

RESOLUCIÓN DE PROBLEMAS COMPUTACIONALES: ANÁLISIS DEL PROCESO DE APRENDIZAJE

Marcia Mac Gaul

Universidad Nacional de Salta – Facultad de Ciencias Exactas
mmacgaul@cidia.unsa.edu.ar

Marcela F. López

Universidad Nacional de Salta – Facultad de Ciencias Exactas
mfflopez@unsa.edu.ar

Paola Del Olmo

Universidad Nacional de Salta – Facultad de Ciencias Exactas
pdelolmo@unsa.edu.ar

RESUMEN

El proceso de resolución de problemas es una situación que genera en el alumno un conflicto cognitivo en la medida que el mismo no dispone de un sistema de estrategias totalmente constituido, que le permita responder de manera inmediata. En el intento de solución reestructurará su sistema conceptual dando lugar a nuevas configuraciones cognitivas.

El equipo de investigación ha efectuado un análisis cuanti-cualitativo de encuestas, evaluaciones y entrevistas aplicadas a alumnos de la Licenciatura en Análisis de Sistemas, a lo largo del primer año de estudios universitarios, para indagar cuestiones asociadas a nuestro objeto de investigación, esto es, ¿Cuál es el proceso mediante el cual los estudiantes

adquieren las competencias necesarias para la resolución de problemas? ¿Qué dificultades aparecen reiteradamente en la resolución de problemas computacionales?, ¿Qué rol desempeñamos los docentes durante este proceso? Planteamos y reflexionamos sobre una serie de fases y subfases por las que el alumno debe transitar con precisión, para obtener una resolución adecuada de éstos problemas, en un proceso espiralado, iterativo e incremental.

Palabras clave:

Problemas computacionales – Carreras de Informática – Aprendizaje – Enseñanza – Algoritmos

Frecuentemente escuchamos de nuestros alumnos, *yo se hacerlo...pero no se cómo se hace.*

INTRODUCCIÓN

Hablar de resolución de problemas como estrategias didáctica en la enseñanza de las ciencias requiere de la precisión de algunos conceptos. ¿Qué entendemos por situación problemática? Es aquella situación que

sostiene la distancia cognitiva adecuada para generar en el alumno un conflicto cognitivo. En el intento de solución reestructurará su sistema conceptual dando lugar a nuevas configuraciones cognitivas. Es una situación problemática en la medida en que el sujeto al que se le plantea, cuenta con los elementos para comprender la situación que el problema describe y no dispone de un sistema de respuestas totalmente constituido que le permita responder de manera inmediata.

Partimos de la premisa que resolver problemas no es una tarea trivial. Resulta paradójico que mientras más experticia desarrollamos para ejecutar una tarea, más difícil nos resulta plasmar su solución mediante un algoritmo adecuadamente especificado como para que un autómatas pueda ejecutarlo. En la solución de problemas se conjuga la comprensión profunda del contenido para relacionarlo a través de inéditas reestructuraciones. Los ejemplos son infinitos, todos sabemos atarnos los cordones de los zapatos, lavarnos los dientes, pagar un determinado importe con billetes de diversa nominación y andar en bicicleta, sin embargo, cuando intentamos que los alumnos escriban la secuencia de pasos necesarios para desarrollar estas actividades y sometamos a prueba tales especificaciones, las cosas no salen como las pensamos cuando actuamos como una computadora, es decir, realizando exactamente la secuencia de órdenes indicadas en el algoritmo, *sin pensarlas*.

El equipo de investigación ha efectuado un análisis cuanti-cualitativo de evaluaciones y entrevistas aplicadas a alumnos de la Licenciatura en Análisis de Sistemas, a lo largo del primer año de estudios universitarios, para indagar cuestiones asociadas a nuestro objeto de investigación, esto es, ¿Cuál es el proceso mediante el cual los estudiantes adquieren las competencias necesarias para la resolución de problemas? ¿Qué dificultades aparecen reiteradamente en la resolución de problemas computacionales?, ¿Qué rol desempeñamos los docentes durante este proceso?

Frecuentemente escuchamos de nuestros docentes, *yo se cómo lo enseño...pero no se cómo lo aprenden*.

OBJETIVOS

Nuestra investigación busca indagar las competencias de los alumnos iniciales de la Lic. en Análisis de Sistemas de la U.N.Sa.,

para la resolución de problemas computacionales, diseño de algoritmos¹ y prueba de los mismos. Para ello, hemos efectuado el análisis de regularidades, conflictos y contradicciones, la continuidad y ruptura de los procesos de los sujetos actuantes. También se indagó en el pensamiento y las actitudes, superando las verbalizaciones inmediatas y habituales, buscando los puntos latentes, los procesos contradictorios en las propias creencias y esquemas cognitivos.

Esta fase investigativa es previa a la de desarrollo de materiales educativos innovadores que faciliten el aprendizaje de los contenidos relativos a resolución de problemas computacionales y diseño de algoritmos. La implementación de estos materiales, a la par de la reflexión y resignificación de las prácticas docentes usando NTICs, son los objetivos últimos de la investigación, que persigue, en definitiva, la mejora de la calidad del proceso educativo y de la práctica educativa.

METODO

Los procesos educativos, como toda práctica social, son complejos y en ellos intervienen una multiplicidad de factores interrelacionados, cuyo aislamiento sólo puede ocasionar la distorsión del conocimiento pretendido. Si al estudiar los fenómenos educativos se disecciona la realidad y se descontextualiza el análisis de la misma “*es fácil que hayamos forjado una imagen de esa realidad irreconocible por los que en ella están inmersos y, por lo tanto, incapaz de producir conocimiento válido para la comprensión, la reflexión y la mejora de la intervención educativa o social*” (Fernández Sierra y Santos Guerra, 1992: 23).

En este trabajo, la investigación se sostuvo con los siguientes instrumentos de recolección de datos:

¹ La representación de los algoritmos se realiza mediante diagramas N-S

1. Encuestas aplicadas a alumnos. El número de alumnos nos permitió indagar en general dificultades, hábitos de estudios, problemas grupales de integración para estudiar, etc.
2. Análisis documental de las producciones de los estudiantes (evaluaciones y producciones de prácticos).
3. Entrevistas semi estructuradas a los alumnos, con el objeto de obtener algunos supuestos y anticipaciones sobre el proceso de conocimiento en particular.
4. Observación sistemática que permitió, por una parte, obtener un registro de "lo cotidiano" como categoría central, teórica y empírica, aquello que aparece como fragmentario, recurrente, contradictorio, divergente de la práctica cotidiana de los docentes. Por otra, permitió examinar el proceso permanente de indagación, reflexión y contraste para captar los significados latentes de los acontecimientos y establecer las relaciones conflictivas, difusas y cambiantes de los alumnos, y sobre todo, obtener algunos indicios de las posibles orientaciones de los alumnos en la construcción del conocimiento computacional.

La aplicación de estos instrumentos dio lugar a matrices de datos cuantitativos, que fueron tratadas estadísticamente usando el software SPSS. Los datos cualitativos se analizaron mediante la herramienta Atlas.ti. Dada la envergadura y sobre todo la complejidad del objeto de indagación que el grupo delimita como problema, se apeló a una variedad de técnicas de recolección de datos.

RESULTADOS

En este apartado reunimos algunos resultados de la aplicación de los cuatro instrumentos ya señalados, tomando de cada uno de ellos lo pertinente a actitudes y aptitudes para la resolución de problemas y el correspondiente diseño de algoritmo.

La encuesta se aplicó a una muestra aleatoria de 83 estudiantes, al momento de finalizar el cursado de la primera asignatura de la carrera en la que se aborda el diseño de Algoritmos. En las siguientes dos tablas se plasman, por una parte, la valoración de los estudiantes relativa a la dificultad que enfrentan a lo largo del proceso de resolución, desde la formulación del problema hasta la corrección de eventuales errores y por otra parte, sus estrategias frente a las dificultades que esta tarea supone.

Paso	Muy difícil	Difícil	Medio difícil	Poco difícil	Sin dificultad
A	12	19	17	25	10
B	1	10	12	31	28
C	5	23	5	32	18
D	7	39	8	20	8
E	6	18	9	37	13
F	23	24	15	17	4
G	4	16	10	27	26
H	11	23	15	26	8

- A (Interpretación de consigna),
- B (Identificación de datos de E/S),
- C (Diseño de casos de prueba),
- D (Identificación de casos extremos de prueba),
- E (Determinación del tipo de variables),
- F (Diseño del algoritmo (diagrama N-S)),
- G (Confección de prueba de escritorio) y
- H (Corrección de error/es)

Tabla 1: Nivel de dificultad en los pasos para diseñar algoritmos

No sorprende que la mayor observación de la valoración *Muy difícil* corresponda al propio diseño del diagrama N-S, tarea en la que confluyen los pasos anteriores y en la que deben reflejarse coordinadamente las decisiones de diseño previas. Tareas tales como la Interpretación de Consignas, en la que casi tres cuartas partes de la población considera de dificultad Media a Difícil, acompañado por un alto registro en la dificultad para la Identificación de Casos Extremos de Prueba, hace presumir que los alumnos realizan un intento superficial durante el proceso de interpretación de consignas, lo que los llevaría a un gran esfuerzo en el diseño

del algoritmo. Por ello, la encuesta indagó acerca de las actitudes del estudiante frente a las dificultades para concretar un diagrama. Los resultados figuran en la tabla 2.

Actitud	Nunca	Casi nunca	A veces	Casi siempre	Siempre
A	12	24	39	6	1
B	2	5	30	33	13
C	33	18	21	7	1
D	12	13	28	17	11
E	7	7	32	15	19
F	23	22	29	4	2
G	6	19	24	25	8
H	9	11	29	21	12

- A (Lo abandona sin resolverlo),
- B (Busca problemas parecidos correctamente resueltos),
- C (Copia la solución elaborada por un compañero),
- D (Intenta resolverlo con otro/s compañero/s),
- E (Copia la solución elaborada por un docente),
- F (Consulta a docente pero no lo resuelve),
- G (Consulta a docente e intenta resolverlo solo) y
- H (Consulta a compañero e intenta resolverlo solo)

Tabla 2. Actitudes adoptadas por los alumnos, ante dificultades para diseñar algoritmos

Interesan particularmente las modas más elevadas en las categorías *Siempre* y *Casi siempre* pues es allí donde se manifiestan las conductas más seguidas por los alumnos. Estas modas señalan que usualmente se apoyan en el estudio de problemas parecidos que estén resueltos correctamente y además que consultan a los docentes buscando orientación que les permita resolver solos los diagramas de mayor dificultad.

Respecto a las producciones de los alumnos, nos hemos detenido en el análisis de una evaluación breve o coloquio, aplicado a 66 estudiantes voluntarios. Este instrumento contaba con dos ítems, uno para evaluar el diseño de un *caso de prueba* apropiado a un problema computacional planteado y el otro, para señalar de una lista de alternativas aquellos componentesⁱⁱ de diagramación

ⁱⁱ Procesos elementales realizados por un autómata. Su característica principal es la de poseer una única funcionalidad, claramente definida.

necesarios para resolver el problema en cuestión.

Nuestra hipótesis establece que los casos de prueba correctamente diseñados, facilitan la identificación de los componentes de diagramación de la solución algorítmica. En este sentido, estudiamos la distribución de los alumnos en una escala de valoración compuesta por *Muy bien*, *Regular* o *Mal*. Se consignan con *Vacío* los coloquios sin desarrollo del caso de prueba. Los resultados se muestran en el gráfico 1.

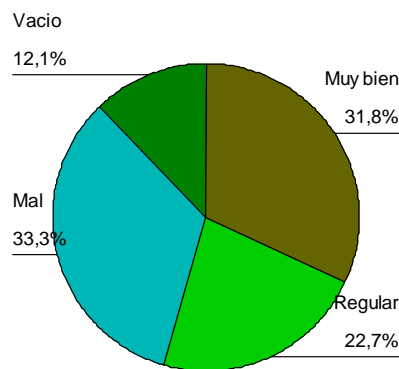


Gráfico 1: Distribución de la valoración de Casos de Prueba diseñados por alumnos

Los problemas computacionales son análogos a los problemas matemáticos bajo la categoría “problemas por resolver” tal como los entiende Polya [9]. Coincidimos con él en que, para resolver un problema se necesitan cubrir cuatro fases, tal como se muestra en la tabla 3. Basándonos en el análisis profundo a partir de los instrumentos aplicados y en la vasta experiencia docente del grupo de investigación, pudimos establecer una estrecha vinculación de las etapas formuladas por Polya con las concebidas en la resolución de problemas computacionales, algunas de las cuales hemos disgregado en nuevas subfases.

En términos de Polya	En términos computacionales	
Etapas	Fases	Subfases
I Comprender el problema	I Comprender el problema	I.1 Reconocer términos
		I.2 Identificar:
		a. Entrada/s
		b. Salida/s
		c. Condición/es
	I.3 Diseñar caso/s de prueba	
II Concebir un plan	II Seleccionar componentes	
III Ejecución del plan	III Diseñar el algoritmo	
IV Examinar la solución obtenida	IV Prueba de escritorio	

Tabla 3: Problemas matemáticos por resolver – Problemas Computacionales

La razón por la que se abren las subfases enunciadas en la tabla 3 es porque consideramos que la comprensión del problema es la piedra basal del proceso destinado al diseño de algoritmos, en el que intervienen los factores convergentes (subfases) de esta interpretación.

FASE I: Comprender el problema

I.1 Reconocer términos: El alumno manifiesta que, durante las primeras aproximaciones al problema, la mayor dificultad reside en identificar los elementos sustantivos de la formulación. Asigna mucha importancia a la ubicación de lo principal por encima de lo accesorio y se asegura de conocer cabalmente el significado de cada término involucrado en la consigna. Así, por ejemplo, si la formulación requiere la determinación de múltiplos de un número entero, el alumno se preguntará, *¿qué es un múltiplo?, ¿qué es un entero?* antes de analizar los procesos en los que actuarán estos términos. En la investigación se ha utilizado el recurso de

incluir memos como el de la figura 1, denominado **Términos/Componentes**, para señalar el posible impacto de la subfase I.1 (reconocer términos) en la posterior fase II (seleccionar componentes). Nuestra hipótesis es que algunos términos inducen al alumno a adoptar componentes relacionados con él, pero que no contribuyen a la solución del problema, tal como parece suceder con el alumno Agüero.

I.2 Identificar Entrada/s, Salida/s y Condición/es:

Al igual que en los problemas matemáticos, el alumno intenta aislar las partes principales del problema. En el contexto de diseño de algoritmos, la incógnita es la salida, los datos constituyen la entrada y las condiciones son las restricciones a las que se someten los datos, tanto de entrada como de salida. En este punto hacemos referencia a la analogía entre el problema matemático y el problema computacional, porque advertimos que el alumno posee cierta experiencia en la identificación de datos y condiciones. Sin duda, sus conocimientos previos se traducen en la competencia lectora e interpretativa necesaria para decodificar las diversas versiones de consigna. Sin embargo, estas competencias también deben madurar a la par que se incorpora la notación computacional y el lenguaje que le es propio. Paulatinamente se construye el sentido de expresiones tales como “...dados N valores enteros correspondientes a las edades ...”, asumiendo que el usuario cuenta con N datos de entrada pero que es imprescindible contar con un autómata capaz de lograr que esos N datos se almacenen en la memoria de la computadora, que los usará para el propósito de la salida. Los valores de la tabla 1 corroboran la escasa dificultad que dicen tener los alumnos para cumplir con esta subfase.

La figura 1 muestra una red de citas de alumnos (entrevistas y coloquios documentados) asociadas a esta fase.

I.3 Diseñar caso/s de prueba: Un caso de prueba está constituido por una colección de

datos de entrada y las condiciones o restricciones que sobre ellos operan, necesarios para obtener la salida del diagrama que modela la solución del problema.

Inculcamos a nuestros alumnos la práctica de diseñar tantos casos de prueba como sean necesarios para ejercitar con ellos la solución manual del problema. Entendemos por solución *manual* aquella que provee la solución a partir de los procesos mentales que ejecuta un sujeto pensante sobre el caso de prueba. En esta fase, la *disposición a pensar* se ejercita sobre un ejemplo (el caso de prueba) cuidadosamente elegido, para indagar los sucesivos pasos a los que debe ser sometido con el fin de llegar a la solución. Podríamos decir que esta fase se inspira en el hecho de que si *manualmente* no podemos resolver el problema, menos aún podremos diseñar un algoritmo ejecutable *automáticamente* por un autómatas.

Las figuras 2 y 3 corresponden a casos de prueba diseñados por dos alumnos para un mismo problema. En el primer caso, vemos que se identifica el dato de entrada (K) y su condición (K es un número natural). En el segundo caso, en cambio, se omite la condición y no se define precisamente el valor de K, con lo cual resulta imposible ejecutar la prueba.

FASE II: Seleccionar componentes

Los conceptos básicos de la Diagramación son enseñados dentro del marco de los problemas que los utilizan. El alumno los aprende, inicialmente, como herramienta de solución asociada únicamente a ese problema y no a toda la *familia* de problemas en la que dicha herramienta puede reutilizarse. No obstante, es justamente con la reutilización de estos recursos, que comienza gradualmente un proceso de resignificación del concepto que lleva al sujeto a asignarle entidad propia como componente de diagramación.

Esta fase se corresponde con la de *concebir un plan*. Para transitar por ella, el alumno busca puntos de contacto con sus conocimientos previamente adquiridos. Intenta recordar lo que le resultó útil en el pasado ante circunstancias parecidas, es decir, intenta reconocer algo *familiar* en lo que examina. Sus conocimientos previos constituyen una colección de herramientas con funcionalidad clara y precisa. En palabras de los propios alumnos “*todos los conceptos para resolverlos están dados*”, haciendo referencia, justamente, a los componentes. Sin embargo, surge de la investigación que, a) o no se comprende cabalmente la funcionalidad de estos autómatas o b) algunos de ellos están tan sobrevalorados que se cree, sirven para infinidad de propósitos.

Del análisis de los coloquios se observa la situación a) en el 33.85% de los casos, en los que, por ejemplo, los alumnos señalan necesitar un componente *Acumulador* cuando en realidad necesitan un *Contador*, cuya funcionalidad es similar pero cada uno posee salidas distintas que lo convierten en una entidad propia. Estos conceptos han sido enseñados enfatizando similitudes y diferencias, pero éstas últimas parecen estar confusas para ellos, aún cuando testimonian *conocer todos los conceptos*. En este punto, interesa sobre todo señalar, que aún transitando con éxito las fases previas, el alumno adopta componentes que nada tienen que ver con el problema.

Respecto a b) surge que el 14.06% elige un componente más complejo que el necesario, siendo este tipo de estrategia inadmisibles desde el punto de vista computacional. Otra vez, en nuestro ejemplo, los alumnos seleccionan el componente *Método de Ordenamiento* siendo que el problema requiere la *Determinación de Mayor* dentro de una colección.

Las evidencias presentadas en las figuras 2 y 3 muestran que el alumno Gutiérrez transitó exitosamente hasta esta fase, mientras que

Agüero no lo logró. En el primer caso, es sugestivo que un caso de prueba bien diseñado y un análisis exhaustivo del mismo, facilite la selección adecuada de los componentes. En el segundo caso, por el contrario, las fases I y II no se transitaron adecuadamente. Analizando las diferencias entre ambos casos y asumiendo que *todos los conceptos están enseñados*, nos inclinamos a atribuir un peso importante a la *disposición a pensar* (ver figura 1), es decir, la abstracción reflexionante como una competencia personal que sostiene los procesos de construcción y reconstrucción de las situaciones problemáticas computacionales.

FASE III: Diseñar el algoritmo

Nuestros alumnos señalan tener la mayor dificultad en la fase III ya que, a partir de componentes de estructura rígida, deben elaborar una combinación articulada de ellos para arribar a un algoritmo solución. Si interpretó correctamente el enunciado, lo plasmó en un caso de prueba y seleccionó correctamente los componentes entendiendo la funcionalidad de los mismos, creemos que podrá ajustarlos de manera que estos componentes se conjuguen en el diagrama N-S, solución del problema. Actualmente, la investigación está dedicada al análisis cualitativo de esta fase, con alumnos ingresantes en el año 2008. Sin embargo, respecto a los casos registrados en el año 2007, los resultados indican que los alumnos con caso de prueba y selección de componentes correctamente elaborados, obtuvieron un rendimiento satisfactorio en la evaluación que acredita los contenidos de Diagramación. El 83% de los sujetos aprobó el parcial con una nota promedio de 75/100. Sólo el 17% restante aprobó con nota destacada en la instancia de recuperación.

FASE IV: Prueba de escritorio

Esta es una fase cuyo análisis no se ha desarrollado todavía. Los alumnos presentan sus diagramas N-S sin adjuntar la correspondiente prueba, debido a que, desde el

punto de vista de la evaluación, sólo es necesario el algoritmo solución para acreditar conocimientos. Actualmente, el grupo de investigación está diseñando los instrumentos adecuados para registrar la evolución de las pruebas de escritorio, presumiendo que la utilización de las mismas como mecanismo de control sobre sus competencias de diagramación, es variable en función del nivel de experticia del sujeto y del grado de complejidad del problema computacional.

El esquema presentado en figura 4 integra las fases de resolución de problemas computacionales, desde su inicio a partir de la formulación hasta su máximo nivel de solución optimizada. Cada rectángulo sugiere la actividad asociada a la fase o subfase. Entendemos que el degradado ilustra la composición de esquemas cognitivos, procedimentales y actitudinales sustentados por los sujetos, conformando una mixtura de elementos que contribuyen a la consecución de la fase. Los rectángulos se grafican con diferentes bases y alturas porque las primeras sugieren el grado de aporte de cada fase respecto de la construcción buscada; mientras que las alturas señalan la contribución que hace esa fase a la solución buscada. Así, una fase de base ancha, como puede ser la fase II indica la importancia de concebir un plan para alcanzar un diagrama solución y una mayor altura denota el nivel de corrección que se adquiere al transitar por la fase. Ahora bien, ¿se arriba a la solución del problema por consecución de una o todas las fases? Creemos que cada rectángulo actúa como pistón que crece a medida que se construyen las etapas de la resolución y decrecen al ritmo de los conflictos cognitivos, los procesos constructivos y la disposición actitudinal. Respecto a la dinámica del esquema, decimos que cada uno de ellos se mueve en paralelo con los otros pero a distinto ritmo, sugiriendo un proceso espiralado, iterativo e incremental.

FUTURAS LINEAS DE ACCIÓN

Este trabajo constituye una primera indagación que abre las puertas a otros posibles trabajos. Las preguntas de investigación son varias, ¿hay transiciones entre fases de mayor dificultad?, si esto fuera afirmativo, ¿cuál es la transición de mayor dificultad?, ¿la dificultad está relacionada con la situación-problema o es independiente de la formulación?; la respuesta a éstas y otras cuestiones nos orientará en la elaboración de estrategias didácticas de enseñanza que deriven en procesos constructivos desde los sujetos de aprendizaje, en la búsqueda de una mejora de la práctica educativa.

BIBLIOGRAFÍA

1. Camilloni, A. y otros (1998) "Debates constructivistas" Ed. Aique. Buenos Aires.
2. Cabero, Bartolomé (Editor) (2000). "Nuevas Tecnologías aplicadas a la Educación". Madrid: Editorial Síntesis.
3. Carretero, Mario (2004) "Introducción a la Psicología Cognitiva". Editorial Aique.
4. López, Marcela otros (2003) "Un sitio para Elementos de Computación" publicado en Libro de Resúmenes de la "III Conferencia Argentina de Educación Matemática". Página: 170. Salta, 9 al 11 de octubre de 2003.
5. López Ruiz, Juan Ignacio (1999) "Conocimiento docente y práctica educativa" Ediciones Aljibe.
6. Mac Gaul, Marcia y otros. (2005) "Las nuevas tecnologías en apoyo del ingreso universitario masivo: alumnos activos versus alumnos presentes". Publicado en actas del Congreso Internacional de Educación Superior y Nuevas Tecnologías.
7. Pedró, Francesc (1999). "Una revisión de las formas de enseñar y aprender en la Universidad". En: Temas y Propuestas. Revista de la Secretaría Pedagógica de la Facultad de Ciencias Económicas de la UBA. Año 8, Nro. 17, octubre de 1999.
8. Perkins, D. y Smith, E. (1990). "Enseñar a Pensar. Aspectos de la aptitud intelectual". Barcelona: Paidós. Temas de Educación. Primera Parte: "El desafío de enseñar a pensar".
9. Polya, G. (1990) "Cómo plantear y resolver problemas". Editorial Trillas.
10. Porlán, Rafael (1991) "Constructivismo y Escuela". Editorial Díada.
11. Pozo Muncio, Ignacio (2003) "Aprendices y Maestros. La nueva cultura del aprendizaje". Ed. Alianza.
12. Raths, L. E y otros (1988) "Cómo enseñar a pensar. Teoría y aplicación". Buenos Aires. Piados Studio. Introducción: Las operaciones del pensamiento.
13. Sautu, Ruth (2003) "Todo es teoría". Editorial Lumiere.

Anexo: Figuras

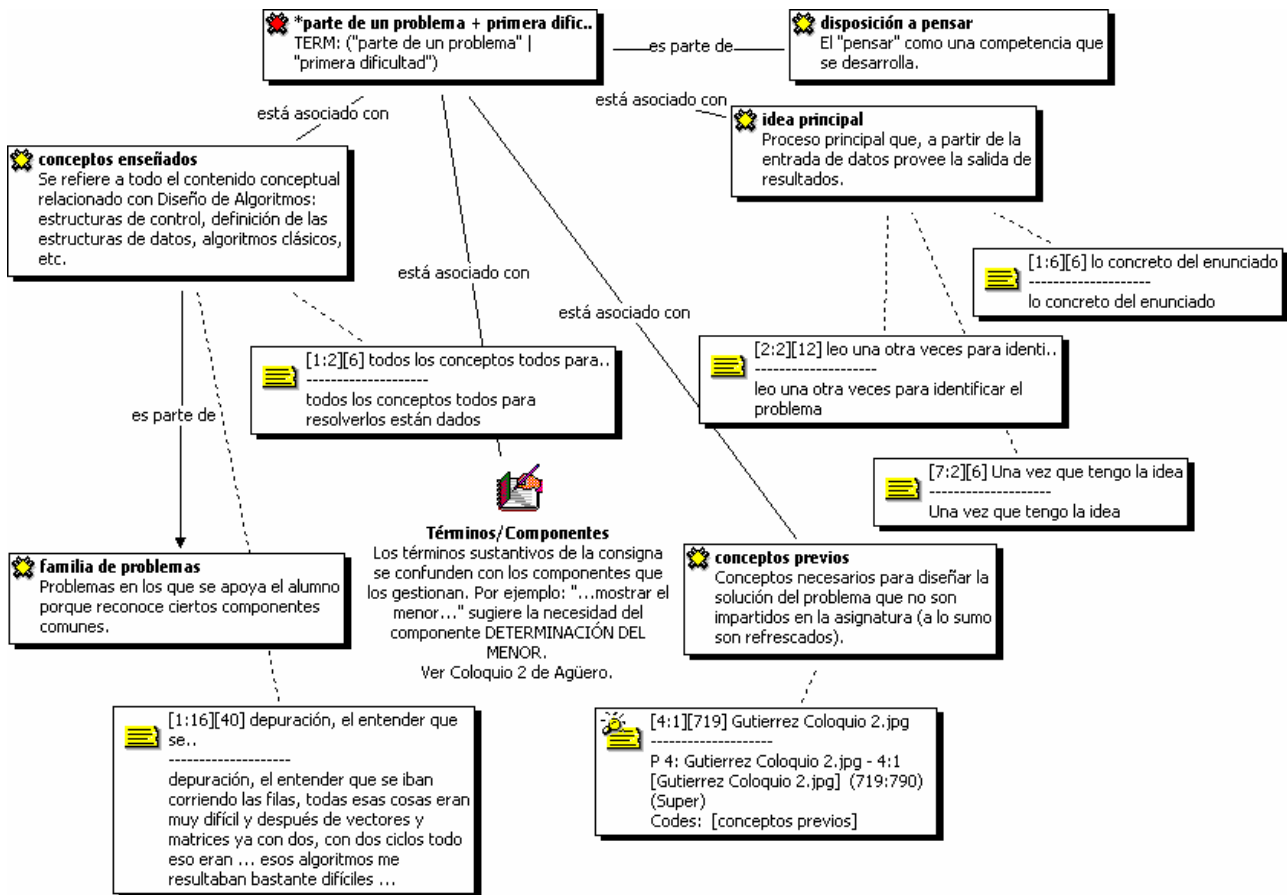


Figura 1: Red de citas asociadas con la fase I

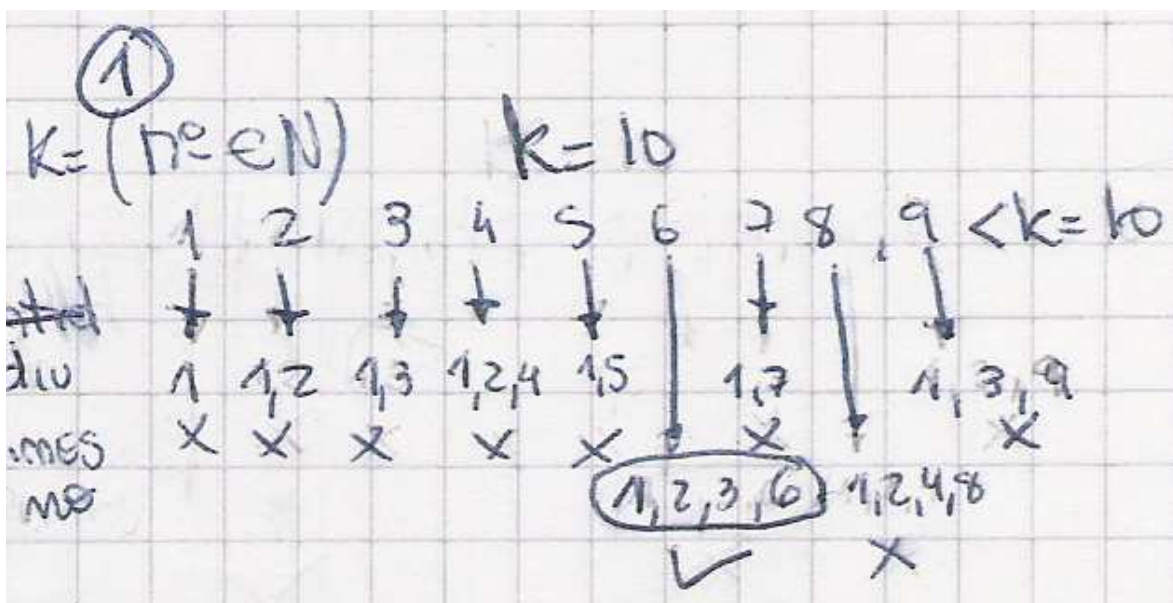


Figura 2: Caso de prueba bien diseñado (alumno Gutiérrez)

$K \in \mathbb{N} / K = 5; 2; 4; 7; 3; 1; 1$

Figura 3: Caso de prueba mal diseñado (alumno Agüero)

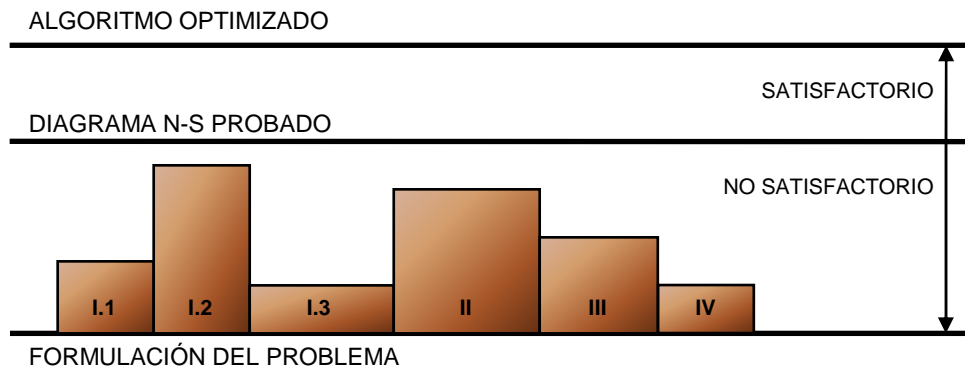


Figura 4: Esquema integrado de fases en la resolución de problemas computacionales