de pensamiento de juego y sus mecanismos para atraer a los usuarios y hacerlos resolver problemas (Zichermann y Cunningham, 2011). Esta definición puede aplicarse a cualquier situación, sin embargo en el ámbito educativo la Gamificación se refiere al uso de elementos del juego para involucrar a los estudiantes, motivarlos a la acción y promover el aprendizaje y la resolución de problemas (Kapp, 2012). Así, la propuesta debe estar basada en juegos (reglas, interactividad, retroalimentación, etc.), incluir niveles, recompensas, insignias y/o puntos.

Existe un enorme aumento en la investigación sobre la gamificación educativa y se argumenta que la educación podría ser un buen ajuste con la gamificación, ya que la gamificación busca atraer a los estudiantes a través de tres áreas cruciales para el compromiso: cognitivas, emocionales y sociales (Lee y Hammer, 2011). Los ambientes de educación en las escuelas pueden tornarse aburridos para una gran cantidad de niños, por lo que el compromiso y la motivación por el aprendizaje normalmente son bajos (Orejuela et al., 2013).

La programación en Scratch se basa en un conjunto de instrucciones icónico/textuales que se combinan o encastran para crear programas, estos bloques sólo se acoplan si la sintaxis es correcta, lo que libera al estudiante de la complejidad de la misma y le permite concentrarse en la solución del problema. Como afirma Pozo (2008) “*puede decirse que si los aprendices se entrenan sólo*

*en completar ejercicios (tareas cerradas o rutinarias para las que han aprendido ya una solución específica) difícilmente aprenderán a resolver problemas (tareas más abiertas para las que hay que buscar vías de solución) [...]. Sólo entrenándose en la solución de problemas se aprende a resolver problemas” (p. 168).*

Como complemento de la estrategia de aprendizaje, se considera integrar algunas técnicas de gamificación e implementaciones finales en el lenguaje educativo de Scratch, donde se integren las soluciones alcanzadas o seleccionadas por el estudiante.

### 3.6 Habilidades algorítmicas

Para conceptualizar las habilidades algorítmicas es necesario comenzar con la definición de Pensamiento Computacional, que es la capacidad de formular y solucionar problemas apoyándose en los conceptos fundamentales de la computación, y usando la lógica sintaxis de los lenguajes informáticos de programación: secuencias básicas, bucles, iteraciones, condicionales, funciones y variables (Román-González, Pérez-González y Jiménez-Fernández, 2015). Proponen un instrumento que mide el pensamiento computacional en sus niveles más bajos de complejidad cognitiva ('reconocer' y 'comprender'), el cual se está utilizando para la investigación (Román-González et al., 2015).

El pensamiento algorítmico es, de alguna manera, un conjunto de habilidades que están conectadas a la construcción y comprensión de algoritmos. Según Futschek (2006), este pensamiento incluye las siguientes capacidades:

- a) Analizar problemas dados.
- b) Especificar un problema de manera precisa.
- c) Encontrar las acciones básicas que son adecuadas para resolver el problema dado.
- d) construir un algoritmo correcto para resolver un problema determinado, utilizando las acciones básicas.
- e) Pensar en todos los posibles casos tanto especiales como normales de un problema.

f) Mejorar la eficiencia de un algoritmo.

Este análisis, previo al trabajo con el entorno de programación Scratch, implica una actividad cognitiva que involucra planificación, formulación de hipótesis, abstracción, comprensión lingüística; es decir, las mismas habilidades identificadas en los estudios sobre pensamiento computacional (Wing, 2006; ISTE, 2011, International Society for Technology in Education). Esto implica la capacidad de definir y enunciar con claridad un problema; descomponerlo en subproblemas más pequeños y manejables y, describir una solución a lograr en un conjunto de pasos bien definido. La capacidad para analizar y dar solución a problemas, es una habilidad importante para todo ser humano, y en especial para los estudiantes (Polya, 1965).

Es fundamental revisar el proceso de la actividad cognitiva del estudiante, que involucra planificación, formulación de hipótesis, abstracción, comprensión plantear diseños e implementaciones en un lenguaje técnico.

En la Tabla 4 se muestran define las habilidades algorítmicas a evaluar en el trabajo para la solución del problema. Se trabaja con factores importantes en el pensamiento algorítmico (habilidades de recordar, analizar y modelar), y solución de problemas (habilidades de analizar, abstraer, modelar, evaluar), tomando en consideración tres dimensiones: habilidades cognitivas básicas, habilidad de abstracción, habilidad de resolución de problemas. Así como la clasificación de las habilidades a desarrollar con base a la taxonomía de Bloom y las posibles tareas que puede hacer el estudiante en cada una de las categorías definidas.

Tabla 4:

*Categorización de las habilidades algorítmicas.*

Dimensiones	Indicadores	Habilidades
<b>Habilidades cognitivas básicas</b>	Recordar	<b>Reconocer la sintaxis de una instrucción.</b>
		<b>Reconocer la semántica de una instrucción.</b>
	Comprender	<b>Interpretar que cambios ocurren al modificar una instrucción.</b>
		<b>Traducir un algoritmo dado a pseudocódigo o diagrama de flujo.</b>
<b>Habilidad de abstracción</b>	Aplicar	<b>Utilizar estructuras con un fin determinado y claro (decisiones, ciclos).</b>
		<b>Modificar un algoritmo.</b>
<b>Habilidad de Resolución de Problemas</b>	Analizar	<b>Predecir los posibles efectos al ejecutar un algoritmo línea por línea.</b>
		<b>Detectar posibles errores en un algoritmo.</b>
<b>Desarrollo y uso de algoritmos</b>	Probar	<b>Evaluar requerimientos a nivel de usuario, sistema, organización y desarrollar una solución algorítmica coherente que los tome en cuenta.</b> <b>Comparar soluciones algorítmicas que resuelven un mismo problema.</b>
		<b>Verificar un conjunto de instrucciones a través de pruebas de escritorio)</b>
	Evaluuar	<b>Generar un algoritmo completo a partir de las especificaciones de un problema</b> Generalizar un algoritmo.

Fuente: Elaboración propia.

### 3.7 Sistema Tutor Inteligente

En la mayoría de los cursos, los docentes se han visto rebasados en relación de atención uno a uno con la cantidad de los estudiantes, además de una variedad en sus niveles de conocimientos previos que tienen los estudiantes (Ausubel et al. 1983). “Los sistemas inteligentes deberían proveer algunas características en función de los propósitos por los que el estudiante recurre a él, tales como: la perspectiva desde la que debe impartir los conocimientos a los alumnos, la forma de adaptación del tutor a los conocimientos previos de los alumnos y la selección de la estrategia de enseñanza más adecuada para el alumno que lo consulta” (Cataldi, Salgueiro, Lage y García-Martínez, 2005).

Los Sistemas Tutores Inteligentes (STI) surgen como una evolución de la enseñanza asistida por computadoras, al combinar conocimientos de pedagogía, psicología, ciencias cognitivas con avances en la ciencia de la computación, en especial de Inteligencia Artificial (IA), usan diversas técnicas de IA en la generación de rutas de aprendizaje, selección de actividades, soporte durante las actividades, evaluación, presentación de los contenidos, entre otras (Woolf, 2009 citado en Altuna, Guibert y Estrada, 2014).

Un STI tienen como meta reproducir el hacer de un profesor humano competente, que adapta técnicas para el aprendizaje de un dominio de enseñanza en función del perfil del estudiante (Conati, 2009), cuando él guíe al alumno deberá tener “reglas” almacenadas para saber qué hacer en casos como: el alumno no puede contestar una pregunta que le hace el tutor o contesta en forma incompleta una pregunta que le hace el tutor (Cataldi et al., 2005). En algunos trabajos se reportan ganancias significativas en el aprendizaje resultado del uso de ST con respecto a otros tipos de sistemas que se usan para apoyar el proceso de enseñanza-aprendizaje (Brawner, Holden et al., 2011).

Los STI permiten la emulación de un tutor humano para determinar qué enseñar, cómo enseñar y a quién enseñar a través de un módulo del dominio: que

define el dominio del conocimiento, un módulo del estudiante: que es capaz de definir el conocimiento del estudiante en cada punto durante la sesión de trabajo, un módulo del tutor: que genera las interacciones de aprendizaje basadas en las discrepancias entre el especialista y el estudiante y finalmente la interface con el usuario: que permite la interacción del estudiante con un STI de una manera eficiente (conocimiento sobre cómo presentar los contenidos) (Cataldi y Lage, 2009).

El éxito tutorial depende más de la investigación científica en el campo de la ciencia cognitiva que se ha estado integrando en los STI. Lo que entonces genera una pregunta: ¿cuál es el futuro de los STI, sabiendo que ya existen herramientas de autor?, o bien ¿cómo generar los modelos flexibles para que sean independientes del área de dominio de conocimiento? Para diseñar un STI (Rodríguez, Castillo y Lira, 2013) se basa en la propuesta de Laureano y De Arriaga (2000) donde se integra el diseño gráfico del conocimiento, diseño del modelo cognitivo (de acuerdo a las características del estudiante), el diseño de la escenarios de acuerdo al estilo del estudiante, diseño del proceso de enseñanza (de acuerdo al tipo de conocimiento y la inteligencia identificada en el estudiante).

Por lo tanto, un STI se puede definir como un sistema computacional que integra técnicas de IA para la implementación de los modelos de estudiante, modelo de tutor, modelo de dominio e interfaz, donde se integren los objetivos a impartir, las estrategias de enseñanza y los contenidos del dominio. El objetivo de los STI es generar una enseñanza asistida y que responda a los cambios de aprendizaje del estudiante a través de retroalimentaciones (Altuna Castillo, 2014; Carbonell, 1970; Cataldi et al., 2005). Actualmente, han pasado de ser herramientas de laboratorio, a integrarse en las aulas y lugares de trabajo.

Existen diferentes herramientas que pueden apoyar al docente a la enseñanza de la programación básica, sin embargo es difícil llevar el seguimiento y avance del aprendizaje de un grupo de más de cuarenta estudiantes. Por lo tanto es importante tener una herramienta tecnológica que apoya a la enseñanza-aprendizaje del estudiante con una atención particular en el tema que se le

dificulte para alcanzar el objetivo de la materia. Es entonces que se elige la creación de una metodología de Sistema Tutor que integre las características necesarias con modelos que apoyen la instrucción, el aprendizaje y midan las habilidades algorítmicas necesarias en la resolución de problemas.

### *3.7.1 Módulos de un Sistema Tutor Inteligente*

Para desarrollar STI con módulos intercambiables y partes reutilizables, se debe efectuar un rediseño de los módulos básicos del modelo propuesto por Carbonell (1970). Se observa que algunos investigadores detectaron que la arquitectura (Salgueiro et al., 2004; Costa, Salgueiro, Cataldi García y Lage, 2005) real implementada en los STI tienen solapamiento de funcionalidades y por lo tanto los módulos no son independientes. Esto se debe a que muchos de los conocimientos particulares del dominio (pertenecientes al módulo de dominio) se encuentran dentro de los módulos del tutor y del estudiante con las consecuentes regiones de superposición entre los módulos. Para evitar esto hay que realizar una definición precisa de las interfaces a fin de diferenciar cada uno de los módulos.

A continuación se describe la funcionalidad de un STI a partir de los cuatro módulos que definen Cataldi y Lage (2009), ya que se define una arquitectura con un módulo más que la arquitectura inicial para trabajar los STI.

En el módulo Tutor se define y aplica una estrategia pedagógica de enseñanza, se integran los objetivos a ser alcanzados y los planes considerados para alcanzarlos. Este módulo selecciona los problemas, monitorea el desempeño, provee asistencia y selecciona el material de aprendizaje para el estudiante. Integra el conocimiento acerca del método de enseñanza, las técnicas didácticas y del dominio a ser enseñado. En este módulo se integran: a) protocolos pedagógicos que pueden estar almacenados en una base de datos, b) planificador de lección: se encarga de organizar los contenidos y c) analizador de perfil: analiza las características del alumno y selecciona la estrategia pedagógica más conveniente.

El módulo estudiante del STI tiene por objetivo realizar el diagnóstico cognitivo del alumno, y el modelado del mismo para una adecuada retroalimentación del sistema. Para el módulo estudiante se han planteado de acuerdo a los objetivos que se requieran analizar como los aspectos de motivación, su nivel cognitivo de del estudiante en la materia, su estilo de aprendizaje, su nivel psico-sociológico entre otros: a) estilos de aprendizaje: está compuesto por una base de datos con los estilos de aprendizajes disponibles en el sistema, los métodos de selección de estilos y las características de cada uno de ellos. b) estado de conocimientos: contiene el mapa de conocimientos obtenido inicialmente a partir del módulo del dominio y que el actualizador de conocimientos irá modificando progresivamente a través de los resultados obtenidos en las evaluaciones efectuadas por el módulo del tutor quien le enviará dichos resultados procesados. A grandes rasgos, el problema del modelado del alumno se puede dividir en dos partes: a) la selección de una estructura de datos (en el sentido de variables, enlaces y parámetros) (Conejo, Millán, Pérez de la Cruz y Trella, 2001) y b) la elección de un procedimiento para efectuar el diagnóstico del estado actual del estudiante.

El módulo dominio tiene el objetivo global de almacenar todos los conocimientos dependientes e independientes del campo de aplicación del STI. El dominio proporciona los conocimientos presentados en forma adecuada para que el alumno pueda adquirir las habilidades y conceptos requeridas, es decir, la capacidad de generar preguntas, explicaciones, respuestas y tareas, y además debe ser capaz de dar respuesta a los problemas y corregir las soluciones presentadas, analizando las diferentes aproximaciones válidas a la solución a través de la intervención del tutor (Conejo et al., 2001). Entre sus elementos básicos están los siguientes: a) parámetros básicos del Sistema: los cuales se almacenan en una base de datos; b) conocimientos: son los contenidos que deben cargarse en el sistema, a través de los conceptos, las preguntas, los ejercicios, los problemas y las relaciones; c) elementos didácticos: imágenes, videos, sonidos, es decir material multimedia que se requiere para facilitarle al alumno apropiarse de conocimiento en la sesión pedagógica.

El módulo de interfaz contiene los mecanismos de representación (imágenes, sonidos, videos, lenguaje oral, etc.) del contenido a mostrar al usuario. Se ha demostrado que el éxito de un software educativo, su calidad y efectividad depende del logro de comunicación en su interacción con el usuario. Moreno y Mayer (2000) demuestran que principios de diseño multimedia como herramienta que potencializa el aprendizaje en sistemas de aprendizaje en el campo de la psicología cognitiva.

Los módulos anteriores serán los que generarán la base estructural de la metodología y sus correspondientes elementos se integran para dar paso a la arquitectura planteada. En la Figura 7 se muestran los cuatro módulos integrados en la arquitectura del Sistema Tutor, donde se ubican las tareas que se definirán en el proceso de la metodología.

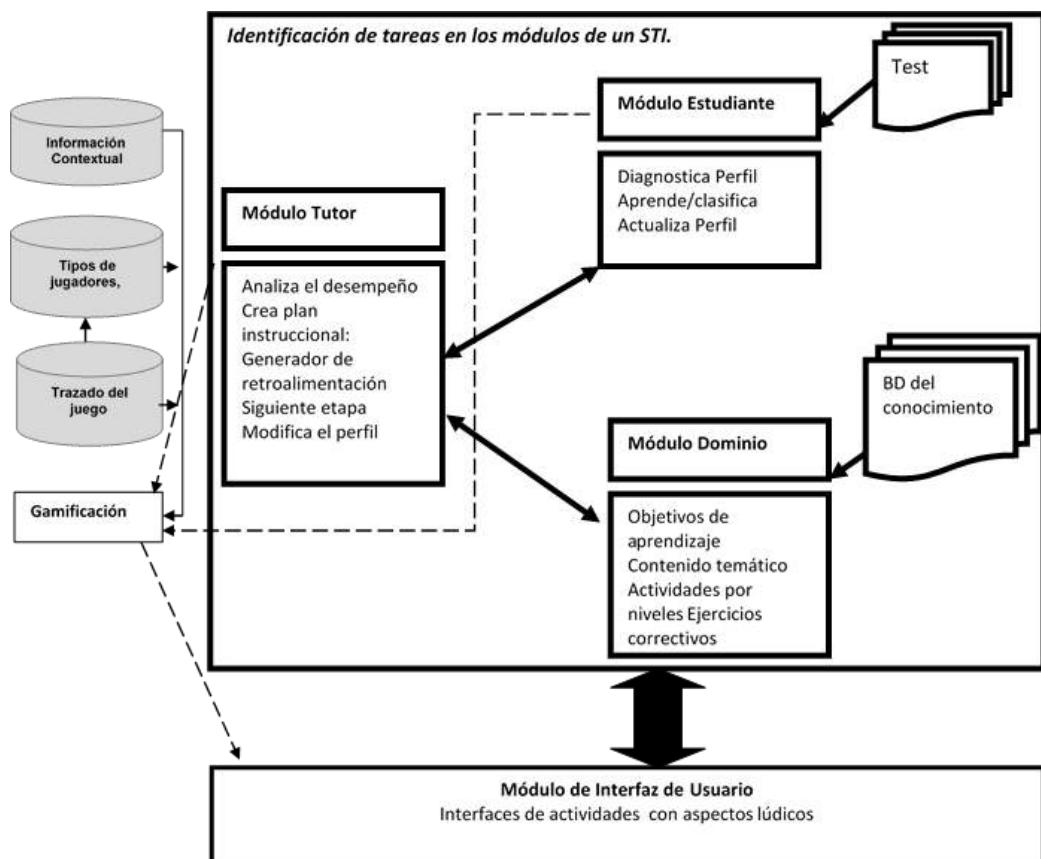


Figura 7. Tareas propuestas en los modelos de un STI.  
Fuente: Adaptado de (Carbonell, 1970 ;González et al., 2016)

### 3.8 Disertación

La metodología para el desarrollo de habilidades algorítmicas se trabaja a partir de la inferencia del perfil de estudiante, donde este se integra en una categoría según su estilo cognitivo y el conocimiento previo (Moroni y Señas, 2005). El ST generará un conjunto de estrategias didácticas que personalizarán el proceso de aprendizaje de acuerdo a las necesidades particulares de cada estudiante, se espera mejorar su aprendizaje y por supuesto su rendimiento académico evitando la deserción escolar (Rosanigo y Paur, 2006).

Los requerimientos esenciales de un ST debe contemplas las evaluaciones, detectar errores, dar sugerencias sobre los ejemplos, plantear en algún momento recomendaciones al estudiante y sobre todo estar constantemente actualizando el modelo de aprendizaje, como lo haría un tutor/docente humano (Ovalle y Jiménez, 2004). El sistema debe ser capaz de comportarse como un experto, tanto en el dominio de conocimiento que enseña (mostrando al alumno cómo aplicar dicho conocimiento), como en el dominio pedagógico (donde es capaz de diagnosticar la situación en la que se encuentra el estudiante y de acuerdo a ello ofrecer una acción o solución que le permita progresar en el aprendizaje) (Carbonell, 1970).

En la Figura 7, se muestran los elementos a incluir los módulos propuestos para el STI. Se tiene contemplado integrar la metodología propuesta en el diseño de los módulos del STI para permitir dar seguimiento individual y gradual, mejorando las habilidades del estudiante en la resolución de problemas generando un aprendizaje significativo que es el objetivo primordial del aprendizaje para que pueda resolver problemas cada vez más complejos de acuerdo a su nivel de avance. Para la elección de los ejercicios de acuerdo al nivel o avance se genera una categorización para que el sistema aprenda de los diferentes comportamientos de entrada y que se trabajarán con técnicas de clasificación ya definidos en Inteligencia Artificial (IA).

Al integrar la estrategia didáctica basada en mecanismos de gamificación en el Sistema Tutor se espera que el estudiante se motive al estar observando su

avance cognitivo y práctico en el proceso de recompensas y retroalimentación para dar solución a los problemas. Un nuevo elemento se ha integrado en la arquitectura es el módulo de interfaz que forma parte del STI para dar una interacción efectiva que aporte en el aprendizaje de los estudiantes. En éste módulo se integrarán principios del diseño, implementación y evaluación de sistemas computacionales interactivos para su utilización por seres humanos (HCI: Human Computer Interaction), y los aspectos de gamificación que implican criterios de usabilidad, es decir con alto grado de facilidad en el uso del sistema interactivo. (Cataldi y Lage, 2009). Algunos elementos básicos de un sistema de gamificación personalizada para la educación propuestos por (González, Toledo y Muñoz, 2016) son: tipos de estudiante-jugador, atributos estáticos y dinámicos del usuario, seguimiento de actividades, comportamiento observable del usuario, factores determinantes del comportamiento (González et al., 2016), de los cuales sólo omitiremos los factores del comportamiento, por el alcance de nuestro trabajo.

### 3.9 Conclusiones

De acuerdo a las teorías mencionadas, cada una de ellas aporta en el proceso de enseñanza aprendizaje en estudiantes novatos, y el análisis de las habilidades algorítmicas al proponer la evaluación con base a la Taxonomía de Bloom en sus dos dimensiones tanto de resolución de problemas, como el nivel de abstracción, y permitiendo generar los tres niveles de estudiantes: novato, aprendiz y experto. Otro aspecto importante que se considera es el estilo de aprendizaje para caracterizar al estudiante con el fin de proporcionar los contenido en el formato indicado de acuerdo al análisis del conocimiento previo y estilo de aprendizaje en el estudiante después de aplicar los test.

Cada uno de los módulos y los elementos considerados se integra en el modelo de tutor y de interfaz aspectos de gamificación, el modelo de Polya para la resolución de problemas y los ejemplos por completar como elementos clave para alcanza medir las dos dimensiones planteada y dar seguimiento en la instrucción

personalizada al ir dosificando y aumentando el nivel de complejidad de los ejemplos, ejercicios y prácticas propuestas durante las sesiones.

Se espera describir las etapas de la metodología y los métodos que acompañaran de forma más amplia en cada una de las etapas, es posible generar aún el test para evaluar la metodología por parte de expertos y analizar los resultados para concretar la propuesta metodología.

## **4. Metodología de diseño de Sistema Tutor para apoyar habilidades Algorítmicas**

Existen diversas metodologías y diseños de Sistemas Tutores, sin embargo su desarrollo e implementación continúa siendo compleja. Uno de los vacíos que se encontró en la literatura es el estudio cualitativo sobre las bases pedagógicas que impliquen el aprendizaje en el dominio dado (Rodriguez, Castilllo y Lira, 2013; Sierra et al., 2004; Tirado, Mancilla y Sánchez, 2009). La forma de interrelacionar las teorías en el proceso de implementación es una de las vulnerabilidades que puede presentar y limitar el sistema.

A continuación se presenta la metodología con los elementos para diseñar un Sistema Tutor (ST) enfocada en las habilidades algorítmicas para estudiantes de nuevo ingreso a la Facultad de Ciencias de la Computación. La integración de la estrategia didáctica se basó en dos dimensiones: las habilidades algorítmicas y el estilo de aprendizaje. Con base en la taxonomía de Bloom (Bloom y Engelhart, 1969) se consideraron las categorías de: Recordar, Describir, Analizar, Aplicar y Evaluar para determinar el nivel cognitivo del estudiante en su dominio de las habilidades algorítmicas. Para realizar el estudio se trabajó con estudiantes de nuevo ingreso en la Facultad de Ciencias de la Computación y que estaban cursando la materia de Metodología de la programación.

En la Figura 8 se muestran las relaciones que hay a partir de los actores importantes y las tareas que realizan: estudiante y profesor.

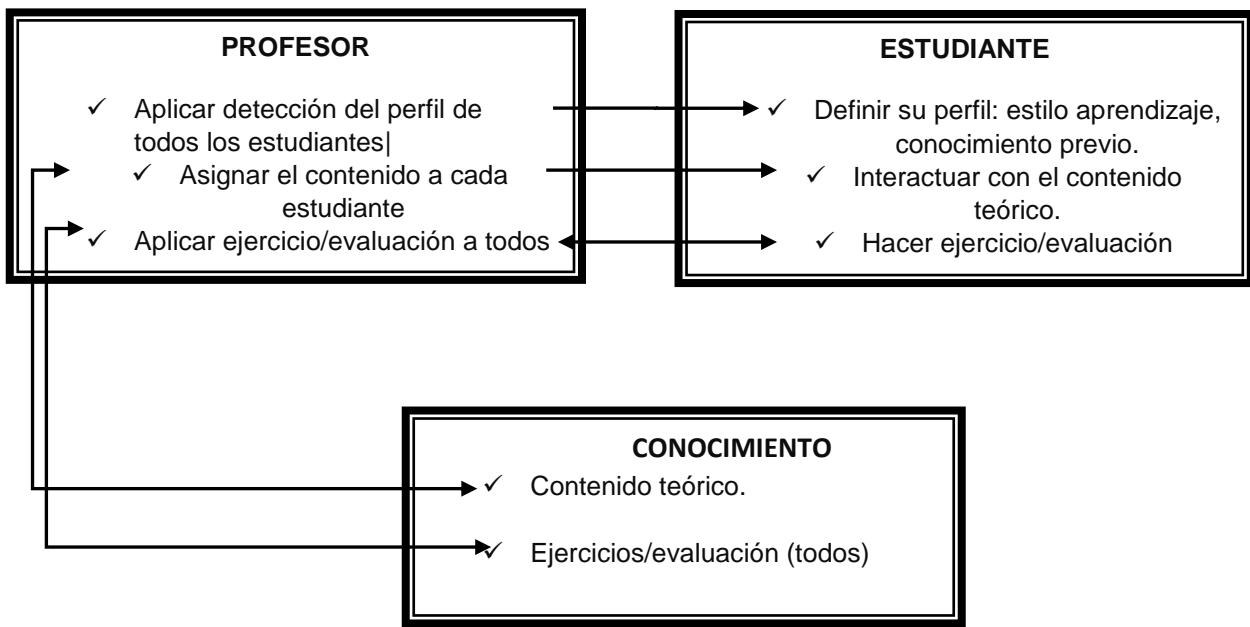


Figura 8: Esquema de tareas para los actores.

Fuente: Elaboración propia.

El objetivo de la investigación es: diseñar una metodología de Sistema Tutor enfocada en las habilidades algorítmicas, orientado a estudiantes de nuevo ingreso a la Facultad de Ciencias de la Computación.

Para explicar el objetivo, el alcance y la aportación se contestan algunas preguntas:

- ¿Cómo dar respuesta al objetivo?
  - Plantear la metodología sustentada en la literatura para el diseño de sistemas tutores y las teorías pedagógicas que aportan en la resolución de problemas y apoyen el desarrollo de habilidades algorítmicas.
- ¿Qué es lo que se necesita adaptar para mejorar la disposición de aprender de los estudiantes?
  - La forma de presentar los contenidos con base a su estilo de aprendizaje y sus habilidades algorítmicas

- ¿Qué se adapta del modelo del sistema tutor?
  - La estrategia didáctica que integra los ejemplos y problemas por completar, integrando técnicas de gamificación para apoyar el proceso de aprendizaje.

La propuesta se basa en los elementos para la caracterización del estudiante con base en el estilo de aprendizaje y las habilidades algorítmicas integrados al modelo de estudiante, además de la estrategia didáctica que se integra en el modelo de tutor, son el aporte de la investigación que se detallan en la metodología para el apoyo de habilidades algorítmicas.

En la Figura 9 se definen las etapas de la metodología y se indican los elementos que se necesitan y resultan en cada etapa denominados como *entrada* y *salida*, posteriormente se darán las secciones correspondientes con las teorías y modelos propuestos que respaldan la estrategia integrada. Finalmente se requerirá de su valoración para evaluar la metodología.

**Etapas de la metodología para desarrollar un Sistema tutor en apoyo de las habilidades algorítmicas**

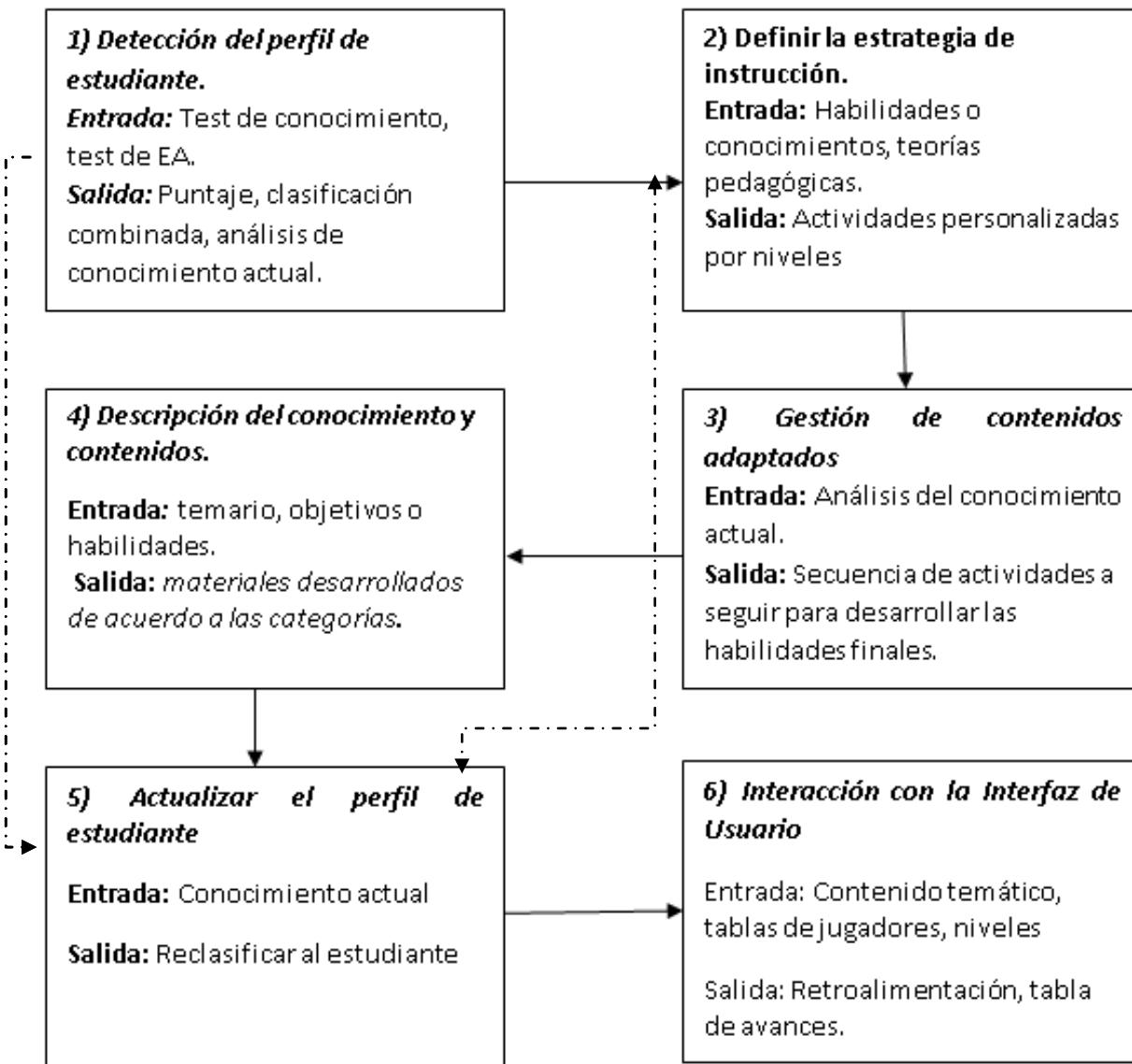


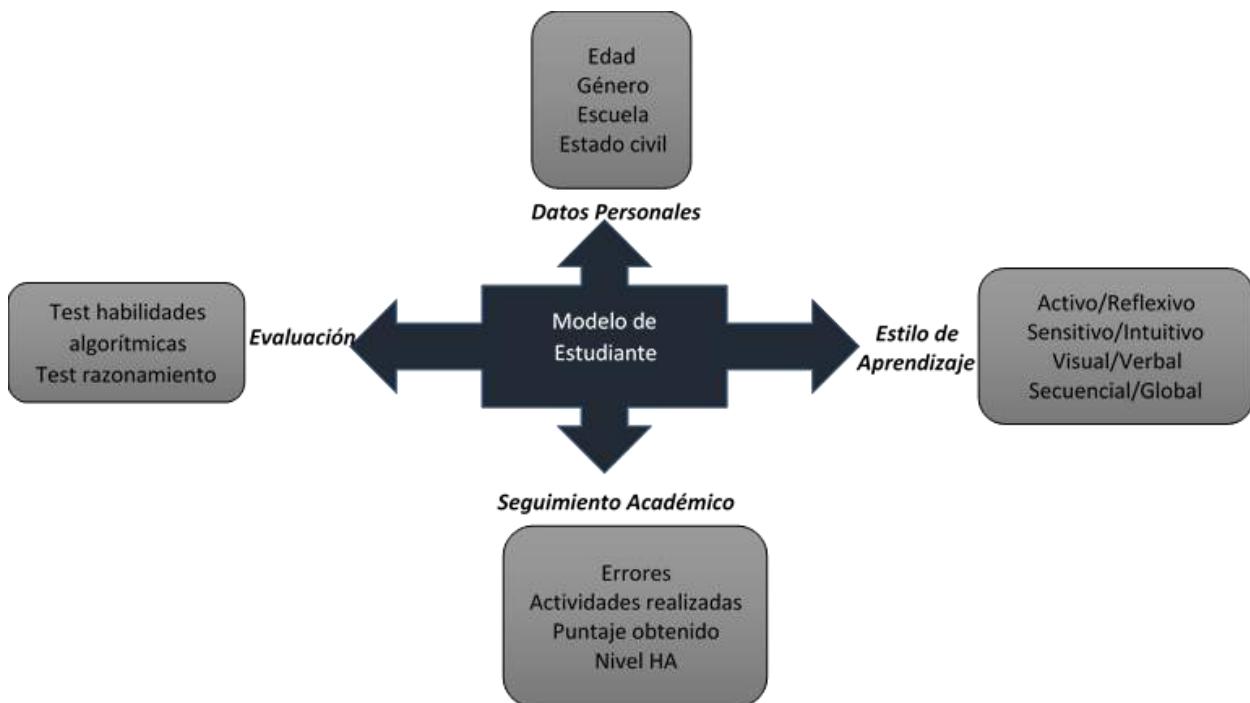
Figura 9: Pasos de la metodología de desarrollo de Sistema Tutor en apoyo de las habilidades algorítmicas.  
 Fuente: Elaboración propia.

En las siguientes secciones se describen cada uno de los pasos con sus objetivos, lo que hace, cómo funcionan, qué implementan y cuáles son los resultados que se espera en cada etapa de la metodología.

#### 4.1 Detección del perfil de estudiante.

El objetivo es definir las características para saber la disposición del estudiante para el aprendizaje en el área de conocimiento. De acuerdo a la literatura existen diversos factores que intervienen en el aprendizaje del estudiante, sin embargo para un estudiante del área de Ciencias de la Computación se requiere que dominen aspectos de lógica, abstracción y resolución de problemas dado que se ha detectado problemas relacionados con esas habilidades (Moroni y Señas, 2005; Robins, Rountree y Rountree, 2003; Rosanigo y Paur, 2006), entre otros factores como el socioeconómico, la motivación, sociodemográficos que son aspectos generales que impactan en los estudiantes en su proceso de aprendizaje (González, Duque y Ovalle, 2008). Sin embargo, este trabajo se centra en aspectos de datos personales como referente de información básica, de estilo de aprendizaje debido a que es importante conocer cómo percibe, procesa, organiza y entiende la información. Existen diferentes estudios sobre estilos de aprendizaje (Dunn y Dunn, 1974; Kolb, 1984; Honey-Murford, 1989; Gardner, 1993). El estilo de aprendizaje indexado (Index of Learning Styles, ILS de Felder y Silverman, 1988) con las teorías de Felder y Silverman consideran los rasgos cognitivos, afectivos, fisiológicos y de personalidad; este test es uno de los más aplicados en el área de Ciencia por las cuatro dimensiones que definen la forma en que manejan la información y que se eligió por los rasgos cognitivos que se derivan de la forma en que es percibida la información, dado que en la literatura se encuentra que los estudiantes del área de computación son preferencialmente visuales-verbales y su forma de recibir la información cognitiva es mediante formatos visuales mediante cuadros, diagramas, gráficos, demostraciones, etc. o en formatos verbales mediante sonidos, expresión oral y escrita, fórmulas, símbolos, etc.

Como se muestra en la Figura 10, el modelo de estudiante está conformado por: datos personales, el estilo de aprendizaje, el seguimiento académico en el área de conocimiento y evaluación.



*Figura 10: Elementos del modelo de estudiante.*

Fuente: Elaboración propia.

En este paso se aplican los test necesarios para identificar el nivel de conocimiento en el que se encuentra el estudiante, para definir el perfil del estudiante se hacen los siguientes pasos:

- Identificar el estilo de aprendizaje del estudiante (aplicar test EA).
- Identificar las habilidades de conocimiento que tiene el estudiante (aplicar test PC).
- Crear el perfil de individual, haciendo la combinación de los resultados.

Es importante conocer las habilidades que debe tener el estudiante para resolver problemas y plasmarlos a través de algoritmos: como se menciona antes, se consideró el estilo de aprendizaje preferente del estudiante de acuerdo test de Felder y Silverman, donde se tienen cuatro pares dicotómicos: activo/reflexivo, intuitivo/sensitivo, visual/auditivo, global/secuencial. Para la obtención de las categorías se tiene la combinación de tipos de estudiante de acuerdo a las habilidades algorítmicas novato, aprendiz y experto. De acuerdo a la investigación exploratoria, se destacan algunas preferencias en los estilos de aprendizaje como el visual/verbal, intuitivo/sensitivo. El modelo se ha utilizado en el desarrollo de

sistemas de educación adaptativos tales como Cs388, Tangow, Lsas, Whurle, entre otros (Parvez y Blank, 2004; Stash, Cristea y De Bra, 2004). Otro motivo es que el cuestionario ha sido validado y probado (Felder y Spurlin, 2005; Litzinger et al., 2005; Zywno, 2003), y esto le proporciona un soporte adicional a la investigación realizada.

La combinación de los tipos de estudiantes novato, aprendiz y experto, con los estilos más predominantes en estudiantes de Computación (Ocampo, Guzmán, Camarena y Luna, 2014) lo cual se corroboró con la aplicación de test de Felder y Silverman, donde se obtuvieron los estilos más predominantes encontrados; como perciben la información se considera verbal-sensitivo, visual-sensitivo, verbal-intuitivo, visual-intuitivo (Figura 11) y muestra el modelo de las combinaciones para el perfil de estudiante. De acuerdo al nivel de estudiante novato, aprendiz o experto, será el tipo de actividad a designar, donde se integran diferentes tipos de actividades como son: ejemplo básico, ejercicio avanzado1, ejercicio avanzado2, test y autoevaluación. Dada esta información, se crea el árbol que combina los estilos de aprendizaje y niveles de estudiantes para tener un modelo general de posibles perfiles (Figura 12). Otra opción que se puede implementar es tener un ranking de los estilos de aprendizaje del estudiante y en caso de no tener todos los contenidos disponibles, ir al siguiente nivel del ranking tratando de tener todos los contenidos disponibles.

Para evaluar los conocimientos se utiliza el test de Pensamiento Computacional (Román-González, Pérez-González y Jiménez-Fernández, 2015) que se basa en la taxonomía de Bloom (Bloom y Engelhart, 1969) para poder medir su avance en las habilidades algorítmicas. Esta información se puede guardar en un ranking de preferencias del estudiante, para el caso en que no exista un material diseñado en el estilo de aprendizaje preferente, se tome el siguiente nivel a fin de cumplir con entregar el contenido más adaptado a su estilo de aprendizaje. Los resultados esperados son la combinación del nivel de conocimiento y estilo de aprendizaje.

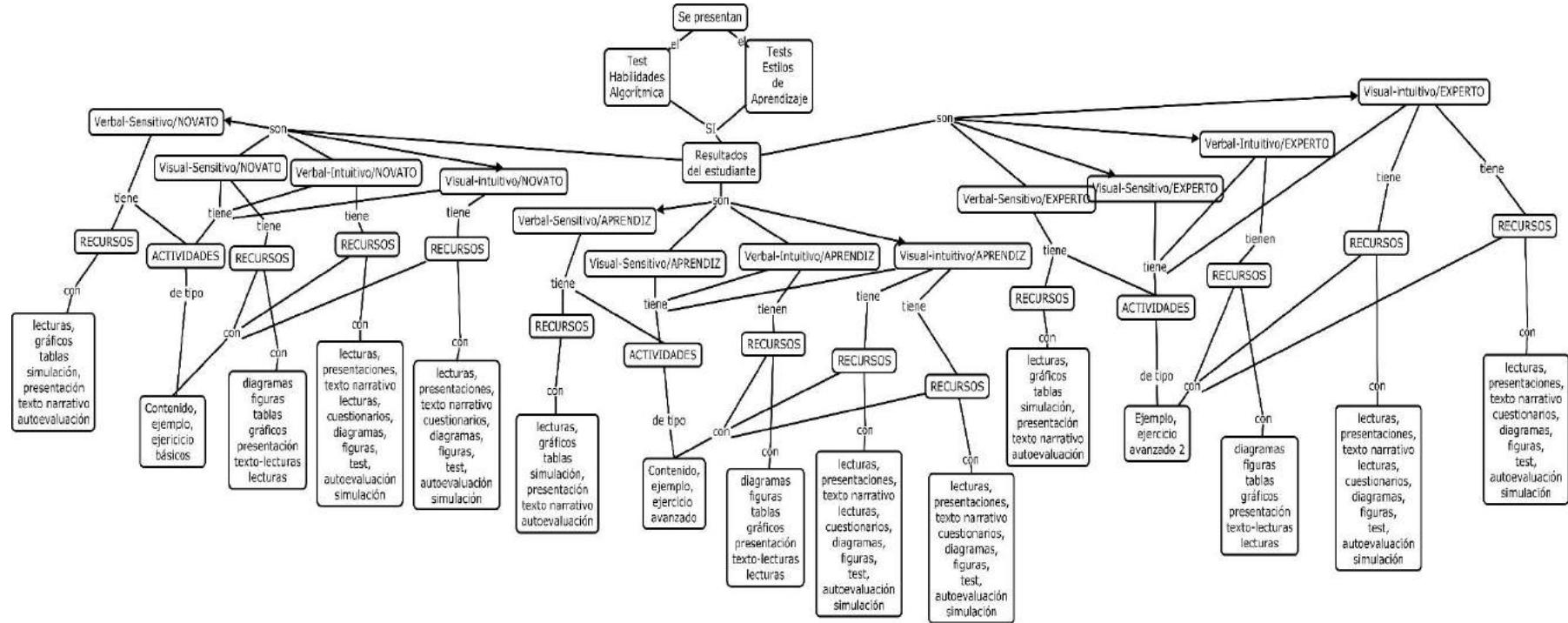


Figura 11 : Modelo de estudiante de acuerdo al estilo de aprendizaje y nivel de conocimiento.

Fuente: Elaboración propia.

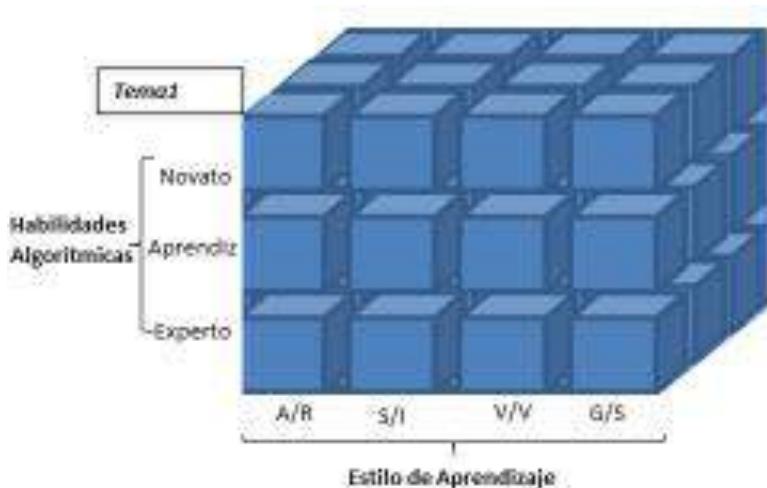


Figura 12. Modelo general de estudiante.

Fuente: Elaboración propia.

## 4.2 Definir la estrategia didáctica

El objetivo de esta fase es establecer la estrategia didáctica que permitirá apoyar el aprendizaje en el área de conocimiento específico. Los pasos a seguir son:

- Definir las habilidades a desarrollar.
- Determinar las teorías pedagógicas que apoyaran en el área de dominio.
- Crear las estrategias a proporcionar de las actividades.
- Con base a ejemplos resueltos y técnicas de gamificación.

De acuerdo a la literatura el estudiante debe ser capaz de identificar las acciones a seguir mediante recursos y procedimientos para generar una autonomía de aprendizaje (Danilov y Skatin, 1985; Pimienta, 2012). Se integran varias teorías que pueden definir una estrategia didáctica en la generación de las actividades, ejemplos y ejercicios que se llevarán a cabo durante el proceso de enseñanza-aprendizaje.

Las teorías integradas para desarrollar las habilidades algorítmicas son: el *modelo de Polya* donde la mayoría de problemas tienen algunos elementos en común: un estado inicial; una meta, lo que se pretende lograr; un conjunto de recursos, lo que está permitido hacer y/o utilizar; y un dominio, el estado actual de conocimientos y habilidades de quien va a resolverlo (Moursund, 1999). Para la

resolución de problemas algunos psicólogos (Schunk, 1997; Woolfolk, 2001) entre otros) opinan que es posible utilizar con éxito estrategias generales, útiles para resolver problemas en muchas áreas. Algunos de los métodos son ensayo y error, iluminación, heurística, modelo de procesamiento de información, análisis de medios y fines, algoritmos, razonamiento analógico, lluvia de ideas, pensamiento lateral. Casi todos los problemas requieren, que quien los resuelve, los divida en submetas que, cuando son dominadas (por lo regular en orden), llevan a alcanzar el objetivo. La solución de problemas también requiere que se realicen operaciones durante el estado inicial y las submetas, actividades (conductuales, cognoscitivas) que alteran la naturaleza de tales estados (Schunk, 1997). Por lo tanto se utiliza el modelo de Polya (1965) para la resolución de problemas que se aproxima mucho al ciclo utilizado para programar computadores. Además permite el planteamiento y las etapas también son flexibles (Wilson, Fernández y Hadaway, 1993) ya que se puede pasar de una etapa a otra, lo que implementado en un sistema puede aportar para dar retroalimentación si es necesario para aquellos estudiantes aprendices (Wilson, et al., 1993; Guzdial, 2000). La programación facilita un diálogo interior en el cual la retroalimentación constante y el éxito gradual empujan a los alumnos a ir más allá de sus expectativas (Stager, 2003); por lo tanto, el paso previo de creación de algoritmos promueve un desarrollo cognitivo y mejorar su nivel de abstracción como parte de las habilidades algorítmicas.

Varios autores de libros en programación, plantean cuatro etapas para elaborar un procedimiento que realice una tarea específica. Estas etapas concuerdan con las operaciones mentales descritas por Polya para resolver problemas:

1. Analizar el problema (Entender el problema)
2. Diseñar un algoritmo (Trazar un plan)
3. Traducir el algoritmo a un lenguaje de programación (Ejecutar el plan)
4. Depurar el programa (Revisar)

Finalmente, como se puede apreciar, hay una similitud entre las metodologías propuestas para solucionar problemas matemáticos (Clements y Meredith, 1992; Díaz, 1993; Melo, 2001; NAP, 2004) y las cuatro fases para solucionar problemas específicos de áreas diversas, mediante la programación de computadores dada la naturaleza de las habilidades algorítmicas.

Desde el punto de vista educativo, la solución de problemas mediante la programación de computadores posibilita la activación de una amplia variedad de estilos de aprendizaje. Los estudiantes pueden encontrar diversas maneras de abordar problemas y plantear soluciones, al tiempo que desarrollan habilidades para: visualizar caminos de razonamiento divergentes, anticipar errores, y evaluar rápidamente diferentes escenarios mentales (Stager, 2003). Así mismo, se pretende lo que Piaget (1965) denominó “la conquista de la difícil conducta de la reflexión” donde se desarrolla el pensar antes de proceder. Además, demanda de los estudiantes planificar, formular hipótesis y anticipar qué sucederá. Los ejercicios justo tratan de guiar al estudiante mediante un tipo de secuencia de pasos a realizar antes de pasar a una formalización, confirmando nuevamente los pasos que plantea Polya (1965).

*La gamificación* es importante para implementar una estrategia de aprendizaje para desarrollar los aspectos de competencia y procesos cognitivos y los objetivos de aprendizaje (Pimienta, 2012). Las técnicas de gamificación ayudan a generar motivación en los estudiantes y trata de engancharlos para las próximas actividades, a través de puntos, clasificación y su nivel de avance. Existe un enorme aumento en la investigación sobre la gamificación educativa y se argumenta que la educación podría ser un buen ajuste con la gamificación, ya que la gamificación busca atraer a los estudiantes a través de tres áreas cruciales para el compromiso; cognitivas, emocionales y sociales (Lee y Hammer, 2011). En este trabajo se utilizan elementos de gamificación que contribuyen al interés del estudiante en el desarrollo de su proceso cognitivo.

Como resultado se integran en conjunto los principios de las teorías de ejemplos por completar, la teoría de Polya y las técnicas de gamificación son la base para construir las actividades que se ajusten a los elementos que aporten al

proceso cognitivo de los estudiantes de novatos en el área de computación. El diseño de las actividades se muestra en la Figura 13, de forma general que fue la forma en que se aplicó en el taller con estudiantes de primer semestre, donde ellos dieron retroalimentación sobre el mismo. Cada uno de los elementos de la estrategia didáctica se da paralelamente dado que las teorías en que se basa se relacionan y permiten diseñar las actividades de forma que se vea el proceso de avance en cada una de las etapas que va avanzando el estudiante.

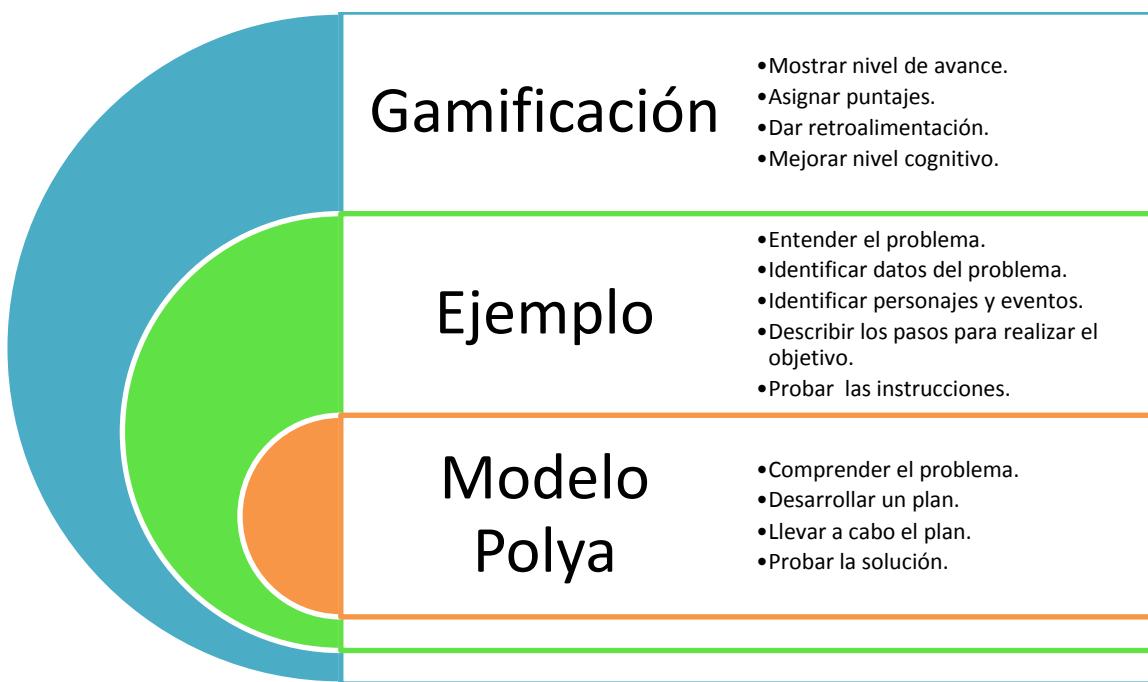


Figura 13: Diseño de una actividad a partir de las teorías.

Fuente: Elaboración propia.

#### 4.3 Gestión y adaptación de contenidos.

El objetivo dirigir al estudiante en su proceso de aprendizaje generando adaptaciones de acuerdo al continuo proceso de aprendizaje y la instrucción adecuada a su estilo de aprendizaje.

Los puntos a llevar acabo son:

- Recibir información actualizada del perfil de estudiante.
- Definir la secuencia de actividades que se le propone a continuar al estudiante.

En esta etapa es dar el plan de contenidos adecuados de acuerdo al perfil del estudiante, se debe definir los contenidos propuestos para llegar al siguiente nivel de forma continua hasta alcanzar el objetivo final. Por lo tanto, lo que se hace en esta etapa es: recurrir a la base de datos de contenidos que definen el contenido a mostrar de acuerdo al perfil del estudiante, además comparar el árbol de conocimiento individual (perfil de estudiante) con el árbol de conocimiento general, y el subárbol resultante dará el camino y contenidos a seguir para llegar alcanzar los objetivos restantes. Es importante mencionar que este proceso se realizará de forma automática y por el momento se plantea hacer superposición de los dos modelos para definir, sin embargo es posible determinar cuál será la mejor técnica a implementar como posible trabajo futuro.

#### 4.4 Descripción del conocimiento y contenidos

En este paso se deben crear los materiales de acuerdo a los objetivos de aprendizaje del área de conocimiento, según las características que se deseen apoyar en el proceso de aprendizaje se tienen que diseñar los recursos multimedia (texto, audio, video, esquemas, figuras, etc.).

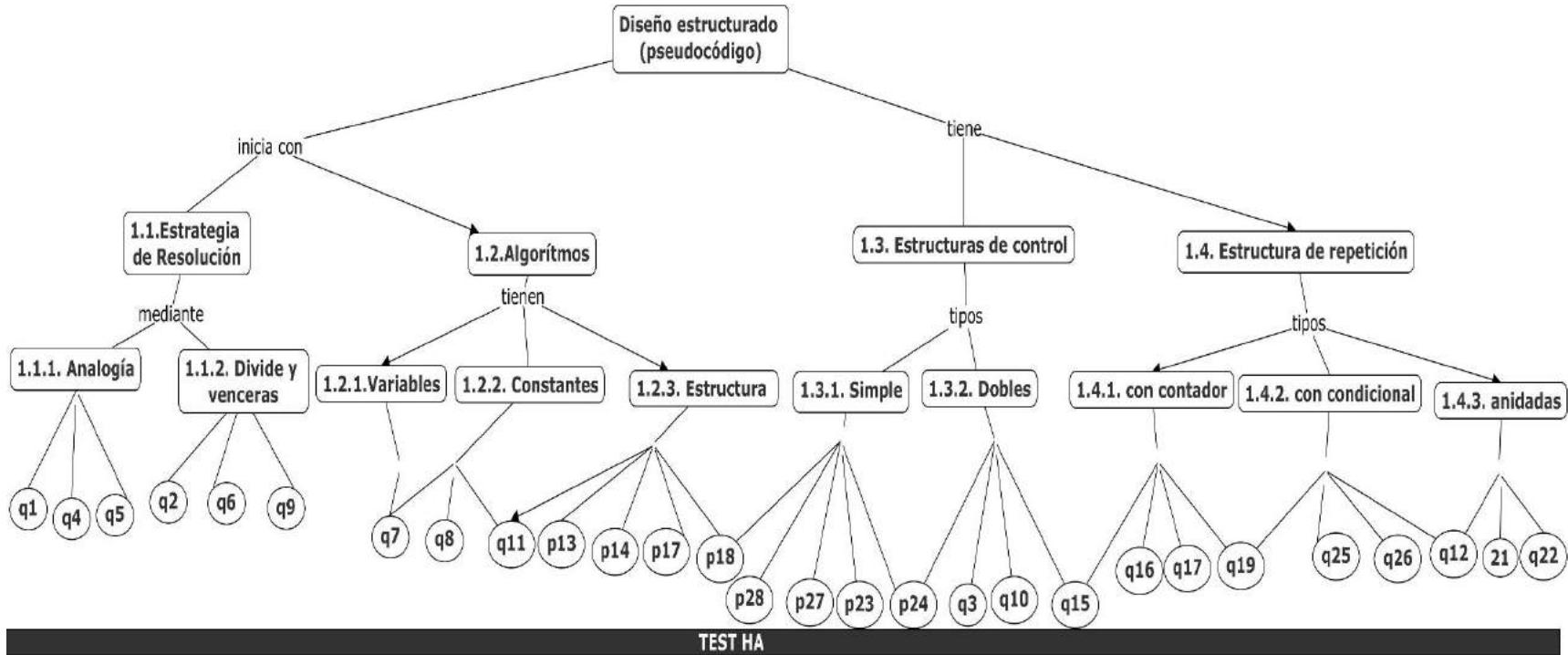
- a) Identificar cuáles son los conocimientos que se deben de adquirir, es decir los temas, subtemas o apartados.
- b) Crear la interconexión de los contenidos (temas, subtemas o apartados).
- c) Evaluar el modelo planteado.

De acuerdo al objetivo de investigación, a continuación se muestra la secuencia de contenidos de los temas integrados en el mapa general de conocimiento que va de lo general a lo particular. Los resultados que se esperan en esta fase es definir las rutas de aprendizaje de acuerdo a los objetivos de cada tema. En la Figura 14 se muestra las ramas y los ítems que permitirán avalar si ya se domina el contenido para mostrar el siguiente contenido, la estructura que se plantea corresponde al contenido de la materia de Metodología de la Programación. De acuerdo a la forma en que está definido el aprendizaje de los fundamentos de programación se debe de aprender conceptos básicos y entender las estructuras de control y repetición para crear un algoritmo funcional. El

teorema de la programación estructurada propuesto por Böhm y Jacopini (1996) dice que “todo programa propio se puede escribir utilizando únicamente las estructuras de control secuencial, condicional e iterativa” (p. 366) a partir de esto se tienen 3 tipos de sentencias.

- a) Una sentencia de acción o de secuencia, que no modifica el flujo de ejecución del programa.
- b) Una sentencia condicional (Si-entonces) que según sea el valor de comparación o relación se ejecutará una u otra alternativa.
- c) Una sentencia de iteración que permita efectuar una operación cero o más veces mientras una expresión lógica sea verdadera (hacer-mientras).

Según el teorema anterior, el objetivo de aprender a estructurar un algoritmo básico para resolver un problema se alcanzará a través del contenido propuesto en el modelo de conceptual de temas (conocimiento). Cabe mencionar que los contenidos están apegados al contenido de la Materia de Metodología de la Programación.



*Figura 14:* Estructura temática de acuerdo a la materia de Metodología de la programación.  
Fuente: Elaboración propia.

El modelo definido al final se basa en la organización de la Figura 4 que está en la sección 2.3, donde se puede observar cómo estarán disponibles los materiales para cada subtema normalmente existirá el concepto, explicación, ejemplo, práctica y test, así como su correspondiente retroalimentación, y de acuerdo siempre al nivel de importancia y de dificultad contemplando el grado de complejidad del subtema que domine el estudiante. La ventaja de tener una estructura de contenidos de acuerdo al tipo de instrucción es poder dar al estudiante un conjunto de elementos progresivamente con más dificultad, desde la base conceptual hasta la aplicación. Al finalizar la evaluación de cada contenido, se aporta una retroalimentación en relación a su avance, así mismo, cada contenido va ligado al estilo de aprendizaje que convenga en los contenidos acorde a su preferencia. Además, en el caso de las características del tema se tienen dos elementos: la importancia y la dificultad del tema, es decir los temas tienen una granularidad que definirá como se dará el tema, subtema o tópico, por otra parte es importante el nivel de dificultad el cual se apega al nivel de conocimiento que se tenga en el área de dominio, de esta forma se puede crear en base a los niveles propuestos para evaluar las habilidades algorítmicas que son novato, aprendiz y experto. Los niveles y habilidades a evaluar se muestran en la Tabla 5: así como los ítems en cada categoría, para determinar en qué momento se pasa de un nivel a otro. El modelo de dominio que define las categorías donde se muestra el grupo de ítems que se deben de cumplir para que un estudiante pase del nivel de novato a aprendiz y finalmente al nivel experto como se ven en la Figura 15.

Tabla 5:

*Habilidades y tipos de estudiantes con base a la taxonomía de Bloom.*

Tipos	Taxonomía	Habilidades
Novato	Recordar	Reconocer la sintaxis de una instrucción.
	Comprender	Reconocer la semántica de una instrucción.
	Aplicar	Interpretar que cambios ocurren al modificar una instrucción.
	Analizar	Traducir un algoritmo dado a pseudocódigo o diagrama de flujo.
Aprendiz	Aplicar	Utilizar estructuras con un fin determinado y claro (decisiones, ciclos).
	Analizar	Modificar un algoritmo.
	Probar	Predecir los posibles efectos al ejecutar un algoritmo línea por línea.
Experto	Analizar	Detectar posibles errores en un algoritmo.
	Probar	Evaluar requerimientos a nivel de usuario, sistema, organización y desarrollar una solución algorítmica coherente que los tome en cuenta.
	Crear	Comparar soluciones algorítmicas que resuelven un mismo problema.
	Probar	Verificar un conjunto de instrucciones a través de pruebas de escritorio)
	Crear	Generar un algoritmo completo a partir de las especificaciones de un problema.
		Generalizar un algoritmo.

Fuente: Elaboración propia

De acuerdo al Test de habilidades algorítmicas, se hizo una clasificación de los tipos de estudiante como se determina en la Tabla 5: sobre su nivel de expertise y sobre la taxonomía de Bloom, que ayuda a reconocer las habilidades en las cuales se tiene necesidad de apoyar.

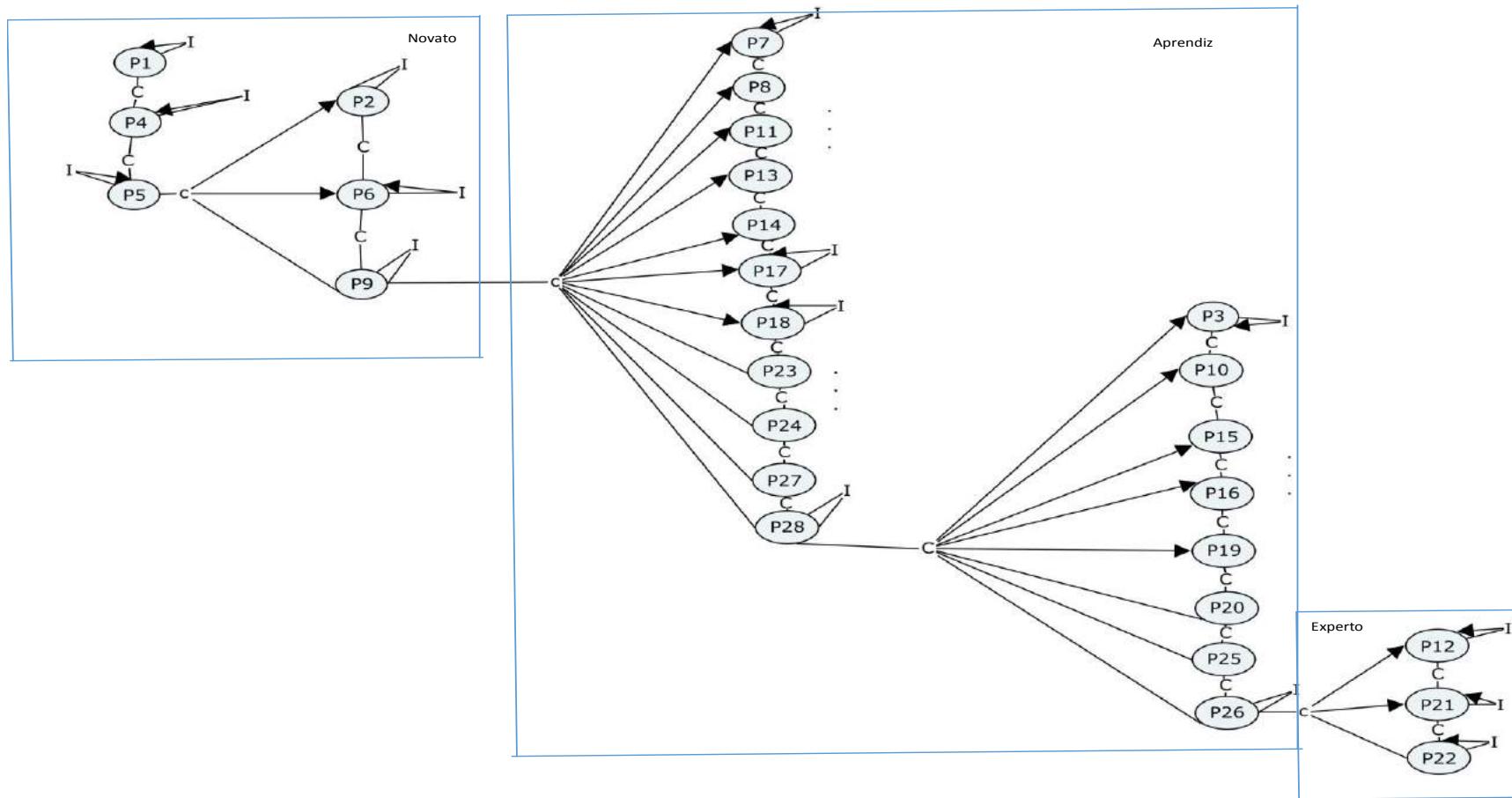


Figura 15: Modelo de dominio sobre la evaluación de test de pensamiento computacional.

Fuente: Elaboración propia.

## 4.5 Actualizar el perfil de estudiante

El objetivo de este paso es tener el nuevo nivel del estudiante en todo momento después de haber interactuado con el sistema.

- a) Obtener el modelo de conocimiento general
- b) Comparar el modelo de conocimiento general con el árbol del perfil del estudiante

Este paso tiene por objetivo cambiar el estado del perfil de estudiante cada que exista un avance en su nivel conocimiento.

Uno de los métodos a aplicarse es tener el árbol de conocimiento general y comparar con el árbol individual, el resultado que se espera es ajustar el árbol individual de acuerdo al avance o mejora del conocimiento.

El resultado de esta etapa es la actualización del árbol o perfil de estudiante en todo momento y detectar su avance.

6) Interacción con la Interfaz de Usuario: El objetivo de este paso es determinar los elementos que se presentarán en la Interfaz de Usuario (IU) y la medida en que ellos abonarán en el análisis de su proceso de aprendizaje.

- a) Plantear las técnicas de gamificación a integrar en la IU.
- b) Definir los elementos de interacción que aporten sobre los errores durante la solución del problema (veces que acceda a un ejercicio, veces que responde incorrectamente a un ejercicio, etc.)
- c) Generar las actividades prediseñadas

A partir de los principios de gamificación, y después elegir las técnicas de retroalimentación, niveles, y puntos (Kapp, 2012), se gamificaron las actividades diseñadas en el paso 3 de la metodología, se crearon prototipos a fin de verificar si los elementos integrados son aceptados y sobre todo generan confianza al acompañamiento del estudiante (Zulma Cataldi y Lage, 2009).

Pérez y Pozo (2009) hacen una afirmación: “nuestro recuerdo y aprendizaje serán el producto de la interacción entre materiales y los conocimientos previos que activamos” (p. 34), definen como otra de las condiciones a la creación de actividades que, basadas en conocimientos previos, propicien los nuevos aprendizajes, es decir “para que los [estudiantes] comprendan no basta con presentarles la información [...] es preciso diseñar actividades o tareas que hagan más probable esa actividad cognitiva...” (Pérez y Pozo, 2009, p.33). Por lo tanto, se pretende identificar los elementos con los que interactúa el estudiante y verificar la posible interacción que realiza el estudiante con el prototipo y recolectar la información sobre los errores o conflictos que se presenten a través de pruebas de usabilidad para determinar el grado de entendimiento y de facilidad en el uso de las interfaces propuestas a través del test como el Cuestionario de Usabilidad en Sistemas Informáticos (CSUQ, por sus siglas en inglés) adaptado y validado al español (Hedlefs, de la Garza, Sánchez y Garza, 2016). Durante la aplicación se valoraron aspectos que no se miden en el cuestionario como la forma de resolver el problema y como se veían interesados en resolver el ejercicio correctamente. A continuación, en la Figura 16 se presentan los prototipos evaluados por alumnos.

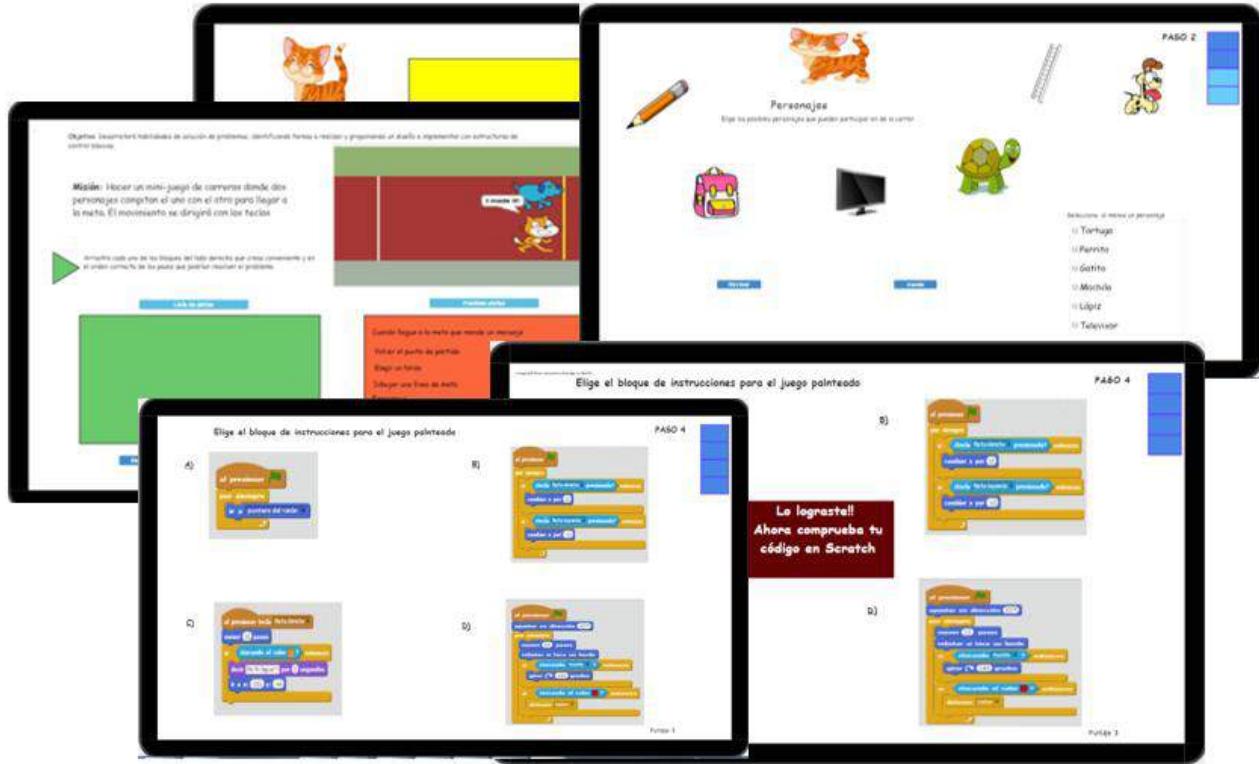


Figura 16: Prototipos de Interfaz de Usuario para las actividades.

Fuente: Elaboración propia.

Los seis pasos descritos en este capítulo integran la metodología que ayudará al diseño e implementación del Sistema Tutor que fomente el desarrollo de las habilidades algorítmicas. Las teorías que se integran se respaldan en la literatura y se contrasta con la percepción y resultados (Capítulo 6) que se obtuvieron después de la aplicación del taller, obteniendo mejoras en las habilidades de abstraer y aplicar. Se sugiere continuar con la implementación como siguiente paso de la investigación.

## **5. Diseño Metodológico**

El presente trabajo se basa en el paradigma de la investigación cualitativa que se caracteriza por comprender el significado de la variedad de realidades por parte de los sujetos que las viven e interpretan. La necesidad de indagar las realidades educativas desde su condición natural, podría suponer que éstas son fieles reproducciones de contextos sociales, independientemente del currículo de las instituciones educativas y de los procesos de enseñanza-aprendizaje, lo cual es un argumento para utilizar los métodos cualitativos. Rodríguez y Valldeoriola (2009) exponen que para comprender las realidades educativas, particularmente volcadas al mundo virtual es conveniente utilizar algunos métodos que son más adecuados: etnografía, estudio de caso, teoría fundamentada, investigación-acción e investigación basada en el diseño. Dada la naturaleza del objetivo, se condujo un estudio cualitativo bajo el enfoque basado en el diseño.

### **5.1 Método**

Dada la naturaleza del objetivo, se condujo un estudio cualitativo bajo el enfoque basado en el diseño. La investigación se fundamenta en la Investigación Basada en el diseño (IBD) (ver Figura 7 ) que integran los pasos del proceso de investigación; esta metodología se centra en planteamientos metodológicos que vincularan investigación, diseño educativo e innovación (Brown y Collins, 1992). La IBD se centra en el diseño y exploración de todo tipo de innovaciones educativas, a nivel didáctico y organizativo, considerando también posibles artefactos (ej. software) como núcleos de esas innovaciones, y contribuyendo a una mejor comprensión de la naturaleza y condiciones del aprendizaje (Bell, 2004). De acuerdo a Kelly (2003) se considera a la IBD como un paradigma emergente en la investigación que explica cómo, cuándo y por qué las innovaciones educativas funcionan en la práctica y que se apega al objetivo general que se persigue: diseñar la metodología de Sistema Tutor enfocada en las habilidades algorítmicas. Este carácter conlleva una dialéctica constante entre la reproducción y cambio de los participantes y sus realidades escolarizadas. La educación, el entrenamiento y el aprendizaje suponen fundamentalmente la

creación de las condiciones más adecuadas para lograr el desarrollo individual, en este trabajo son importantes los objetivos instruccionales que definen las condiciones que se diseñen de acuerdo al campo en que son aplicadas.

Se tienen cuatro características de los estudios de diseño de acuerdo a Rinaudo, Chiecher y Donolo (2010):

- La decisión de ubicar la investigación en el contexto natural en que ocurren los fenómenos estudiados.
- El propósito de producir cambios específicos en ese contexto.
- La opción por los enfoques sistémicos es decir estudios que tratan a las variables como interdependientes y transaccionales.
- El carácter cíclico e iterativo de los diseños.

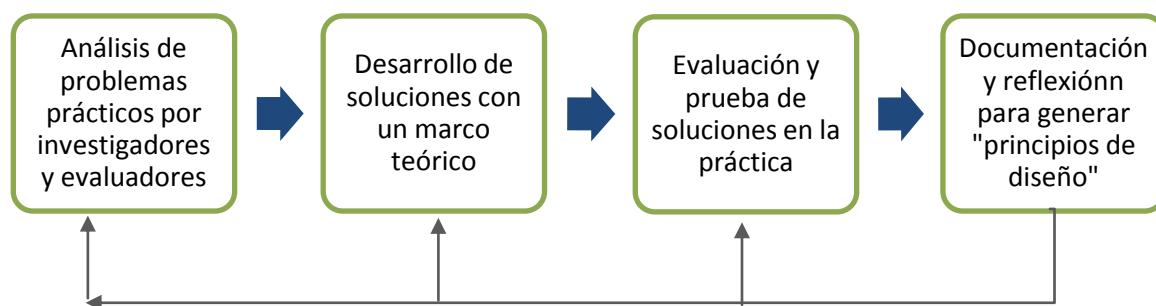


Figura 17. Pasos para la Investigación Basada en el Diseño (IBD).

Fuente: Adaptado de Reeves (2000).

Los diseños de investigaciones basadas en IBD pueden darse en contextos muy variados en su tipología y amplitud. Por su parte Cobb, Confrey, Disessa y Schauble (2003) la caracterizan como iterativa, centrada en procesos, intervencionista, colaborativa, multinivel, orientada a la utilidad, y fundamentada en la teoría.

Cada una de las fases (Figura 18) de la investigación basada en el diseño se trabajó de forma independiente con el fin de interactuar con los estudiantes y docentes/investigadores expertos en áreas de pedagogía, psicología, ciencias de

la computación. Al finalizar se obtiene una retroalimentación del diseño así como generar la validez de expertos y estudiantes.

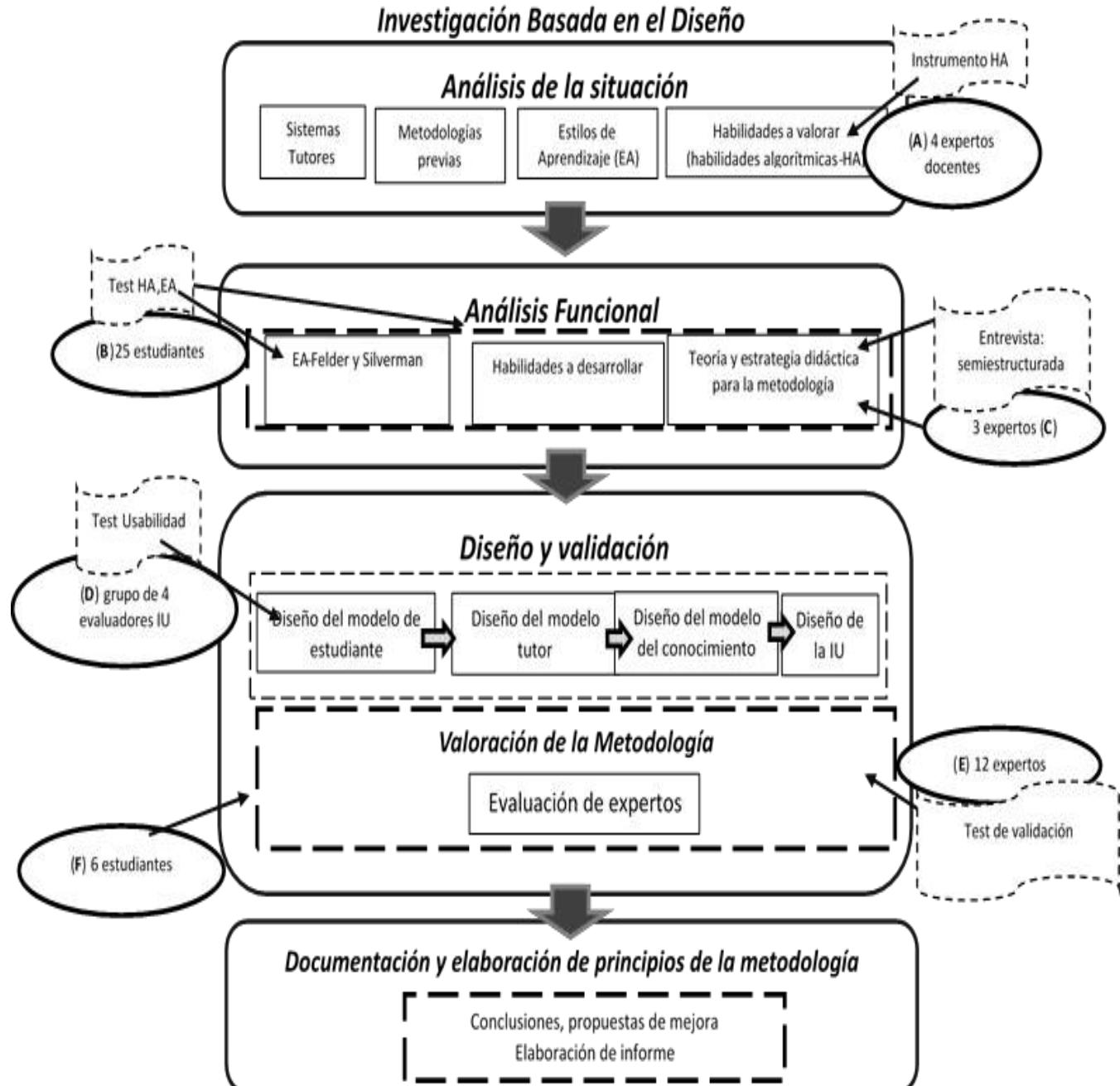


Figura 18. Proceso de la investigación general.  
Fuente: Elaboración propia.

A continuación se describirá el procedimiento que se llevó a cabo en cada una las fases.

## 5.2 IBD Fase 1

### 5.2.1 Análisis de la situación (A)

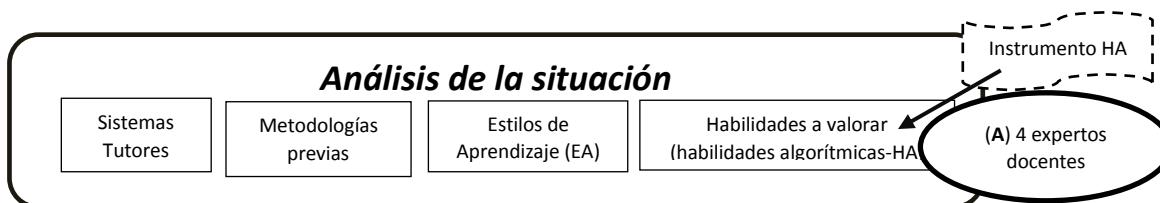


Figura 19 . Análisis de la situación.

Fuente: Elaboración propia.

En esta etapa se revisan las metodologías previas (Anexo C) que existen en el diseño de Sistemas Tutores para delimitar la problemática, con el fin de fundamentar la orientación de la investigación enfocada en las habilidades algorítmicas (Anexo A) de los estudiantes de nuevo ingreso a la FCC.

#### Objetivo

Definir los conocimientos y habilidades algorítmicas a desarrollar con base en la taxonomía de Bloom.

#### Población

Docentes expertos en el área de programación y pedagogía.

#### Participantes

Docentes que han impartido la materia de Metodología de la Programación.

#### Muestra

Se toma una muestra por conveniencia de

- Dos docentes de la FCC del área de programación básica y con Maestría en Ciencias de la Computación

- Dos docentes con Maestría en Educación Superior.

En ambos casos los docentes cuentan con una experiencia de ocho años impartiendo clases en programación básica en la FCC.

#### Técnica e instrumentos

Se utiliza un cuestionario para la validación del instrumento que se va a utilizar en la investigación. La valoración se hace con base a la taxonomía de Bloom (Anexo B) donde se califican los ítems propuestos y la categoría en la que se clasifica a través de la validación de expertos.

#### Procesamiento de datos

Se inicia con el análisis de las metodologías (Figura 19) que se han desarrollado hasta la fecha, en la cual se determinan los huecos que existen en la investigación, en el cuál se genera una tabla comparativa para destacar los huecos en el área de investigación.

Posteriormente se hace una evaluación de expertos (Anexo B) para adaptar el instrumento de Habilidades Algorítmicas (Román-González et al., 2015) con base a la taxonomía de Bloom. Para ello se realizaron los siguientes pasos:

1. Se dio el instrumento impreso y la tabla de categorización de las habilidades a los expertos con base a la taxonomía de Bloom donde colocan los ítems en la categoría que consideran adecuada.
2. Se recopilan los datos de acuerdo a su clasificación para la validez de expertos con las categorías de suficiencia, coherencia, relevancia y claridad.
3. Se concentra la información en una hoja de cálculo y se adaptan los ítems de acuerdo a la coincidencia de la mayoría de expertos.
4. Se realizó una estimación de la medida de acuerdo entre jueces a través del estadístico Kappa y el análisis de preguntas cualitativas que permitió recuperar información para mejorar en la redacción del instrumento y apoyar su validez de contenido.

## Resultados

De acuerdo a los datos obtenidos en la validación de expertos sobre el instrumento, los ítems del instrumento de pensamiento computacional (Román-González et al., 2015) se evaluaron obteniendo una concordancia evaluada con el coeficiente de Kappa en las siguientes categorías: 0.965 en suficiencia, 0.830 en coherencia, en relevancia 0.862 y 0.827 en claridad; por lo tanto, los ítems quedan en el nivel indicado por los docentes, se recomendó hacer mejoras de redacción en algunos ítems o bien adaptar la legibilidad en algunas imágenes. El instrumento adaptado de pensamiento computacional se integra como uno de los elementos para ayudar a caracterizar a estudiantes de nuevo ingreso a la FCC.

Después de realizar los cambios propuestos por los expertos se aplicó el instrumento para verificar su nivel de confiabilidad. La validación de éste instrumento se realizó a través del jueceo de expertos para tener nuestros propios niveles de estudiante novato, aprendiz y experto (Román-González et al., 2015). En la Tabla 6: se muestran las validaciones realizadas al test para evaluar las habilidades algorítmicas que se adapta a los requerimientos de la investigación.

Tabla 6:  
*Validaciones de los instrumentos*

	<b>Test de Pensamiento Computacional</b>
<b>Validado</b>	Si
<b>Pilotaje</b>	Si
<b>Validación de Contenido</b>	Jueceo de expertos
<b>Validación Criterio</b>	Rendimiento Académico (Examen Final)

Fuente: Elaboración propia.

## 5.3 IBD Fase 2

### 5.3.1 Análisis funcional (B)

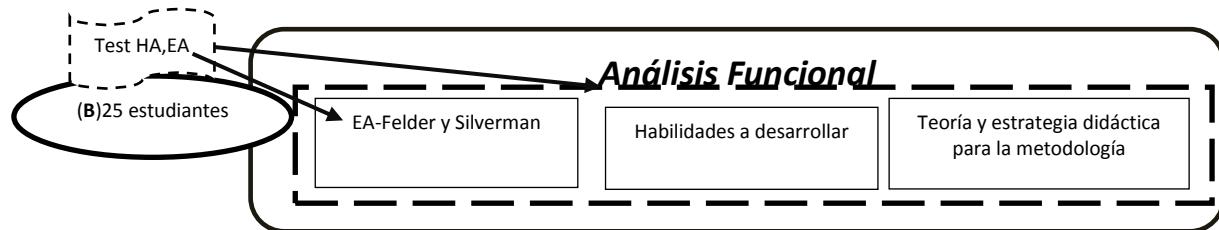


Figura 20. Análisis funcional, subfase B.

Fuente: Elaboración propia.

Después de seleccionar las teorías y el instrumento que medirá las habilidades algorítmicas Román-González et al., 2015), en esta segunda fase B (Figura 20) se selecciona el test de estilo de aprendizaje (Felder y Silverman, 1988) ya validado, e identifica sobre la preferencia en el manejo y procesamiento de información, ambos aspectos ayudan a determinar las características del estudiante.

#### Objetivo

Caracterizar a los estudiantes de nuevo ingreso a computación de la FCC, BUAP de acuerdo a su nivel de: habilidades algorítmicas y estilo de aprendizaje.

#### Población

La principal población son los estudiantes de nuevo ingreso al nivel superior del área de Computación, en particular en la Facultad de Ciencias de la Computación en las carreras de Ingeniería en Ciencias de la Computación y la Licenciatura en Ciencias de la Computación. La Facultad está en un área urbana de la ciudad de Puebla, la cual consta de 5 edificios y cuentan con laboratorios de 35 computadoras con acceso a Internet. Se tiene una población de 500 jóvenes que ingresan anualmente en periodo de otoño, la mayoría tienen entre 18 y 23 años.

#### Participantes

Se trabajó con estudiantes de primer semestre que cursan la materia de Metodología de la Programación los cuales aportan para el proceso de diseño de

la metodología al identificar su caracterización en cuanto al estilo de aprendizaje y sus habilidades algorítmicas, que es la unidad de análisis en esta etapa.

### Muestra

En esta fase de exploración utiliza una muestra por conveniencia de un grupo de 22 estudiantes de nuevo ingreso, el grupo ya está conformado previamente de forma aleatoria por la Secretaría Académica. Dada la naturaleza de la investigación y el contexto en el que se desarrolla, los estudiantes deben cumplir con un conjunto de características:

- Son estudiantes de nuevo ingreso (1º o 2º cuatrimestre).
- Tienen entre 18 y 22 años.
- Cursan la materia de metodología de la programación

Los criterios de exclusión es que cursen materias donde ya utilicen algún lenguaje de programación, que ya sean de cuarto cuatrimestre o posterior, es decir ya tienen desarrolladas las habilidades algorítmicas básicas.

### Técnica e instrumento

Durante la primera fase de exploración de la población se utiliza el cuestionario tipo test con preguntas cerradas de opción múltiple (Román-González et al., 2015) y el test de estilos de aprendizaje (Felder y Silverman, 1988) que contiene respuestas dicotómicas. El instrumento se aplicó a un grupo de estudiantes de nuevo ingreso para diagnosticar las características del grupo en sus dos dimensiones: estilo de aprendizaje y habilidades algorítmicas.

### Procesamiento de datos

La labor realizada por los estudiantes en esta investigación fue contestar a los test que definen sus características de aprendizaje, así como el conocimiento previo en el desarrollo de algoritmos.

Para el análisis diagnóstico se procesó la información en una hoja de cálculo de acuerdo a los datos recolectados en los instrumentos de estilos de

aprendizaje y habilidades algorítmicas aplicados a los estudiantes. Los pasos para el procesamiento son:

1. Se introducen los datos recabados de los test aplicados en línea a través de google forms en una hoja de cálculo y se procede a su análisis.
2. Se utilizan representaciones gráficas para presentar los resultados.
3. Se analiza el comportamiento y se contrasta con la literatura, que contribuye a la justificación de la problemática y la orientación de la investigación.

La caracterización del estudiante servirá de información en el diseño del sistema para determinar la estrategia de aprendizaje y las habilidades que es necesario reforzar en cada estudiante.

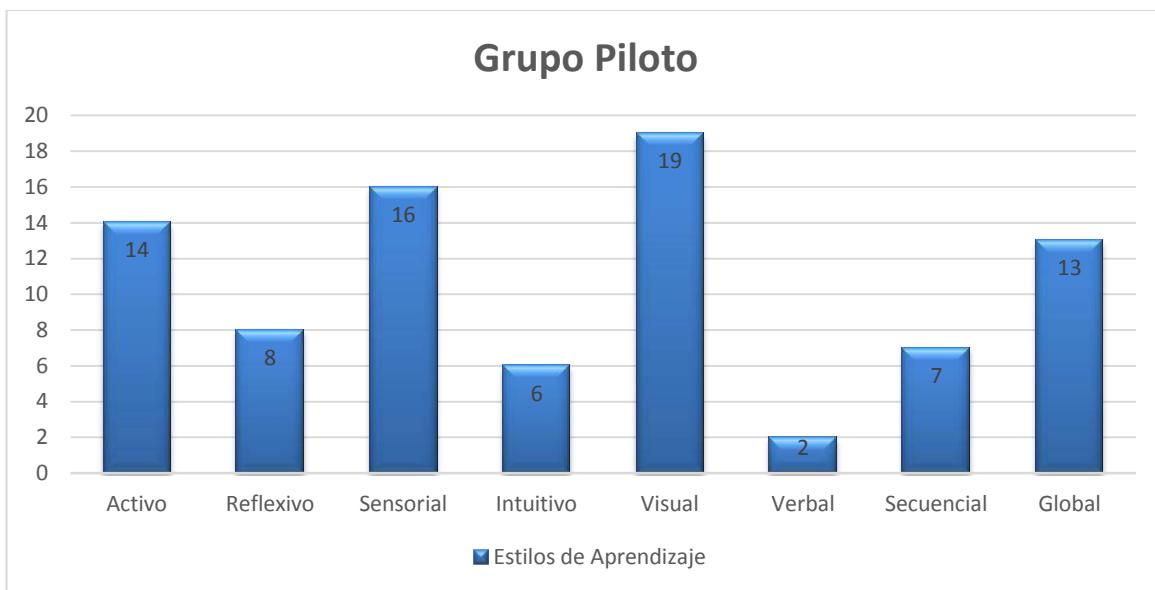
### Resultados

Después de aplicar los dos instrumentos (Anexos D, E): el test de estilos de aprendizaje, el test de Pensamiento Computacional. El test de estilos de aprendizaje está validado y se utilizarán sin ninguna modificación (Anexo D), en el caso del test de Pensamiento Computacional ya fue validado en la fase anterior.

Se trabajó con un grupo de estudiantes de 22 alumnos que ya han cursado la materia de Metodología de la Programación anteriormente, sin embargo se encontraron datos interesantes para la investigación. Se utilizaron 2 sesiones de 45 minutos, para una sesión se aplicó el test de estilos de aprendizaje, y en otra sesión se aplicó el test de pensamiento computacional, los dos instrumentos se aplicaron en el laboratorio a través de un formulario de Google Forms donde se integraron las preguntas de cada test.

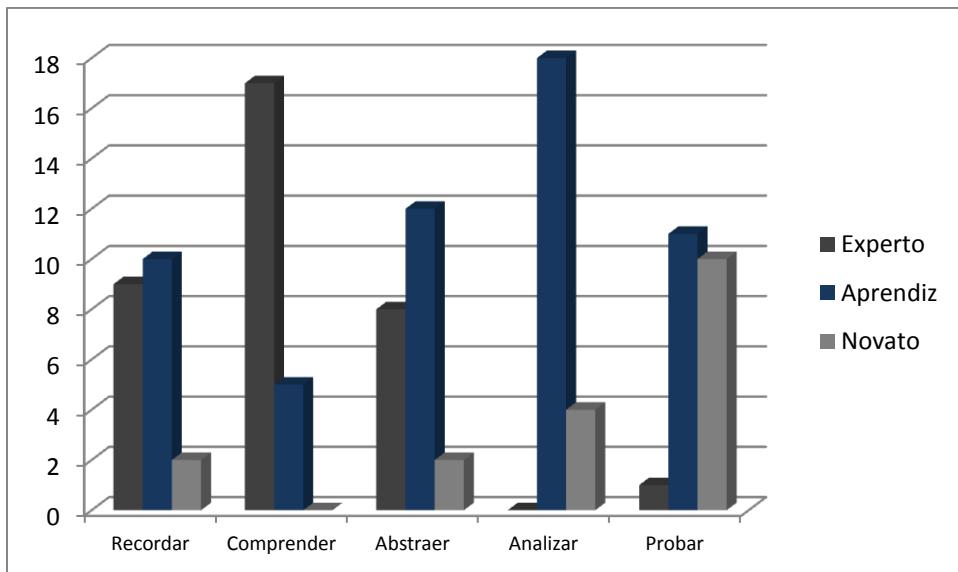
Después de aplicar el instrumento de estilos de aprendizaje (Felder y Silverman, 1988) al grupo de estudiantes, los resultados obtenidos muestran (Gráfica 2) que los estudiantes de nuevo ingreso tienen tendencia a manejar la información de forma: activa, sensorial, visual y global. El resultado de la aplicación del test ayuda a tomar las características de los estilos predominantes

para definirlas como base en la generación del material a integrar en el taller percibiendo como el estudiante procesa, gestiona y recupera la información.



Gráfica 2. Estilos de aprendizaje de estudiantes de nuevo ingreso.

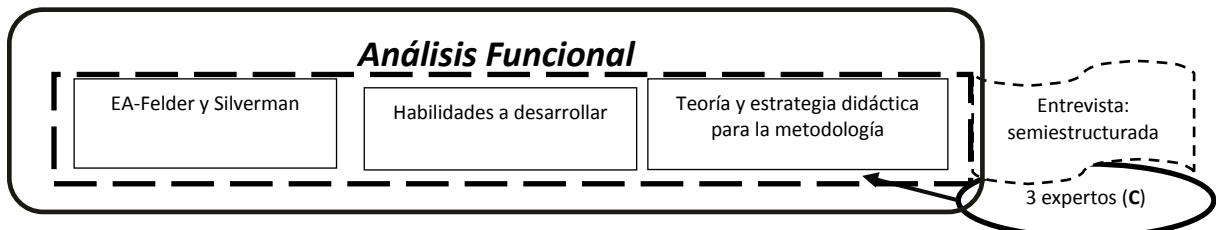
Después de aplicar el instrumento de pensamiento computacional como se puede observar en la Gráfica 3 que las habilidades de recordar, comprender y abstraer los estudiantes tienen mayor porcentaje en el nivel de experto, sin embargo en las dos últimas habilidades de analizar y probar el porcentaje de experto disminuye considerablemente y repunta el aprendiz.



*Gráfica 3.* Nivel de habilidades algorítmicas.

Por lo tanto, se realizó una investigación exploratoria donde se obtiene las características de los estudiantes de nuevo ingreso a la FCC.

### 5.3.2 Análisis funcional (C)



*Figura 21.* Análisis funcional, subfase C.

Fuente: Elaboración propia.

Después de identificar los tests para caracterizar a los estudiantes, en la fase C se plantean los elementos didácticos enfocados a las habilidades algorítmicas que servirán de base al diseño de la metodología de ST.

#### Objetivo

Diseñar la estrategia didáctica y de retroalimentación para desarrollar habilidades algorítmicas.

#### Población

Profesores de áreas de Programación básica e investigadores en temas relacionados en ST o Inteligencia Artificial

### Participantes

Se tomó un grupo focal de docentes con experiencia en programación y diseño de Sistemas Tutores quienes ayudarán en el proceso de diseño y evaluación de la metodología. Estos docentes expertos aportan al diseño de la metodología.

### Muestra

Los profesores expertos fueron seleccionados mediante el procedimiento de bola de nieve, en donde se identifican participantes clave y se agregan a la muestra (Martínez, 2012). Los expertos fueron elegidos a través de la interacción con algunos expertos por parte del investigador responsable de este proyecto, los criterios que se utilizaron son:

- Ser identificados como expertos en el área de sistemas tutores.
- Docentes con experiencia en el área de programación e Interfaces de Usuario.
- Docentes con experiencia en el área de pedagogía con experiencia en educación superior.
- Poseen conocimientos como investigadores en el diseño y desarrollo de sistemas tutores.
- Que acepten ser grabados en la entrevista (presencial y en línea).

### Técnica e instrumentos

Se aplicó una entrevista teniendo como instrumento el cuestionario con preguntas abiertas semiestructuradas.

### Procesamiento de datos

En esta etapa (Figura 21) se estructura la metodología, los pasos y los modelos, con la teoría analizada en las etapas previas, además se realizó una prueba piloto para la creación de la entrevista semiestructurada que se aplicó al grupo focal de expertos para obtener las categorías a evaluar de la metodología.

Para el análisis de la metodología para el desarrollo de ST se aplicaron entrevistas piloto utilizando como instrumento el cuestionario de preguntas abiertas semiestructuradas, con el fin de identificar categorías que permitan crear el cuestionario para la entrevista final y medir la metodología a través de expertos en el área de programación y Sistemas Tutores. El análisis de datos a partir de las entrevistas fue crear códigos (Tabla 7:) y categorías de forma abierta, axial y selectiva para generar las categorías de análisis. A continuación se listan los puntos relevantes de las entrevistas a expertos y predominan para generar las categorías.

1.- Aspectos en que coinciden los expertos son las características de estilo de aprendizaje y conocimiento previo en el área, los expertos coinciden en que el razonamiento es un aspecto importante para desarrollar habilidades algorítmicas. Además, mencionan la motivación y actitud como características importantes a observar; con base a estos aspectos se genera la categoría: *Modelo de estudiante* a través de los códigos ME01, ME02, ME03, ME05, ME06, ME08, ME09, ME10.

2.-Otro aspecto en lo que refieren los expertos es el diseño instruccional y las estrategias instruccionales para alcanzar los objetivos del área de conocimiento. Con base a estos aspectos para la enseñanza-aprendizaje de habilidades algorítmicas. Se genera la categoría: *Modelo pedagógico*. MP01, MP02, MP03, MP04, MP05, MP06, MP07, MP08.

3.- Concuerdan en que la metodología debe cumplir con aspectos pedagógicos y tecnológicos que se puedan replicar en otros contextos. El aspecto que les pareció muy importante son la validación y comparación de resultados, así como la aplicación del modelo en otros contextos. Consideran que puede existir una brecha entre los aspectos pedagógicos y tecnológicos, por eso es importante conocer las restricciones que tendrá la metodología. Se genera la categoría *general* sobre la Metodología de diseño para ST conformada por los códigos. MG01, MG02, MG03, MG04, MG05, MG06, MG07.

4.-Otro aspecto en que coincidieron fueron los contenidos y materiales didácticos a generar de acuerdo al campo de conocimiento que se apeguen a una variedad de acuerdo a la combinación de las características del estudiante. *Modelo de conocimiento* reconocida a partir de los códigos MC01, MC02.

5.- Los expertos coinciden en que las tecnologías, los lenguajes y las técnicas de IA, así como los módulos son aspectos a considerar después de la metodología: *Modelo de implementación* identificada por códigos MI01, MI02, MI03, MI04, MI05, MI06.

6.- La categoría de modelo de interfaz de usuario se introduce en el momento que se muestran los prototipos de las actividades, y en las cuales surgen varias coincidencias entre ellas visualizar los elementos de apoyo, retroalimentación y trazado de la actividad. Trata de recolectar elementos perceptores a través del mouse, teclado, entre otros para saber que está haciendo el usuario. MIU01, MIU02.

En la Figura 22 se indican las categorías finales en un mapa general, este contiene los códigos permite identificar las categorías que se presentan en el mapa general y que se respaldan en los pasos de la metodología integrando algunas sugerencias de los expertos.

Algunos aspectos en que coinciden los expertos 1,2 y 3 son las características de estilo de aprendizaje, conocimiento previo y como un elemento importante mencionan el razonamiento abstracto como aspecto importante para desarrollar habilidades algorítmicas, expresan que la motivación y la actitud como características importantes a observar. En base a estos aspectos se genera la categoría: *Modelo de estudiante* donde coinciden los códigos ME01, ME02, ME03, ME05, ME06, ME08.

Tabla 7:  
*Categorización y codificación de las entrevista*

Entrevistado	Preguntas	Respuestas textuales	Subcategoría	Código	Definición
<b>Entrevistado 1 (E1): Nivel académico: cDoctor en Ingeniería; Expertise: Trabaja con Metodologías y Diseño de Sistemas Tutores</b>	En su opinión ¿Consideras que las características de EA, conocimiento previo y razonamiento científico son viables para el diagnóstico del perfil de estudiante al inicio de un curso de acuerdo a tu experiencia?	"considero que están acordes dentro de lo que en los sistemas de tutorías se está trabajando actualmente veo muy novedosa del aspecto razonamiento que estás haciendo por que digamos las variables que tenemos de estilos de aprendizaje y conocimientos previos son variables que vienen en los últimos años siendo han sido muy investigadas y consideró que el involucrar una variable asociada al razonamiento le da un diferenciador muy interesante al trabajo que tú estás haciendo. " ""taxonomía de características que se pueden ser incorporadas en el aspecto de tutoría normalmente se trabajan con estilo de aprendizaje con el nivel de conocimiento que es una variable asociada al perfil del estudiante pero también se incluyen variables como por ejemplo psicológica, de acceso y cuando hablo de acceso tiene que ver directamente con la infraestructura tecnológica disponible para los estudiantes para los usuarios del sistema de tutoría, eso yo creo que son un poco más complejas como por ejemplo de tipo cognoscitivos o de perfil psicológico pero nuevamente lo considero que lo que tu agregas, genera un modelo de estudiante muy interesante para hacer un diagnóstico y hacer adaptaciones"	variable asociada al razonamiento las variable de conocimiento previo y estilos de aprendizaje se han trabajado en los últimos años Variables psicológicas, cognoscitivas Variables de acceso (Infraestructura tecnológica) Realizar una práctica Realizar una serie de pasos (proceso)	ME01 ME02 ME03 ME04 MP02 MP03 MP04	Modelo de estudiante: Percepción que tiene el profesor respecto al conocimiento, estilos de aprendizaje y razonamiento.

Fuente: Elaboración propia.

Tabla 8:  
*Categorización y codificación de las entrevista (Continuación)*

Entrevistado	Preguntas	Respuestas textuales	Subcategoría	Código de subcategorías	Definición
<b>Entrevistado 2(E2): Doctorado en Sistemas de Información con la especialidad en el área de Interacción Humano Computadora. Expertise: Sistemas Interactivos para cualquier contexto de uso.</b>	<b>En su opinión ¿Consideras que las características de EA, conocimiento previo y razonamiento científico son viables para el diagnóstico del perfil de estudiante al inicio de un curso de acuerdo a tu experiencia?</b>	<p>El conocimiento previo casi siempre es el gran problema, de que partimos del supuesto que ya traen ciertos conocimientos ciertas habilidades o cierta capacidad de razonamiento que en el área de programación es fundamental persiste y es muy decepcionante y al final sus profesores tendemos a decir nosotros estimamos describir que es algo necesario, considero que es algo adecuado la parte el conocimiento previo la parte esta de saber si tienen el razonamiento". "característica en los estudiantes que pudiera ser parte del conocimiento para el profesor para saber cómo vienes, algo extra"? "nos gustaría saber es su actitud, la parte actitudinal , no, bueno, cuando son grupos de pequeños y le vas a dar continuidad a los grupos es bueno saber tal o cual alumnos trae cierta tendencia a criticar a descalificar este otro, ofende este otro como juez , para saber cómo es el grupo de estrategias tomar para control optar por el salón entonces las actitudes respecto a... mira fulanito te puede ayudar a ser un buen estratega no se para apoyar tus compañeros, entonces este es quizás eso es necesario para que tú como profesor pues como que apuntalar respecto a la parte esta de los estilos de aprendizaje pues sí. Sería ideal que me ayudaras a entender que significa que alguien sea kinestésico? muchas veces da la impresión de que debe de estar bailando, pero sobre todo que significa para los que no son expertos en aspectos de pedagogía y cómo se pueden abordar estos estilos de aprendizaje".</p>	Perfil de estudiante: Conocimientos o habilidades para resolver problemas mediante algoritmos.	ME05	

<p>¿De acuerdo a tu experiencia docente consideras viable el modelo pedagógico-instruccional para desarrollo de habilidades algorítmicas?</p>	<p>como está alineado el cómo debe ser y lo que esperaría yo que fuera, de hecho se parecen mucho, se parece mucho porque es un mundo de lo incremental parte de... te explicó enseño, te doy un ejemplo y al final espero que logre cierta madurez no?, también tiene sustento teórico que eso también fundamental porque vez con uno propone esto demostrar que funciones es complicado, en el caso tuyo pues ya tienes un par de pilares que ya están muy bien documentados y que han demostrado tener éxito y en general a mí me gusta el tema de la gamificación es algo en lo que creo y es algo considero que puede aportar mucho a lograr estos objetivos". "el tema de los objetos de aprendizaje y he aprendido que por ejemplo para un cierto tema puedes tener diversos caminos para aprender lo mismo entonces representaciones alternativas en concepto o estrategias diferentes para aprender lo mismo a veces resulta una forma muy interesante de aprender por ejemplo ahorita tienes la representación de aprender con Scratch o con el pac-man, que pasaría si en lugar de aprender con el pac-man tratáramos de aprender con legos, tratamos de aprender lo mismo no sé, con hojitas papel es un ejemplo. Entonces a mí me gusta tratar de buscar esas asociaciones paralelas para que algo quizás parecieran no tan obvio, no tan importante y que si alguien se ha dado la tarea de investigar múltiples formas de aprender algo es porque es importante y aprender de varias formas me ha resultado muy enriquecedor que ellos mismos han contrastado que ellos mencionen "yo aprendí así" y "de esta manera es mejor" y como que complementan lo que ya sabe entonces muy enriquecedor."</p>	<p>Estrategia didáctica: Son las teorías y procesos que ayudan a desarrollar las habilidades algorítmicas entendiendo el proceso de aprendizaje del área.</p>	<p>MP05 MP06 MP07</p>
---	--	---	-------------------------------

Tabla 9:  
*Categorización y codificación de las entrevista (Continuación)*

Entrevistado	Preguntas	Respuestas textuales	Subcategoría	Código de subcategorías	Definición
<b>Entrevistado 3</b> <b>(E3): Dr. en Ciencias de la Computación con especialidad.</b> <b>Expertise:</b> <b>Sistemas Inteligentes y reconocimiento de patrones minería de datos. Trabajo en la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla</b>	En su opinión ¿Consideras que las características de EA, conocimiento que son viables para el diagnóstico del perfil de estudiante al inicio de un curso de acuerdo a tu experiencia?	Claro que sí es importantísimo porque mientras más conozco a un alumno de cómo puede aprender cómo visualiza lo que es el conocimiento vamos a poder facilitar este proceso de aprendizaje y es sumamente útil". que "vienen siendo la forma es decir el carácter que tiene los mismos estudiantes que puede ser sumamente importante porque por porque en ocasiones hay alumnos que por su mismo carácter anímico hay que tratar los de una manera diferente que con alumnos que son más decididos a afrontar retos y no que los más decididos puedan lograr más cosas puede ser que los tímidos necesiten un empujoncito para desarrollarse con mucho éxito en lo que vienen siendo diferentes tipos de proyectos y es ahí donde es posible detectar a través de un trato ya personal con los alumnos	Perfil de estudiante: Conocimientos o habilidades para resolver problemas mediante algoritmos.	ME09 ME10	Percepción que tiene el profesor respecto al conocimiento, estilos de aprendizaje y razonamiento.
	¿De acuerdo a tu experiencia docente consideras viable el modelo pedagógico-instruccional para desarrollo de habilidades algorítmicas?	Si yo considero que tienen muchos módulos muy importantes así como se está planteando está considerando ahora si lo que viene siendo una línea de aprendizaje y eso es lo que de alguna manera puede apoyar el proceso de enseñanza-aprendizaje	Estrategia didáctica: Son las teorías y procesos que ayudan a desarrollar las habilidades algorítmicas entendiendo el proceso de aprendizaje del área	MP08	Modelo tutor: Contienen la línea que llevará el proceso de enseñanza-aprendizaje.

### Resultados

En el mapa general (Figura 22) como se puede observar se integran en las categorías los códigos que se identificaron en la primer entrevista piloto y que sirvió para generar la entrevista final. Por otra parte, se comienzan a ver elementos que concuerdan con la literatura sobre los Sistemas Tutores y que dan pauta a integrar los elementos de la fase de análisis en el diseño de la metodología.

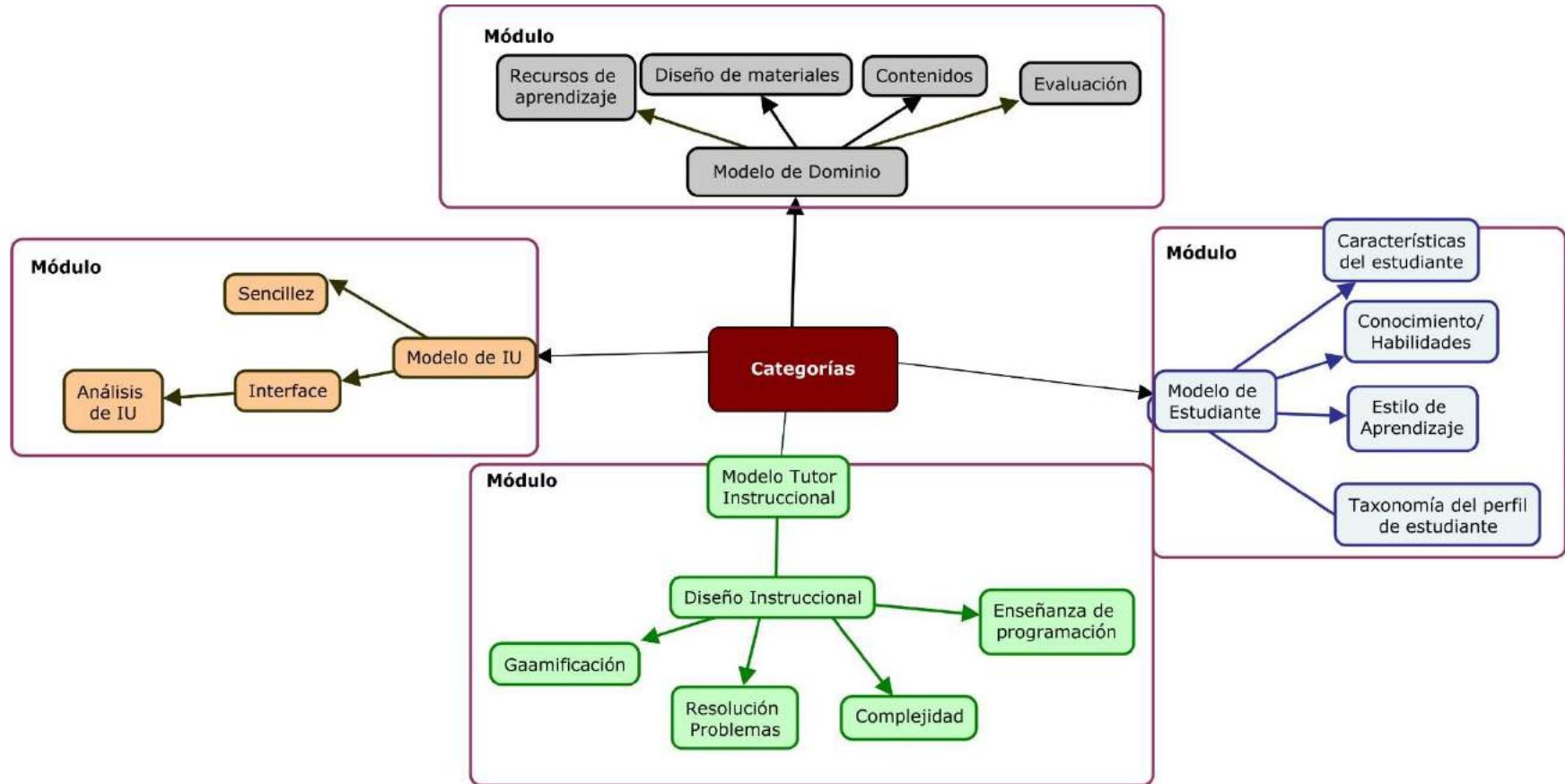


Figura 22. Mapa general de unidades de análisis en las categorías.

Fuente: Elaboración propia

## 5.4 IBD Fase 3

### 5.4.1 Diseño y validación (D)

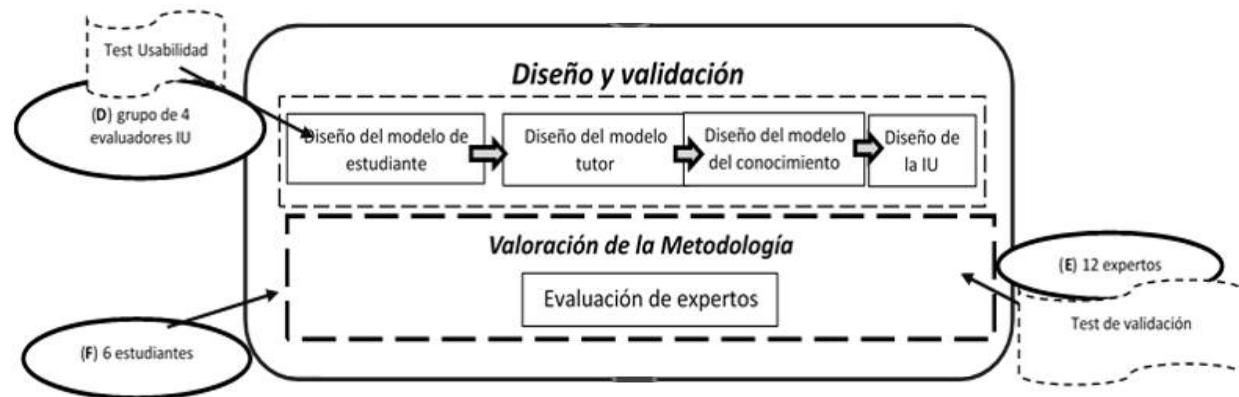


Figura 23. Diseño de la metodología, subfase D.

Fuente: Elaboración propia.

Como complemento en el diseño de la metodología se realizaron prototipos de Interfaz de Usuario (IU) de las actividades que integran elementos con los cuales posteriormente interactuará el usuario.

#### Objetivo

Diseñar una metodología de Sistema Tutor que cumple con las especificaciones identificadas en la literatura y valoración de expertos.

#### Población

Estudiantes de nuevo ingreso en la Facultad de Ciencias de la Computación.

#### Participantes

Estudiantes que se integraron en el semestre 2018 de Ingeniería o Licenciatura en Ciencias de la Computación.

#### Muestra

Se formó un grupo focal de cuatro estudiantes de nuevo ingreso al semestre 2018 de Ingeniería.

#### Técnicas e instrumentos

Se utiliza el Cuestionario de Usabilidad en Sistemas Informáticos (CSUQ, por sus siglas en inglés) adaptado y validado al español (Hedlefs et al., 2016) y se realizan la

técnica de entrevista grupal para recuperar la experiencia de usuario y dar su retroalimentación sobre la usabilidad.

### Procesamiento de datos

Para la evaluación de los prototipos propuestos en este trabajo (Figura 24), se llevaron a cabo pruebas de usabilidad con cuatro usuarios, todos alumnos del primer semestre de la Licenciatura en Ingeniería en Tecnologías de la Información de la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla. Las pruebas fueron en un ambiente controlado en el que se les pidió a los usuarios simular la interacción con los prototipos realizando la actividad de diseño de un algoritmo, que consiste en leer las instrucciones y completar la secuencia de pasos a resolver en el primer paso de la actividad debe arrastrar los enunciados de la derecha tratando de organizar la lista de pasos generales para resolver el problema. En el segundo paso se presentan diversas figuras que representan los personajes que deben de elegir, en cualquier momento se puede pedir retroalimentación de la actividad o ayuda sobre el uso de la interfaz. En el tercer paso se procede a identificar los eventos que manipularán el personaje y en el último paso se le presenta al usuario los bloques que manipulan al personaje y él debe elegir el bloque correcto, al final se muestra el resultado de su desempeño durante la actividad (puntos, nivel de avance y categoría).

### Resultados

Los resultados presentados en la Tabla 10, como se puede observar en todas las categorías (Uso del sistema, Calidad de la Información, Calidad de la IU, General) el valor mínimo no está por debajo de 4 y la media obtenida se encuentra entre 5 y 6, por lo que se considera un nivel de satisfacción buena pero se puede mejorar (Lewis, 1995).

Tabla 10:  
*Resultados de evaluación de usabilidad de prototipos de IU*

Rango	Uso del Sistema	Calidad de la Información	Calidad de la Interfaz	Estimado General
Alta	6.450	6.898	7.294	6.707
Baja	4.382	4.351	4.038	4.792
Promedio	5.416	5.625	5.666	5.75

Fuente: Elaboración propia.

Los prototipos fueron evaluados arriba del promedio aceptable, sin embargo se hicieron comentarios de mejora que se tomarán en consideración para la posterior integración en el prototipo como trabajo futuro. Durante la aplicación también se valoraron aspectos que no mide el cuestionario, como la forma de resolver el problema y el interés en resolver el ejercicio correctamente. Algunos de los comentarios de los usuarios son:

Usuario 1: Creo que se puede mejorar los botones en la parte inferior y usar palabras más fáciles de comprensión. Se podrían usar iconos para una mejor comprensión. Al arrastrar las instrucciones al otro cuadro, estaría bien poder enumerar las opciones.

Usuario 2: La interfaz es buena y muy amigable pero le falta que proporcione información más precisa de qué es lo que se debe realizar sería muy bueno resaltar los botones.

Usuario 3: En general el sistema me pareció muy bien pero creo que me gustaría que me llevara más de la mano en cuanto a la explicación de cada botón. En la parte de las respuestas donde aparece la parte del Scratch me diera la oportunidad de explicarme cada parte así poder dar una respuesta más adecuada.

Usuario 4: A mi parecer en la interfaz de retroalimentación para el usuario deberían de retirarse los botones de revisar y ayuda, considero se serían iconos visuales.

Por lo tanto, las interfaces se modificarán para integrar las mejoras que proponen los estudiantes y mejorar su nivel y experiencia de usuario, con el fin de que se centren en la solución del problema.

#### 5.4.2 Diseño y validación (E)

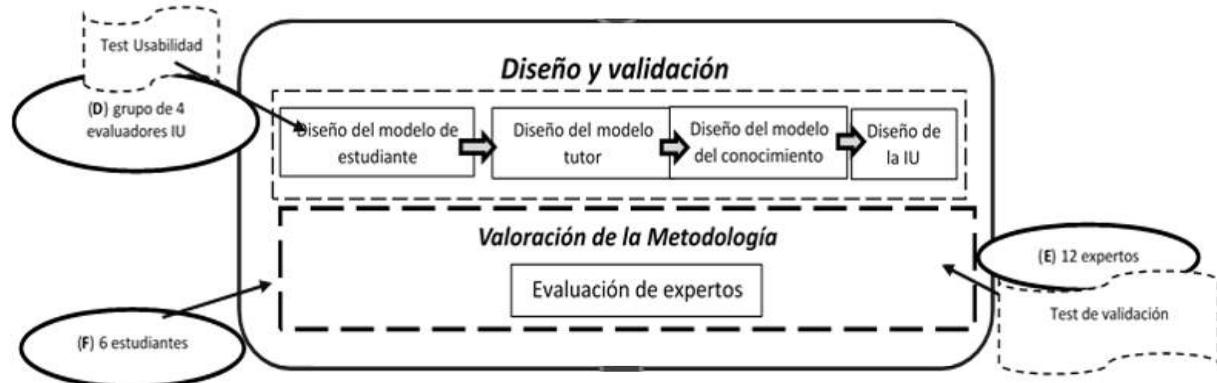


Figura 24. Validación de expertos sobre la metodología, subfase E.

Fuente: Elaboración propia.

En la subfase E se deben concretar los elementos que sustentarán la metodología, por lo tanto se contempla la evaluación de expertos para la validación de la metodología (Figura 24).

#### Objetivo

Diseñar una metodología de Sistema Tutor que cumple con las especificaciones identificadas en la literatura y valoración de expertos.

#### Población

Profesores investigadores que estén impartiendo clases en el nivel superior en áreas de psicología, pedagogía y computación.

#### Participantes

Se tomó un grupo focal de docentes que estén inmersos en alguna de las áreas de pedagogía, psicología, computación y programación o diseño de Sistemas Tutores quienes ayudarán en el proceso de diseño y evaluación de la metodología.

#### Muestra

Los profesores expertos fueron seleccionados mediante el procedimiento de bola de nieve, en donde se identifican participantes clave y se agregan a la muestra

(Martínez, 2012). Los expertos fueron elegidos a través de la interacción con algunos expertos por parte del investigador responsable de este proyecto, los criterios que se utilizaron son:

- Identificados como expertos en el área de sistemas tutores, pedagogía o psicología.
- Docentes con experiencia en el área de programación e Interfaces de Usuario.
- Docentes con experiencia en el área de pedagogía con experiencia en educación superior.
- Poseen conocimientos como investigadores en el diseño y desarrollo de sistemas tutores.
- Que acepten ser grabados en la entrevista.

#### Técnica e instrumentos

Se tiene un grupo de enfoque (expertos), donde se aplicó una entrevista (anexo F) teniendo como instrumento el cuestionario con preguntas abiertas semiestructuradas y de escala Likert que evalúan la metodología.

#### Procesamiento de datos

De acuerdo a los datos generados en la entrevista tenemos dos momentos para el análisis de la entrevista, por un lado las preguntas en escala Likert y por otro las preguntas abiertas. Para determinar los resultados obtenidos comenzamos con la descripción por categorías que se hace sobre los datos a través de la frecuencia de las respuestas de los expertos y el porcentaje en cada una de las escalas de Likert a través de gráficas.

#### Resultados

Posteriormente, se presenta la metodología a un grupo de expertos con el objetivo de encontrar información de personas idóneas a los objetivos temáticos que impulsan la investigación en diferentes áreas como programación, pedagogía, psicología y de desarrollo de Sistemas Tutores, a quienes se les hace una entrevista semiestructurada de preguntas abiertas y de escala Likert

Para la categoría de modelo de estudiante los expertos coincidieron un 64% en estar totalmente de acuerdo con los elementos del estilo de aprendizaje y el diagnóstico del estudiante en sus habilidades algorítmicas que se integran en este modelo en un 64%, en contraste con un 2% que integran otros elementos en su práctica docente. Para la categoría del modelo de dominio los expertos coincidieron un 56% en estar totalmente de acuerdo con la forma de abordar el contenido y de disponer el avance de los estudiante en las categorías definidas de novato, aprendiz y experto, sin embargo un 2% define que el diseño de los contenidos es una tarea ardua y que algunos docentes pueden ser renuentes a su creación.

En la categoría de modelo pedagógico, los expertos coincidieron en que los elementos de la estrategia pedagógica apoya a las habilidades algorítmicas coincidiendo un 78% en estar totalmente de acuerdo además resaltaron que la retroalimentación en cada etapa y apoyados con las técnicas básicas de gamificación orientan y motivan al estudiante, un 2% les gustaría ver la actividad en el sistema.

Para la categoría de evaluación general, un 67% de los expertos están totalmente de acuerdo en que la metodología cumple con los aspectos teóricos y apoya a las habilidades algorítmicas, sin embargo un 3% resalta que se puede integrar otros aspectos en el módulo de estudiante.

#### 5.4.3 Diseño y validación (F)

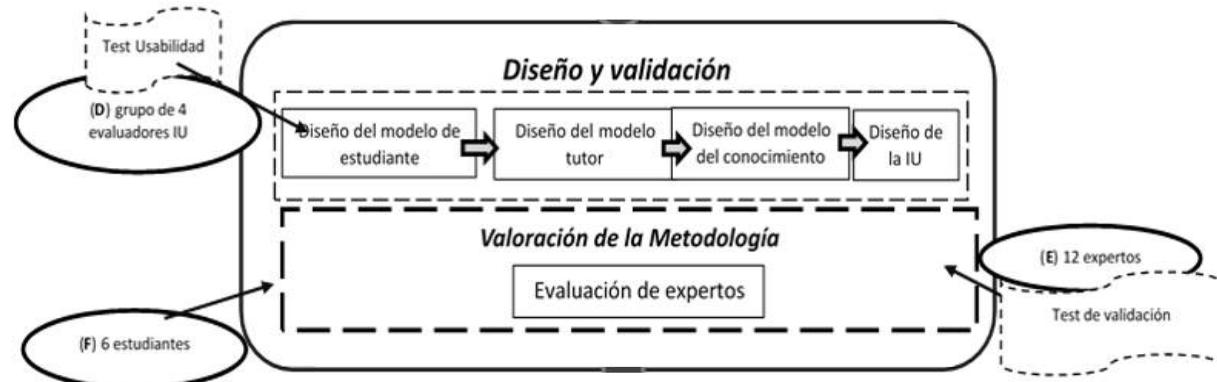


Figura 25. Validación de la metodología con estudiantes, subfase F.

Fuente: Elaboración propia.

En esta etapa se define el proceso de validación de la metodología a través de la aplicación de la estrategia didáctica a alumnos de nuevo ingreso, haciendo un análisis cualitativo sobre la intervención que se efectuó (Figura 25).

#### Objetivo

Diseñar una metodología de Sistema tutor que cumple con las especificaciones identificadas en la literatura y valoración de expertos.

#### Población

La principal población son los estudiantes de nuevo ingreso al nivel superior del área de Computación, en particular en la Facultad de Ciencias de la Computación en las carreras de Ingeniería en Ciencias de la Computación y la Licenciatura en Ciencias de la Computación. La facultad está en un área urbana de la ciudad de Puebla, la cual consta de 5 edificios y cuentan con laboratorios de 35 computadoras y con acceso a Internet. Se tiene una población de 500 jóvenes que ingresan anualmente en periodo de otoño, la mayoría tienen entre 18 y 23 años.

#### Participantes

Los estudiantes que cursan la materia de Metodología de la programación y son de nuevo ingreso.

#### Muestra

Se trabajó con un grupo focal de acuerdo a las características que se obtuvieron en la fase de caracterización, en este caso se eligieron los 6 estudiantes que obtuvieron bajo puntaje y son principalmente visuales.

Los estudiantes cumplen las siguientes características:

- Son estudiantes de nuevo ingreso (1er semestre).
- Tienen entre 18 y 22 años.
- Cursan la materia de metodología de la programación.
- Obtener de 15 a 20 aciertos en el test de habilidades algorítmicas
- Su estilo de aprendizaje sea preferentemente visual.

Los criterios de exclusión para no integrarlos es que cursen materias donde utilicen algún lenguaje de programación. Los criterios de exclusión es que ya sean de segundo cuatrimestre o posterior, es decir ya tienen desarrolladas las habilidades algorítmicas básicas. La participación de los estudiantes aportó a la validación de la estrategia didáctica como parte de la metodología, además dieron retroalimentación de la estrategia didáctica propuesta, lo cual se discutirá en el capítulo de resultados.

### Técnicas e instrumentos

- a) Se aplica el test de pensamiento computacional
- b) Se aplica un cuestionario con preguntas semiestructuradas y preguntas de tipo Likert para valorar el taller.

Se utilizó la técnica de entrevista para la aplicación del cuestionario sobre el taller y en el caso del test se aplicó el cuestionario en línea.

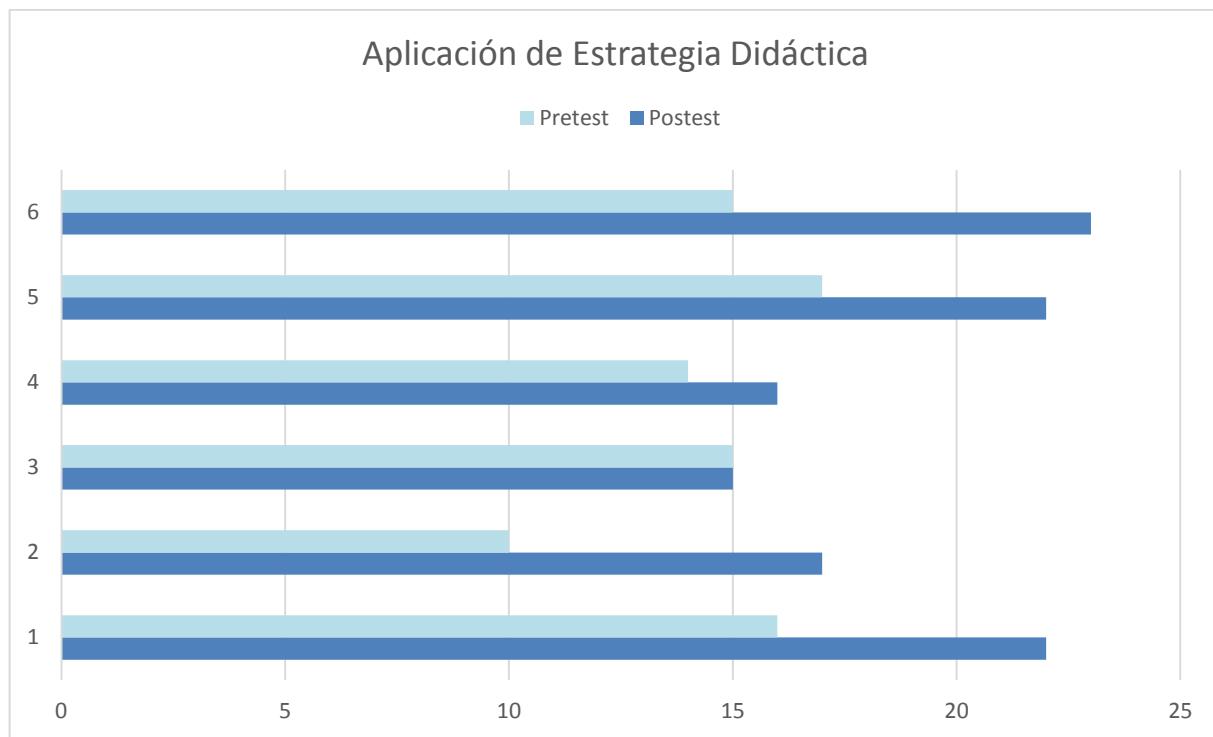
### Procesamiento de datos

Al finalizar el taller se les aplica el postests y los datos se recaban del formulario que respondieron en línea. Después, se hace la entrevista utilizando un cuestionario con preguntas de forma individual sobre la valoración del taller. Se integran los datos en una hoja de cálculo para compararlos los resultados del pretest.

### Resultados

Después de llevar a cabo la estrategia instruccional en el desarrollo de habilidades algorítmicas aplicando el test de habilidades algorítmicas utilizando un cuestionario de preguntas de respuesta múltiple en línea (Román-González, Pérez-González y Jiménez-Fernández, 2015), se comparan los resultados del pretest y postest en las cuales se despliegan las diferencias en los dos momentos, destacando una mejora en cinco de seis estudiantes, quedando un estudiante en el mismo nivel general.

A continuación se muestra en la Gráfica 4 los resultados de los seis estudiantes, donde los datos arrojados del postest muestran que un 85% de los estudiantes han mejorado sus habilidades algorítmicas y sólo el 15% de ellos quedó con el mismo resultado que el pretest.



*Gráfica 4.* Resultados de la Aplicación de la estrategia (Pretest y Postest).

Algunos de los comentarios de los estudiantes para mejorar el taller fue ampliar el tiempo de aplicación dado que les gustó la temática, el integrar la gamificación fue algo que los motivo a interesarse y aprender por sí mismos, otro aspecto fue que si el curso se puede dar para preparar a los estudiantes que piensen entrar a la carrera como curso propedéutico.

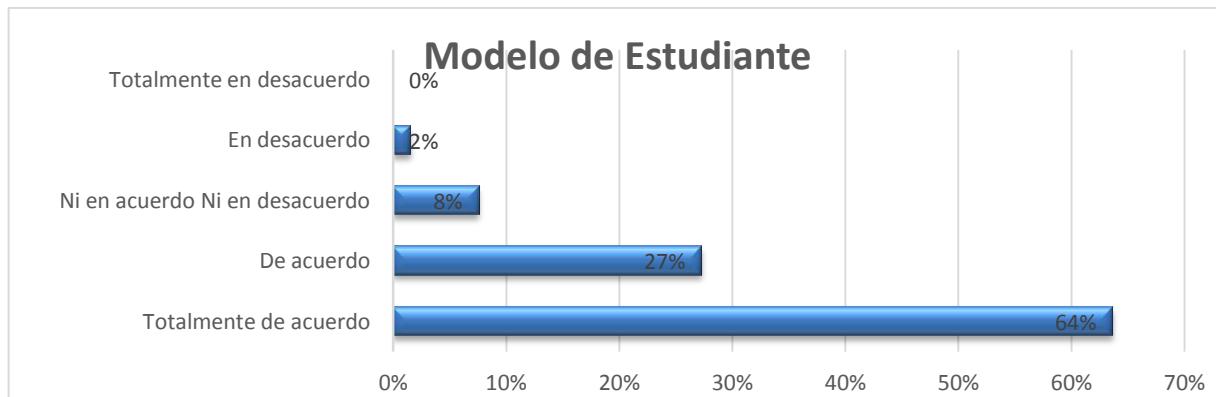
Los resultados de la validación de las entrevistas de preguntas abiertas realizadas a ambos grupos (expertos y estudiantes) se profundizan en el siguiente capítulo, dando los resultados de la investigación a fondo.

## 6. Resultados

En este capítulo se describirán los datos obtenidos para evaluar la metodología; se tienen dos momentos: la evaluación de expertos en áreas de pedagogía, psicología, en ST y otro momento es la validación de un grupo focal de estudiantes. A partir de ambas validaciones se determina la aceptación y mejora de la metodología a partir de los aportes hechos de ambos grupos. Por otra parte, los resultados del taller a los estudiantes se analizan para comparar el pre y postest de sus habilidades algorítmicas, así como el aporte de sus respuestas en la estrategia didáctica.

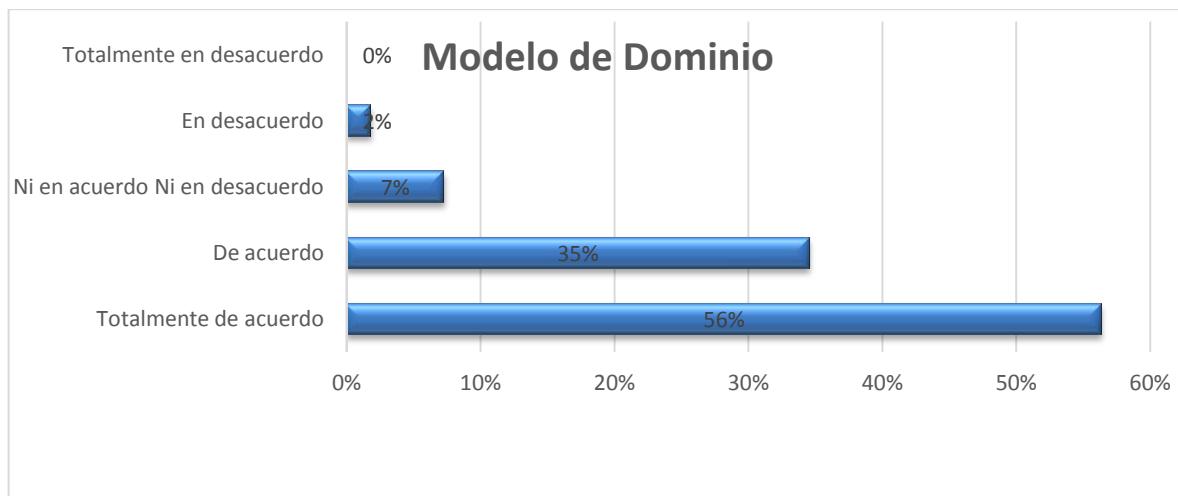
### 6.1 Validación de expertos

Después de exponer la metodología en el Capítulo 4, se obtienen los resultados de la en la Fase E a través de la entrevista a expertos (Anexo F) generando las categorías con base en la frecuencia de las respuestas de los expertos y el porcentaje en cada una de las escalas Likert. En la categoría de modelo de estudiante (Gráfica 5) los expertos coincidieron un 64% en estar totalmente de acuerdo, 27% mencionó estar de acuerdo con los elementos diagnósticos como el estilo de aprendizaje y las habilidades algorítmicas particularmente identificar en cuáles requiere mayor dedicación; en contraste con un 2% que integran otros elementos en su práctica docente y por lo tanto mencionan que sería recomendable integrar aspectos para medir la motivación.



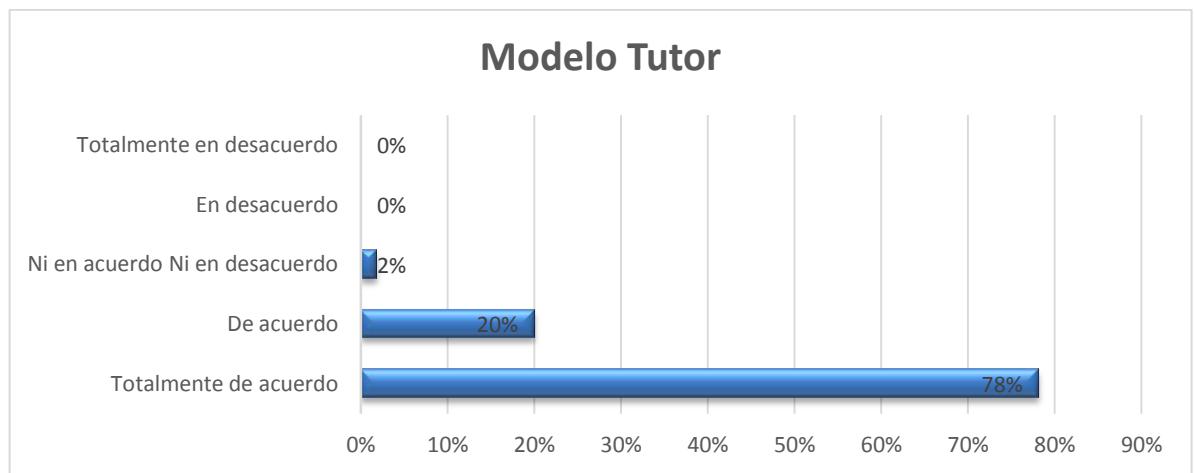
Gráfica 5. Evaluación de la categoría modelo de estudiante.

Para la categoría del modelo de dominio (Gráfica 6) los expertos coincidieron un 56% en estar totalmente de acuerdo y un 35% está de acuerdo con la forma de abordar el contenido a través de la dosificación de los ejercicios para minimizar la carga cognitiva y de disponer el avance de los estudiante en las categorías definidas de novato, aprendiz y experto, sin embargo un 2% define que el diseño de los contenidos es una tarea ardua y que algunos docentes pueden ser renuentes a su creación.



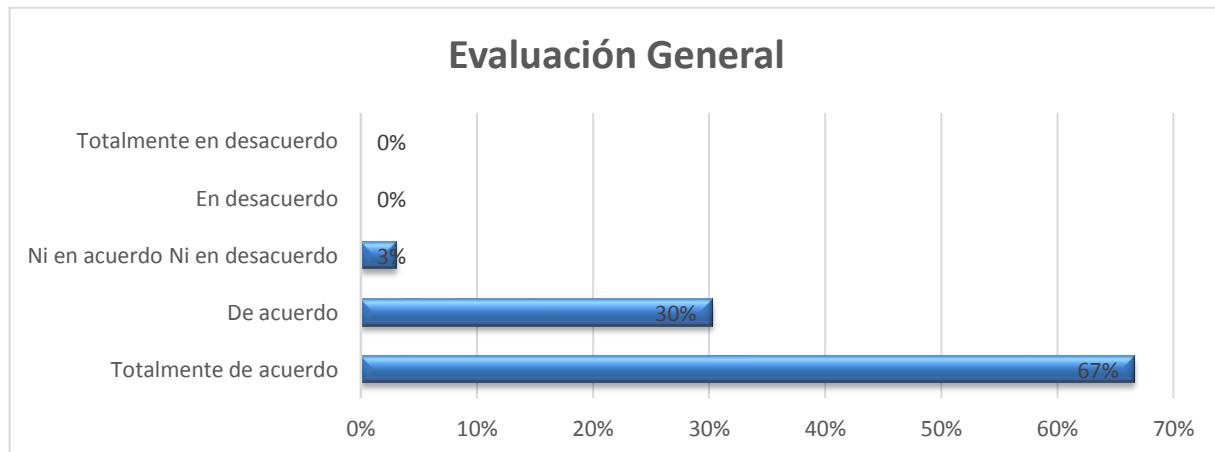
*Gráfica 6. Evaluación de la categoría de modelo de dominio.*

En la categoría de modelo tutor (Gráfica 7), los expertos están totalmente de acuerdo en un 78% en que los elementos de la estrategia pedagógica apoyan a las habilidades algorítmicas, un 20% están de acuerdo y resaltaron que la retroalimentación en cada etapa y con el apoyo de las técnicas básicas de gamificación orientan y motivan al estudiante, un 2% quieren ver la actividad en el sistema.



*Gráfica 7.* Evaluación de la categoría de modelo tutor.

Para la categoría de evaluación general (Gráfica 8), un 67% de los expertos están totalmente de acuerdo en que la metodología cumple con los aspectos teóricos y apoya a las habilidades algorítmicas, sin embargo un 30% resalta que se puede integrar otros aspectos en el módulo de estudiante sobre todo aspectos demográficos y socioeconómicos que consideran pueden dar mayor información sobre el estudiante. El 3% supone que es viable, pero que requiere otra validación o el sistema para dar una respuesta más concreta.

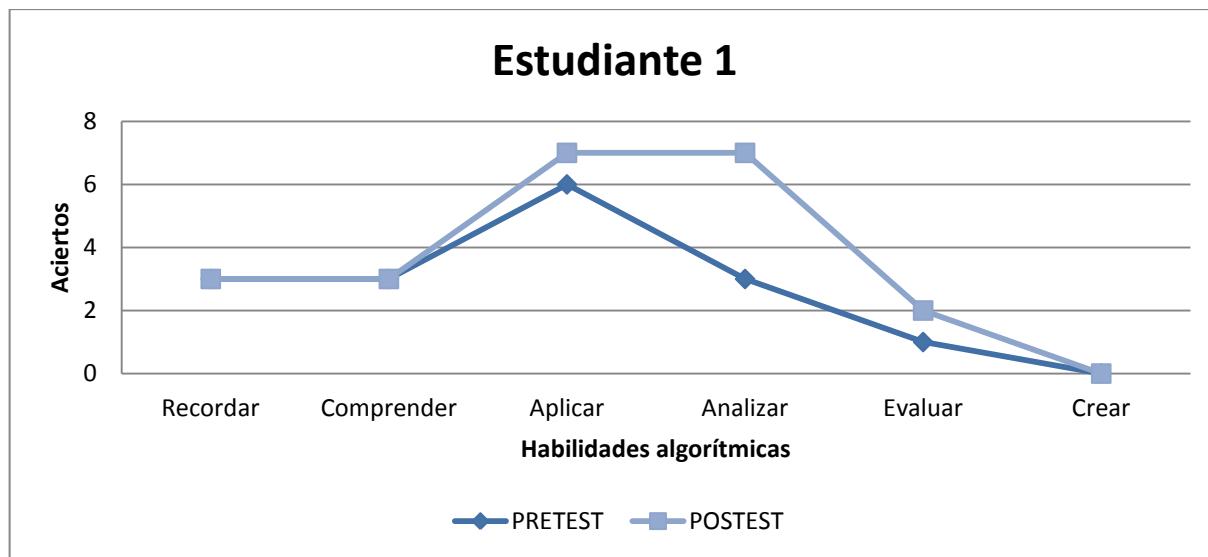


*Gráfica 8.* Evaluación de la categoría general.

## 6.2 Validación de estudiantes

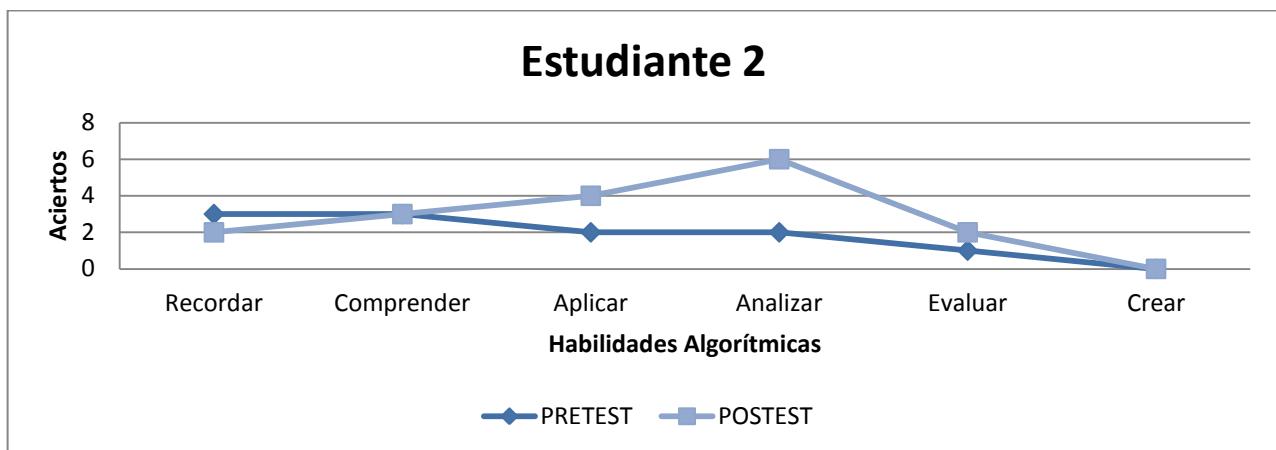
Los resultados que se obtuvieron del pretest y postest después de aplicar el taller a los estudiantes se muestran de forma individual en gráficas que muestran el avance en cada una de las habilidades algorítmicas que se valoraron.

En la Gráfica 9 se muestra como el estudiante uno mejoró en las habilidades de Aplicar, Analizar y Evaluar, teniendo 3 aciertos en el pretest y 7 aciertos en el postest en la habilidad de Analizar. Es importante resaltar que dada las categorías se pueden identificar que además de saber en qué nivel está, se determina la habilidad que requiere mayor atención para focalizar las actividades.



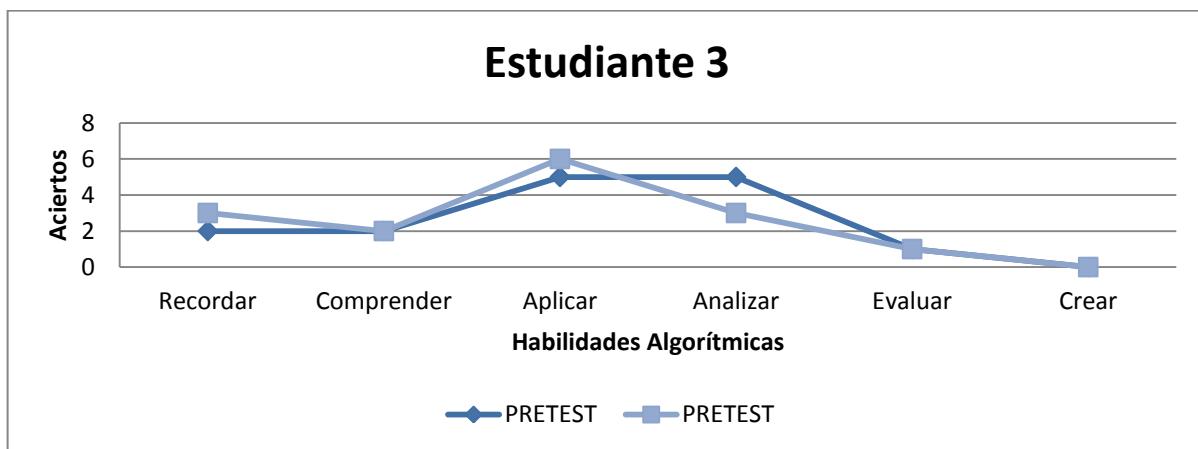
Gráfica 9: Resultados del pretest y postest de las habilidades del estudiante 1

Para el estudiante dos (Gráfica 10) incrementa su puntaje de 2 a 6 aciertos en el nivel de Analizar, sin embargo sigue en el nivel de Aprendiz, mejorando sus habilidades resaltando principalmente la de Analizar, a pesar de que mejoró en siete puntos en total, aún le falta más actividades para fortalecer sus habilidades de Evaluar y Crear.



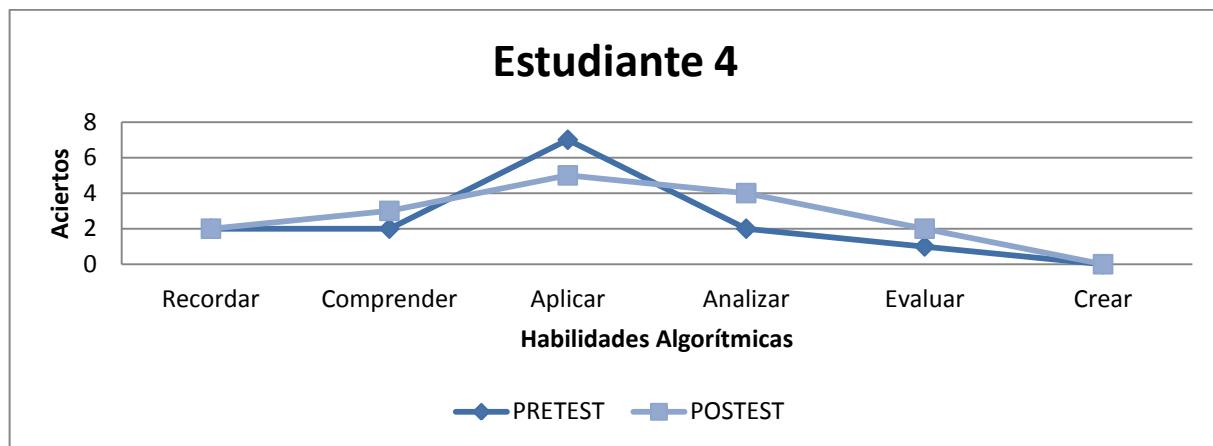
Gráfica 10. Resultados del pretest y postest de habilidades del estudiante 2.

El estudiante tres (Gráfica 11) mejoró la habilidad de Aplicar, sin embargo en las preguntas de Analizar disminuye su puntaje, lo cual precede a que la habilidad de Evaluuar y Crear que quedan en el mismo nivel.



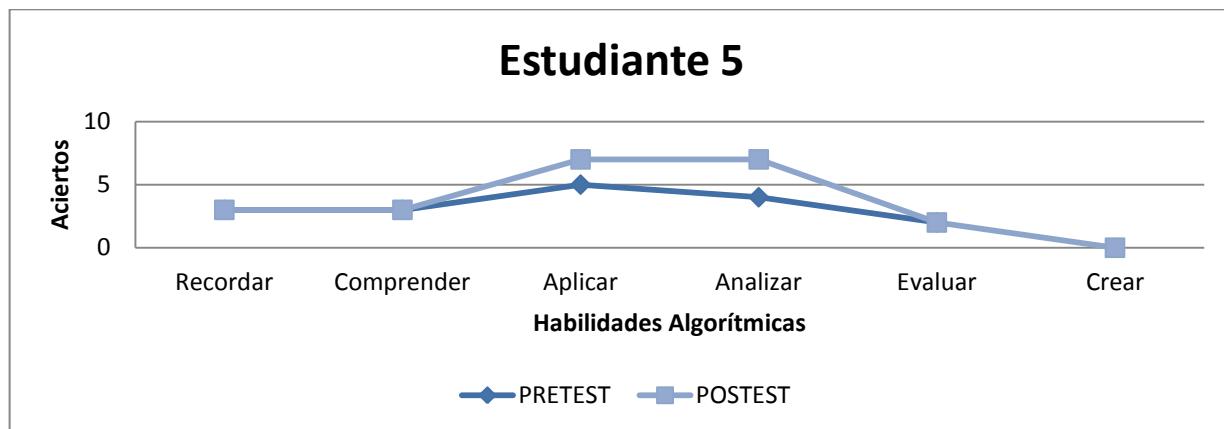
Gráfica 11 . Resultados del pretest y postest de habilidades del estudiante 3.

En la Gráfica 12 se indican los resultados del estudiante cuatro; quien incrementó en las habilidades de Analizar y Evaluuar, sin embargo tiene menos puntaje en la habilidad de Aplicar, por lo que para llegar a la habilidad de Crear requiere mayor trabajo.



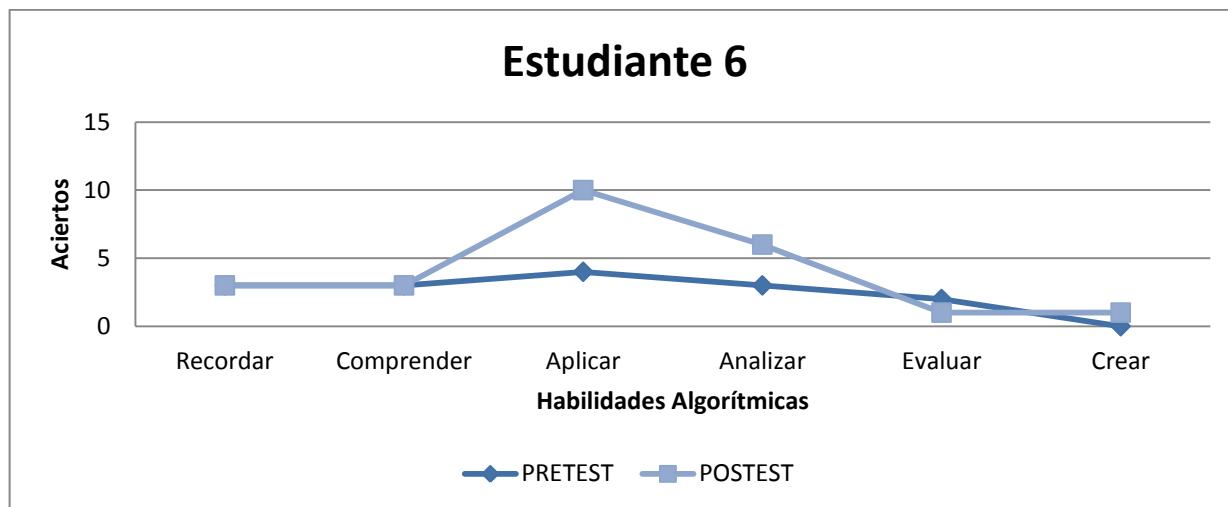
Gráfica 12 . Resultados del pretest y postest de habilidades del estudiante 4.

En la Gráfica 13 el estudiante cinco mejora ascendentemente en cada una de las siguientes categorías después de Comprender, lo que muestra un avance correspondiente a las habilidades de Aplicar, Analizar y Evaluar requeridas en esa misma tendencia a mejorar la última habilidad de Crear.



Gráfica 13 . Resultados del pretest y postest de habilidades del estudiante 5.

Se muestra en la Gráfica 14 como el estudiante seis incrementa en las habilidades de Aplicar y Analizar, además de que es el único que resuelve satisfactoriamente la pregunta en la que crean completamente las instrucciones.



*Gráfica 14.* Resultados del pretest y postest de habilidades del estudiante 6

### 6.3. Discusión de resultados.

De acuerdo al resultado de la investigación se obtuvo la metodología de diseño de Sistema Tutor enfocado a las habilidades algorítmicas para estudiantes de nuevo ingreso, y en particular de la Facultad de Ciencias de la Computación. A partir del problema de investigación sobre el apoyo a las habilidades algorítmicas y a través del apoyo tecnológico de los Sistemas Tutores, se encuentran cuatro unidades de análisis: Modelo de conocimiento, Modelo de estudiante, Modelo de pedagógico y Modelo de tutor. De acuerdo a las diferentes propuestas de metodologías que se han estudiado existen diferentes tareas a definir en cada uno, las cuales abordaremos en las siguientes tablas a partir de las categorías, ejes de análisis y lexías encontradas haciendo el contraste con la teoría.

Tabla 11:  
*Modelo de estudiante*

Categoría	Eje de análisis	Lexías	Análisis
<b>Metodología de Modelo de estudiante.</b> <b>Sistema Tutor.</b>	Representa la visión de cada Proceso que define los elementos que componen la base teórica de los módulos de estudio, área de dominio, ritmo de dominio, tutor e interfaz de usuario. La metodología debe estar definida de acuerdo al área de conocimiento particular para llevar a cabo la estrategia instruccional en el desarrollo de	<p>❖ <i>La taxonomía de características que pueden ser incorporadas en el aspecto de tutoría normalmente se trabajan con estilo de aprendizaje y con el nivel de conocimiento que es una variable asociada al perfil del estudiante, pero también se incluyen variables como por ejemplo psicológica, de acceso y cuando hablo de acceso tiene que ver directamente con la infraestructura tecnológica disponible para los usuarios del sistema de tutoría, eso yo creo que son un poco más complejas como por ejemplo de tipo cognoscitivos o de perfil psicológico.</i></p> <p>❖ <i>Se tiene que garantizar que exista la tecnología en el caso de la implementación para tener el acceso cuando lo requieran.</i></p>	Los elementos que perciben los expertos para poder caracterizar al estudiante de computación se enfocan fundamentalmente en su estilo de aprendizaje y su nivel cognitivo respecto a las habilidades algorítmicas, donde se recomienda conocer esta información que caracteriza a cada uno de los estudiantes como apoyo al docente y su nivel de dominio en el tema para guiar su aprendizaje en las habilidades algorítmicas. Los resultados concuerdan con Moroni y Señas ( 2005) en que aspectos de conocimiento previo y nivel cognitivo son relevantes para identificar el perfil del estudiante. Los expertos consideran importante

habilidades algorítmicas.	<p>pragmático, este modelo debe contener los datos que requieran los procesos que lo manipulen y representarlos de la forma más adecuada para ello (Self, 1990). Este módulo debe ser capaz de reconocer las carencias y fortalezas del alumno. Puede ser visto como la unión de dos submódulos, uno que contenga la información referente a los estilos de aprendizaje y el otro que contenga los datos que van a ser actualizados, en dependencia de la interacción del estudiante con el sistema, se define así el nivel de conocimiento (Tarongí, 2010). En general es la habilidad para identificar y modelar el nivel cognitivo de cada estudiante. (Rosado, Sánchez y García, 2016).</p>	<ul style="list-style-type: none"> <li>❖ <i>El conocimiento previo casi siempre es el gran problema de que partimos del supuesto que ya traen ciertos conocimientos ciertas habilidades o cierta capacidad de razonamiento que en el área de programación es fundamental.</i></li> <li>❖ <i>Característica en los estudiantes que pudiera ser parte del conocimiento para el profesor para saber cómo vienes, algo extra.</i></li> <li>❖ <i>Es quizás necesario para que tú como profesor, pues, como que apuntalar respecto a la parte esta de los estilos de aprendizaje pues sí, un poco...para uno cuando no sabe uno muy bien el tema sería ideal que me ayudaras a entender ¿Qué significa que alguien sea kinestésico?</i></li> <li>❖ <i>Mientras más conozco a un alumno de cómo puede aprender cómo visualiza, lo que es el conocimiento, vamos a poder facilitar este proceso de aprendizaje y es sumamente útil.</i></li> <li>❖ <i>Vienen siendo la forma es decir el carácter que tiene los mismos</i></li> </ul>	<p>la revisión del conocimiento previo y el estilo de aprendizaje, entre otras variables psicológicas y cognitivas que se pueden generalizar para tener una caracterización del estudiante. Conejo et al., (2001) mencionan características del alumno como (a) aspectos observables, en este caso el carácter anímico o aspectos de motivación y entusiasmo que se demostraba en la realización de los ejercicios; (b) las características internas que debe ser inferidas en base a la información obtenida y que son importantes para el aprendizaje, por lo tanto la preferencia del estilo de aprendizaje es importante para dar el seguimiento del alumno.</p> <p>A pesar de manifestar el factor anímico como un elemento que afecta su aprendizaje, por el momento queda fuera del alcance de esta investigación.</p>
---------------------------	---	---	---

---

*estudiantes que puede ser sumamente importante porque en ocasiones hay alumnos que por su mismo carácter anímico hay que tratar los de una manera diferente que con alumnos que son más decididos a afrontar retos y no que los más decididos puedan lograr más cosas puede ser que los tímidos necesiten un empujoncito para desarrollarse con mucho éxito.*

---

Fuente: Elaboración propia.

Tabla 12  
*Modelo de Tutor*

Categoría	Eje de análisis	Lexías	Análisis
<b>Metodología de Modelo Tutor</b>	<p><b>Sistema Tutor.</b> Proceso que define los elementos que componen la base teórica de los módulos de dominio, tutor e interfaz de usuario. La metodología debe estar definida de acuerdo al área de conocimiento particular para llevar a cabo la estrategia instruccional en el desarrollo de habilidades algorítmicas.</p> <p>De acuerdo a Cataldi y Lage (2009) el modelo pedagógico o tutor debería estar diseñado desde una concepción epistemológica acerca de lo que significa enseñar en el estudiante, área programación. Según Sierra et. al, (2004) es una representación de los métodos que se usarán en el tutor inteligente para proveer información al estudiante. Este modelo es complejo pues está pensado para dirigir al estudiante en su proceso de aprendizaje y efectuar ajustes en esta dirección a medida que el estudiante progresá.</p>	<ul style="list-style-type: none"> <li>❖ Los elementos fundamentales es decir identifica como a partir de un problema en el cual el estudiante tiene que realizar una práctica y tiene que realizar una serie de pasos y con base en los resultados que va obteniendo en el proceso, entonces obtiene una retroalimentación, es decir rescata el proceso de instrucción que debe estar acompañado de la retroalimentación en cada paso de la actividad.</li> <li>❖ Resalta nuevamente el aspecto didáctico que se integra en el diseño de las actividades, caracterizándolos como incrementales, que es un aspecto importante para los estudiantes, cómo está alineado el cómo debe ser y lo que esperaría yo que fuera; de hecho, se parece mucho porque es un mundo que va de lo incremental, parte de té explicó enseño, te doy un ejemplo y al final espero que logre cierta madurez, ¿no?</li> <li>❖ Concretamente en el aspecto de la por lo que la estrategia</li> </ul>	<p>De acuerdo a la literatura y las percepciones sobre el modelo pedagógico, se coincide con el aspecto de la retroalimentación en las etapas de diseño de la estrategia didáctica para apoyar al estudiante. Otro aspecto que se respalda en la literatura (Moroni y Señas, 2005; Rosanigo y Paur, 2006) es el tener diversas actividades para atender los diferentes niveles de estudiante. Además, la propuesta que se hace va de la mano con la recomendación de Cataldi y Lage (2009), al diseñar las actividades desde el aspecto epistemológico,</p>

---

*gamificación como un factor que puede ayudar al estudiante a lograr los objetivos de aprendizaje, expresando “tienes un par de pilares que ya están muy bien documentados y que han demostrado tener éxito y en general a mí me gusta el tema de la gamificación es algo en lo que creo y es algo considero que puede aportar mucho a lograr estos objetivos. Además, menciona la forma en que se puede atender con diversidad de contenidos y formas de llevar a cabo la enseñanza a través de la estrategia didáctica, tener diversos caminos para aprender lo mismo entonces representaciones alternativas en concepto o estrategias diferentes para aprender lo mismo a veces resulta una forma muy interesante de aprender.*

fundamentada en el desarrollo de habilidades algorítmicas y en la resolución de problemas, acompañada de aspectos de gamificación complementa la estrategia de didáctica que se respalda con la literatura.

---

Fuente: Elaboración propia.

Tabla 13:  
Modelo de dominio

Categoría	Eje de análisis	Lexías	Análisis
<b>Metodología de Sistema Tutor.</b>	<p><b>Modelo de dominio.</b></p> <p>El módulo del dominio, que está compuesto por la ruta de aprendizaje que viene definida por el módulo del tutor.. Anderson (1983) concuerda en que este modelo contiene el conocimiento sobre la materia que debe ser dominio, tutor e aprendida, y que el modelo interfaz de usuario. La metodología cuanta más abundancia de debe estar definida de acuerdo al área de conocimiento particular para llevar a cabo la estrategia instruccional en el desarrollo de habilidades algorítmicas.</p> <p>El módulo del dominio, que viene definida por el módulo del tutor.. Anderson (1983) concuerda en que este modelo contiene el conocimiento sobre la materia que debe ser dominio, tutor e aprendida, y que el modelo interfaz de usuario. La metodología cuanta más abundancia de debe estar definida de acuerdo al área de conocimiento particular para llevar a cabo la estrategia instruccional en el desarrollo de habilidades algorítmicas.</p>	<ul style="list-style-type: none"> <li>❖ Que diseñen todas las combinaciones de contenido que diría que sí que no habría ningún inconveniente siempre y cuando digamos sea un proceso como paulatino, digamos hagamos con combinaciones de una sola variable luego entonces hagamos 2, pero, entrar el reto de hacer todo de una vez está un poco como complejo</li> <li>❖ Si nos ubicamos en un escenario en el cual todos tenemos la convicción que esto nos va contribuir a muchos problemas que estamos enfrentando actualmente yo diría que es posible que los docentes inicien este proceso obviamente eso tiene una curva de aprendizaje lento, ¿correcto? “.</li> <li>❖ Porque acá es la clave eso, un alumno cuando se considera novato que instrumento se utiliza no? en el modelo que va haber o cómo va trabajarla quizás por el conocimiento adquirido pasa de un nivel novato a aprendiz o experto pero</li> </ul>	<p>En el modelo de dominio se logra corroborar que las categorías de aprendizaje en las habilidades algorítmicas deben de estar bien definidas. Los expertos se refieren a que el diseño de contenidos puede ser laborioso individualmente, pero no imposible si se hace conciencia al docente o se integra como un trabajo colaborativo e interdisciplinario. Según Donnamaria y Lage, 2009 el proceso de elaboración de los recursos didácticos tiene que ser lo más completo posible para dar el contenido y material adecuados. Además, Conejo et al. (2001) consideran que los conocimientos se deben de presentar de forma adecuada a través de explicaciones, ejercicios y contenidos, por lo tanto se deben de proponer diferentes ejercicios y soluciones para dar acompañamiento en los diferentes niveles.</p>

---

*esas evaluaciones son sumamente importantes en la ruta que efectivamente se le pueda dar una vez avance en su proceso de aprendizaje, acá lo interesante es como se enlazan esos casos esos módulos como evalúas que un alumno debe transitar de un nivel a otro, este experto refiere a que la forma en que se represente el contenido y a su vez se certamente el proceso de cambio entre niveles de aprendizaje.*

---

Fuente: Elaboración propia.

Tabla 14:  
Modelo de Interfaz de Usuario

Categoría	Eje de análisis	Lexías	Análisis
<b>Metodología de Sistema Tutor.</b> Proceso que define los elementos que componen la base teórica de los módulos de estudiante, dominio, tutor e interfaz de usuario. La metodología debe estar definida de acuerdo al área de conocimiento particular para llevar a cabo la estrategia instruccional en el desarrollo de habilidades algorítmicas.	<b>Modelo de Interfaz de Usuario.</b> Este contiene los mecanismos de representación (imágenes, sonidos, videos, lenguaje oral, etc.) del contenido a mostrar al usuario. Moreno y Mayer (2000) demuestran que principios de diseño multimedia como herramienta que potencializa el aprendizaje en sistemas de aprendizaje en el campo de la psicología cognitiva. A través de ella, se lleva a cabo la interacción hombre-máquina. Es necesario un esfuerzo adicional en el desarrollo de esta parte de la arquitectura, haciéndola intuitiva y transparente a los ojos del usuario alumno. El diseño de	<ul style="list-style-type: none"> <li>❖ Aquí están los perceptores es ahí al final de cuentas para ti solamente es captar cuando mueva el mouse o cuando haga clic ¿a qué les dio enter? para ti ¿qué es lo más importante, qué es lo que vas a recolectar de la interfaz de usuario? “Habría que ver al menos desde el punto de vista conceptual, recuperar todos esos atributos que son importantes para saber que está haciendo ya que al final son datos relevantes si tú después quisiera saber hacer un “post análisis de los datos, haber porque habrá dado tantos clics? se equivocó, esta torpe, que no entendió de la interfaz? que lo están llevando equivocarse entonces podría tomar el análisis del tiempo de respuesta sí tengo tantos componentes en la interfaz cuál es la carga cognitiva de este agente u objeto”, no se son cuestiones que debes de tener en consideración.</li> <li>❖ Simplemente como parte conceptual</li> </ul>	Si bien sólo se generaron los prototipos de la IU y a su vez se eligió un test que se acompañan de problemas representados con imágenes con el fin de hacer una carga extrínseca menor y que el estudiante realmente se concentre en la recopilación de información para alcanzar su objetivo, es decir en la carga relevante (Chong, 2005; Shaffer et al., 2003). Según las ideas de la Teoría de la Carga Cognitiva, es necesario disminuir la carga cognitiva extrínseca para aumentar el espacio de la carga relevante en la memoria de trabajo (Chong, 2005; Shaffer et al., 2003). Por definición, la carga relevante es la directamente responsable de contribuir al aprendizaje. Se constituye a partir de procesos cognitivos adecuados, como las abstracciones y las elaboraciones. Este tipo de carga está relacionada con el diseño de la interfaz, puesto que la manera como se presente la información y el tipo de actividades que se sugieran, puede

<p>la interfaz. cómo se presente la información y el tipo de actividades que se sugieran, puede favorecer el aprendizaje del individuo</p>	<p><i>sería recolectar esta información entonces que eso sería válido porque finalmente estás tratando de presentar algo que sustituye todos los sentidos no la interfaz de usuario más la entrada o el medio de entrada para el agente este donde sustituye lo que vemos lo que oímos lo que tocamos lo que degustamos y lo que olemos no? entonces muchos sentidos quedan completamente descartados en un sistema este tipo entonces el tema motor en el manejo del mouse y de presionar teclas más el de la vista que requieren de cierto procesamiento entonces yo porque valía la pena que dijeras yo tengo estos atributos y son los que yo debería recolectar en mi herramienta.</i></p>	<p>favorecer el aprendizaje del individuo. Aunque la carga relevante también se suma a la carga cognitiva total, esta representa los recursos invertidos directamente al aprendizaje del material, como la construcción de esquemas (Artino, 2008).</p>
--	---	---

Fuente: Elaboración propia.

Tabla 15:  
Valoración general

Categoría	Eje de análisis	Lexías	Análisis
<b>Metodología de Sistema Tutor.</b> Proceso que define los elementos que componen la base teórica de los módulos de estudiante, dominio, tutor e interfaz de usuario. La metodología debe estar definida de acuerdo al área de conocimiento particular para llevar a cabo la estrategia instruccional en el desarrollo de habilidades algorítmicas.	<b>Valoración General.</b> Esta categoría se refiere a los elementos metodología incorporados para el apoyo de las habilidades algorítmicas y la percepción de la aplicación de la metodología.	<p>❖ <i>Si totalmente e incluso eso o lo pensé al inicio de tu presentación que dije lo planteas en un escenario muy completo en un dominio muy específico cierto? pero yo creo que a futuro tu podrías plantear una metodología que sea independiente del dominio, quiere decir obviamente tú estás haciendo tu validación para las habilidades algorítmicas, uno finalmente tiene que delimitar su escenario de validación, pero creo que la metodología sin ningún problema se puede adaptar de forma tal en cualquier dominio de conocimiento la puede retomar, claro que sí.</i></p> <p>❖ <i>Se tienen que considerar restricciones y ahí cuando tú presentes lo que viene siendo la propuesta metodológica. Tendrías que decir cuándo y en qué casos se puede aplicar, tu misma debes definir cuáles son esas restricciones porque obviamente las metodologías por muy generales que sean no aplican a todos los casos, a toda las áreas de</i></p>	<p>Finalmente se llega a una aceptación considerable en los módulos elegidos por parte de los expertos, de acuerdo Altuna Castillo, 2014, Carbonell, 1970 y Cataldi et al., 2005, ellos mencionan que se tienen al menos cinco módulos bien definidos a los cuales en esta investigación se integran las técnicas de gamificación como se ve en la arquitectura propuesta adaptada de González et al., (2016).</p> <p>Dado que la estrategia permite diferentes etapas de detección del avance en las habilidades, ofrece flexibilidad en la estructura de las actividades (Wilson, Fernández y Hadaway, 1993) ya que se puede pasar de una etapa a otra, aportando retroalimentación si es necesario para aquellos estudiantes aprendices.</p> <p>La programación facilita un diálogo interior en el cual la retroalimentación constante y el éxito gradual empujan a los alumnos a ir más allá de sus expectativas (Stager, 2003); por lo tanto, el paso previo de creación de algoritmos promueve un desarrollo cognitivo</p>

---

*estudio, cada área de estudio llega a y mejorar su nivel de abstracción como tener sus particularidades, en este caso parte de las habilidades algorítmicas. pues habría que ver si está se podría adaptar áreas mediante meramente Al alinear las actividades diseñadas permite aprender a los estudiantes a través su preferencia en obtener el conocimiento técnicas o a otras áreas de conocimiento. como lo mencionan Najjar y Mayer, 2003.*

*Particularmente esta metodología fue probada para habilidades algorítmicas, sin descartar una generalización, pero como un trabajo futuro es posible continuar con las validaciones necesarias en otras áreas de conocimiento para hacer generalizaciones.*

---

Fuente: Elaboración propia.

Finalmente, de acuerdo a la validación del instrumento para evaluar la metodología, los expertos están de acuerdo con los módulos elegidos y los elementos para definir las características del estudiante y sobre todo la estrategia didáctica que acompaña y desmenuza las actividades para los diferentes tipos de estudiantes.

Después de hacer la categorización se aplicó la entrevista con preguntas abiertas y escala Likert a docentes investigadores de nivel superior especialistas en pedagogía, psicología, docentes de programación y en Ciencias de la Computación. A continuación se extraen los conceptos generales por especialidad de los expertos. Algunos docentes se limitan en su área pero a su vez aportan de acuerdo a su expertise, por lo que sus recomendaciones son sumamente valiosas para la evaluación de la metodología.

*Pedagogía:*

- Existe una coherencia de las teorías integradas en la estrategia didáctica y en los modelos teóricos.
- Fundamenta el acompañamiento en relación a la atención de las necesidades del estudiante.
- Genera las bases para un estudio experimental.

Recomiendan:

- Medir algunos aspectos como: hábitos de estudio, esfuerzo, lógica matemática.
- Comparación de niveles de logro (ranking).

*Psicología:*

- Se logra definir el proceso de acompañamiento y los elementos creados en la Interfaz para minimizar la carga extrínseca.
- Se recomienda la implementación de la metodología a través de algún prototipo.

### Programación:

- Integra una estrategia clara y de acompañamiento a partir de las características necesarias para el estudiante de nuevo ingreso.
- La metodología presenta un proceso completo que se debe implementar.
- Respaldan el grado de vinculación en el proceso de pedagogos, psicólogos vinculados en el uso de TICs para la creación de los contenidos.
- Recomiendan extender la metodología a otros contenidos (materias más avanzadas).

### Ciencias de la Computación

- La metodología es completa teóricamente.
- Se recomienda integrar elementos para su posterior implementación.

Los datos obtenidos en las entrevistas que se hicieron a los estudiantes después de haber terminado el taller se muestran en las siguientes tablas donde se describen las categorías, los ejes de análisis, las lexías y su análisis de acuerdo a la teoría.

Tabla 16:  
*Experiencia durante la estrategia didáctica*

Categoría	Eje de análisis	Lexías	Análisis
<b>Módulo Tutor:</b> Se encarga de definir el tipo de materiales y contenido que se presenta al estudiante de acuerdo al nivel de conocimiento que tenga. La estrategia didáctica es integrada de actividades, ejercicios, elementos de gamificación (puntaje), desarrollo y seguimiento de algoritmos.	<b>Experiencia en la estrategia didáctica:</b> Es el proceso de llevar a cabo actividades de tipo académicas que ayudan al estudiante a desarrollar habilidades y conocimientos.	<ul style="list-style-type: none"> <li>➤ <i>Fue agradable y entretenida ya que aprendimos los fundamentos de la programación de una manera más amena, agradable o cómoda que mediante teoría y algoritmos o precódigo.</i></li> <li>➤ <i>Fue algo nuevo para mí, y muy interesante. Pues conocí un poco de lo mucho que me falta por aprender.</i></li> <li>➤ <i>Ha sido interesante el trabajar con una plataforma en la cual creas pequeños juegos y valoras el trabajo que hay atrás para poder lograr que funcione perfectamente.</i></li> <li>➤ <i>Muy satisfactorio, respecto a lo que nos enseñó en el poco tiempo.</i></li> <li>➤ <i>Excelente, sobre todo la dinámica de generar actividades interactivas, en este caso los juegos que programamos.</i></li> <li>➤ <i>Mi experiencia que tuve fue de gran utilidad ya que fue interesante este curso que me impartió la maestra, debido a que reforcé aún más los conocimiento independientemente de la materia que estoy cursando, para en un futuro esto me sirva de mucho.</i></li> <li>➤ <i>Nos ayudó a ir mejorando nuestras habilidades de</i></li> </ul>	<p>Consideran que fue amplio el panorama de contenidos en el poco tiempo que se impartió. El integrar aspectos de gamificación y realizar actividades de tipo juego hacen la estrategia interactiva (Astudillo, Bast y Willging, 2016).</p> <p>Los estudiantes rescatan que su experiencia en el taller les ayuda a mejorar las habilidades de razonamiento, mencionan que su panorama y habilidad para crear algoritmos se han incrementado. Les permite crear patrones para llegar al resultado. De acuerdo a los comentarios, se considera que tomar como base el Modelo de Polya (1965) para la resolución de problemas que les ayudo a estructurar sus ideas, escribirla,</p>

---

*razonamiento para la solución de problemas tanto analizarlas y probarlas. de forma individual como en grupo.*

- *Si me dio una idea más de todo lo que podemos crear.*
  - *Ayuda a realizar un patrón para poder generar un resultado.*
- 

Fuente: Elaboración propia.

Tabla 14:  
*Experiencia durante la estrategia didáctica (Continuación...)*

Categoría	Eje de análisis	Lexías	Análisis
<b>Módulo</b> <b>Tutor:</b> Se encarga de definir el tipo de materiales y contenido se presenta al estudiante de acuerdo al nivel de conocimiento que tenga. La estrategia didáctica es integrada de actividades, ejercicios, elementos de gamificación (puntaje), desarrollo y seguimiento de algoritmos.	Ventajas de la estrategia	<ul style="list-style-type: none"> <li>➤ Pues como ventaja yo creo que todos los niños o los no tan niños, pero las personas que hacen algo por primera vez se guían por instinto o curiosidad y como ya había dicho, tanto un niño como un novato en algún tema aprenden sobre todo jugando y tratando de comprender el ¿Por qué? de las cosas o como es que funcionan o que hacen, en cuanto a la secuencia de aprendizaje me pareció bien ir poniendo los pasos, instrucciones o fórmulas de creación del juego al principio y después ir omitiendo partes y tratar de que el alumno intente razonar que falta o como mejorar el juego o programa por su cuenta sin apoyo del maestro/a.</li> <li>➤ La plataforma que usamos era muy cómoda y accesible, aunado a eso el apoyo de la maestra.</li> <li>➤ Que puedes ir poco a poco y de una manera divertida</li> <li>➤ Considero prudente el hecho de realizar actividades en una plataforma que tenga el grado de complejidad básico para comenzar en el mundo de la programación, además de estar</li> </ul>	Consideran los estudiantes una estrategia buena para que los novatos aprendan o comprendan como funciona, la secuencia de pasos apoya a generar las instrucciones, e ir omitiendo los pasos. El agregar un reto más se alienta a razonar sin la ayuda del docente. La escritura de bloques de instrucciones simples para generar algoritmos cada vez más complejos (Gerjet et al., 2004; Renkl et al., 2004; Renkl, et al., 2009). El avance paulatino de la

muy completa y como a partir de códigos básicos podemos generar algo mayormente complejo.

- Ayuda a aprender y asumir conceptos matemáticos como por ejemplo: coordenadas, variables, algoritmos, sonidos al crear escenarios reales.
- Ayuda a aprender los fundamentos de la programación.

complejidad y la gamificación como parte del diseño de las actividades con puntaje y llevar a cabo el procedimiento.

---

Tabla 14:  
Experiencia durante la estrategia didáctica (Continuación...)

Categoría	Eje de análisis	Lexías	Análisis
<b>Módulo Tutor:</b> Se encarga de definir el tipo de materiales y contenido se presenta al estudiante de acuerdo al nivel de conocimiento que tenga. La estrategia didáctica es integrada de actividades, ejercicios, elementos de gamificación (puntaje), desarrollo y seguimiento de algoritmos.	Sugerencia en las actividades	<ul style="list-style-type: none"> <li>➤ <i>Tal vez incluirlo en el curso de inducción a la carrera y que dure mínimo un mes como el curso de inducción, si es posible incluirlo como una materia de primer semestre, Al incrementar la complejidad de taller libre para primer semestre o complementarla para los los ejercicios dar más puntaje (o chicos tanto si son de nuevo ingreso como de recurso de puntos extras o cuando encuentre metodología de la programación, ya que si me parece una otra forma de hacerlo en menos actividad fascinante.</i></li> <li>➤ <i>Aumentar el número de sesiones, para así avanzar un instrucciones, por lo tanto se van poco más en el manejo de la plataforma porque en lo mejorando sus habilidades a personal es realmente interesante y atractiva.</i></li> <li>➤ <i>Ir subiendo de a poco la complejidad según se observe complejidad de los ejemplos y que el alumno puede o no resolver los problemas agregando puntaje en las planteados, ir poniendo puntaje a los pasos del ejercicios y actividades que implican aspectos puntos extras si se descubre otra forma de hacerlo o se de gamificación (Kapp, 2012) e hace el ejercicio y se quiere mejorar o añadir más cosas a interacción con el usuario (Cataldi la idea original y se logran hacer..</i></li> <li>➤ <i>Sobre todo por el tiempo, el grado de complejidad en lo potencialización del sistema de personal está muy bien ya que es algo que se puede aprendizaje (Moreno y Mayer aprender en pocas sesiones y si llegara a aumentar esto (2000) igual que las sesiones, sería fantástico.</i></li> <li>➤</li> </ul>	

Tabla 16:  
*Experiencia durante la estrategia didáctica (Continuación...)*

Categoría	Eje de análisis	Lexías	Análisis
<b>Módulo Tutor:</b> Se encarga de definir el tipo de materiales y contenido se presenta al estudiante de acuerdo al nivel de conocimiento que tenga. La estrategia didáctica es integrada de actividades, ejercicios, elementos de gamificación (puntaje), desarrollo y seguimiento de algoritmos.	Dificultades	<ul style="list-style-type: none"> <li>➤ <i>Fueron un reto en el sentido de que es la primera vez que usamos un lenguaje de programación aunque este orientado a niños, en el caso de un mayor grado de complejidad, al ir resolviendo un problema para un juego simple. Al comienzo aunque era algo nuevo para mí, fue aumentando en su grado de complejidad.</i></li> <li>➤ <i>Al principio se me dificultó pero después de partir de las instrucciones, es decir la explorar todas las herramientas me gustó y me ayudó</i>. habilidad de crear es un elemento</li> <li>➤ <i>Sí, porque primeramente esta carrera de Ciencias importante para desarrollar la actividad de la Computación para mí es muy interesante y en para escribir los algoritmos, lo que se base a esto también me interesa saber, aprender y contrasta con el aumento de la superarme específicamente en programación, en complejidad en las actividades, que pocas palabras para mí fue y sigue siendo un reto para permitir generar retos cognitivos que son importantes para resolver problemas a aprender más.</i></li> <li>➤ <i>No tanto en las estrategias, sino más bien en el tiempo que se le invirtió a las sesiones además de estar distanciadas, en general considero que es un excelente taller para conocer lo complejo del mundo de la programación a partir de una herramienta más básica pero que al conocerse bien se puede desarrollar algo muy complejo</i>.</li> </ul>	<p>Su reto fue comenzar a escribir "código" a</p> <p>habilidad de crear es un elemento</p> <p>permite generar retos cognitivos que son importantes para resolver problemas a</p> <p>(Gerjet et al., 2004; Renkl et al., 2004; Renkl, et al., 2009).</p>

Tabla 16:  
*Experiencia durante la estrategia didáctica (Continuación...)*

Categoría	Eje de análisis	Lexías	Análisis
<b>Módulo Tutor:</b> Se encarga de definir el tipo de materiales y contenido se presenta al estudiante de acuerdo al nivel de conocimiento que tenga. La estrategia didáctica es integrada de actividades, ejercicios, elementos de gamificación (puntaje), desarrollo y seguimiento de algoritmos	Resultados positivos de la estrategia	<ul style="list-style-type: none"> <li>➤ <i>Al ir relacionando la teoría con la práctica, mediante la creación de juegos de mediana dificultad.</i></li> <li>➤ <i>En las clases de metodología se me resulta familiar las estructuras de los diferentes tipos de ciclos.</i></li> <li>➤ <i>Podía ir a la par con algunas cosas que vamos viendo en clase.</i></li> <li>➤ <i>Creo que si lo sigo practicando a futuro puede ayudarme.</i></li> <li>➤ <i>Es algo que no había realizado antes y fue una excelente plataforma para implementar conocimiento nuevo.</i></li> <li>➤ <i>Se podría decir de alguna manera si porque pude y comprendo más estos temas de programación.</i></li> </ul>	<p>La creación de actividades que se relacionarán con aspectos de interés de los estudiantes como los juegos y la manipulación de objetos para realizar una tarea les permitió vincular lo aprendido con lo real. Zichermann y Cunningham (2011) precisan que los elementos de juego promueven y motivan a la resolución de problemas. Por lo tanto, se confirma que a los estudiantes les ayudó a implementar los ejercicios con la estrategia didáctica a través de la gamificación (Cataldi y Lage, 2009), integrando conocimiento nuevo y generaron relación de los temas vistos con las clases (Moroni y Señas, 2005).</p>

Tabla 14:

*Experiencia durante la estrategia didáctica (Continuación...)*

Categoría	Eje de análisis	Lexías	Análisis
<b>Módulo Tutor:</b> Se encarga de definir el tipo de materiales y contenido se presenta al estudiante de acuerdo al nivel de conocimiento que tenga. La estrategia didáctica es integrada de actividades, ejercicios, elementos de gamificación (puntaje), desarrollo y seguimiento de algoritmos	Ánimo durante la actividad	<ul style="list-style-type: none"> <li>➤ <i>Al principio nervioso o frustrado ya que nunca en la vida había programado, después de ver que Lee y Hammer, 2011, argumentan que la crearíamos nuestros propios juegos y la forma de ser gamificación busca atraer a los estudiantes a y explicar de la profesora volvieron la actividad más través de áreas cognitivas, sociales y amena y divertida, la motivación la encontré en el emocionales, y los estudiantes mencionan punto de hacer funcionar mi juego como primer varios estados de ánimo, desde el nerviosismo punto e irlo mejorando de a poco hasta lograr un al enfrentarse a cosas que no habían hecho juego jugable y con todos o incluso más datos y pensando que era muy difícil programar, variables de los que la profesora pedía.</i></li> <li>➤ <i>Motivada, con un poco de frustración a la vez, tarea que tenían que desarrollar, y motivados porque hubiera querido tener bases de ello para cuando veían que funcionaba lo que ellos sentirlo más fácil y haber resuelto todo completo".</i></li> <li>➤ <i>Fue una mezcla de frustrado, alegre, motivado, pero alcanzaban los puntos extras o bien ellos creen que eso te impulsa a que quieras seguir mismos generaban adaptaciones a los juegos aprendiendo más y más.</i></li> <li>➤ <i>De pronto un poco cansado por el horario, pero lejos Cunningham, 2011). Por lo tanto, se han de eso las actividades me mantuvieron activo, pude involucrado aspectos cognitivos y emocionales, resolver ciertas dudas con ayuda de la profesora.</i></li> <li>➤ <i>Al principio un poco nervioso, porque la verdad fondo en esa investigación pero es un aspecto pensé que esto de la programación era difícil, pero que se está detectando dentro de los resultados. después de estas sesiones ahora me siento motivado.</i></li> </ul>	<p>frustrados cuando no les salía la lógica de la</p> <p>habían creado, y finalmente alegres cuando</p> <p>creaban sus propios retos (Zichermann y</p> <p>aunque estos últimos no se han abordado a</p> <p>fondo en esa investigación pero es un aspecto</p>

Fuente: Elaboración propia.

La metodología que se plantea en el Capítulo 4, se validó a través de expertos que evaluaron cada uno de los módulos en una escala Likert de aceptable y haciendo algunas observaciones desde distintas perspectivas, a fin de continuar con líneas de investigación para este proyecto.

Por otra parte, se hizo la validación de la estrategia didáctica planteada en la metodología en un grupo focal de estudiantes de nuevo ingreso a la FCC. Los datos obtenidos se integraron en las tablas haciendo el análisis de lexías para determinar la importancia de la secuencia didáctica, sus ventajas, sus beneficios, su dificultad, si apoya en sus habilidades; son algunos aspectos que se observaron en su participación durante el taller. Además, los resultados que se identificaron en la comparación de resultados del pretest y postest como se observa en las tablas 9 a 14, los estudiantes que llegaron en un nivel de aprendiz con habilidades a mejorar, no necesariamente mejoraron pasando a otro nivel, sino que mejoraron sus habilidades de Aplicar, Analizar y Evaluar, sin embargo aún falta trabajar la habilidad de Crear para alcanzar en nivel de experto. Lo cual demuestra que la intervención fue satisfactoria y que la metodología funciona para apoyar a las habilidades algorítmicas de los estudiantes de nuevo ingreso.

## 7. Conclusiones

Los estudios que se realizaron en esta investigación muestran que las habilidades de los estudiantes universitarios que ingresan a carreras del área de Ciencias de la Computación son deficientes. Existe ausencia del dominio de los aspectos cognitivos e instrumentales relacionados con éstas habilidades para resolver problemas mediante algoritmos, tanto para su enseñanza y evaluación (Moroni y Señas, 2005; Robins, Rountree, y Rountree, 2003; Rosanigo y Paur, 2006).

Algunas de las preguntas a contestar durante este trabajo son: ¿Cómo plantear la metodología sustentada en la literatura para el diseño de sistemas tutores? ¿Qué es lo que se necesita adaptar para mejorar la disposición de aprender de los estudiantes de nuevo ingreso? ¿Qué se adapta del modelo del sistema tutor para apoyar las habilidades algorítmicas de los estudiantes?

Los aspectos a considerar para el aprendizaje son las características de los estudiantes y en qué área se requiere enseñar; de tal manera que de acuerdo a la epistemología del área se individualice el proceso de enseñanza-aprendizaje y se focalice la naturaleza del área de conocimiento. La investigación abona al área educativa puntualmente al área de instrucción en el aprendizaje de las habilidades básicas de programación.

Este trabajo se basa en el paradigma de la investigación cualitativa que se caracteriza por comprender el significado de la variedad de realidades por parte de los sujetos que las viven e interpretan, dado que el aprendizaje de cada persona es diferente y cambiante, este paradigma ayuda a detectar aspectos que complementan otras investigaciones en el área del aprendizaje de programación. Rodríguez Gómez y Valldeoriola Roquet (2009) exponen que para comprender las realidades educativas, particularmente volcadas al mundo virtual exponen algunos métodos que en su opinión son los más adecuados: etnografía, estudio de caso, teoría fundamentada, investigación-acción, e investigación basada en el diseño. Este trabajo se fundamenta en la investigación basada en el diseño (IBD), la cual se centra en planteamientos metodológicos que vincularan investigación, diseño

educativo e innovación (Brown y Collins, 1992), particularmente se apoya en la fundamentación de la teoría.

En el modelo de estudiante por el momento se integraron dos factores: el estilo de aprendizaje y el nivel cognitivo. Sin embargo, se encontraron diferentes factores que pueden afectar el proceso de aprendizaje: el razonamiento y la motivación, esta última fue mencionada por los estudiantes durante la aplicación del taller. El factor de razonamiento fue un factor detectado por los docentes expertos ya que lo consideran como un elemento valioso para las habilidades algorítmicas y la capacidad de resolver problemas en el nivel superior.

La investigación propone una estrategia didáctica que se integra en la propuesta metodológica expuesta en el capítulo 5, en apoyo a las habilidades algorítmicas particularmente para los estudiantes de nuevo ingreso en la Facultad de Ciencias de la Computación.

Un hallazgo que se determinó fue en la parte metodológica, dado que el instrumento elegido se adaptó a las necesidades de evaluar las habilidades para desarrollar algoritmos en el área de la programación básica. Además de identificar el nivel novato, aprendiz o experto, la clasificación que se generó ayuda a determinar cuál es la habilidad que necesita fortalecer el estudiante y dar el seguimiento puntual de su desarrollo cognitivo en la resolución de problemas mediante el algoritmos.

Los expertos definían que es más fácil que la mente sea capaz de retener y recuperar información a través de imágenes, por lo tanto, para el nivel inicial en que se trabaja es necesario hacer la vinculación de los procesos básicos para la resolución de problemas y generar menor carga extrínseca, por lo que el instrumento que se eligió tiene preguntas acompañadas de imágenes a analizar (Paivio, 1990; Sadoski y Paivo, 2005), contrastando con los resultados de la aplicación del taller donde a los estudiantes se les notaba mayor interés y les permitía analizar mejor el problema.

Durante el trabajo de campo se detectó la necesidad de fortalecer las necesidades cognitivas de acuerdo a las habilidades que el estudiante domina,

sugiriendo un proceso de aprendizaje previo en los casos de estudiantes que no habían tenido un aprendizaje en el área. Al aplicar la estrategia en un grupo que ya había tomado la materia se determinaba que no tenía mucha dificultad los ejercicios propuesto, lo cual argumenta la necesidad identificar su nivel de conocimiento y el referente de la teoría de la carga cognitiva para no inhibir su aprendizaje, sino por el contrario proponer ejemplos con mayor grado de complejidad. El hallazgo es que se tienen que integrar la estrategia didáctica inicialmente con ejemplos resueltos para desarrollar paulatinamente las habilidades y posteriormente integrar prácticas guiadas que ameriten mayor interacción con el estudiante. (Gerjets, Scheiter y Catrambone, 2004; Renkl, Atkinson y Grobe, 2004; Renkl, Hilbert y Schworm, 2009).

En los resultados de las entrevistas a los expertos con base a los cuestionamientos en las categorías propuestas, se detectó una categoría extra denominada Tecnológica, en el sentido de separar la teoría que fundamenta la metodología y por otra parte los aspectos de implementación, que por el momento quedan fuera del alcance de este trabajo pero que es importante como trabajo futuro.

Los estudiantes de nuevo ingreso a nivel superior del área de Computación son los principales beneficiados debido a que se propone una instrucción diseñada para el área de desarrollo de habilidades algorítmicas. Al desarrollar las habilidades algorítmicas, tendremos la certeza de que podrán resolver problemas más complejos tratando de crear nuevas estrategias posteriormente cuando conozcan algún lenguaje de programación.

## 7.1. Trabajo futuro

En esta investigación se propone la metodología de diseño de Sistema Tutor en apoyo a las habilidades algorítmicas, en un escenario particular con los estudiantes de nuevo ingreso a la carrera de Ciencias de la Computación de la BUAP. Las categorías y modelos generados incorporan los elementos claves para el desarrollo

del Sistema Tutor como una herramienta para apoyar al docente y al alumno la cual se desarrollará a futuro, por lo que los puntos a continuar son:

- Crear el prototipo y realizar pruebas del diseño creado.
- Realizar el diseño de la metodología incluyendo técnicas para la implementación, en la categoría que definieron los expertos como Tecnológica.
- Integrar la evaluación de usabilidad para saber la satisfacción del uso de la herramienta y su eficiencia.
- Identificando la preferencia del estilo de aprendizaje del estudiante es posible que se pueda dar ésta información a los docentes para mejorar su práctica educativa.
- Los resultados obtenidos de forma local pueden expandir la propuesta a las áreas de ciencias e ingeniería en el sistema educativo de nivel superior.
- Después de tener el sistema, se pueden medir los eventos de interacción con el usuario para ver su disponibilidad, su nivel de entendimiento, su aprendizaje, sus errores o el tiempo para resolver el ejercicio.

# Referencias

- Abu-Naser, S. (2008). Developing an Intelligent Tutoring System for Students Learning to Program in C++. *Information Technology Journal*, 7(7), 10055–1060.
- Abu Naser, S. (2009). Evaluating the effectiveness of the CPP-Tutor, an intelligent tutoring system for students learning to program in C ++. *Journal of Applied Sciences Research*, 5(1), 109–114.
- Abu-Naser, S., Ahmed, Al-Masri, N., Deeb, a, Moshtaha, E., y Abu Lamdy, M. (2011). An Intelligent Tutoring System for Learning Java Objects. *International Journal of Artificial Intelligence y Applications*, 2(2), 68–77. <https://doi.org/10.5121/ijaia.2011.2205>
- Aguirre-Chaves, M. (2007). Algunas reflexiones en torno al estilo de aprendizaje empleado por estudiantes universitarios. *Revista Electrónica Educare*, 11(2), 1–15. Recuperado a partir de <http://www.revistas.una.ac.cr/index.php/EDUCARE/article/view/1336>
- Alba, J. Q., Elola, J. J. C., y Luffiego, G. M. (2008). *Las competencias básicas en las áreas de Ciencias* (Consejería). Consejería de Educación de Cantabria. Retrieved from [http://www.educantabria.es/docs/info\\_institucional/publicaciones/2008/Cuadernos\\_Educacion\\_4.pdf](http://www.educantabria.es/docs/info_institucional/publicaciones/2008/Cuadernos_Educacion_4.pdf)
- Aleven, V., Baker, R., Wang, Y., Sewall, J., y Popescu, O. (2016). Bringing Non-programmer Authoring of Intelligent Tutors to MOOCs. In *Proceedings of the Third ACM Conference on Learning @ Scale* (pp. 313–316). <http://doi.org/10.1145/2876034.2893442>
- Aleven, V., McLaren, B., y Sewall, J. (2006). Tutorial on rapid development of intelligent tutors using the Cognitive Tutor Authoring Tools (CTAT). In *6th International Conference on Advanced Learning Technologies, ICALT 2006* (Vol. 2006). Retrieved from <http://www.scopus.com/inward/record.url?eid=2-s2.0-34247173574&partnerID=40ymd5=b81eb6f38e7f78efb9ff14fcf0ba47e2>
- Aleven, V., McLaren, B. M., Sewall, J., y Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems : Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19, 105–154. Retrieved from <http://iospress.metapress.com/index/X3760U67H22LM111.pdf>
- Alonso C.M., Gallego, D.J. y Honey, P. (1999). Estilos de Aprendizaje. Bilbao: Ediciones Mensajero.
- Altuna, C. E. J., Guibert, E. L., y Estrada, S. V. (2014). Método Para La Construcción Del Modelo De Dominio En Un Tutor Inteligente De Programación. *Revista Cubana de Ciencias Informáticas*, 8(4), 100–115. Retrieved from <http://scielo.sld.cu/pdf/rcci/v8n4/rcci06414.pdf>
- Aparicio, F. A., Gutierérrez, V. L. F., González, J. L., y Isla, M. J. L. (2012). Método de análisis y aplicación de la gamificación.
- Astudillo, G. J., Bast, S. G., y Willging, P. A. (2016). Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación A game-based approach for learning a programming language, 12(2015), 125–142.
- Artino, A. R. (2008). Cognitive Load Theory and the Role of Learner Experience: An Abbreviated Review for Educational Practitioners. Association for the Advancement of Computing In Education Journal, AACE Journal, 16 (4), 425-439. Disponible en: [editlib.org/d/25229](http://editlib.org/d/25229).

- Ausubel, D., Novak, J. y Hanesian, H. (2000). Psicología educativa. Un punto de vista cognositiva. México: Trillas.
- Baker, R. S., Corbett, A. T., Koedinger, K. R., y Wagner, A. Z. (2004). Off-task behavior in the cognitive tutor classroom: When students “game the system.” In E. Dykstra-Erickson y M. Tscheligi (Eds.), Proceedings of the 2004 conference on human factors in computing systems (pp. 383–390). New York, NY: ACM Press.
- Barrón-Estrada, M. L., Zatarain-Cabada, R., Aranda-Ortega, M., Gómez-Pérez, H. E., y Mejía-Arredondo, J. (n.d.). Un tutor inteligente , afectivo y configurable para el aprendizaje de números naturales de 3er grado. In *Research in Computing Science* (Vol. 77, pp. 45–54).
- Bartó, C. A., y Díaz, L. C. (2012). Proyecto : sistemas inteligentes aplicados a la enseñanza de la programación en ingeniería Resumen Contexto Introducción. *XIV Workshop de Investigadores En Ciencias de La Computación*, pp. 1051–1055.
- Bell, P.(2004). "On the theoretical breadth of design-based research in Education". *Educational Psychologist*. Vol. 4, núm. 39, pp. 243-253
- Böhm, C., Jacopini, G., Comm. (1966). Teorema Fundamental de la programación estructurada, ACM vol.9, nº5, 366-371
- Bloom, B., and Engelhart, M., (1969). Taxonomy of Educational Objectives: The Classification of Educational Goals. David McKay.
- Byrne, P. y Lyons, G. (2001)."The effect of student attributes on success in programming". *Proceedings of ITiCSE*, pp. 49-54. ACM Press. NY, NY, USA.
- Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions. *Journal of the Learning Sciences*, 2, 141–178.
- Bruno, O. R. (2005). incorporación de un sistema tutor inteligente como facilitador del aprendizaje de algoritmia ., 2(4), 1–31.
- Camacho, J. S, "Sami:Sofbot de charla e desarrollo con la técnica de razonamiento Basado en Casos".2006.
- Carbonell, J. . (1970). AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction. *Man-Machine Systems, IEEE Transactions*, 11(4), 190–202. Retrieved from <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&yearNumber=4081977>
- Cataldi, Z., Salgueiro, F., Lage, F., y García-Martínez, R. (2005). Sistemas tutores inteligentes: los estilos del estudiante para selección del tutorizado. In *VII Workshop de Investigadores en Ciencias de la Computación* (pp. 66–70). Buenos Aires, Argentina. Retrieved from <http://sedici.unlp.edu.ar/handle/10915/21144>
- Cataldi, Z., Figueroa, N., Méndez, P., Lage, F. J., Vigliecca, M. E., y Kraus, G. (2006). Herramienta automatizada para la determinación de los estilos de aprendizaje en ingresantes a cursos de Programación Básica. In *Congreso Argentino en Ciencias de la Computación - CACIC 2006* (pp. 952–961).
- Cataldi, Z., y Lage, F. (2009). Línea de investigación: Sistemas Tutores Inteligentes orientados a la

enseñanza para la comprensión. In *XI Workshop de Investigadores en Ciencias de la Computación* (pp. 650–654). Buenos Aires, Argentina. Retrieved from <http://sedici.unlp.edu.ar/handle/10915/19849>

Cataldi, Z., y Lage, F. (2009). Sistemas Tutores Inteligentes Orientados a la Enseñanza para la comprensión. En Edutec Revista Electrónica de Tecnología. pp. 650– 654. Recuperado a partir de [http://edutec.rediris.es/Revelec28/articulos\\_n28\\_pdf/Edutec-E\\_Cataldi\\_Lage\\_n28.pdf](http://edutec.rediris.es/Revelec28/articulos_n28_pdf/Edutec-E_Cataldi_Lage_n28.pdf)

Chong, T. S. (2005). Recent Advances in Cognitive Load Theory Research: Implications for Instructional Designers. *Malaysian Online Journal of Instructional Technology (MOJIT)*, 2 (3), 106-117.

Clements, D.H. y Meredith J., (1992). Research on Logo: Effects and Efficacy. Logo Fundation. State University of New York at Buffalo. Retrieve from: [http://el.media.mit.edu/logo-foundation/resources/papers/pdf/research\\_logo.pdf](http://el.media.mit.edu/logo-foundation/resources/papers/pdf/research_logo.pdf)

Collins, A. (1992) Toward a design science of education. In E. Scanlon y T. O'Shea (Eds.), *New directions in educational technology*. Berlin: Springer-Verlag.

Conejo, R., Millán, E.; Pérez de la Cruz, J. y Trella, M. (2001) Modelado del alumno: un enfoque bayesiano . ETSI Informática, Universidad de Málaga, España.

Costa, G.; Salgueiro, F. A., Cataldi, Z., García Martínez, R. y Lage, F. J. 2005. Sistemas inteligentes para el modelado del estudiante. Proceedings.

Danilov, M.; Skatin, M. (1985). Didáctica de la escuela Média. Havana, Pueblo y Educación.

Dunn, R. Dunn K. and Price, G. (1985). Manual: Learning Style Inventory, KS: Price Systems, Lawrence.

Díaz Pulecio, Laura Jeannette (1993): Recreto matemático 5, cuaderno de actividades; Editorial Voluntad, Bogotá.

Dunn, R., Dunn, K. y Price, G. E. (1979). Identifying Individual Learning Styles. En National Association of Secondary School Principals (US). *Student learning styles: Diagnosing and prescribing programs*. Reston, Virginia: Natl Assn of Secondary School, pp. 39-54.

Dunn, R. Dunn K. and Price, G. (1985). Manual: Learning Style Inventory, KS: Price Systems, Lawrence.

Felder, R. M., y Silverman, L. K. (1988). Learning and Teaching Styles. *Engr. Education*, 78(7), 674– 681.

Felder, R. M. y Spurlin, J. E. (2005). Applications, reliability, and validity of the Index of Learning Styles. *International Journal of Engineering Education*, 21(1), 103

Felder, R. M. y Solomon, B. A. (2004). Index of learning styles questionnaire. North Carolina State University. Consultado de: <http://www.enr.ncsu.edu/learningstyles/ilsweb.html>

Gardner, H (1993). *Inteligencias Múltiples: La teoría en la práctica*. Paidós. Barcelona, Buenos Aires, México.

Gardner, H. (2010). *Estructuras de la mente*. México: Fondo de Cultura Económica.

Godino, J. et al (2004), "Didáctica de las matemáticas para docentes". Proyecto Edumat Docentes. Universidad.

González, C., Mora, A., y Toledo, P. (2014). Gamification in intelligent tutoring systems. In *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM '14* (pp. 221–225). <http://doi.org/10.1145/2669711.2669903>

González, C. S., Toledo, P., y Muñoz, V. (2016). Enhancing the engagement of intelligent tutorial systems through personalization of gamification. *International Journal of Engineering Education*, 32(1), 532–541. Retrieved from <http://www.scopus.com/inward/record.url?eid=2-s2.0-84959421037&partnerID=tZOTx3y1>

González, Duque, M. N., y Ovalle, C. D. (2008). Modelo del Estudiante para Sistemas Adaptativos de Educación Virtual. *Revista Avances En Sistemas E Informática*, 5(1), 199–206. Retrieved from <http://www.redalyc.org/articulo.oa?id=133114993004>

Guzdial, M. (2000). Soporte tecnológico basado en proyectos. En: Aprendiendo con tecnologías. Buenos Aires, Paidos.

Guzmán, E. y Conejo, R. (2004). A model for student knowledge diagnosis through adaptative testing. Proceedings: 7th International Conference Intelligent Tutoring Systems, ITS2004, Maceió, Brasil.

Harrer, A., McLaren, B. M., Walker, E., Bollen, L., y Sewall, J. (2005). Collaboration and Cognitive Tutoring: Integration, Empirical Results, and Future Directions. In *Artificial Intelligence in Education: Supporting Learning Through Intelligent and Socially Informed Technology* (Vol. 125, pp. 266–273).

Hedlefs, A. M. I., de la Garza, G. A., Sánchez, M. M. P., y Garza, V. A. A. (2016). Adaptación al español del Cuestionario de Usabilidad de Sistemas Informáticos CSUQ / Spanish language adaptation of the Computer Systems Usability Questionnaire CSUQ. *RECI Revista Iberoamericana de Las Ciencias Computacionales E Informática*, 4(8), 84–99.

Hicks, A. (2012). Creation, evaluation, and presentation of user-generated content in community game-based tutors. In *Proceedings of the International Conference on the Foundations of Digital Games - FDG '12* (p. 276). ACM Press. Retrieved from <http://dl.acm.org/prox.lib.ncsu.edu/citation.cfm?id=2282338.2282397>.

Hederich., C. (2004). *Estilo Cognitivo en la dimensión Independencia - Dependencia de campo, Influencias culturales e implicaciones para la educación*. Tesis Doctoral. Universidad Autónoma de Barcelona. 2004. recuperado de [http://www.tdx.cesca.es/tesis\\_uab/available/tdx-1128105-155731//chm1de1.pdf](http://www.tdx.cesca.es/tesis_uab/available/tdx-1128105-155731//chm1de1.pdf)

Honey, P. y Mumford, A. (1989). *Learning styles questionnaire*. Organization Design and Development, Incorporated.

Honey, P. y Mumford, A. (1992). *The Manual of Learning Styles*. Maidenhead, Berskshire. Ardingly House.

Hooshyar, D., Ahmad, R. B., Yousefi, M., Fathi, M., Abdollahi, A., Horng, S. J., y Lim, H. (2016). A solution-based intelligent tutoring system integrated with an online game-based formative assessment: development and evaluation. *Educational Technology Research and Development*. <http://doi.org/10.1007/s11423-016-9433-x>

Kapp, K (2012) The gamification of learning: case-Based Methods and strategies for training and

education, New York, John Wiley y Sons.

Kessler, A. M., Stein, M. K., y Schunn, C. D. (2015). Cognitive Demand of Model Tracing Tutor Tasks: Conceptualizing and Predicting How Deeply Students Engage, 20(3), 317–337.  
<http://doi.org/10.1007/s10758-015-9248-6>

Kim, S., Park, J., Kim, J. W., y Oh, A. (2016). Elice: An online CS education platform to understand how students learn programming. In *Proceedings of the Third ACM Conference on Learning @ Scale* (pp. 225–228). <http://doi.org/10.1145/2876034.2893420>

Kolb, D. A., (1984) Experiential Learning: Experience as the source of learning and development. Nueva York: Prentice Hall.

Kollu, K. (2011). *Prototype of a intelligent tutoring system using the java expert system shell*. UMI. Retrieved from  
<http://search.proquest.com.etcchconrcyt.idm.oclc.org/docview/853328559/fulltextPDF/DE0DC20E55A440A4PQ1?accountid=163027>

Laureano, A. y F. de Arriaga (2000) *Reactive Agent Design for Intelligent Tutoring Systems. Cybernetics and Systems (an International Journal)*, TAYLOR y FRANCIS (Ed.) 31: 1-47

Lemus, S. C. (2010). *Estrategia metodológica para el desarrollo de un sistema tutor inteligente: módulo del dominio*. Revista Tecnológica. 19–24

Lewis, J. R. (1995). *IBM computer usability satisfaction questionnaires : Psychometric evaluation and instructions for use IBM Computer Usability Satisfaction Questionnaires : Psychometric Evaluation and Instructions for Use*. International Journal of Human- Computer Interaction (Vol. 7).

Litzinger, T. A., Lee, S. H., Wise, J. C. y Felder, R. M. (2005). *A Study of the reliability and validity of the Felder-Solomon Index of Learning Styles*. Proceedings of the 2005 American Society for Engineering Education Annual Conference y Exposition, EE. UU., 1-16.

López, R. J., Hernández, R. C., y Farran, L. Y. (2011). *Una plataforma de evaluación automática con una metodología efectiva para la enseñanza/aprendizaje en programación de computadores*. Ingeniare. Revista Chilena de Ingeniería, 19(2), 265–277. <http://doi.org/10.4067/S0718-33052011000200011>

Lee, J.J. and Hammer, J. (2011) “Gamification in Education: What, How, Why Bother?”, Academic Exchange Quarterly, vol. 15, no. 2, pp. 1-5, [Online], Available:  
<http://www.gamifyingeducation.org/files/Lee-Hammer-AEQ-2011.pdf>

Martínez, M. (2012). *La investigación cualitativa etnográfica en educación. Manual teórico-práctico* (Tercera ed.). México: Editorial Trillas.

Matsuda, N., Cohen, W. W., Sewall, J., Lacerda, G., y Koedinger, K. R. (2008). *Why Tutored Problem Solving May be Better Than Example Study : Theoretical Implications from a Simulated-Student Study*. Springer, (1), 111–121.

Melo, R. C. (2001); *Dominios 5, matemáticas para básica primaria*; Editorial Escuelas del Futuro, Bogotá.

Mitrovic, A., Koedinger, K. R., y Martin, B. (2003). *A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modeling*. In P. Brusilovsky, A. Corbett, y F. de Rosis (Eds.), *User Modeling*

- 2003 (pp. 313–322). Springer Berlin Heidelberg. Retrieved from [http://link.springer.com/chapter/10.1007/3-540-44963-9\\_42](http://link.springer.com/chapter/10.1007/3-540-44963-9_42)
- Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J., y McGuigan, N. (2009). ASPIRE: An Authoring System and Deployment Environment for Constraint-Based Tutors. *Int. J. Artif. Intell. Ed.*, 19(2), 155–188.
- Mitrovic, A., McGuigan, N., Martin, B., Suraweera, P., Milik, N., y Holland, J. (2008). Authoring constraint-based tutors in ASPIRE: a case study of a capital investment tutor.
- Mitrovic, A., Suraweera, P., Martin, B., Zakharov, K., Milik, N., y Holland, J. (2006). Authoring Constraint-Based Tutors in ASPIRE. In M. Ikeda, K. D. Ashley, y T.-W. Chan (Eds.), *Intelligent Tutoring Systems* (pp. 41–50). Springer Berlin Heidelberg. Retrieved from [http://link.springer.com/chapter/10.1007/11774303\\_5](http://link.springer.com/chapter/10.1007/11774303_5)
- Mitrovic, A., Martin, B., y Suraweera, P. (2007). Intelligent Tutors for All: The Constraint-Based Approach. *Intelligent Educational Systems*, 38–45.
- Moreno, R. y Mayer R.E. (2000). A learner-centered approach to multimedia explanations: Deriving instructional design principles from cognitive theory. *Interactive Multimedia. Electronic: Journal of Computer Enhanced Learning*
- Moroni, N., y Señas, P. (2005). Estrategias para la enseñanza de la programación. *Primeras Jornadas de Educación En Informática Y TICS En Argentina*, pp. 254–258. Bahía Blanca, Argentina.
- Moursund, D. (1999). Project-based learning using information technology. Eugene, OR: International Society for Technology in Education.
- Najjar, M., y Mayers, A. (2003). A computational cognitive-based approach to represent knowledge within intelligent tutoring systems. In *Proceedings - 3rd IEEE International Conference on Advanced Learning Technologies, ICALT 2003* (pp. 66–71). <http://doi.org/10.1109/ICALT.2003.1215028>
- NAP (2004): Computer Science: Reflections on the field, reflections from the field; [consulta en línea: National Academy Press; <http://nap.edu/catalog/11106.html>]
- Ovalle, D. and Jiménez, J., Millennium: A Learning Framework based on Integrating Model of Intelligent Tutoring Systems and Computer Supported Collaborative Learning. Proceeding 1st LEDGRAPH Workshop of 7th International Conference on ITS2004, 2004
- Orejuela, H. F., García, A. A., Hurtado, J. A., y Collazos, C. (2013). Analizando y Aplicando la Gamificación en el Proceso ChildProgramming, 7–23.
- Ocampo B. F., Guzman A. A.; Camarena G. P. y De Luna C. R. (2014). Identificación de estilos de aprendizaje en estudiantes de ingeniería. RMIE [online]. 2014, vol.19, n.61, pp.401-429. ISSN 1405-6666.
- Parvez, S. M. y Blank, G. D. (2008). Individualizing tutoring with learning style based feedback. En B. Woolf, E. Aimeur, R. Nkambou, y S. Lajoie (Eds.), *Lecture notes in computer Science: Intelligent tutoring systems Vol. 5091* (pp. 291-301). Alemania: Springer-Verlag Berlin Heidelberg.
- Peña, C. I., Marzo, J.-L., de la Rosa, L. J., y Fabregat, R. (2002). *Un sistema de tutoría inteligente*

*adaptativo considerando considerando estilos de aprendizaje. Universidad de Girona España* (Vol. 1). Retrieved from  
[http://bcds.udg.es/papers/un\\_sistema\\_de\\_tutoria\\_inteligente\\_adaptativo\\_considerando\\_estilos\\_de\\_aprendizaje.pdf](http://bcds.udg.es/papers/un_sistema_de_tutoria_inteligente_adaptativo_considerando_estilos_de_aprendizaje.pdf)

Pérez, J., y Pozo, E. (2009). *Aprender para comprender y resolver problemas.* (E. Morat, Ed.) (First). Madrid.

Pimienta, P. J., (2012). *Estrategias de enseñanza-aprendizaje. Docencia Universitaria basada en competencias.* Primera Edición. Ed. Pearson. ISBN 978-607-32-0752-2

Piaget, J. (1978) *La Equilibración de las Estructuras Cognitivas*, Madrid, Ed. Siglo XXI.

Pires, J. M., y Cota, M. P. (2016). "Intelligent" adaptive learning objects applied to special education needs: Extending the eLearning paradigm to the uLearning environment. In 2016 11th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1–6).  
<http://doi.org/10.1109/CISTI.2016.7521504>

Polya, G. (1965). "Como plantear y Resolver problemas", Trillas, 1965.

Rinaudo, M. C; A. Chiecher y D. Donolo, (2010). "La investigación basada en diseños en el estudio de los contextos virtuales de aprendizaje". Ponencia presentada en Simposio Internacional Para La Socialización De Buenas Prácticas E Investigación En Red. CIAFIC, Buenos Aires. Abril de 2010.

Robins, A., Rountree, J., y Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2):137-173

Rodriguez, A. R., Castillo, G. L. J., y Lira, C. A. L. (2013). Diseño de un sistema tutorial inteligente, 5(2013), 1–13. Recuperado a partir de  
<http://www.udgvirtual.udg.mx/apertura/index.php/apertura/rt/printerFriendly/371/310>

Rodriguez, A. R., Castillo, G. L. J., y Lira, C. A. L. (2013). Diseño de un sistema tutorial inteligente, 5(2013), 1–13. Retrieved from  
<http://www.udgvirtual.udg.mx/apertura/index.php/apertura/rt/printerFriendly/371/310>

Román-González, M., Pérez-González, J. C., y Jiménez-Fernández, C. (2015). Test de Pensamiento Computacional : diseño y psicometría general. In *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015)* . Madrid, España.  
<http://doi.org/10.13140/RG.2.1.3056.5521>

Rosado, G. C., Sánchez, G. D., y García, S. R. (2016). Sistema 4MAT apoyado con tutores inteligentes en estudiantes de Ingeniería, 1–8.

Rosanigo, Z., y Paur, A. (2006). Estrategias para la enseñanza de Algorítmica y Programación. TEyET'06, pp. 117–124.

Samsonovich, A. V, Kitsantas, A., Brien, E. O., y Jong, K. A. De. (2015). Cognitive Processes in Preparation for Problem Solving. *Procedia - Procedia Computer Science*, 71, 235–247.  
<http://doi.org/10.1016/j.procs.2015.12.218>

Salgueiro, F., Costa, G., Cataldi, Z., Lage, F., García-Martínez, R. (2005). Redefinition of basic modules of an intelligent tutoring system: the tutor module. WICC 2005. WICC 2005. 13 y 14 de mayo. Universidad Nacional de Río Cuarto. Córdoba.

- Sánchez-Román, G. (2016). Sistema Tutor para el desarrollo de habilidades algorítmicas. En 11 Congreso Colombiano de Computación.
- Sánchez, R. G., Guerrero, G. J., y Mocencahua, M. D. (2015). Hacia un Sistema Tutor Inteligente en Algoritmia. En Aportaciones en el uso de las Tecnologías para el Aprendizaje: (pp. 622–623).
- Self, J. (1990). Bypassing the intractable problem of student modeling. In C. Frasson y G. Gauthier (Eds.), Intelligent Tutoring Systems: At the Crossroads of Artificial Intelligence and Education. New Jersey: Ablex.
- Shaffer, D.; Doube, W. y Tuovinen, J. (2003). Applying Cognitive Load Theory to Computer Science Education. Paper presented at the 15th Workshop of the Psychology of Programming Interest Group, Keele UK. Disponible en: <http://www.ppig.org/papers/15th-shaffer.pdf>
- Sierra, E., Gracia-Martinez, R., Cataldi, Z., y Hossian, A. (2004). Fundamentos para una metodología de diseño de sistemas tutoriales inteligentes centrada en la reparación de mecanismos. In X Congreso Argentino de Ciencias de la Computación. Retrieved from <http://hdl.handle.net/10915/22414>
- Stash, N. V., Cristea, A. I. y De Bra, P. M. (2004). Authoring of learning styles in adaptive hypermedia: problems and solutions. En S. Feldman, M. Uretsky, M. Najork, y C. Wills (Eds.), Proceedings of the 13th International World Wide Web conference, 114-123.
- Sundararaja, S. C., y Nitta, S. V. (2015). Designing engaging intelligent tutoring systems in an age of cognitive computing, 59(6), 1–9. <http://doi.org/10.1147/JRD.2015.2464085>
- Tarongí, V. A. (2010). Sistema Tutor Inteligente Adaptativo para Laboratorios virtuales y remotos. Tesis de Master Universitario en Automática e Informática Industrial, Valencia: Universidad Politécnica de Valencia.Urretavizcaya Loinaz, M. (2001). Sistemas inteligentes en el ámbito de la educación. Revista Iberoamericana de Inteligencia Artificial, 5 (12), pp. 55-12. Recuperado de <http://www.redalyc.org/pdf/925/92551202.pdf>
- Tirado, H. J., Mancilla, E. E., y Sánchez, C. I. (2009). Metodología para realizar un Tutor Inteligente usando Razonamiento Basado en Casos ( RBC ) (pp. 1–8). MICAI. Retrieved from <http://www.micai.org/2009/proceedings/complementary/cd/ws-wile/WILE09-02.pdf>
- Tirado, H. J., Mancilla, E. L., y Cruz, S. I. (2009). Metodología para realizar un Tutor Inteligente usando Razonamiento Basado en Casos ( RBC ). 8th Mexican International Conference on Artificial Intelligence, 1–8. Retrieved from <http://www.micai.org/2009/proceedings/complementary/cd/ws-wile/WILE09-02.pdf>
- Trella, M. (2006). MEDEA : MEtodologías y herramientas para el Desarrollo de entornos inteligentes de Enseñanza y Aprendizaje. Revista Iberoamericana de Inteligencia Artificial, 10(32), 77–80. Retrieved from <http://www.redalyc.org/articulo.oa?id=92503210>
- URBANCIC, J. y TRAMPUS, M. Putka - A Web Application in Support of Computer Programming Education. International Journal Olimpiads in Informatics, 2012, 6(1): p. 205-211.
- Vygotsky, L. (1978) El desarrollo de los procesos psicológicos superiores, Harvard University Press.
- Vygotsky, L. (1978) El desarrollo de los procesos psicológicos superiores, Harvard University Press.
- Vygotsky, L.S. (1979). El desarrollo de los procesos psicológicos superiores. Barcelona: Crítica.

Walker, E., Rummel, N., McLaren, M., y Koedinger, K. R. (2007). The student becomes the master: Integrating peer tutoring with cognitive tutoring. In CSCL'07 Proceedings of the 8th international conference on Computer supported collaborative learning (pp. 751–753). Retrieved from [http://delivery.acm.org/10.1145/1600000/1599737/p751-walker.pdf?ip=148.228.23.96&id=1599737&acc=ACTIVE&SERVICEkey=6F4CCF05E2930152.35F2A2F115BE1A28.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=598070848yCFTOKEN=80658943y\\_acm\\_=1459991980\\_b85e71aed6d1791991c11](http://delivery.acm.org/10.1145/1600000/1599737/p751-walker.pdf?ip=148.228.23.96&id=1599737&acc=ACTIVE&SERVICEkey=6F4CCF05E2930152.35F2A2F115BE1A28.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=598070848yCFTOKEN=80658943y_acm_=1459991980_b85e71aed6d1791991c11)

Wilson, J. W., Fernandez, M. L. y Hadaway N. (1993). Mathematical problem solving. New York: MacMillan. <http://jwilson.coe.uga.edu/emt725/PSsyn/PSsyn.html>

ZHAO, Q.; WANG, F., et al. Arbiter: the Evaluation Tool in the Contests of the China NOI International Journal Olimpiads in Informatics, 2013, 7(1): p. 180-185.

Zatarain, C. R., y Barrón, E. L. M. (2011). Herramienta de autor para la identificación de estilos de aprendizaje utilizando mapas auto-organizados en dispositivos móviles. Revista Electrónica de Investigación Educativa, 13(1), 43–55. Retrieved from <http://redie.uabc.mx/vol13no1/contenido-zatarainbarron.htm>

Zatarain C., Barrón E., M. L., y Reyes G. (2011). Herramienta de autor para la identificación de estilos de aprendizaje utilizando mapas auto-organizados en dispositivos móviles. Investigación Científica, 13(1), 43–55. Retrieved from <http://www.redalyc.org/articuloBasic.oa?id=15519374003>

## Anexo A: Tabla de operacionalización de habilidades algorítmicas

VARIABLE	Dimensiones	Habilidades	Categorías/Tareas
<p><b>VD: Habilidad algorítmica:</b> Se refiere al desarrollo y uso de algoritmos que puedan ayudar a resolver un tipo de problema o realizar un tipo específico de tarea.</p> <p>.Contempla capacidades de abstracción y resolución de problemas en el alumno. El estudiante pone en práctica a partir de un conjunto de pasos sistémicos habilidades cognitivas como: recordar, comprender, abstraer, analizar, probar y crear, para la solución de algún tipo de problema, (Noguera, 1997)(Durand, 1988) Moursund,2006) (Duran,2012)</p>	<p><b>Capacidad de Abstracción</b></p> <p><b>Capacidad de Resolver Problemas</b></p>	H1.Recordar	<p><b>Reconocer la</b> sintaxis de una instrucción.</p> <p><b>Reconocer la</b> semántica de una instrucción.</p>
		H2.Comprender/Interpretar	<p><b>Interpretar</b> que cambios ocurren al modificar una instrucción.</p> <p><b>Traducir</b> un algoritmo dado a pseudocódigo o diagrama de flujo.</p>
			<p><b>Utilizar</b> estructuras con un fin determinado y claro (decisiones, ciclos).</p> <p><b>Modificar</b> un algoritmo.</p>
		H3.Abstraer(Aplicar)	<p><b>Predecir</b> los posibles efectos al ejecutar un algoritmo línea por línea.</p>
			<p><b>Detectar</b> posibles errores en un algoritmo.</p>
		H4.Analizar	<p><b>Evaluar</b> requerimientos a nivel de usuario, sistema, organización y desarrollar una solución algorítmica coherente que los tome en cuenta.</p> <p><b>Comparar</b> soluciones algorítmicas que resuelven un mismo problema.</p>
			<p><b>Verificar</b> un conjunto de instrucciones a través de pruebas de escritorio)</p>
		H5.Probar(Evaluar)	<p><b>Generar</b> un algoritmo completo a partir de las especificaciones de un problema</p>
			<p><b>Generalizar</b> un algoritmo.</p>

## Anexo B: Cuestionario de validación para el instrumento de Habilidades algorítmicas

Respetado juez:

Usted ha sido seleccionado para evaluar el instrumento “**HABILIDADES ALGORÍTMICAS**” que hace parte de la investigación “**PROTOTIPO DE UN SISTEMA TUTOR INTELIGENTE PARA EL DESARROLLO DE HABILIDADES ALGORÍTMICAS**”. La evaluación de los instrumentos es de gran relevancia para lograr que sean válidos y que los resultados obtenidos a partir de éstos sean utilizados eficientemente. Agradecemos de antemano su valiosa colaboración.

NOMBRES Y APELLIDOS DEL JUEZ: \_\_\_\_\_

FORMACIÓN ACADÉMICA: \_\_\_\_\_

AREAS DE EXPERIENCIA PROFESIONAL \_\_\_\_\_

TIEMPO \_\_\_\_\_ CARGO ACTUAL \_\_\_\_\_

INSTITUCIÓN \_\_\_\_\_

Objetivo de la investigación: *Desarrollar las habilidades algorítmicas de los estudiantes de nuevo ingreso en Ciencias de la Computación.*

Objetivo del juicio de expertos: *Evaluuar el grado de claridad, coherencia y relevancia de cada ítem del test para evaluar las habilidades algorítmicas en estudiantes con base a la Taxonomía de Bloom.*

Objetivo de la prueba: *Examinar las habilidades algorítmicas del estudiante de nuevo ingreso.*

Muchas gracias por su colaboración.

ATENTAMENTE

**Estudiante:** Mtra. Guillermina Sánchez Román

**Director:** Dra. Josefina Guerrero

De acuerdo con los siguientes indicadores califique cada uno de los ítems según corresponda.

CATEGORIA	CALIFICACIÓN	INDICADOR
<b>SUFICIENCIA</b> Los ítems que pertenecen a una misma dimensión bastan para obtener la medición de ésta.	1 No cumple con el criterio	Los ítems no son suficientes para medir la dimensión.
	2. Bajo Nivel	Los ítems miden algún aspecto de la dimensión pero no corresponden con la dimensión total.
	3. Moderado nivel	Se deben incrementar algunos ítems para poder evaluar la dimensión completamente.
	4. Alto nivel Los ítems son suficientes	Los ítems son suficientes
<b>COHERENCIA</b> El ítem tiene relación lógica con la dimensión o indicador que está midiendo.	1 No cumple con el criterio	El ítem no tiene relación lógica con la dimensión
	2. Bajo Nivel	El ítem tiene una relación tangencial con la dimensión.
	3. Moderado nivel	El ítem tiene una relación moderada con la dimensión que está midiendo.
	4. Alto nivel	El ítem se encuentra completamente relacionado con la dimensión que está midiendo.
<b>RELEVANCIA</b> El ítem es esencial o importante, es decir debe ser incluido.	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión
	2. Bajo Nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado nivel	El ítem es relativamente importante
	4. Alto nivel	El ítem es muy relevante y debe ser incluido
<b>CLARIDAD</b> El ítem se comprende fácilmente, es decir, su sintáctica y semántica son adecuadas.	1 No cumple con el criterio	El ítem no es claro
	2. Bajo Nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por la ordenación de las mismas.
	3. Moderado nivel	Se requiere una modificación muy específica de algunos de los términos del ítem.
	4. Alto nivel	El ítem es claro, tiene semántica y sintaxis adecuada

Dimensiones	Tareas en realización de Algoritmos	Habilidades	Ítem	Suficiencia	Coherencia	Relevancia	Claridad
Capacidad de Abstracción	Reconocer la sintaxis de una instrucción.	Conocer	1				
	Reconocer la semántica de una instrucción.		4				
	Interpretar que cambios ocurren al modificar una instrucción.		5				
	Traducir un algoritmo dado a pseudocódigo o diagrama de flujo.	Comprender/Interpretar	2				
	Utilizar estructuras con un fin determinado y claro (decisiones, ciclos).		6				
	Usar rutinas para llevar a cabo una tarea específica.		9				
	Modificar un algoritmo.		7				
			8				
			11				
			13				
Capacidad de Resolver Problemas	Predecir los posibles efectos al ejecutar un algoritmo línea por línea.	Abstraer(Aplicar)	14				
	Detectar posibles errores en un algoritmo.		17				
	Evaluar requerimientos a nivel de usuario, sistema, organización y desarrollar una solución algorítmica coherente que los tome en cuenta.		18				
	Comparar soluciones algorítmicas que resuelven un mismo		23				
			24				
			27				
			28				
			3				
			10				
			15				
	Analizar	Probar(Evaluar)	16				
			19				
			20				
			25				
			26				
	12						
	21						

problema. <b>Verificar</b> un conjunto de instrucciones a través de pruebas de escritorio)		22			
<b>Generar</b> un algoritmo completo a partir de las especificaciones de un problema	Modelar/Formalizar(Crear)	29			
<b>Generalizar</b> un algoritmo.					

¿Hay alguna dimensión que hace parte del constructo y no fue evaluada?  
 ¿Cuál? \_\_\_\_\_

---

\*Para los casos de equivalencia semántica se deja una casilla por ítem, ya que se evaluará si la traducción o el cambio en vocabulario son suficientes.

## Anexo C: Tabla de artículos del Estado del Arte

Artículo	Estrategia de resolución de problemas	Técnicas de gamificación	Estudiante de nivel superior	Ambiente Individual	Ambiente Colaborativo	Modulo estudiante	Programación	Validación del sistema	Arquitectura	Metodología Cuantitativa	Herramienta CTAT
(Kollu, 2011)	X		X					X	X	X	
(Abu-Naser, 2008)	X		X			X	X		X	X	
(Vincent Aleven, McLaren, Sewall, y Koedinger, 2009)	X			X					X	X	X
(Walker, Rummel, McLaren, y Koedinger, 2007)					X			X		X	
(Harrer, McLaren, Walker, Bollen, y Sewall, 2005)	X				X					X	X
(Bruno, 2005)						X	X	X			X
(Zatarain y Barrón, 2011)				X		X					X
(Lemus Serrano Carlos, 2010)				X			X		X		
(Peña, Marzo, de la Rosa, y Fabregat, 2002)				X					X	X	
(Z Cataldi et al., 2005)	X					X				X	
(Zulma Cataldi et al., 2006)			X				X				
(Bartó y Díaz, 2012)	X		X	X		X				X	X
(Barrón-Estrada, Zatarain-Cabada, Aranda-Ortega, Gómez-Pérez, y Mejía-Arredondo, n.d.)				X		X				.	
(Najjar y Mayers, 2003)						X				X	X
(V Aleven, McLaren, y Sewall, 2006)											X
(Altuna Castillo, 2014)			X				X			.	X
(Kim, Park, Kim, y Oh, 2016)					X	X				X	
(V Aleven, Baker, Wang, Sewall, y Popescu, 2016)				X							X
(C. González et al., 2014)		X		X		X			X	X	
(Matsuda, Cohen, Sewall, Lacerda, y Koedinger, 2008)	X		X			X				X	

# Anexo D: Instrumento de estilos de aprendizaje de Felder y Silverman

*Nombre:* \_\_\_\_\_

*Edad:* \_\_\_\_\_ *Lugar de origen:* \_\_\_\_\_

*Sexo:* \_\_\_\_\_ *Matricula:* \_\_\_\_\_

**Inventario de estilos de aprendizaje de Felder y Silverman** (Felder y Silverman, 1988)

## Descripción.

El inventario de estilos de aprendizaje de Felder y Silverman está diseñado a partir de cuatro escalas bipolares relacionadas con las preferencias en el estilo de aprendizaje. Estas son: **Activo-Reflexivo**, **Sensorial-Intuitivo**, **Visual-Verbal** y **Secuencial-Global**.

## Instrucciones.

En cada uno de los ítems califica con 1 (uno) aquella opción que se aplica **más frecuentemente** a la forma en que aprendes. La otra opción califícalo con 0 (cero).

## Ejemplo:

1	<i>Entiendo mejor algo:</i>		
	a)	si lo practico.	1
	b)	si pienso en ello.	0

**Observación.** Evita calificar ambas opciones con uno o con cero.

En la parte inferior de la tabla encontrarás tus resultados; anótalos y analízalos y entregalos al docente

#			1/0
1	<i>Entiendo mejor algo:</i>		
	a)	si lo practico.	0
	b)	si pienso en ello.	1
2	<i>Me considero:</i>		
	a)	Realista	0
	b)	Innovador	0
3	<i>Cuando pienso en lo que hice ayer es más probable que lo haga:</i>		
	a)	empleando imágenes.	0
	b)	empleando palabras.	0
4	<i>Tengo tendencia a:</i>		
	a)	Entender los detalles de un tema pero no ver claramente su estructura completa.	0

	b)	Entender la estructura completa pero no ver claramente los detalles.	0
5	Cuando estoy aprendiendo algo nuevo me ayuda:		
	a)	Hablar de ello.	0
	b)	Pensar en ello.	1
6	Si yo fuera profesor preferiría dar un curso:		
	a)	Que trate sobre hechos y situaciones reales de la vida.	0
	b)	Que trate con ideas y teorías.	0
7	Prefiero obtener información nueva de:		
	a)	Imágenes, diagramas, gráficas o mapas.	0
	b)	Instrucciones escritas o información verbal.	0
8	Una vez que entiendo:		
	a)	todas las partes, entiendo el total.	0
	b)	el total de algo, entiendo cómo encajan las partes.	0
9	En un grupo de estudio que trabaja con un material difícil, es más probable que:		
	a)	participe y contribuya con ideas.	0
	b)	no participe y sólo escuche.	0
10	Es más fácil para mí:		
	a)	aprender hechos.	0
	b)	aprender conceptos.	0
11	En un libro con muchas imágenes y gráficas es más probable que:		
	a)	revise cuidadosamente las imágenes y las gráficas.	0
	b)	me concentre en el texto escrito.	0
12	Cuando resuelvo problemas de matemáticas:		
	a)	generalmente trabajo sobre las soluciones un paso a la vez.	0
	b)	frecuentemente sé cuáles son las soluciones pero tengo dificultad para imaginarme los pasos para llegar a ellas.	0
13	En las clases a las que he asistido:		
	a)	he llegado a saber cómo son la mayoría de los estudiantes.	0
	b)	pocas veces he llegado a saber cómo son la mayoría de los estudiantes.	0
14	Cuando leo temas que no son de ficción, prefiero:		
	a)	que me enseñe nuevos hechos o me diga cómo hacer algo.	0
	b)	que me dé nuevas ideas en qué pensar.	0
15	Me gustan los maestros:		
	a)	que utilizan muchos esquemas en el pizarrón.	0
	b)	que utilizan mucho tiempo para explicar.	0
16	Cuando estoy analizando un cuento o una novela:		
	a)	pienso en los incidentes y trato de acomodarlos para configurar los temas.	0

	b)	me doy cuenta de cuáles son los temas cuando termino de leer y luego tengo que regresar y encontrar los incidentes que los demuestran.	0
17	Cuando comienzo a resolver un problema de tarea, es más probable que:		
	a)	comience a trabajar en su resolución inmediatamente.	0
	b)	primero trate de entender completamente el problema.	0
18	Prefiero la idea de:		
	a)	certeza.	0
	b)	teoría.	0
19	Recuerdo mejor:		
	a)	lo que veo.	0
	b)	lo que oigo.	0
20	Es más importante para mí que un profesor:		
	a)	exponga el material en pasos secuenciales claros.	0
	b)	me dé un panorama general y relacione el material con otros temas.	0
21	Prefiero estudiar:		
	a)	en un grupo de estudio.	0
	b)	solo.	0
22	Me considero:		
	a)	cuidadoso en los detalles de mi trabajo.	0
	b)	creativo en la forma en la que hago mi trabajo.	0
23	Cuando alguien me da la dirección de un nuevo lugar, prefiero:		
	a)	un mapa.	0
	b)	instrucciones escritas.	0
24	Aprendo:		
	a)	a un paso constante. Si estudio con ahínco consigo lo que deseo.	0
	b)	con interrupciones. Me llevo a confundir y súbitamente lo entiendo.	0
25	Prefiero primero:		
	a)	hacer algo y ver qué sucede.	0
	b)	pensar cómo voy a hacer algo.	0
26	Cuando leo por diversión, me gustan los escritores que:		
	a)	dicen claramente lo que desean dar a conocer.	0
	b)	dicen las cosas en forma creativa e interesante.	0
27	Cuando veo un esquema o bosquejo en clase, es más probable que recuerde:		
	a)	la imagen.	0
	b)	lo que el profesor dijo de ella.	0
28	Cuando me enfrento a un cuerpo de información:		
	a)	me concentro en los detalles y pierdo de vista el total de la misma.	0
	b)	trato de entender el todo antes de ir a los detalles.	0
29	Recuerdo más fácilmente:		
	a)	algo que he hecho.	0
	b)	algo en lo que he pensado mucho.	0
30	Cuando tengo que hacer un trabajo, prefiero:		

	a)	dominar una forma de hacerlo.	0
	b)	intentar nuevas formas de hacerlo.	0
31	Cuando alguien me enseña datos, prefiero:		
	a)	gráficas.	0
	b)	resúmenes con texto.	0
32	Cuando escribo un trabajo, es más probable que:		
	a)	lo haga (piense o escriba) desde el principio y avance.	0
	b)	lo haga (piense o escriba) en diferentes partes y luego las ordene.	0
33	Cuando tengo que trabajar en un proyecto de grupo, primero quiero:		
	a)	realizar una "lluvia de ideas" donde cada uno contribuye con ideas.	0
	b)	realizar una "lluvia de ideas" en forma personal y después reunirme con el grupo para comparar las ideas.	0
34	Considero que es un gran elogio para una persona llamarla:		
	a)	sensible.	0
	b)	imaginativa.	0
35	Cuando conozco personas en una fiesta, es más probable que recuerde:		
	a)	cómo es su apariencia.	0
	b)	lo que dicen de sí mismos.	0
36	Cuando estoy aprendiendo un tema, prefiero:		
	a)	mantenerme concentrado en ese tema, aprendiendo lo más que pueda de él.	0
	b)	hacer conexiones entre ese tema y temas relacionados.	0
37	Me considero:		
	a)	abierto.	0
	b)	reservado.	0
38	Prefiero cursos que dan más importancia a:		
	a)	material concreto, como hechos y datos.	0
	b)	material abstracto, como conceptos y teorías.	0
39	Para divertirme prefiero:		
	a)	ver televisión.	0
	b)	leer un libro.	0
40	Algunos profesores inician sus clases haciendo un bosquejo de lo que enseñarán. Esos bosquejos son:		
	a)	algo útiles para mí.	0
	b)	muy útiles para mí.	0
41	La idea de hacer una tarea en grupo con una sola calificación para todos:		
	a)	me parece bien.	0
	b)	no me parece bien.	0
42	Cuando hago grandes cálculos:		
	a)	tiendo a repetir todos mis pasos y revisar cuidadosamente mi trabajo.	0
	b)	me cansa hacer su revisión y tengo que esforzarme para hacerlo.	0
43	Tiendo a recordar lugares en los que he estado:		
	a)	fácilmente y con bastante exactitud.	0
	b)	con dificultad y sin mucho detalle.	0
44	Cuando resuelvo problemas en grupo, es más probable que yo:		

	a)	piense en los pasos para la solución de los problemas.	0
	b)	piense en las posibles consecuencias o aplicaciones de la solución en un amplio rango de campos.	0

## RESULTADOS

En la parte inferior de la tabla encontrarás tus resultados; anótalos y analízalos y entregalos al docente

Activo	0	
Reflexivo	2	
Tendencia: REFLEXIVO con	2	puntos

Sensorial	0	
Intuitivo	0	
Tendencia: INTUITIVO con	0	puntos

Visual	0	
Verbal	0	
Tendencia: VERBAL con	0	puntos

Secuencial	0	
Global	0	
Tendencia: GLOBAL con	0	puntos

### NIVELES

Si el puntaje está entre 1-3: representa un equilibrio (**NEUTRO**) apropiado entre los dos estilos de aprendizaje.

Si el puntaje está entre 5-7: representa una preferencia **MODERADO** hacia uno de los extremos de la escala y se aprenderá más fácilmente empleando recursos de ese tipo.

Si el puntaje está entre 9-11: representa una preferencia **muy FUERTE** hacia uno de los extremos de la escala. Se pueden presentar dificultades para aprender en ambientes diferentes a esta modalidad.

# Anexo E: Instrumento de Habilidades Algorítmicas.

Adaptado de (Román-González et al., 2015a)

El test está compuesto por 28 preguntas, distribuidas en 7 páginas con 4 preguntas en cada una de ellas. Todas las preguntas tienen 4 opciones de respuesta (A, B, C ó D) de las cuales sólo una es correcta. A partir de que comience el test dispones de 45 minutos para hacerlo lo mejor que puedas. MUY IMPORTANTE: cuando acabes o finalice el tiempo debes avanzar hasta la última página y pinchar sobre el botón 'Enviar' para que se guarden tus respuestas. Si necesitas ampliar alguna pregunta para verla más grande, haz 'Ctrl+' con el teclado para verla más grande o 'Ctrl-' para verla más pequeña. ¡ÁNIMO Y SUERTE!



'Pac-Man'



Fantasma



Artista

Pregunta 1

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A  
→ → ↓

Opción B  
→ ↓ ↓

Opción C  
→ → ↓ ↓

Opción D  
↓ ↓ →

Marca la opción correcta

- A)    B)    C)    D)

## Pregunta 2

¿Qué orden falta en la secuencia para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

Opción B

Opción C

Opción D

## Pregunta 2

Marca la opción correcta

- A)    B)    C)    D)

## Pregunta 3

Para llevar a 'Pac-Man' hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un **error**?

avanzar → Paso A

girar a la izquierda ↉ ↓ → Paso B

avanzar

avanzar → Paso C

girar a la izquierda ↉ ↓ → Paso D

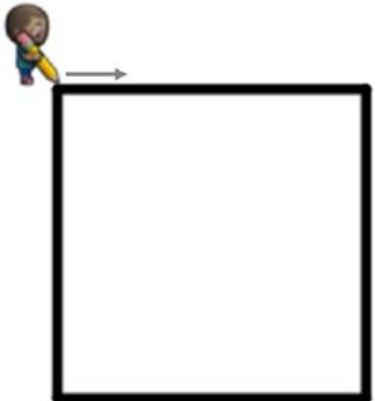
avanzar

## Pregunta 3

Marca el paso en el que hay error

- A)    B)    C)    D)

#### Pregunta 4

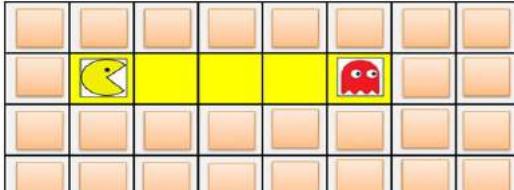
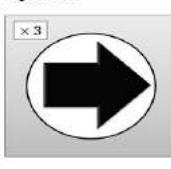
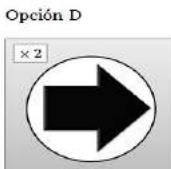
<p>¿Qué órdenes debe ejecutar el artista para dibujar el cuadrado? Cada uno de los lados del cuadrado mide 100 píxeles.</p> 	<p>Opción A</p> <pre>move hacia adelante 100 píxeles girar a la derecha por 90 grados move hacia adelante 100 píxeles girar a la izquierda por 90 grados move hacia adelante 100 píxeles girar a la derecha por 90 grados move hacia adelante 100 píxeles</pre>	<p>Opción B</p> <pre>move hacia adelante 25 píxeles girar a la derecha por 90 grados move hacia adelante 25 píxeles girar a la izquierda por 90 grados move hacia adelante 25 píxeles girar a la derecha por 90 grados move hacia adelante 25 píxeles</pre>
	<p>Opción C</p> <pre>move hacia adelante 50 píxeles girar a la derecha por 90 grados move hacia adelante 50 píxeles girar a la derecha por 90 grados move hacia adelante 50 píxeles girar a la derecha por 90 grados move hacia adelante 50 píxeles</pre>	<p>Opción D</p> <pre>move hacia adelante 100 píxeles girar a la derecha por 90 grados move hacia adelante 100 píxeles girar a la derecha por 90 grados move hacia adelante 100 píxeles girar a la derecha por 90 grados move hacia adelante 100 píxeles</pre>

#### Pregunta 4

Marca la opción correcta

- A)    B)    C)    D)

#### Pregunta 5

<p>¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?</p> 	<p>Opción A</p> 	<p>Opción B</p> 
	<p>Opción C</p> 	<p>Opción D</p> 

#### Pregunta 5

Marca la opción correcta

- A)    B)    C)    D)

### Pregunta 6

¿Cuántas veces se debe repetir la secuencia para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A  
× 2

Opción B  
× 1

Opción C  
× 4

Opción D  
× 3

### Pregunta 6

Marca la opción correcta

- A)    B)    C)    D)

### Pregunta 7

Para que el artista dibuje una vez el siguiente rectángulo (50 píxeles de ancho y 100 píxeles de alto), ¿en qué paso de la siguiente secuencia de órdenes hay un **error**?

**Paso A**

```

repetir [4] veces
  hacer
    mover hacia adelante ▾ [50] píxeles
    girar a la izquierda ▾ por [90] grados → Paso B
    mover hacia adelante ▾ [100] píxeles → Paso C
    girar a la izquierda ▾ por [90] grados → Paso D

```

### Pregunta 7

Marca el paso en el que hay error

- A)    B)    C)    D)

Pregunta 8

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?


Opción A

```

repetir 4 veces
haz repetir 3 veces
haz avanzar
girar a la derecha ⚡
avanzar

```

Opción B

```

repetir 3 veces
haz repetir 4 veces
haz avanzar
girar a la derecha ⚡
avanzar

```

Opción C

```

repetir 3 veces
haz repetir 4 veces
haz avanzar
girar a la derecha ⚡
avanzar

```

Opción D

```

repetir 4 veces
haz avanzar
repetir 3 veces
haz girar a la derecha ⚡
avanzar

```

Pregunta 8

Marca la opción correcta

- A)    B)    C)    D)

Pregunta 9

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?


Opción A

Opción B

Opción C

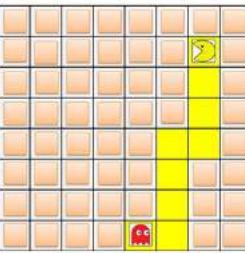
Opción D

Pregunta 9

Marca la opción correcta

- A)    B)    C)    D)

Pregunta 10

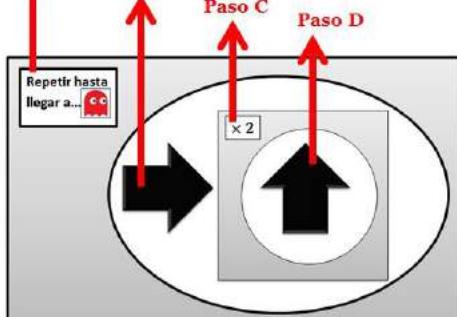
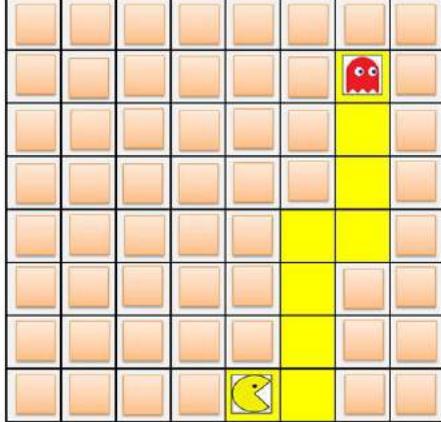
<p>¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' llegue hasta el fantasma por el camino señalado?</p>  	<p>Opción A <b>girar a la izquierda</b></p>	<p>Opción B <b>girar a la derecha</b></p>
	<p>Opción C <b>avanzar</b></p>	<p>Opción D <b>No falta ningún bloque</b></p>

Pregunta 10

Marca la opción correcta

- A)    B)    C)    D)

Pregunta 11

<p>Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un <b>error</b>?</p>  	<p><b>Paso A</b></p> <p><b>Paso B</b></p> <p><b>Paso C</b></p> <p><b>Paso D</b></p>
---	---

Pregunta 11

Marca el paso en el que hay error

- A)    B)    C)    D)

## Pregunta 12

¿Qué secuencia de órdenes debe ejecutar el artista para dibujar la escalera que llegue hasta la flor? Cada peldaño sube 30 píxeles.



Opción A

```

Repetir hasta la flor
haz | repetir 3 veces
haz | mover hacia adelante ▲ 30 píxeles
haz | girar a la derecha ▯ por 90 grados
salir hacia adelante ▲ 30 píxeles

```

Opción B

```

Repetir hasta la flor
haz | repetir 4 veces
haz | mover hacia adelante ▲ 120 píxeles
haz | girar a la derecha ▯ por 90 grados
salir hacia adelante ▲ 30 píxeles

```

Opción C

```

Repetir hasta la flor
haz | repetir 3 veces
haz | mover hacia adelante ▲ 30 píxeles
haz | girar a la derecha ▯ por 90 grados
salir hacia adelante ▲ 210 píxeles

```

Opción D

```

Repetir hasta la flor
haz | repetir 7 veces
haz | mover hacia adelante ▲ 30 píxeles
haz | girar a la derecha ▯ por 90 grados
salir hacia adelante ▲ 30 píxeles

```

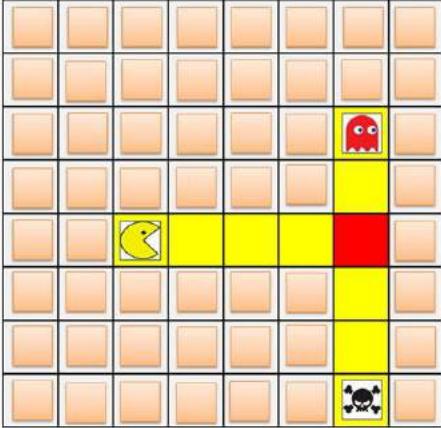
## Pregunta 12

Marca la opción correcta

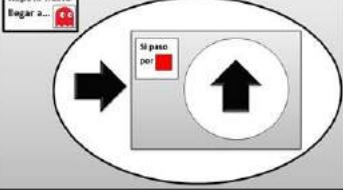
- A)    B)    C)    D)

## Pregunta 13

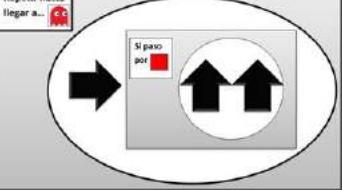
¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?



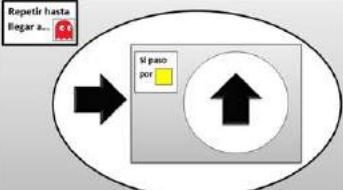
Opción A



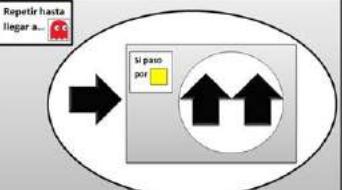
Opción B



Opción C



Opción D



## Pregunta 13

Marca la opción correcta

- A)    B)    C)    D)

Pregunta 14

<p>¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?</p>	<p>Opción A</p> <pre>Repetir hasta llegar a... 🐍 hacer avanzar si hay camino a la derecha ↘ hacer girar a la derecha ↙</pre>	<p>Opción B</p> <pre>Repetir hasta llegar a... 🐍 hacer girar a la derecha ↘ si hay camino a la derecha ↘ hacer avanzar</pre>
	<p>Opción C</p> <pre>Repetir hasta llegar a... 🐍 hacer avanzar si hay camino a la derecha ↘ hacer girar a la izquierda ↙</pre>	<p>Opción D</p> <pre>Repetir hasta llegar a... 🐍 hacer avanzar si hay camino a la izquierda ↙ hacer girar a la izquierda ↙</pre>

Pregunta 14

Marca la opción correcta

- A)    B)    C)    D)

Pregunta 15

<p>¿Qué falta en la siguiente secuencia de órdenes para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?</p> 	<p>Opción A</p> <p>Opción B</p> <p>Opción C</p> <p>Opción D</p> <p>Tanto la opción A como la opción C son correctas</p>
--	--

Pregunta 15

Marca la opción correcta

- A)    B)    C)    D)

### Pregunta 16

Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un **error**?

```

    Repetir hasta llegar a... [ghost]
    hacer avanzar
    si hay camino a la izquierda ↗
    hacer girar a la izquierda ↙
    si hay camino a la derecha ↘
    hacer avanzar
  
```

**Paso A:** hacer avanzar  
**Paso B:** hacer girar a la izquierda  
**Paso C:** hacer avanzar  
**Paso D:** hacer avanzar

### Pregunta 16

Marca el paso en el que hay error

- A)    B)    C)    D)

### Pregunta 17

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

<b>Opción A</b> <pre>         Repetir hasta llegar a... [ghost]         hacer si hay un camino delante ↗         hacer avanzar         sino girar a la izquierda ↙       </pre>	<b>Opción B</b> <pre>         Repetir hasta llegar a... [ghost]         hacer si hay un camino delante ↗         hacer avanzar         sino girar a la derecha ↘       </pre>
<b>Opción C</b> <pre>         Repetir hasta llegar a... [ghost]         hacer si hay camino a la derecha ↘         hacer girar a la derecha ↙         sino avanzar       </pre>	<b>Opción D</b> <pre>         Repetir hasta llegar a... [ghost]         hacer si hay camino a la izquierda ↗         hacer girar a la izquierda ↙         sino avanzar       </pre>

### Pregunta 17

Marca la opción correcta

- A)    B)    C)    D)

Pregunta 18

<p>¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?</p>	<p>Opción A</p> <pre> Repeticion [repetir hasta llegar a... fantasma]     hacer [si hay un camino delante]         avanzar     sino [girar a la izquierda] </pre>	<p>Opción B</p> <pre> Repeticion [repetir hasta llegar a... fantasma]     hacer [si hay un camino delante]         avanzar     sino [girar a la derecha] </pre>
	<p>Opción C</p> <pre> Repeticion [repetir hasta llegar a... fantasma]     hacer [si hay camino a la derecha]         girar a la derecha     hacer [girar a la derecha]     sino [avanzar] </pre>	<p>Opción D</p> <pre> Repeticion [repetir hasta llegar a... fantasma]     hacer [si hay camino a la izquierda]         girar a la izquierda     hacer [girar a la izquierda]     sino [avanzar] </pre>

Pregunta 18

Marca la opción correcta

- A)    B)    C)    D)

Pregunta 19

<p>Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un <b>error</b>?</p>	<pre> repeticion [repetir hasta [fantasma]]     [haz [si hay un camino delante] [avanzar] --&gt; Paso A]     [sino [si hay camino a la derecha] [girar a la derecha] --&gt; Paso B]         [haz [girar a la izquierda] --&gt; Paso C]     [sino [girar a la derecha] --&gt; Paso D] </pre>
---	---

Pregunta 19

Marca el paso en el que hay error

- A)    B)    C)    D)

Pregunta 20

¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' llegue hasta el fantasma por el camino señalado?

```

repeat [until (there is a ghost)]
    if [path ahead?]
        move [1 step]
    else
        turn right
    end
end

```

**Map:**


**Opción A:** avanzar

**Opción B:** girar a la derecha □ ▼

**Opción C:** girar a la izquierda □ ▼

**Opción D:** No falta ningún bloque

Pregunta 20

Marca la opción correcta

- A)    B)    C)    D)

Pregunta 21

¿Qué órdenes llevan a 'Pac-Man' por el camino señalado hasta las fresas e indican a 'Pac-Man' que se coma el número de fresas indicado?

```

while [there is a path ahead?]
    move [1 step]
end
repeat [3 times]
    move [1 step]
    eat [1 strawberry]
end

```

**Map:**


**Opción A:**

```

mientras [haya camino delante]
    hacer [avanzar]
    repetir [3 veces]
        hacer [Comer 1 fresa]
    fin
fin

```

**Opción B:**

```

mientras [haya camino delante]
    hacer [avanzar]
    repetir [4 veces]
        hacer [Comer 1 fresa]
    fin
fin

```

**Opción C:**

```

mientras [haya camino delante]
    hacer [avanzar]
    repetir [5 veces]
        hacer [Comer 1 fresa]
    fin
fin

```

**Opción D:**

```

mientras [haya camino delante]
    hacer [avanzar]
    repetir [3 veces]
        hacer [Comer 1 fresa]
    fin
fin

```

Pregunta 21

Marca la opción correcta

- A)    B)    C)    D)

## Pregunta 22

¿Qué órdenes van llevando a 'Pac-Man' por el camino señalado e indicándole que se coma el número de fresas correspondiente?

		3	3	3	3	3

Opción A

```
mientras haya camino delante
haz
    repetir 5 veces
        hacer avanzar
    repetir 3 veces
        hacer Comer 1 fresa
```

Opción B

```
mientras haya camino delante
hacer avanzar
repetir 3 veces
haz Comer 1 fresa
```

Opción C

```
mientras haya camino delante
haz
    repetir 3 veces
        hacer avanzar
    repetir 5 veces
        hacer Comer 1 fresa
```

Opción D

```
mientras haya camino delante
hacer avanzar
repetir 3 veces
haz Comer 1 fresa
```

## Pregunta 22

Marca la opción correcta

- A)    B)    C)    D)

## Pregunta 23

¿Qué falta en la siguiente secuencia de órdenes para que 'Pac-Man' avance por el camino señalado comiendo el número de fresas indicadas?

		1				

```
mientras haya camino delante
haz
    repetir ? veces
        hacer avanzar
    si hay alguna fresa
        haz Comer 1 fresa
```

Opción A  
1 vez

Opción B  
2 veces

Opción C  
3 veces

Opción D  
5 veces

## Pregunta 23

Marca la opción correcta

- A)    B)    C)    D)

## Pregunta 24

¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' avance por el camino señalado comiendo el número de fresas indicadas (número desconocido)?

```

mientras haya camino delante
  hacer avanzar
    Si hay alguna fresa
      hacer [comer 1 fresa v]

```


Opción A  
**Mientras haya camino delante**

Opción B  
**Mientras no haya camino delante**

Opción C  
**Mientras haya alguna fresa**

Opción D  
**Mientras no haya ninguna fresa**

## Pregunta 24

Marca la opción correcta

- A)    B)    C)    D)

## Pregunta 25

Si tenemos el siguiente conjunto de órdenes, al que llamamos 'my function', y que dibuja un cuadrado de 100 píxeles de lado:

```

Función
my function
repetir [4] veces
  haz [mover hacia adelante v 100 píxeles]
  girar a la derecha v por [90] grados

```

¿Qué secuencia debe ejecutar el artista para dibujar el siguiente diseño? Cada uno de los lados de cada cuadrado mide 100 píxeles.

Opción A

```

repetir [3] veces
  haz [my function]
  girar a la derecha v por [120] grados

```

Opción B

```

repetir [3] veces
  haz [my function]
  girar a la derecha v por [120] grados

```

Opción C

```

repetir [4] veces
  haz [my function]
  girar a la derecha v por [90] grados

```

Opción D

```

repetir [4] veces
  haz [my function]
  girar a la derecha v por [90] grados

```

## Pregunta 25

Marca la opción correcta

- A)    B)    C)    D)

## Pregunta 26

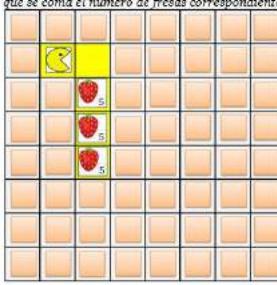
<p>Si tenemos el siguiente conjunto de órdenes, al que llamamos 'my function', y que dibuja un triángulo de 50 píxeles de lado:</p>  <p>¿Qué le falta a la siguiente secuencia para que el artista dibuje el siguiente diseño? Cada uno de los lados de cada triángulo mide 50 píxeles.</p>  	Opción A	Opción B
	15	5
<p>Opción C</p>	4	Opción D
	3	

## Pregunta 26

Marca la opción correcta

- A)    B)    C)    D)

## Pregunta 27

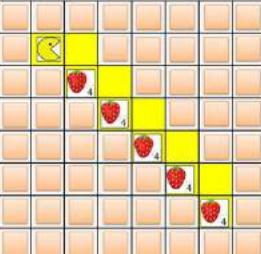
<p>Si tenemos el siguiente conjunto de órdenes, al que llamamos 'get 5':</p>  <p>¿Qué órdenes van llevando a 'Pac-Man' por el camino señalado e indicándole que se coma el número de fresas correspondiente?</p> 	Opción A	Opción B
	<pre>avanzar girar a la derecha repetir (3) veces haz avanzar get 5</pre>	<pre>avanzar girar a la derecha repetir (3) veces haz get 5 avanzar</pre>
<p>Opción C</p>	Opción D	
	<pre>avanzar girar a la derecha repetir (5) veces haz avanzar get 5</pre>	<pre>avanzar girar a la derecha repetir (5) veces haz get 5 avanzar</pre>

## Pregunta 27

Marca la opción correcta

- A)    B)    C)    D)

Pregunta 28

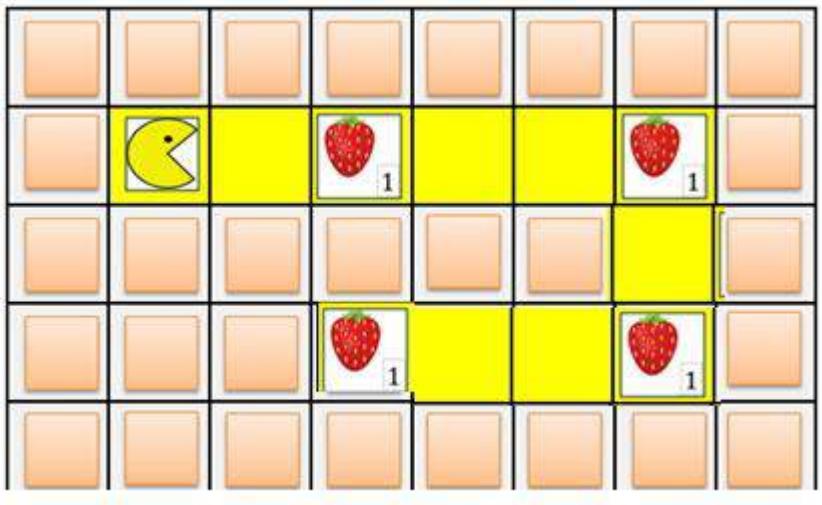
<p>Si tenemos el siguiente conjunto de órdenes, llamado 'move and get 4':</p> <p>Función</p> <pre> move and get 4 avanzar girar a la derecha avanzar repetir 4 veces     haz Comer 1 fresa     girar a la izquierda </pre>	<p>Opción A</p> <p><b>3</b></p>	<p>Opción B</p> <p><b>4</b></p>
<p>¿Qué falta en la siguiente secuencia para llevar a Pac-Man por el camino señalado hasta las fresas, comiendo el número de fresas indicado?</p>  	<p>Opción C</p> <p><b>5</b></p>	<p>Opción D</p> <p><b>6</b></p>

Pregunta 28

Marca la opción correcta

- A)    B)    C)    D)

29.- Escribe las instrucciones para llevar al PacMan por el camino señalado hasta las fresas comiendo el número de fresas indicado



# Anexo F: Cuestionario de validación de la Metodología de ST

## *Cuestionario*

Este cuestionario se aplicará a expertos de diferentes áreas para validar la metodología propuesta.

El trabajo tiene como objetivo: *Proponer una metodología de Sistema Tutor en apoyo a las habilidades algorítmicas de los estudiantes de nuevo ingreso en la Licenciatura de Ciencias de la Computación de la Facultad de Ciencias de la Computación de la BUAP.*

Validación por expertos

Fase 1: Pedagogos

Fase2: Lic. En Ciencias de la Computación

La entrevista puede ser grabada de acuerdo a su disponibilidad, usted puede detener la grabación en el momento que lo desee. Los resultados de esta entrevista serán analizados con fines educativos y de la investigación propia.

ATENTAMENTE

**Estudiante:** Mtra. Guillermina Sánchez Román

**Directora:** Dra. Josefina Guerrero García

**Codirector:** Dr. Daniel Mocenahua Mora

**Instrucciones:** Usted responderá a cada pregunta de acuerdo a su experiencia y dará su opinión al respecto de mejoras o necesidades de la metodología. A continuación se enlista cada una de las categorías que se encontraron:

- 1) Modelo de dominio: Se refiere al manejo de contenidos o materiales que derivan del conocimiento a trabajar de acuerdo a los objetivos que se pretenden alcanzar.
- 2) Modelo de estudiante: Son los elementos que caracterizan y definen al estudiante del área de ciencias de la Computación.
- 3) Modelo de Interacción con el usuario: Se definen los elementos que se observan en la Interfaz de Usuario y con los cuales interactúa el estudiante.
- 4) Modelo pedagógico (tutor): Describe las estrategias didácticas a seguir para cada uno de los estudiantes y que apoyen a las habilidades algorítmicas.
- 5) Generales (otros): En esta categoría se enfoca a la metodología completa.

Solicito su autorización para grabar esta entrevista la cual tomará aproximadamente 60 minutos en los cuales expongo la metodología en 15 minutos y la entrevista lleva las preguntas en escala de Likert y solicito sus comentarios a cada una de ellas.

Categoría	Preguntas de entrevista					
CONOCIMIENTO	1. ¿Se identifican los niveles de logro de los estudiantes?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	2. ¿Las categorías propuestas definen los niveles de logro del estudiante para la evaluación de sus habilidades?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	3. ¿El modelo propuesto muestra la relación entre subtemas y las categorías obtenidas?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	4. ¿Considera posible que los docentes diseñen todas las combinaciones de contenidos?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	5. ¿Los pasos propuestos que integran el seguimiento de contenido de acuerdo a la categoría de estudiante son pertinentes para apoyar las habilidades algorítmicas?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	6. ¿Qué elementos considera que se deben incorporar en la categoría de dominio (conocimiento)?					

<b>ESTUDIANTE</b>	7. ¿Cuáles son las características que se contemplan para determinar el perfil de ingreso de los estudiantes al área de ciencias de la computación?					
	8. ¿Considera que los fundamentos de los conocimientos previos y los estilos de aprendizaje propuestos permiten caracterizar al estudiante de nuevo ingreso a Computación?	<b>Totalmente De acuerdo</b>	<b>De acuerdo</b>	<b>Ni en acuerdo ni en desacuerdo</b>	<b>En desacuerdo</b>	<b>Totalmente en desacuerdo</b>
	9. ¿Los niveles de estudiantes se definen con base a las características académicas del estudiante?	<b>Totalmente De acuerdo</b>	<b>De acuerdo</b>	<b>Ni en acuerdo ni en desacuerdo</b>	<b>En desacuerdo</b>	<b>Totalmente en desacuerdo</b>
	10. Al modelo de estudiante, ¿se incorporan las características para detectar el perfil de ingreso?	<b>Totalmente De acuerdo</b>	<b>De acuerdo</b>	<b>Ni en acuerdo ni en desacuerdo</b>	<b>En desacuerdo</b>	<b>Totalmente en desacuerdo</b>
	11. ¿El test propuesto de pensamiento computacional se considera adecuado para identificar las habilidades algorítmicas del estudiante?	<b>Totalmente De acuerdo</b>	<b>De acuerdo</b>	<b>Ni en acuerdo ni en desacuerdo</b>	<b>En desacuerdo</b>	<b>Totalmente en desacuerdo</b>
	12. ¿El test propuesto de Felder y Silverman se considera adecuado para identificar los estilos preferentes del estudiante de Ciencias de la Computación?	<b>Totalmente De acuerdo</b>	<b>De acuerdo</b>	<b>Ni en acuerdo ni en desacuerdo</b>	<b>En desacuerdo</b>	<b>Totalmente en desacuerdo</b>
	13. ¿Considera que se está midiendo el conocimiento previo para detectar habilidades?	<b>Totalmente De acuerdo</b>	<b>De acuerdo</b>	<b>Ni en acuerdo ni en desacuerdo</b>	<b>En desacuerdo</b>	<b>Totalmente en desacuerdo</b>
	14. ¿Qué elementos considera que se deben incorporar a la categoría de modelo de estudiante?					

<b>INTERFAZ DE ESTUDIANTE</b>	15. ¿Los elementos del prototipo integran aspectos de la teoría pedagógica?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	16. ¿Cuáles elementos de la interacción pueden recopilar información sobre el avance del estudiante?					
	17. ¿Qué otros elementos considera que se deben incorporar en esta categoría?					
<b>PEDAGOGICO</b>	18. ¿La estrategia presentada integra aspectos de aprendizaje individual?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	19. ¿La estrategia didáctica propuesta considera el acompañamiento al estudiante?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	20. ¿La estrategia didáctica promueve la abstracción y resolución de problemas?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	21. ¿Las teorías propuestas conllevan a una estrategia favorable en apoyo a las habilidades algorítmicas?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	22. De acuerdo a tu experiencia, ¿considera viable el modelo instruccional para el desarrollo de habilidades algorítmicas?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	23. ¿Qué elementos considera que se deben incorporar en el módulo pedagógico?					

GENERAL	24. ¿Consideras suficientes los pasos propuestos en la metodología y su secuencia?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	25. ¿Cómo considera la metodología respecto a los elementos para detectar el perfil de estudiante?					
	26. ¿Considera que el diseño de la estrategia a implementar en el ST puede promover confianza al estudiante?					
	27. ¿Considera que la metodología cumple con las especificaciones identificadas en la literatura?					
	28. ¿Considera que la arquitectura corresponde con la teoría propuesta?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	29. ¿Crees que los módulos propuestos son suficientes?	Totalmente De acuerdo	De acuerdo	Ni en acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo
	30. ¿Consideras agregar alguna categoría?					

En escala de 0 a 100, ¿cómo evaluarías a la metodología con base en el objetivo, se alcanzó?