

RISCE Revista Internacional de Sistemas Computacionales y Electrónicos

MARZO 2010
Número 2, Volumen 2, Año 2



e
a
n
d
r

RISCE Revista Internacional de Sistemas Computacionales y Electrónicos; es una publicación bimestral del Instituto Politécnico Nacional, Av. Luis Enrique Erro S/N, unidad “Profesional Adolfo López Mateos”, Del. Gustavo A. Madero, C.P. 07738, México D.F. a través de la Escuela Superior de Computo; Av. Juan de Dios Bátiz S/N esquina Miguel Othón de Mendizábal. “Unidad Profesional Adolfo López Mateos”. Col. Lindavista C.P. 07738, México, D. F. tel. 57296000 ext. 52000. Certificado de reserva de Derechos al uso Exclusivo del título No. 04-2008-062613190500-203, ISSN en trámite. Los artículos son responsabilidad exclusiva del autor y no reflejan necesariamente el criterio de la institución, a menos que se especifique lo contrario. Se autoriza la reproducción total o parcial, siempre y cuando se cite explícitamente la fuente.

La revista se especializa en el área de los sistemas computacionales y electrónicos; tanto en el desarrollo, como en la investigación en:

- Ciencias de la Computación
- Cómputo educativo
- Cómputo Móvil
- Comunicaciones
- Disciplinas Emergentes
- Electrónica
- Física Electrónica
- Ingeniería de Cómputo
- Ingeniería de Software
- Innovación Tecnológica
- Inteligencia artificial
- Matemática computacional
- Procesamiento de señales
- Robótica y cibernética
- Sistemas de Información
- Tecnologías de la Información

Distribución

La revista cuenta con 300 ejemplares que se distribuyen en:

Europa, Asia y América Hispana; mediante CD ROM y correo electrónico

Directorio



INSTITUTO POLITÉCNICO NACIONAL

DRA. YOLOXÓCHITL BUSTAMANTE DÍEZ
DIRECTORA GENERAL

ING. JUAN MANUEL CANTÚ ALVAREZ
SECRETARIO GENERAL

DR. EFREN PARADA ARIAS
SECRETARIO ACADEMICO

DR. JAIME ALVAREZ GALLEGOS
SECRETARIO DE INVESTIGACIÓN Y POSGRADO

ING. ERNESTO MERCADO ESCUTIA
SECRETARIO DE SERVICIOS EDUCATIVOS

ING. OSCAR JORGE SÚCHIL VILLEGAS
SECRETARIO DE EXTENSIÓN E INTEGRACION SOCIAL

M. EN C. FERNANDO ARELLANO CALDERON
SECRETARIO DE GESTION ESTRATEGICA

C.P. ROBERTO ALVAREZ ARGUELLES
SECRETARIO DE ADMINISTRACION

LIC. JUDITH CLAUDIA RODRIGUEZ ZUÑIGA
DEFENSORA DE DERECHOS POLITECNICOS



ESCUELA SUPERIOR DE CÓMPUTO

ING. APOLINAR FRANCISCO CRUZ LÁZARO
DIRECTOR

DR. FLAVIO ARTURO SÁNCHEZ GARFIAS
SUBDIRECTOR ACADÉMICO

DR. JESÚS YALJÁ MONTIEL PÉREZ
JEFE DE LA SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

LIC. ARACELI LOYOLA ESPINOSA
SUBDIRECTORA DE SERVICIOS EDUCATIVOS E INTEGRACIÓN SOCIAL

M. EN C. JUAN VERA ROMERO
SUBDIRECTOR ADMINISTRATIVO

DR. EDUARDO BUSTOS FARÍAS
EDITOR DE RISCE

Miembros del comité Revisor

(Todo el comité técnico está formado por doctores en ciencias o su equivalente)

Francisca Losavio de Ordaz (Venezuela) (Universidad Central de Venezuela)

Alfredo Matteo (Venezuela) (Universidad Central de Venezuela)

Emmanuel F. Moya Anica (México)

Edgardo Manuel Felipe Riverón (Cuba) (México) (CIC)

Luis Enrique Palafox Maestre (México)

Eduardo F. Caicedo Bravo (Colombia)

Hilda Ángela Larrondo (Argentina)

Guillermo Leopoldo Kemper Vásquez (Perú)

Elizabeth León Guzmán (Colombia)

María Cecilia Rivera (Chile)

Satu Elisa Schaeffer (Finlandia) (UANL)

Rafael Canetti (Uruguay)

Javier Echaiz (Argentina)

Pablo Belzarena (Uruguay)

Carlos Beltrán González (Italia) (Università di Genova)

Elena Fabiola Ruiz Ledesma (México)

Jonatan Gómez (Colombia)

Armando De Giusti (Argentina)

Juan José Torres Manríquez (México)

Jesús Yaljá Montiel Pérez (México)

Luis Alfonso Villa Vargas (México)

Marco Antonio Ramírez Salinas (México)

Félix Moreno González (España) (UPM)

Salvador Godoy Calderón (México) (CIC)

INDICE

Avances en la Integración de Planificación y Scheduling	6
Ma. de Guadalupe García-Hernández, Gerardo Enrique Canedo-Romero, Donato Hernández-Fusilier	
Herramienta computacional para elaborar software educativo	19
Saturnino Job Morales Escobar, Dora María Martínez Magaña, Ivonne Rodríguez Pérez, Sara Lilia	
Desarrollo de un sonómetro digital.....	30
Pablo Roberto Lizana Paulin, José de Jesús Negrete Redondo, Sergio Beristain	
Sistema de detección del nivel de glucosa en sangre para la administración de insulina	38
Rubén Ortega González, José Sánchez Juárez, Fabiola Ocampo Botello, Roberto de Luna Caballero	
Instrucciones para los autores	43

Avances en la Integración de Planificación y Scheduling

Ma. de Guadalupe García-Hernández, Gerardo Enrique Canedo-Romero, Donato Hernández-Fusilier

División Ingenierías del Campus Irapuato Salamanca, Universidad de Guanajuato
Comunidad de Palo Blanco s/n, C.P. 36885. Salamanca, Guanajuato, México. Tel. 52-464-6479940 Ext.2467
[.garcia, canedo, donato}@salamanca.ugto.mx](mailto:{garcia, canedo, donato}@salamanca.ugto.mx)

Resumen

Dentro de la Inteligencia Artificial existen dos grandes áreas de conocimiento: Planificación y Scheduling, las cuales han sido estudiadas y aplicadas en forma aislada. La primera se centra en la selección de las acciones para obtener un buen plan y la segunda en la asignación de recursos para ejecutar adecuadamente dicho plan. Sin embargo, al tratar de resolver problemas del mundo real, la comunidad de planificación en IA ha visto la necesidad de combinar ambos procesos. En el estado del arte existen dos perspectivas que intentan resolver este problema: planificación extendida y Scheduling extendido. En el primer caso resulta mucha complejidad computacional y en el segundo no se puede replanificar debido a que maneja plantillas rígidas de planes. En este artículo estamos proponiendo un modelo integrado que aprovecha homogénea y simultáneamente las capacidades de ambos procesos para poder abordar la resolución de problemas reales. Este enfoque contiene dos módulos (uno para cada proceso), que trabajan juntos de una forma fuertemente acoplada para resolver los conflictos que se presentan durante la revisión de un plan, cuyas acciones son muy dependientes de sus restricciones numéricas. Este artículo también presenta la estructura y el flujo de información del modelo integrado, acompañándolos con un ejemplo que muestra su adecuada aplicabilidad al mundo real.

Abstract

Inside Artificial Intelligent there are two great knowledge areas: Planning and Scheduling, they have been studied separately. The first one studies action selection for obtaining a good plan. The second one studies the suitable resources application in that plan. However, the AI Planning Community, when had tried to solve real-world problems, have seen necessary to combine both processes. At the State-of-the-Art there are two approaches trying to solve this problem: extended Planning and extended Scheduling. The first case results in high computational complexity and the second case cannot modify the initial plan because it uses strong templates of plans. In this paper we propose an integrated model who uses in an homogeneous way the capabilities of both processes for tackling real-world problems. This model has two modules (one for each process), where they work together in a strongly coupled form for solving conflicts in a plan, whose actions are highly dependent of their numeric features. This paper presents the structure and the information flow of the integrated model too, along with an example that shows its suitable applicability to real-world.

Palabras clave: Integración de Planificación y Scheduling, Satisfacción de Restricciones

Keywords: Planning Scheduling Integration, Constraint Satisfaction

I. Introducción y motivación.

Dentro de Inteligencia Artificial (IA) existen dos grandes áreas de estudio: Planificación y Scheduling (calendarización), que han sido vistas en forma aislada, pues la primera se ocupa de seleccionar acciones para obtener un buen plan y la segunda de asignar recursos para ejecutar adecuadamente dicho plan. Sin embargo, al tratar de resolver problemas del mundo real, la comunidad de planificación en IA ha visto la necesidad de aplicar simultáneamente las capacidades de ambos procesos, pues estos problemas han requerido del manejo de la duración de tareas, costos de ejecución, uso limitado de recursos, criterio de optimización, restricciones temporales, persistencia de efectos, etc. Al respecto, algunos de los problemas que se han tratado de resolver han sido: la automatización de procesos de manufactura, la elaboración de planes logísticos, la actuación ante

catástrofes, la optimización de rutas turísticas, la navegación de robots autónomos, el control de tráfico terrestre en aeropuertos, la distribución de energía, etc. [1], [2], [3]. Por lo tanto es prácticamente imposible separar las características de Planificación y de Scheduling durante la resolución de un plan correspondiente al mundo cotidiano, pues la elección de las acciones está influenciada por las restricciones impuestas sobre sus recursos, mientras que la elección de los recursos a utilizar está influenciada por las acciones planificadas que los requieren [4]. Por lo tanto, se ha demostrado que estos procesos se complementan perfectamente, lo que motivó al diseño de un modelo general que sea aplicable a cualquier dominio. De igual manera que sea flexible para que integre al máximo las capacidades de ambos procesos, especialmente si se considera que ambos procesos usan técnicas similares, como la búsqueda en grafos, uso de heurísticas, manejo y razonamiento de restricciones, etc. [5].

Después de algunos intentos de integración de estos procesos [6], [7], [8], se ha planteado la siguiente cuestión: ¿Cuál debe ser el diseño del modelo que, homogénea y simultáneamente, aproveche las capacidades de ambos procesos? Para responderla se debe poner énfasis en los siguientes aspectos: estructura del modelo de integración, tecnología utilizada por cada proceso y su trabajo en equipo para la resolución de conflictos. Para abordar ese reto, este artículo presenta un modelo de integración de dichos procesos que, a nuestro juicio, resulta ser muy novedoso debido a que intercala a un planificador y un scheduler utilizando al máximo sus capacidades. Consideramos que nuestras principales contribuciones son: i) la propuesta de una estructura integrada de ambos procesos, ii) el uso de una red de acciones que representa a las acciones con sus enlaces causales y restricciones numéricas, iii) la descripción del flujo de información en este modelo integrado y su algoritmo resultante, e iv) un ejemplo que muestra la aplicabilidad de esta estructura a problemas del mundo real.

II. Descripción del problema de Planificación y Scheduling.

La resolución de un problema de Planificación y de Scheduling deberá involucrar la ejecución de un plan parcial, cuyas acciones tendrán que satisfacer algunas restricciones (tanto de tiempo como de disponibilidad de recursos). Entonces, para modelar un problema de Planificación y Scheduling se necesita definir lo siguiente:

- Estado inicial y las metas del problema, con la información que es verdadera al inicio del mismo y los hechos que necesitan ser obtenidos al final del mismo, respectivamente.
- Acciones, con su duración, condiciones y efectos, que pueden proporcionar diferentes alternativas para la obtención de las metas del problema.
- Recursos, los cuales deberán estar disponibles para la ejecución de las acciones planificadas. Estos pueden ser consumibles o reutilizables [9], [10].
- Función métrica, como una función de multicriterio que necesita ser optimizada, para balancear los requerimientos heurísticos de cada proceso.
- Restricciones del problema sobre los requerimientos de las acciones planificadas, que pueden ser obligatorias (duras) o de preferencia (suaves) [11].

Por otro lado, no existe un lenguaje común entre Planificación y Scheduling. Desde la perspectiva de Planificación, se utiliza el Lenguaje de Definición de Dominio cuya última versión es PDDL3.0 [11]. El dominio consiste en un listado de todas las acciones que se pueden utilizar, estableciendo sus duraciones, condiciones y efectos. El problema define la situación inicial y las metas, así como todos los objetos que están involucrados en el problema. Sin embargo, este lenguaje presenta limitaciones para describir las características del Scheduling tales como: uso de recursos, tiempos límite para que suceda algo dentro del plan, restricciones temporales complejas (persistencia de efectos, lapso de tiempo entre dos acciones, etc.), que modelarían adecuadamente a un problema del mundo real.

III. Desarrollo del Modelo de Integración propuesto.

Una vez establecida la dependencia entre Planificación y Scheduling, se procede a conformar la estructura del plan (razonamiento del planificador sobre sus condiciones, efectos, ordenamiento y enlaces causales) y la satisfacción de sus restricciones numéricas (razonamiento del scheduler sobre la línea del tiempo y el uso de

recursos). Puede presentarse el conflicto por violación de alguna de estas restricciones cuando se tienen acciones muy dependientes de sus condiciones numéricas y para resolverlo será necesario modificar la estructura del plan. Por ejemplo, en un problema de aplicación de *Rover* (dominio utilizado en *IPC-02, International Planning Competition*, <http://ipc.icaps-conference.org>), se tiene un explorador interplanetario que requiere de una batería que le provee de energía, dicha batería puede ser recargada solamente en los sitios donde se recibe luz solar (los cuales están muy limitados) y, si no se considera el consumo de energía durante la ejecución de sus acciones, entonces el explorador nunca requerirá captar energía solar para su batería y, en consecuencia, le será imposible conseguir el objetivo deseado.

Una alternativa posible para el manejo de problemas reales es la relajación de características numéricas en las acciones, separando así la parte proposicional (tarea del planificador) del manejo de recursos y sus características numéricas (tarea del scheduler) [12], [4]. Otras formas de integración también no muy exitosas de estos procesos han sido llevadas a cabo [13], [8]. La aproximación más ampliamente utilizada, pero con la desventaja de no poder enfrentar restricciones de tiempo y/o de recursos, es la planificación temporal, tales como O-Plan [14], TOSCA [15], IxTeT [16], LPG [17], [18], [19] y TPSYS [20], entre otros. También en el estado del arte se encontraron dos problemas del mundo real que han sido resueltos satisfactoriamente mediante un enfoque integrado: ASPEN [7] y HSTS [8]. Su inconveniencia radica en que no proporcionan un modelo de integración general, pues se han construido específicamente para un problema en concreto. Esto pone de manifiesto la necesidad de estudiar y proponer técnicas para un modelo independiente del dominio de aplicación [4].

3.1 Estructura del Modelo de Integración

El modelo de integración que se muestra en la Figura 1 [4] contempla un esquema de dos módulos: uno para Planificación y otro para Scheduling. Como antes se señaló, se requiere como entrada la definición de un dominio y de un problema. Este modelo también requiere de un plan, que si se alimenta un plan vacío (es decir que sólo contiene al estado inicial y las metas) se ha comprobado que aumenta mucho la complejidad computacional [4]. Actualmente existen muchos planificadores en el Estado del Arte que pueden generar planes ejecutables proposicionalmente (o sea que sus proposiciones, como son las condiciones y los efectos, se “enganchan” perfectamente, también son denominados planes *Strips*) de una forma eficiente [21], [22]. Se comprobó que con este tipo de plan como entrada al módulo de integración se simplificó mucho su procesamiento. De cualquier manera esto no limita a dicho modelo, pues incluso puede tener como entrada a un plan relajado [23], [24] o a una secuencia de actividades elaborada manualmente, pues no es condición necesaria que el plan de entrada sea un plan *Strips*, debido a que el módulo de planificación puede reparar condiciones no soportadas por los efectos de otras acciones del mismo plan [4].

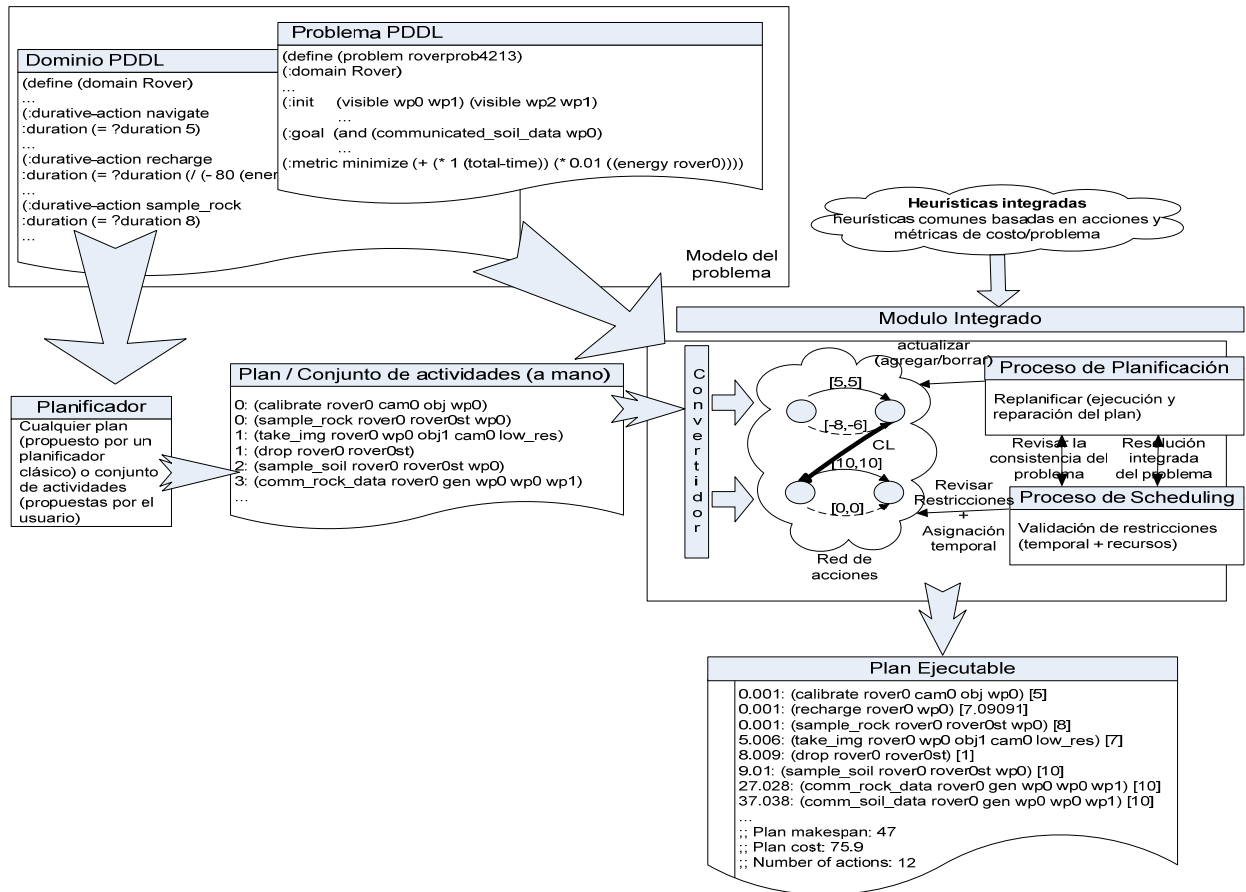


Figura 1 Arquitectura del modelo propuesto de integración

Una de las novedades que presenta este modelo de integración es que utiliza un convertidor del plan a una red de acciones (*Action Network*, AN), que transforma al plan de entrada en una red que contiene a sus todas sus acciones, enlaces causales y restricciones sobre tiempo y recursos. Se usa esta red para poder plasmar toda la información métrico-temporal, la cual sigue la filosofía de la Red de Restricciones Temporales (*Temporal Constraint Networks*), TCN [25] y de la Red de Recursos Consumibles (*Consumable Resource Network*), RCN [9]. Los nodos de dicha red son puntos en el tiempo: uno para el inicio (.on) y otro para el término (.off) de cada acción. También contiene tres tipos de arcos o aristas etiquetadas (unión de dos nodos) que representan: i) uso de un recurso entre los nodos .on y .off de la misma acción (representado mediante etiquetas numéricas), ii) enlaces causales entre nodos de diferentes acciones e iii) restricciones temporales entre nodos.

Un ejemplo de AN se muestra en la Figura 2, ubicado en el ya mencionado dominio *Rovers*, donde la acción *calibrate (rover0 camera0 obj1 w0)* incrementa el tiempo en cinco unidades, que es la duración de dicha acción, y decrementa la energía del explorador *rover0* en dos unidades, definida en el intervalo de la arista entre sus puntos de inicio y término, como puede observarse en la Figura 2. La acción *calibrate (rover0 camera0 obj1 w0).off* tiene un enlace causal con *take_image (rover0 wp0 obj1 cam0 low_res).on* debido a que la primera acción genera la proposición (*at end(rover w0)*) que la segunda necesita en (*over all (rover w0)*) para su ejecución, es decir, que el explorador siempre se encuentre en el punto *w0* para tomar la imagen solicitada. Finalmente, las restricciones temporales permiten definir restricciones adicionales como es el límite de tiempo (*deadline*) para la ejecución de una acción, por ejemplo *sample_soil (rover0 rover0store w0)* debe terminar antes del punto de tiempo 50'.

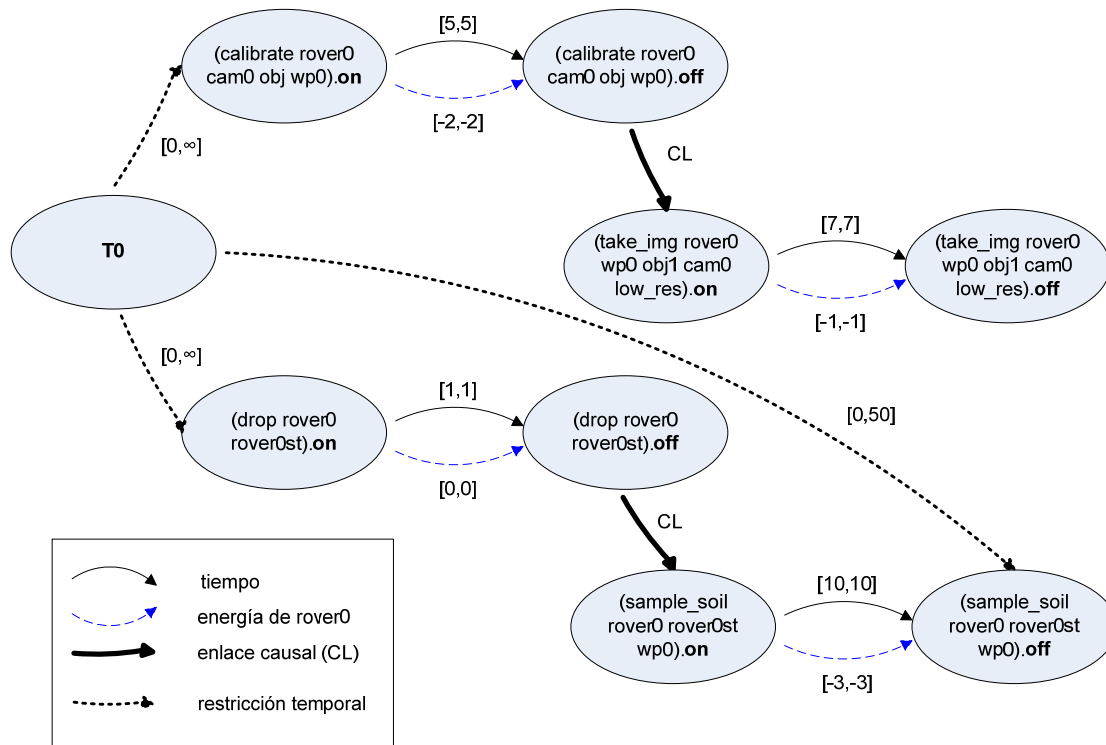


Figura 2 Ejemplo de una red de acciones

Siguiendo con la Figura 2, las líneas delgadas sólidas representan al tiempo y las punteadas representan el uso del recurso (que en este caso es la energía del explorador) por las acciones (donde el signo negativo significa decremento del mismo), las líneas gruesas sólidas son los enlaces causales, las líneas gruesas punteadas son las restricciones temporales y T0 identifica al inicio del tiempo. Esta red de acciones resultó muy útil para la representación de toda la información requerida para buscar la solución de un problema mediante la integración de Planificación y Scheduling [4].

3.2 Tecnología disponible para el manejo de conflictos

Cuando los procesos de Planificación y Scheduling están tratando de resolver sus tareas por separado, es común que se presenten conflictos en ellas. Para resolver estos conflictos, cada proceso dispone de algunas técnicas ya muy estudiadas en el estado del arte. En el caso de planificación, una técnica muy útil es la rutina de replanificación o reparación de fallos (*flaw repair*) [30], que es aplicada cuando un plan está siendo ejecutado y el mundo cambia de una manera inesperada. Durante la replanificación, el planificador trata de utilizar la mayor parte del plan ya hecho, buscando mejores alternativas para resolver el problema que se le presenta después de haber cambiado el mundo. Sin embargo, cuando se trata con planes que manejan restricciones de tiempo y de recursos, las necesidades de reparación del plan pueden ser causadas por la violación en alguna restricción. Por lo tanto, un fallo está siempre asociado con un estado o con una atribución de recurso (el cual puede estar agotado o no haber aún sido liberado). De ahí que, dado un plan parcial en construcción, se distinguen tres tipos de fallos [10]:

- submetas pendientes de obtener o condiciones no soportadas, que se pueden resolver mediante la inserción de nuevas acciones que lleven a obtener dichas submetas o que den soporte a dichas condiciones.
- amenaza de exclusión mutua entre una acción candidata al plan y una o más acciones del plan en construcción, que puede ser solucionada mediante promoción, democión o separación de alguna de las acciones protagonistas.

- conflicto de recursos, que se puede dar por insuficiencia de éstos durante su instanciación en las acciones del plan en construcción. Se puede resolver mediante restricciones de precedencia, sustitución por una acción menos consumidora o inserción de una acción productora del recurso en conflicto.

Por otro lado, dentro de Scheduling existe el Problema de Satisfacción de Restricciones, conocido como el paradigma CSP (*Constraint Satisfaction Problem*) [26], [29]. En general, el CSP es resuelto mediante dos aproximaciones complementarias: búsqueda hacia atrás y algoritmos de consistencia en redes. También se han propuesto técnicas de descomposición y de revisión heurística [25].

3.3 Flujo de información del Modelo de Integración

Una vez estudiada la estructura del modelo integrado, en esta sección se hace un análisis detallado del flujo de la información entre ambos procesos, el cual es presentado en la Figura 3 [4]. Ahí se pueden apreciar claramente dos módulos (uno para cada proceso) así como su flujo de información. Cuando el módulo de planificación detecta un conflicto significa que existe al menos una condición no soportada de alguna de las acciones pertenecientes a AN o también que existe una amenaza entre dos acciones de AN. Cuando el módulo de Scheduling detecta un conflicto significa que se ha consumido más cantidad de recurso de la que inicialmente había o que el recurso está siendo utilizado por otra acción. Como el tiempo también es un recurso, también puede presentar conflicto durante la asignación de acciones en el plan parcial. Cuando el conflicto es por recursos o tiempo insuficientes, el scheduler se lo comunica al planificador utilizando los siguientes parámetros heurísticos [4],

- R: recurso en conflicto,
- q: cantidad que se necesita del recurso,
- ti: punto del tiempo (nodo) donde el recurso es necesario,
- {A}: conjunto de acciones susceptibles de ser borradas para reparar el conflicto,
- {C}: conjunto de restricciones sobre el recurso que debe satisfacerse.

El Algoritmo 1 [4] presenta la secuencia de pasos necesarios para resolver dichos conflictos y las subtarear requeridas para la obtención de las metas establecidas por el problema, donde cada proceso trata con su respectivo subproblema [27].

- 1: rev_PLN \leftarrow IS (situación inicial)
- 2: rev_SCH \leftarrow \emptyset
- 3: **mientras** \exists a (una acción) \in AN | a \notin rev_SCH **haga:**
- 4: {planificador; razonamiento sobre enlaces causales}
- 5: a \leftarrow acción disponible en este instante para ser ejecutada en AN
- 6: rev_PLN \leftarrow rev_PLN \cup {a}
- 7: **si** el número de conflictos en rev_PLN $>$ 0 **entonces,**
- 8: *solve_PLN_conflict* (rev_PLN)
- 9: {scheduler; razonamiento sobre tiempo y recursos}
- 10: **para** toda $a_i \in$ rev_PLN | $a_i \notin$ rev_SCH **haga:**
- 11: **si** \exists una colocación consistente de a_i en rev_SCH **entonces,**
- 12: rev_SCH \leftarrow rev_SCH \cup { a_i }
- 13: **en caso contrario,**
- 14: *solve_SCH_conflicts* (rev_PLN, parámetros heurísticos)

Algoritmo 1. Esquema general para la integración de Planificación y Scheduling

El planificador arregla todos los conflictos que se presentan, aplicando la rutina Reparación de Fallos arriba mencionada. Entonces el planificador llama al scheduler para que le sugiera qué debe de hacer para tomar la mejor decisión. Los puntos clave de este comportamiento funcional son dos llamadas al proceso de Planificación y las tareas derivadas de esas llamadas (siguiendo al Algoritmo 1):

1. llamada a resolver conflictos de planificación, *solve_PLN_conflicts* (paso 8)
2. llamada a resolver conflictos de *scheduling*, *solve_SCH_conflicts* (paso 14)
3. aplicación de un criterio de decisión para ejecutar el problema de reparación,

4. recomendación provista por el *scheduler* para ayudar al proceso de búsqueda en planificación.

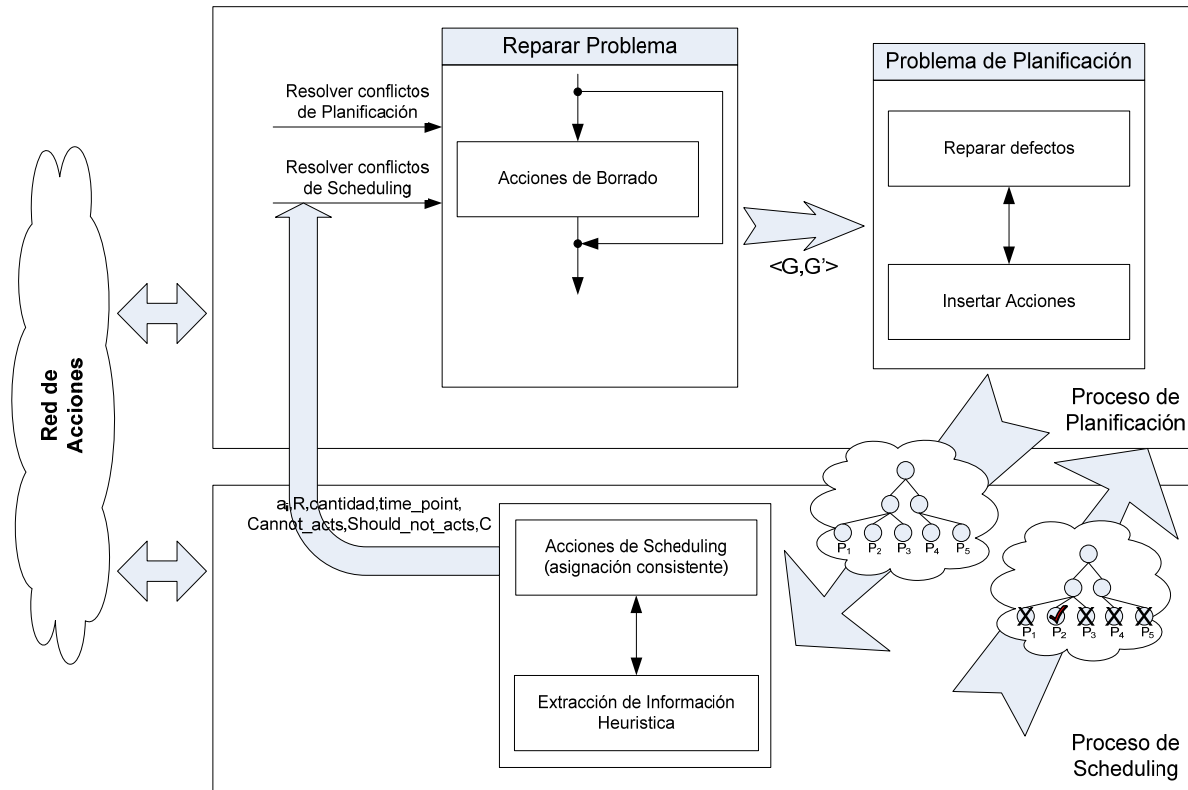


Figura 3 Subtareas derivadas de los pasos 8 y 14 del Algoritmo 1.

Su descripción es la siguiente [4],

1. *solve_PLN_conflicts* <rev_PLN> cuando algunos conflictos de planificación son detectados en el conjunto de acciones revisadas por el planificador *rev_PLN*, éste se invoca a sí mismo para que solucione el problema. Esta tarea requiere de menos información heurística. La idea global es resolver “un problema de planificación desde una perspectiva de planificación”.
2. *solve_SCH_conflicts* <acción en conflicto, recurso, cantidad, nodo de la acción del conflicto, acciones imposibles de borrar, acciones que no se deberían borrar, restricciones sobre tiempo y sobre recurso> Cuando el scheduler detecta una violación en alguna de las restricciones, entonces extraerá los parámetros del conflicto aquí señalados, siendo ésta la tarea más difícil pues implica resolver “un conflicto de scheduling desde una perspectiva de planificación”. Esta información entrará al módulo de planificación a través de la llamada *solve_SCH_conflicts*. Para guiar la búsqueda lo mejor posible, el scheduler puede proporcionar un conjunto adicional de restricciones que el planificador debe conocer para resolver el problema, por ejemplo: “no tomes más de 'n' unidades de recurso Rj para resolver este problema”.
3. Cuando el planificador recibe cualquiera de las dos llamadas arriba mencionadas, analiza la información que le ha sido transferida y lo transforma en sus propios términos a un problema de reparación (Figura 4). La idea es formular un problema de reparación con las submetas que serán obtenidas y con las que serán mantenidas. El planificador usa esta información y sus propias estrategias para decidir si es conveniente borrar algunas acciones o insertar nuevas acciones.
4. Las sugerencias provistas por el scheduler se enfocan al proceso de búsqueda de planificación. Esta etapa

corresponde a la Planificación Clásica, donde el planificador generará distintos planes, resultantes de los diferentes puntos de decisión en el árbol de búsqueda. El scheduler ayudará a determinar el mejor plan de las alternativas mostradas por el planificador, en cuanto al tiempo y recursos.

De acuerdo con el Algoritmo 1 y con la Figura 3, la dinámica del modelo integrado es la siguiente:

1. Proceso de Planificación

Inicia el planificador tomando una acción del AN (paso 5), después la revisa (paso 7) si está libre de conflictos de planificación (es decir, que todas sus condiciones estén soportadas o que no represente una amenaza con otras acciones ya planificadas) y la inserta (paso 8) en *rev_PLN*. Si éste conjunto está libre de conflictos, entonces el planificador lo entrega (paso 9) al scheduler para que éste proceda a su verificación de consistencia de tiempo y recursos. Si al añadir una nueva acción al *rev_PLN* en éste aparecen conflictos, entonces el planificador se llama a sí mismo mediante *solve_PLN_conflicts*, para que repare este conflicto en el *rev_PLN* actual (paso 8). Entonces el planificador tratará de arreglar el conflicto: si es por una amenaza, entonces lo podrá resolver mediante una nueva ordenación de las acciones involucradas. Si se trata de condiciones no soportadas de alguna acción, entonces tendrá que añadir nuevas acciones, generando de esta manera un subplan adicional que logre reparar el conflicto en cuestión. Si no tuviera otra alternativa, eliminará de *rev_PLN* la acción que causa el conflicto. En cualquier caso, el proceso de planificación no terminará hasta que haya arreglado todos los conflictos de planificación. Por tanto, su salida es *rev_PLN* libre de conflictos de planificación.

2. Proceso de Scheduling

La entrada al scheduler es *rev_PLN* (paso 10) con sus restricciones temporales entre acciones, tiempo límite para la ejecución de una acción y para la obtención de (sub)objetivos, persistencia de efectos, etc. Obviamente la tarea del scheduler es comprobar la consistencia de *rev_PLN*. Si se comprobara toda la red, se obtendría un resultado positivo o negativo de consistencia y no se tendría información acerca de cuál acción es la que hace que aparezca un conflicto y en qué cantidad de recurso. Por lo tanto, se decidió ir comprobando la consistencia de las acciones del *rev_PLN* de una en una (paso 11). Si la acción es consistente, entonces se adiciona a *rev_SCH* (paso 12) y se continúa con el proceso de revisión. Si la acción es inconsistente (paso 13), entonces el scheduler llama al planificador mediante *solve_SCH_conflicts* para que repare este conflicto de scheduling en el *rev_PLN* actual. Esta llamada es similar a *solve_PLN_conflicts* pero ahora con más información debido a que el problema a resolver es más complejo, y es utilizada para arreglar un problema de scheduling mediante el módulo de reparación que pertenece al proceso de planificación. En dicho módulo el planificador construye, a partir del borrado de la acción del conflicto y/o la inserción de la acción productora/liberadora del recurso, todos los posibles planes que obtienen el mismo objetivo establecido. Posteriormente, el scheduler ayudará a seleccionar al *rev_SCH* libre de conflictos de ese árbol de planes que le envía el planificador, es decir, al mejor plan alternativo que resulte consistente.

3.4 Beneficios de la arquitectura propuesta

Este modelo de integración ayudará significativamente a resolver problemas muy similares con la realidad, es decir, con una especial complejidad al contener restricciones numéricas. Naturalmente, las capacidades del modelo integrado dependen de la implementación de los módulos individuales de estos procesos. Hemos identificado algunas ventajas generales del modelo:

- **Flexibilidad.** En un esquema fuertemente acoplado, ambos procesos se comunican y colaboran a cada paso durante la construcción del plan parcial.
- **Modularidad.** Como ambos procesos permanecen en módulos separados, cada uno conserva sus capacidades al máximo y puede aplicarlas en el momento que lo requiera o le sean requeridas.
- **Generalidad.** La arquitectura propuesta contempla el uso de este modelo para dominios y problemas genéricos, pues aplica técnicas ya existentes en el Estado del Arte para cada proceso que pueden utilizar cualquier planificador y cualquier *scheduler*.
- **Detección temprana/oportuna de conflictos.** En un esquema de colaboración mutua, ambos procesos van construyendo y verificando al plan paso a paso, es decir, acción por acción, etiquetando a las acciones colocadas una vez revisadas por cada módulo, por lo que oportunamente se percatarán de la presencia de

un conflicto y de qué tipo es éste, ya sea por condiciones no soportadas o por acciones/proposiciones mutuamente exclusivas (ambos conflictos de Planificación) o por insuficiencia de recursos (conflicto de Scheduling).

- **Resolución inmediata de conflictos.** En cuanto el sistema detecta un conflicto, procede inmediatamente a su resolución aplicando técnicas propias de cada área, por ejemplo: la replanificación mediante la rutina de Reparación de Fallos [30] y la búsqueda de consistencia del Problema de Satisfacción de Restricciones [25].
- **Información operativa disponible entre los módulos.** Este modelo presenta dos puntos claves de intercambio de información necesaria para que operen ambos módulos, no pudiendo actuar uno de ellos sin la información del otro.
- **Compartición de información heurística.** El scheduler deberá extraer los parámetros heurísticos de planificación para poder guiar al planificador durante la obtención de planes alternativos.
- **Compartición de un criterio de optimización.** Estos módulos pueden intercambiar información acerca del criterio de optimización que cada uno debe aplicar.
- **Formalización de la interfaz.** Mediante una interfaz compuesta por un traductor del lenguaje de definición de dominio en planificación y el vocabulario e identificadores utilizado por cualquier scheduler, ambos módulos se pueden entender y dar seguimiento a las tareas solicitadas.
- **Capacidad de razonar ante restricciones duras y suaves.** De acuerdo con los requerimientos de la última versión del lenguaje formal de planificación PDDL3.0 [11], este modelo contempla proveer razonamiento sobre restricciones duras (exigencias) y suaves (preferencias).
- **Simplicidad de la arquitectura.** Esta es una gran ventaja, pues permite la inserción de módulos adicionales que sean requeridos para la obtención de buenos planes.

IV. Resultados experimentales: Ejemplo de aplicación.

Para ayudar a comprender el funcionamiento de este modelo integrado, se utilizó un problema con restricciones sobre tiempo y recursos en el dominio *Rovers* ya indicado previamente, el cual requiere que cada explorador *rover0* navegue hasta la superficie de un planeta para que ahí recoja y analice muestras de roca y de polvo. De igual manera, requiere que obtenga imágenes de objetivos ya establecidos. Toda esta información la debe comunicar a su base de control. El explorador consume energía en algunas de sus actividades y puede recargarla en los sitios que reciben luz solar. Se seleccionó un problema que impone las siguientes restricciones:

- el explorador tiene acciones con diferente duración y consumo de energía,
- su nivel de energía inicial: 10 unidades (insuficientes para toda la misión),
- que el Sol solamente esté disponible para recargar batería en dos puntos: *w1* y *w3*.

Con el afán de darle más similitud con la realidad, se le adicionaron las siguientes restricciones complejas:

- que la submeta (*have_image objective1*) no pueda ser obtenida antes de 20',
- que el efecto (*calibrated camera0*) sólo persista durante 30',
- que la meta (*communicated_image*) deba ser obtenida antes del tiempo límite de 40'.

Siguiendo al Algoritmo 1, las acciones son seleccionadas del AN una a una, y revisadas por el planificador así como por el *scheduler*. Mientras ningún conflicto aparece, los conjuntos *rev_PLN* y *rev_SCH* son actualizados. Este proceso continúa hasta que, revisando en el nodo de inicio de la acción *communicate_image (w0 w1)*, se produce un conflicto por inconsistencia debido al sobreconsumo de energía, pues esta acción requiere de 6 unidades de energía y sólo están disponibles 4 unidades (resultante de $RCN = \text{energía inicial} - \text{consumo} = 10 - 3 - 2 - 1$), presentándose entonces un déficit de 2 unidades de energía. Al detectar el conflicto, el scheduler ejecuta la subtarea *solve_SCH_conflicts* cuyos parámetros son <acción en conflicto: *comm_image (w0 w1)*, recurso: energía del *rover0*, cantidad disponible de energía: 2 unidades, nodo del conflicto: *comm_image.on*, acciones imposibles de ser borradas: todas, pues son obligatorias, acciones susceptibles de ser borradas: ninguna, restricción del recurso: energía inicial del *rover0 = 10*>. Es notorio que el conjunto *cannot_acts* contiene a todas las acciones del AN (pues ellas son la única manera de

obtener las submetas), mientras que el conjunto *Should_not_acts* está vacío en consecuencia. Como la reparación del plan exige mantener las mismas submetas, el planificador tratará de reparar este conflicto insertando las siguientes acciones: *recharge (w1)* y *recharge (w3)*, así que creará planes alternativos que puedan ser insertados en diferentes puntos del tiempo, como se puede apreciar en la Figura 4, donde P1 es [*navigate (w0 w1)*, *recharge (rover w1)*, *navigate (w1 w0)*] y P1' es idéntico a P1, sólo que el punto destino es w3 para recargar ahí la batería.

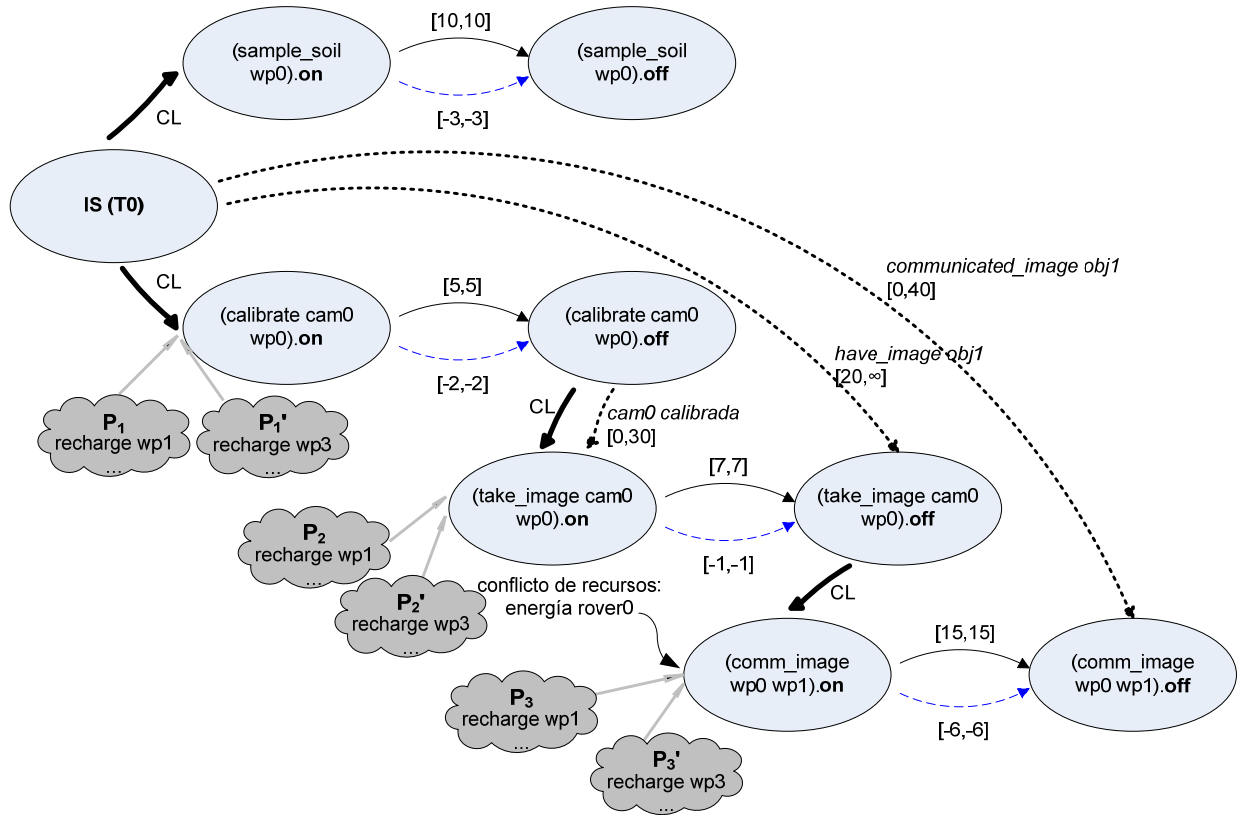


Figura 4 Representación del AN con todos los posibles planes generados durante la reparación.

Como resulta un número considerable de planes alternativos, entonces el planificador se los comunicará al scheduler, que los verificará e informará al planificador cuál de ellos es consistente. Si hubiera dos ó más planes consistentes, el planificador escogería a uno de acuerdo con la métrica de optimización del plan. Una vez resuelto el conflicto, el algoritmo seguirá estudiando las acciones del AN hasta que ninguna acción de la misma necesite ser revisada. El conjunto de acciones contenidas en el *rev_SCH* es el plan final, libre de conflictos.

V. Discusión de Resultados.

En el estado del arte se encontraron dos perspectivas de combinación que han tratado de abordar problemas muy similares con la realidad. La primera es una aproximación de Planificación extendida para que maneje tiempo y recursos (también conocida como planificación temporal) [10], [28], [21]. La segunda es una aproximación de Scheduling extendido para que ayude a obtener una selección dinámica de acciones (planificación con capacidades embebidas en Scheduling) [5]. En la Planificación extendida, ambos procesos se mezclan de tal manera que no es posible distinguirlos, donde las decisiones de colocación de acciones y de recursos son consideradas mientras se planifica en la línea del tiempo. Una gran desventaja que ha presentado

este enfoque es que la mezcla de estos procesos ha incrementado mucho la complejidad computacional, haciendo prácticamente imposible la resolución de los problemas con fuerte componente numérico, no pudiendo enfrentar la escalabilidad de los problemas [4]. En el Scheduling extendido, la planificación es una subtarea de scheduling, requiriendo como entrada un plan proveniente de un conjunto de plantillas de planes rígidos, además de la definición del problema. La principal desventaja de este enfoque es que, cuando dichas actividades tienen mucha dependencia con las características numéricas de scheduling (lo cual es bastante común en la realidad), se puede presentar conflicto por insuficiencia de recursos y este enfoque no tiene la capacidad de replanificar, es decir, de modificar al plan de entrada. Por el contrario, este modelo integrado trata de ser general al utilizar técnicas muy estudiadas en cada proceso, aplicadas a cualquier dominio. De igual forma es flexible pues intercala a ambos procesos, teniendo ambos un rol similar dentro de su dinámica, donde cada componente puede aplicar al máximo sus propias capacidades [4].

VI. Conclusiones y trabajos futuros

Durante la resolución de un problema del mundo real, el extender el proceso de Planificación o de Scheduling no ha resultado buena alternativa, según se ha visto, pues ha presentado mucha complejidad computacional y no ha podido replanificar cuando se ha presentado conflicto por insuficiencia de recursos. De ahí que fue necesario proponer un modelo integrado que tuviera un enfoque semidistribuido (fuertemente acoplado) haciéndolo entonces flexible [4]. La propuesta de integración que aquí se estudia logra una comunicación total entre ellos, en oposición con las aproximaciones existentes en que están completamente aislados. En este modelo, cada proceso beneficia al otro desde su conocimiento por separado [27], tales como la obtención de planes *Strips*, la ejecución del plan y su reparación ante la presencia de conflictos, la colocación de acciones en el tiempo, el manejo de restricciones temporales y de recursos, el problema de la satisfacción de restricciones, etc. Obviamente el planificador juega el rol central en este trabajo colaborativo, guiado inteligentemente por la información heurística del scheduler [4]. La arquitectura y funcionalidad de este modelo resultaron factibles en las trazas realizadas.

Una vez establecida la estructura y el comportamiento funcional de este modelo de integración, las tareas pendientes para lograr darle las capacidades suficientes para modelar problemas del mundo real y para hacerlo funcionar en forma automatizada son las siguientes: i) incorporación del convertidor a AN en el modelo integrado, ii) implementación de un algoritmo de búsqueda de satisfacción de restricciones sobre el AN. Por otro lado, de acuerdo con las nuevas capacidades aportadas por el lenguaje formal de Planificación versión PDDL3.0, se le deberá proveer de razonamiento extendido sobre restricciones suaves o preferencias.

VII. Referencias

- [1] Long, D., Fox, M., Progress in AI Planning Research and Applications, The European Online Magazine for the IT Professional, 3(5), (2002), pages 10-24.
- [2] Smith, D., Frank, J., Jónsson, A., Bridging the gap between planning and scheduling, Knowledge Engineering Review, 15 (1), (2000), pages 47-83.
- [3] Weld, D., Recent advances in AI Planning, AI, Magazine, 20(2), (1999), pages 93-123.
- [4] Garrido, A., Garcia-Hernández, M. de G., Onaindía, E., Towards an efficient Integration of Planning and Scheduling, Proc. of the 24th Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG 2005), pages 58-65, London, England, December (2005).
- [5] Smith, S., Zimmerman, T., Planning tactics within scheduling problems, Proc. ICAPS 2004, Workshop on Integrating Planning and Scheduling, (2004), pages 83-90.
- [6] Bartok, R., "Integrating planning into production scheduling: a formal view", in Proc. of ICAPS-2004, Workshop on Integrating Planning into Scheduling, (2004), pages 1-8.
- [7] Chien, S., Rabideau, G., Knight, R., Sherwood, R., Engelhardt, B., Mutz, D., Estlin, T., Smith, B., Fisher, F., Barrett, T., Stebbins, G., Tran, D., ASPEN-automating space mission operations using automated planning and scheduling, Proc. SpaceOps2000 (2000).
- [8] Muscettola N., HSTS: Integrating planning and scheduling, Zweben & Fox Editor, Intelligent Scheduling, pages 169-212, Morgan Kaufmann, San Mateo, CA, USA (1994).
- [9] Wallace, R.J., Freuder, E.C., "Supporting dispatchability in schedules with consumable resources", Journal of Scheduling, 8(1), (2005) pages 7-23.

- [10] Ghallab, M., Nau, D., Traverso, P., Automated Planning: Theory and Practice. Morgan Kaufmann (2004).
- [11] Gerevini, A., Long, D., "Plan constraints and preferences in PDDL3", Technical Report, University of Brescia, Italy (2005).
- [12] Halsey, K., The workings of CRIKEY-a temporal planner. Proc. Int. Conf. on Automated Planning and Scheduling, ICAPS 2004, International Planning Competition, (2004), pages 35-37.
- [13] Gervasio, M., De Jong, D., Morgan Kaufmann, "A Completable Approach to Integrating Planning and Scheduling", Proc. of the 1st International Conference (AIPS-92), Artificial Intelligence Planning Systems, (1992) pages 275-276.
- [14] Currie, K., Tate, A., O-Plan: the Open Planning Architecture Artificial Intelligence Journal, 52(1), (1991), pages 49-86.
- [15] Beck, H., TOSCA: A novel approach to the management of job-shop Scheduling Constraints, Proc.9th CIM-Europe Annual Conference, Realizing CIM's Industrial Potential, (1993), pages 138-149.
- [16] Ghallab M., Laruelle H., Representation and control in IxTeT, a temporal planner, Proc. 2th International Conference on AI Planning Systems (AIPS-94), (1994), pages 61-67.
- [17] Gerevini, A. and Serina, I., LPG: a planner based on local search for planning graphs. AIPS-2002, (2002), pages 281-290.
- [18] Gerevini, A., Saetti, A. and Serina, I., Planning Through Stochastic Local Search and Temporal Action Graphs in LPG, Journal of Artificial Intelligence Research, 20, (2003), pages 239-290.
- [19] Gerevini, A., Saetti, and Serina, I., Planning with numerical expressions in LPG, ECAI-2004, (2004), pages 667-671.
- [20] Garrido, A. and Long, D., Planning with Numeric Variables in Multiobjective Planning, Proc. European Conference on AI (ECAI-2004), (2004), pages 662-666.
- [21] Edelkamp, S., Symbolic pattern databases in heuristic search planning. AIPS-2002, (2002), pages 195-204.
- [22] Fox, M., Long, D., PDDL2.1: an extension to PDDL for expressing temporal planning domains, Journal of Artificial Intelligence Research, Special Issue on the 3rd International Planning Competition (2003).
- [23] Bonet, B., Geffner, H., Planning as heuristic search, Artificial Intelligence, 129, (2001), pages 5-33.
- [24] Hoffmann, J., Nebel, B., The FF planning system: Fast plan generation through heuristic search, Journal of Artificial Intelligence Research, 14, (2001), pages 253-302.
- [25] Dechter, R., Meiri, I. and Pearl, J., Temporal Constraint Networks, Artificial Intelligence, 49, (1991), pages 61-95.
- [26] Tsang, E., Foundations of constraint satisfaction, Academic Press, 1993.
- [27] Pecora, F., Cesta, A., "Evaluating plans through restrictiveness and resource strength", Proc. Workshop on Integrating Planning into Scheduling (AAAI) (2005).
- [28] Chen, Y., Hsu, C.W. and Wah, B.W., SGPlan: Subgoal Partitioning and Resolution in Planning. Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS-2004) - International Planning Competition, (2004), pages 30-32.
- [29] Do, M.B., Kambhampati, S., Planning as constraint satisfaction: Solving the planning graph by compiling it into CSP, Artificial Intelligence, 132, (2001), pages 151-182.
- [30] Yang, Q., "Intelligent Planning: A decomposition and Abstraction Based Approach", Springer Ed., Germany (1997).

VII. Extracto Curricular de los Autores

Ma. de Guadalupe García-Hernández hizo sus estudios de Doctorado en Reconocimiento de Formas e Inteligencia Artificial por la Universidad Politécnica de Valencia, España. Obtuvo la Maestría en Ingeniería Mecánica por la Universidad de Guanajuato en 1992. Se graduó de la Licenciatura en Ingeniería Química por la Universidad de Guanajuato en 1985. Actualmente es Profesora de Tiempo Completo del Cuerpo Académico de Telemática en la División Ingenierías, Campus Irapuato Salamanca de la Universidad de Guanajuato. Ha presentado 18 publicaciones en eventos científicos internacionales y 12 publicaciones en eventos científicos nacionales. Tiene 7 colaboraciones en revistas indiciadas, así como 4 colaboraciones en revistas nacionales. Su área de interés es Planificación en Inteligencia Artificial.

Gerardo Enrique Canedo Romero hizo sus estudios de Doctorado en Reconocimiento de Formas e Inteligencia Artificial por la Universidad Politécnica de Valencia, España. Obtuvo la Maestría en Ingeniería Eléctrica por la Universidad de Guanajuato en 1981. Se graduó de la Licenciatura en Ingeniería en Comunicaciones y Electrónica por la Universidad de Guanajuato en 1979. Actualmente es Profesor de Tiempo Completo del Cuerpo Académico de Telemática en la División Ingenierías, Campus Irapuato Salamanca de la Universidad de Guanajuato. Su área de interés es la Inteligencia Artificial.

Donato Hernández Fusilier hizo sus estudios de Doctorado en Reconocimiento de Formas e Inteligencia Artificial por la Universidad Politécnica de Valencia, España. Obtuvo la Maestría en Ingeniería Eléctrica por la Universidad de Guanajuato. Se graduó de la Licenciatura en Ingeniería en Comunicaciones y Electrónica por la Universidad de Guanajuato en 1985. Actualmente es Profesor de Tiempo Completo del Cuerpo Académico de Telemática en la División Ingenierías, Campus Irapuato Salamanca de la Universidad de Guanajuato. Su área de interés es la Inteligencia Artificial.

Herramienta computacional para elaborar software educativo

*Saturnino Job Morales Escobar*¹, *Dora María Martínez Magaña*², *Ivonne Rodríguez Pérez*³, *Sara Lilia García Pérez*⁴

^{1,2,3,4}Centro Universitario Valle de México de la UAEMéx

Blvd. Universitario s/n Predio San Javier, Atizapán de Zaragoza, Estado de México, México
e-mail: sjobmorales@yahoo.com.mx¹, doramartinezm@gmail.com², irodriguezperez@yahoo.com³,
saraliliag@yahoo.com.mx⁴

Resumen: Se desarrolló un prototipo que asiste a los profesores en el diseño y elaboración de software educativo, de esta manera se atiende la necesidad de enfatizar los procesos de aprendizaje y abstracción que faciliten la adquisición de conocimiento por parte de los estudiantes. La herramienta permite elaborar material educativo con una estructura y contenidos que tomen como base resultados de la investigación experimental sobre las teorías psicológicas de la educación, donde se considera que el conocimiento se puede lograr utilizando estrategias de enseñanza y aprendizaje enfatizados en el alumno. De esta manera el sistema, asiste al docente durante todo el proceso, ofreciendo la posibilidad de incorporar documentos de diversos tipos de acuerdo a las necesidades de cada área de conocimiento.

Palabras clave: Software educativo, aprendizaje, teorías psicológicas de la educación, proceso enseñanza aprendizaje, tutorial.

Abstract: A prototype was developed in order to assist teachers in the design and building of educational software, this way the need to emphasize the learning and abstraction processes in order to ease the knowledge acquisition among the students is attended. This tool allows us to create educational material with structure and contents that have the experimental research about psychological educational theories as a base, where it is considered that knowledge can be acquired using teaching and learning strategies that are focused on the student. This way the system assists the teacher during all the process, offering the possibility to incorporate several types of documents, according to the needs of each area of knowledge.

Keywords: Educational software, learning, educational psychological theories, learning teaching process, tutorial.

Introducción

En esta era de la informática las exigencias pedagógicas obligan a buscar nuevos métodos encaminados a potenciar la creatividad, a impulsar el desarrollo de la capacidad de pensar y de aprender a aprender, en contraposición a la pedagogía tradicional, caracterizada por el enciclopedismo, la memorización, el bajo nivel de crítica, clases expositivas y el autoritarismo en la enseñanza. Las necesidades pedagógicas insisten en el uso de recursos o medios audiovisuales que sirvan como vía para la estimulación sensorial. Sin embargo, los medios y recursos audiovisuales están condicionados al entorno social y a las condiciones culturales en que se realiza el proceso de aprendizaje. En este sentido se continúa la búsqueda de propuestas de solución a la tendencia predominante de la pedagogía actual de considerar como eje central del proceso educativo al aprendizaje, entendido como un proceso permanente, continuo y generalizado a todos los niveles y modalidades de la vida social.

Una herramienta que ha resultado importante en este proceso educativo es la computadora, pues desde sus inicios se ha utilizado en el proceso de enseñanza, recibiendo el nombre de enseñanza asistida por computadora, entendida como la aplicación de la tecnología informática para proporcionar enseñanza y como una solución de la tecnología informática al proceso de instrucción individualizada. La aplicación de la computadora en este campo se inicia en los años 50's. Patrick Suppes con otros investigadores fueron de los primeros en usar la computadora en la educación, al crear Computer Curriculum Corporation (CCC) con la intención de producir material educativo a través de lecciones tutoriales para el aprendizaje de la matemática y de idiomas.

Otro de los que siguieron la línea de elaboración de lecciones tutoriales asistidas por computadora fue Robert B. Davis, matemático y profesor quien tenía a su cargo el programa de formación de profesores de matemáticas de escuelas primarias en los 60's, Davis opinaba diferente que Suppes, ya que para él, el mecanismo de aprendizaje se da en mayor medida por medio del proceso de descubrimiento, que como resultado del esfuerzo. Con la aportación de Davis y su equipo, se iniciaron nuevas investigaciones en la educación por computadora.

Robert Davis y David Kibbey, citados en [1], también se consideran pioneros en este campo, pues con el proyecto PLATO marcaron la pauta al presentar programas de muy diversa índole. En su trabajo, Davis propone un conjunto de juegos para el aprendizaje de números, operaciones, relaciones, etc. dando origen a una perspectiva de aplicación en donde algunos autores [2, 3, 4], le han dedicado un apartado especial al proceso de clasificación. Los juegos se clasificaron en la siguiente forma: de contenido (content games), donde el propósito es la enseñanza de algún hecho particular de una materia, y los de proceso (process games), que pretenden la enseñanza de un cognitivo mayor, es decir, de estrategias para la resolución de problemas aplicados en varias materias.

Por otro lado, se considera que los modelos educativos son una alternativa a los retos que plantea la educación superior en nuestros días, como son la atención a la demanda, la respuesta a las exigencias sociales de los programas de estudio relacionados con la globalización y la dinámica de diversificación de perfiles profesionales, problemas cuya solución se traduce en tendencias tales como programas académicos transcurriculares y flexibles y la educación personalizada. Estas tendencias convergen y se fortalecen con la educación virtual y la educación a distancia.

En este contexto, una de las actividades más comunes e importantes en el proceso de enseñanza-aprendizaje enfatizada en el aprendizaje, es la planeación y desarrollo de los materiales para impartir un curso, el reto es mayor si estos materiales se realizan con la intención de ser autocontenidos, que favorezcan la participación proactiva del estudiante y que el proceso enseñanza aprendizaje se realice con los medios tecnológicos actuales y en particular utilizando la computadora. Esto último es lo que caracteriza lo que se ha llamado software educativo, y su desarrollo se sustenta en la suposición de que las tecnologías de información ofrecen grandes posibilidades y pueden facilitar el aprendizaje de conceptos, ayudar a resolver problemas y contribuir a desarrollar las habilidades cognitivas.

Sin embargo, hasta hoy, una gran parte del software educativo desarrollado corresponde al tipo denominado tutorial inteligente, que intenta reproducir la forma de enseñanza basada en el diálogo con un tutor. En estos diálogos basados en preguntas, el tutor provoca la reflexión en el alumno, haciendo que construya por sí mismo las respuestas correctas y, en definitiva, que aprenda los conceptos objeto de estudio. Aquí, la actividad del alumno está controlada por la computadora y la estrategia pedagógica utilizada es del tipo "ejercicios" o tutorial. Las bases de un buen programa tutorial son el diagnóstico y la corrección de errores en las fases de respuesta y reenseñanza, lo que impide que los errores de aprendizaje se acumulen.

Así, los sistemas tutoriales inteligentes, recrean un ambiente de aprendizaje y la interacción del sistema con el estudiante tiene un cierto comportamiento inteligente. Estos sistemas se guían a partir de dos premisas: La sustitución total del maestro, lo cual obliga a tener un modelo de enseñanza, y la creación en la computadora de un modelo de estudiante, que implica el desarrollo de una teoría del conocimiento.

Otra alternativa de software educativo se basa en simulaciones, que corresponde al concepto de entornos libres, en los que el control del proceso es detentado por el alumno y no por la máquina, como ocurre en los tutoriales. En este caso se dice que se aplican conceptos del constructivismo que corresponden al estilo de aprendizaje guiado por el alumno inmerso en un entorno de enseñanza.

Actualmente continúa la búsqueda de herramientas, técnicas y estrategias para la aplicación de las tecnologías de información en el proceso de enseñanza aprendizaje, enfatizado en el aprendizaje.

Con base en estas necesidades, el diseño e implantación de ambientes informáticos de aprendizaje tienen como premisas fundamentales: La eficiencia, rapidez y exactitud en la interacción hombre-máquina que parten de la consideración de las características individuales del que aprende, pretendiendo desarrollar al alumno cognoscitivamente y promover su autonomía personal. Un ejemplo de ello es el uso de Software educativo, con características de: Tutorial inteligente, multimedia, hipertexto, adaptativo y con la integración de animaciones, imágenes de video y sonidos.

La educación utilizando software educativo, se considera como un proceso de enseñanza aprendizaje que tiene como misión transmitir al individuo, conocimientos, habilidades, actitudes, destrezas y valores, para su aplicación en la sociedad, y en la búsqueda de perfeccionar los métodos educativos, con la finalidad de

transmitir la misma cantidad de conocimientos en menos tiempo, sin sacrificar la profundidad y la calidad de la enseñanza.

Así, el desarrollo de la Inteligencia Artificial, la Pedagogía, la Psicología Cognitiva y la Tecnología Multimedia, han hecho posible la elaboración de software educativo que permite la enseñanza interactiva, motivante e individualizada y además adaptable a las necesidades y ritmo de aprendizaje del estudiante [5].

Sin embargo, el software educativo que hasta el momento se ha producido con las características ya mencionadas no garantiza el aprendizaje significativo de los alumnos, sino que, en su mayoría siguen el método de educación conductista de enseñanza aprendizaje. Por tanto, se considera muy importante que el software educativo actual integre características que permitan un aprendizaje significativo en el estudiante. Esto se puede lograr, fundamentalmente y de manera particular en la forma de cómo se seleccionan y se presentan los contenidos a los alumnos a través del software.

Hacer bien este cometido requiere un gran esfuerzo individual y un trabajo importante de planificación y coordinación de un equipo de profesores, pero se tiene la oportunidad de cubrir esa necesidad y crear una enseñanza de forma que teoría, abstracción, diseño y experimentación estén integrados.

Como se puede apreciar las posibilidades de desarrollo son muy amplias. Con este trabajo en el Centro Universitario Valle de México, perteneciente a la Universidad Autónoma del Estado de México (CUVM-UAEM), se inicia la producción de materiales didácticos a través de software educativo que de respuesta a las necesidades propias de nuestro modelo educativo basado en el desarrollo de competencias y que en un momento dado se puede extender para atender a nuevos requerimientos.

Se partió de la idea que la elaboración de materiales educativos realizados por los profesores, se puede transformar en un proceso de construcción de software educativo que puede ser asistido por un sistema computacional que facilite, por una parte la estructuración adecuada de los materiales a ser presentados y por otra facilitar la adquisición del conocimiento por parte de los estudiantes.

Se desarrolló la primera versión de un sistema que apoya a los profesores en la elaboración de cursos completos autocontenidos para ser presentados en un equipo de cómputo, y a la vez permite al estudiante personalizar su aprendizaje y facilitar la adquisición del conocimiento utilizando ese material.

En este documento se presenta un modelo de aprendizaje que considera las aportaciones de las teorías psicológicas de la educación, los requerimientos mínimos que debe cubrir un sistema para la elaboración de software educativo y la propuesta de un sistema para realizarlo.

Aportaciones de las teorías psicológicas de la educación a la elaboración de software educativo

Las teorías cognitivas, constructivistas y conductistas utilizan estrategias que pueden dividirse en dos grupos: De instrucción o de enseñanza y de aprendizaje, que han sustituido y perfeccionado las técnicas y hábitos de estudio. Las estrategias seleccionadas han demostrado, en diversas investigaciones [6] su efectividad al ser introducidas como apoyo en la elaboración de materiales como: Textos académicos y materiales educativos, así como en la dinámica de la enseñanza.

Las estrategias de instrucción o de enseñanza son utilizadas para diseñar entornos de enseñanza, y en la planificación se deben ajustar a los contenidos de enseñanza, a la organización intelectual y a los esquemas mentales de los estudiantes para mejorar y facilitar el proceso enseñanza-aprendizaje.

En la tabla 1, se presenta en forma sintetizada, una conceptualización de dichas estrategias de enseñanza basadas en las investigaciones de West, Mayer, Farmer y Wolf citadas en [6].

Tabla 1: Estrategias de enseñanza

Estrategia	Definición
Objetivos o propósitos	Enunciado que establece condiciones, tipo de actividad y forma de evaluación del aprendizaje
Resumen	Síntesis y abstracción de la información más relevante de un discurso oral o escrito. Hace énfasis en las palabras claves principios, términos y argumentos centrales.
Organizador previo	Información de tipo introductoria y contextual. Es elaborado con un nivel superior de abstracción, generalidad e inclusividad que la información que se aprenderá. Tiende un puente cognitivo entre la información nueva y la previa.
Ilustraciones	Representación visual de los conceptos, objetos o situaciones de una teoría o tema específico (fotografías, dibujos, esquemas, gráficas, dramatizaciones, etcétera).
Analogías	Proposición que indica que una cosa o evento (concreto y familiar) es semejante a otro (desconocido y abstracto o complejo).

Preguntas intercaladas	Preguntas insertadas en la situación de enseñanza o en un texto. Mantienen la atención y favorecen la práctica, la retención y la obtención de información relevante.
Pistas topográficas y discursivas	Señalamientos que se hacen en un texto o en la situación de enseñanza para enfatizar y/u organizar elementos relevantes del contenido por aprender.
Mapas conceptuales y redes semánticas	Representación gráfica de esquemas de conocimiento (indican y relacionan conceptos, proposiciones y explicaciones).
Uso de estructuras textuales	Organizaciones y retóricas de un discurso oral o escrito, que influyen en el recuerdo.
Uso de átomos de conocimiento	Un átomo de conocimiento es la cantidad mínima de material que se puede presentar en forma de un texto corto, un esquema, un dibujo, una foto.

Por otro lado, las estrategias de aprendizaje son habilidades, procedimientos, hábitos, técnicas y destrezas utilizadas por el estudiante para facilitar su aprendizaje, permitiéndole una mejor asimilación, comprensión y recuerdo de la información. Además, le permiten aprender con mayor eficacia los contenidos curriculares o extracurriculares que se presentan, y de esta manera se responsabiliza de su proceso de aprendizaje y puede mejorar su rendimiento académico.

Los objetivos particulares de cualquier estrategia de aprendizaje, afectan la forma en cómo se selecciona, adquiere, organiza o integra el nuevo conocimiento, e incluso modifican del estado afectivo o motivacional del estudiante. En la tabla 2 se presentan las estrategias de aprendizaje con base en lo citado en [6].

Tabla 2: Estrategias de aprendizaje

Proceso	Estrategia	Finalidad
Aprendizaje Memorístico	Recirculación de la información	Repaso simple Apoyo al repaso (Seleccionar)
Aprendizaje Significativo	Elaboración	Procesamiento simple Procesamiento complejo
	Organización	Clasificación de la información Jerarquización y organización de la información
Recuerdo	Recuperación	Evocación de la información

De cada una de las teorías de enseñanza-aprendizaje se obtuvieron elementos a considerar en el diseño de software educativo, entre las que destacan los siguientes: De la teoría Neoconductista de B. F. Skinner, las unidades básicas elementales, objetivos de aprendizaje y la enseñanza programada; de la teoría Epistemológica genética de J. Piaget, la organización mediante esquemas y estructuras; del Aprendizaje Significativo de D. Ausubel, los organizadores previos, planteamiento de problemas, situaciones desafiantes, mapas y esquemas conceptuales, uso del pensamiento visual, resúmenes e ideas o palabras claves, formulación de preguntas por parte de los estudiantes, búsqueda de analogías y metáforas, uso de la fantasía y de juegos de simulación, lectura crítica (analizando, valorando) y lectura creativa y técnicas de la estructura del texto; de la teoría de Zona de Desarrollo Potencial de L.S. Vygotsky, el papel del maestro como guía experto; del Procesamiento de la información de R. Gagné, el uso de condiciones internas que intervienen en el proceso y de condiciones externas que pueden favorecer un aprendizaje óptimo; y de la Teoría de la Instrucción de Merrill el desarrollo de modelos prescriptivos para la elaboración de materiales educativos informáticos.

Para incorporar los elementos mencionados en los materiales educativos por medio del sistema, el profesor debe reflejar los siguientes procesos en su preparación: Observación, análisis, ordenación, clasificación, representación, memorización, interpretación y evaluación.

Observación: Implica presentar casos resueltos para visualizar la ruta que se siguió al resolver un problema.

Análisis: En la presentación de contenidos se sugiere la incorporación de materiales de acuerdo a la estructuración definida, los cuales pueden ser: Redes semánticas, mapas conceptuales, presentaciones, texto.

Ordenación: Implica organizar la información por medio del índice de contenidos, para que el estudiante adquiera de una manera eficaz el aprendizaje.

Clasificación: Identificar la información más importante.

Representación: Utilizar diversas estrategias de enseñanza para una visión más completa sobre el desarrollo de cada unidad del material educativo.

Memorización: Es de gran importancia para retener conocimiento con base en esta capacidad, como en el aprendizaje de leyes y reglamentos.

Interpretación: Utilizar diferentes opciones de presentación de los contenidos, para que el alumno pueda razonarlos y explicar con sus propias palabras lo que quiere decir el texto.

Evaluación: Implica preparar ejercicios y prácticas para que el alumno al realizar auto evaluación, esté consciente de lo aprendiendo con el material educativo.

Estos planteamientos permiten considerar que se puede ofrecer tutoría guiada tanto para la generación de materiales como para el estudiante que los utilice. Por consiguiente, la propuesta del sistema para el desarrollo de software educativo está basada en ambos aspectos, integrando las tecnologías de información y las teorías Psicológicas de la educación para enriquecer el aprendizaje del estudiante. Esto da como resultado que al utilizar el software educativo el alumno se responsabilice de su propio proceso de asimilación de conocimiento, y que el maestro obtenga una herramienta de apoyo en el proceso de construcción de material educativo.

En principio, un tutorial puede ser un curso completo de alguna unidad de aprendizaje o curso especial, donde su principal característica debe ser facilitar el aprendizaje y el conocimiento.

Así, en forma genérica un tutorial se puede definir como un pequeño manual que mediante una serie de pasos da a conocer un tema específico o los temas de una unidad de aprendizaje y por lo tanto el punto de partida para su elaboración es tener los propósitos del programa del curso.

El procedimiento para elaborar tutoriales está dirigido a profesores y basado en el trabajo presentado por Jesús Figueroa Nazuno [7].

Procedimiento para realizar tutoriales

Este procedimiento parte del hecho que se tiene el programa de la unidad de aprendizaje que se desea presentar en un tutorial.

PASO 1.- Desarrolle la portada del tutorial, esto es la primera hoja del tutorial deberá contener: Nombre del programa educativo, nombre de la materia, nombre del profesor (es), semestre y fecha de elaboración.

PASO 2.- El programa de la unidad de aprendizaje debe convertirse en un texto en donde los capítulos, secciones, subsecciones, y átomos de conocimiento están numerados en forma progresiva, y de forma jerárquica, por ejemplo:

1.1.1

1.1.1.1

1.1.1.2

1.1.2

El objetivo es que este índice detallado represente la estructura general del tutorial, mirándose de la siguiente forma:

1 Capítulo

1.1 Primer tema del capítulo

1.1.1 Primer subtema

1.1.1.1 Primera sección

1.1.1.1.1 Primera subsección

1.1.1.1.1.1 Átomo de conocimiento

En este momento no es relevante el nombre de cada elemento del capítulo, lo esencial es cuidar que las secciones correspondan a un desglose cuidadoso y detallado del tema.

El primer número, corresponde al capítulo y cada número adicional es un nivel jerárquico más bajo que aumenta el nivel de refinamiento, se recomienda no excederse de cinco o seis niveles. Como resultado se tendrá un índice muy detallado, donde en el nivel de jerarquía más bajo del refinamiento, se tendrá un átomo de conocimiento¹.

Un átomo de conocimiento puede igual ser a una diapositiva o acetato de una presentación y por lo que debe seguir la norma de 7 x 7, con base en las investigaciones de Ausubel [5], que indica que el texto de una diapositiva deberá contener un máximo de siete renglones con siete palabras en cada renglón y que este puede estar acompañado por un dibujo, una foto o esquema.

En algunos casos estos átomos de conocimiento requieren de varias diapositivas o acetatos, llamadas "cadenas de átomos".

¹ Un átomo de conocimiento es la cantidad mínima de material que se puede presentar en forma de un texto corto, un esquema, un dibujo una foto

PASO 3.- Debe recopilar todo el material que tenga al respecto a la Unidad de Aprendizaje que pueden ser: Notas de clase, definiciones importantes, esquemas, dibujos fotos y ejercicios para convertirlos en átomos de conocimiento, utilizando algún procesador de textos o de presentaciones, ya que la parte más laboriosa de un tutorial es convertir todo el material acumulado en unidades básicas o átomos de conocimiento siguiendo cuidadosamente el índice elaborado.

PASO 4.- Elabore la mayor cantidad de redes semánticas naturales pequeñas y detalladas e introdúzcalas como guías en su material para el ordenamiento y búsqueda de relaciones entre diferentes átomos de conocimiento.

PASO 5.- Se recomienda construir gran cantidad de ejercicios, problemas, preguntas, ejemplos, y modelos para que el alumno los resuelva.

PASO 6.- En el índice original, debe subrayar todos los conceptos que son importantes y que tienen una definición técnica en la unidad de aprendizaje que se está trabajando. Estos conceptos tienen que estar subrayados en el texto para que posteriormente se pueda hacer el enlace al glosario.

PASO 7.- Construya la mayor cantidad posible de mapas conceptuales con los conceptos que se subrayaron en el paso anterior y colocarlos antes y después de cada subsección, mientras más mapas, mucho mejor.

PASO 8.- Cuando ya tenga todo el material ordenado e ingresado en el índice, debe revisar y tratar de repetir con otras palabras y en otra forma, la mayor cantidad posible de átomos de conocimiento.

PASO 9.- Debe hacer pequeñas introducciones en forma de texto de las grandes secciones o subsecciones del material e introducirlas en el índice general.

PASO 10.- Basado en el material original utilizado para hacer los átomos de conocimiento, debe hacer una lista por capítulo de las referencias importantes que pueden ser introducidas como recomendaciones de lecturas complementarias en el índice para fortalecer el tutorial.

PASO 11.- De ser posible capture en algún formato electrónico, los materiales recomendados como lectura y almacénelos en un archivo indexado para que puedan ser consultadas dentro del tutorial. Resumiendo se tiene que proporcionar una antología básica con los textos indispensables directamente relacionados con el material del tutorial y una antología complementaria con las lecturas recomendadas pero no indispensables.

PASO 12.- En caso de existir capítulos que sean temáticas opcionales en el tutorial o que se tenga que remitir constantemente al estudiante a que revise secciones anteriores del tutorial, una forma de hacerlo, es elaborando un mapa de dependencias y/o precedencia del material presentado. Estos mapas de precedencia deben ser proporcionados al estudiante para que conozca las relaciones de toda la información presentada.

Requerimientos para el sistema

A partir de lo expuesto en la sección anterior, se especificaron los requisitos que debía cumplir el sistema para el desarrollo de software educativo, los cuales se dividieron en tres grupos: Pedagógicos, funcionales y técnicos. El sistema desarrollado cumple con estos requisitos. En los siguientes párrafos se presenta cada uno de ellos y cómo el sistema los satisface.

Requisitos pedagógicos

Versatilidad: El sistema facilita que el material desarrollado sea portable, es decir, el material educativo podrá ser utilizado por el alumno en los equipos de cómputo que cumplan las especificaciones para instalar el sistema.

Adecuación de usuarios: El contenido y la profundidad del material están bajo la responsabilidad del docente que lo desarrolla y entre las facilidades que ofrece el sistema están, ayuda en línea, que puede consultar como una guía para realizar su tutorial, cómo hacer un índice, introducir contenido, cómo ligarlo con otros archivos, etc. El diseño de las pantallas se basó en la psicología del color y el texto en pantalla es de tamaño adecuado para facilitar su lectura.

Capacidad de motivación: El sistema ofrece una interfaz gráfica fácil de usar por medio de botones con nombres o íconos relativos a la acción que realiza.

Potencialidad de recursos didácticos: El sistema permite la comunicación con el usuario por medio de mensajes de texto que le especifican la acción que va a realizar el sistema y que el usuario puede cancelar o no.

Evaluación: El sistema permite introducir ejercicios, problemas, preguntas, ejemplos y actividades de autoevaluación.

Enfoque aplicativo y creativo: El sistema admite la presentación de entornos aplicativos, por ejemplo, las prácticas que se introducen en un tema, para que el alumno desarrolle la creatividad al aplicar el conocimiento adquirido.

Fomento de la iniciativa y autoaprendizaje: El material desarrollado se podrá entregar al estudiante como apoyo, siendo a la vez una forma de responsabilizarlo de su proceso de aprendizaje, ya que él decidirá que ejercicios y tareas desea realizar y en qué momento.

En la autoevaluación obtendrá una realimentación sobre el aprendizaje obtenido.

Requisitos funcionales

Facilitar el uso e instalación: El sistema es amigable para el usuario, además, es consistente, por lo que en cada pantalla aparece el nombre de la unidad de aprendizaje y la sección a la que pertenece, por ejemplo, si está introduciendo la bibliografía de un tema, este tema es el que aparece en cada pantalla para referenciar el sitio en que está. Asimismo, tiene botones de regresar, anterior, siguiente, salir, guardar, etc., para facilitar la navegación del usuario. Además, ofrece ayuda sobre las dudas que pueden surgir en la elaboración del material.

Versatilidad: Se tiene el manual técnico y código abierto para su modificación. Permite realizar el material en varias sesiones y facilita la impresión del contenido.

Requisitos técnicos

Calidad del entorno visual: El sistema presenta pantallas con texto mínimo, con letra clara y legible de un color suave que no es molesto a la vista al momento de leerlas.

Calidad y cantidad de los elementos multimedia: El sistema ofrece al docente la posibilidad de incorporar diferentes tipos de archivos, sin embargo, los contenidos dependerán del docente que los realiza.

Calidad y estructura de contenido: El sistema estructura los contenidos a partir del índice de manera jerárquica, considerando de izquierda a derecha, primero capítulos, secciones, subsecciones, hasta llegar al átomo de conocimiento que se refiere a un texto corto, un esquema o un dibujo.

El índice proporciona la **estructura de navegación** y el sistema indica en cada momento la ubicación de los usuarios y cómo seguir adelante en el desarrollo o cómo utilizar el material.

Metodología utilizada en el desarrollo del sistema para la elaboración de software educativo

Con base en los requerimientos para el desarrollo del sistema se utilizó la metodología RUP [8], cuyas fases son: Comienzo, Elaboración, Construcción y Transición

Fase de comienzo

Esta fase se orientó hacia la detección de los requerimientos de los usuarios y definir los alcances del sistema. Las actividades realizadas fueron: Estudio de las teorías psicológicas de la educación, identificación de los requerimientos y actores, así como las restricciones principales del sistema y la elaboración de modelos de casos de uso.

Se utilizó el lenguaje de modelado unificado (UML) [9], para definir documentar y construir el modelado, con la ayuda de diagramas, para modelar los diferentes aspectos de las entidades a representar entre los que se encuentran: Diagramas de estructura, de comportamiento y de interacción.

Para modelar el comportamiento se utilizó el diagrama de casos de uso en los cuales se representa la forma en cómo un cliente o usuario (actor) opera con el sistema, y diagramas de actividad, que describen las acciones que realizan los actores con el sistema.

En principio el sistema interactúa con dos actores: Profesor y alumno.

El profesor podrá realizar las siguientes actividades con el sistema: Elaborar portada, introducir el índice, introducir contenido, realizar prácticas y ejercicios de auto evaluación, proponer bibliografía y solicitar ayuda. A cada actividad le corresponde un módulo en el sistema, ver figura 1.

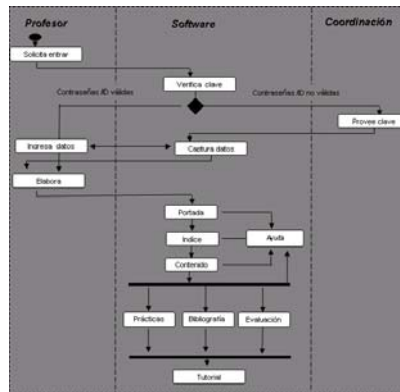


Figura 1: Diagrama de Actividades del profesor

El alumno está limitado para modificar contenidos, pero necesariamente ingresará información al sistema en la realización de prácticas y ejercicios de autoevaluación como se ilustra en la figura 2.

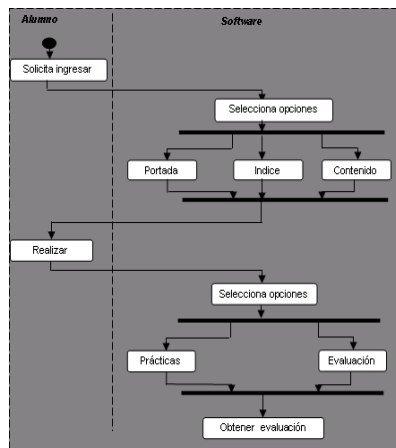


Figura 2: Diagrama de Actividades del alumno

Con base en el procedimiento para la construcción de tutoriales, la secuencia de elaboración del tutorial utilizando el sistema debe darse a partir de la creación de la portada y finalizando con la bibliografía. A continuación se da una breve explicación de lo que realizará cada módulo que conforma el sistema.

Portada: Este módulo permite generar una portada de la unidad de aprendizaje, cuyos ítems son: Nombre del tutorial (unidad), nombre del profesor, periodo, fecha y competencias a desarrollar.

Índice: Es importante que se tome en cuenta los puntos sugeridos en el módulo de ayuda para realizar el índice, ya que permiten definir la estructura de los contenidos que se manejarán en el tutorial.

Contenido: En este módulo el profesor introducirá la información que debe contener el tutorial y que previamente evaluó y analizó.

Prácticas: Como resultado del estudio de las teorías psicológicas de la educación, varios autores, mencionan la importancia de reforzar el aprendizaje realizando prácticas y ejercicios, por este motivo el módulo, proporciona la manera de ingresarlas de acuerdo a las necesidades específicas de cada unidad de aprendizaje, además de ofrecer la incorporación de vínculos con el procesador de texto (Word), la hoja de cálculo (Excel), el programa de presentaciones (Power Point), entre otros.

Evaluación: Es probable que el profesor quiera implementar evaluaciones para tener un indicativo del nivel de aprendizaje logrado, por lo tanto, el sistema permitirá que el profesor formule esta actividad las veces que considere necesarias.

Ayuda: El software maneja un módulo de ayuda en donde se explica paso a paso la elaboración del tutorial y se dan ejemplos para facilitar su uso.

Establecer un diseño para el sistema tutorial

Como se sabe la interfaz de usuario es un elemento muy importante de un sistema o producto de cómputo, si no está bien diseñada, el usuario se verá limitado para aprovechar toda la aplicación y puede llevar al fracaso, en este caso se tienen 2 tipos de usuarios, el profesor quién utilizará el sistema para la elaboración de tutoriales, y el alumno, quién utilizará el tutorial producido por el sistema, por lo tanto es importante señalar que la interfaz elegida funciona para ambos usuarios.

Entre las características importantes de la interfaz con el usuario se pueden considerar: Es una interfaz gráfica con manejo de menú y comandos, incluye ayuda en línea, de fácil navegación, con tiempo de respuesta razonable, en ambiente Windows. La interfaz logra un equilibrio entre estética y funcionalidad. Se decidió manejar los colores institucionales de la UAEM, el nuevo logotipo de la universidad, letra tipo arial con diferentes tamaños en color verde y negro, cumpliendo el objetivo de tener un punto medio entre la funcionalidad y la estética.

Fase de Elaboración

Con base en los requerimientos que se detectaron en la fase de comienzo, se trabajó con los usuarios para afinar los casos de usos y de esa manera describir las actividades que realizarán. Además, se construyeron un prototipo del sistema y una primera versión, el tiempo aproximado para el desarrollo fue de 1 año.

Fase de construcción

En esta fase se desarrollaron los componentes del software de cada caso de uso atendido y refinado en las fases anteriores para hacerlos operativos, esto requirió que el análisis y diseño del cual se partió estuviera bien definido, se implementó el código fuente en el lenguaje Visual Basic 6.0, y se realizaron las pruebas de acuerdo a las estrategias de prueba del software para arquitecturas convencionales [10], que considera recorrer 4 fases de un espiral de adentro hacia fuera: Pruebas de unidad, de integración, de validación y del sistema, en donde se define el grado de abstracción en cada vuelta.

Las pruebas por módulo permitieron detectar errores en cada componente del sistema, es decir, cada módulo se comprobó unitariamente para descubrir errores y depurarlos.

Después de terminadas las pruebas de integración, se aplicaron las **Pruebas de validación**, que permitieron verificar que el objetivo del software se hubiera cumplido. Una vez terminada la prueba de validación, se trabajo con los usuarios finales en este caso fueron los profesores, para realizar la prueba **alfa**.

La primera prueba aplicada fue la alfa, esta prueba se realizó con 7 profesores del Centro Universitario UAEM Valle de México. Se les informó el objetivo de la prueba y se les dio una pequeña introducción del manejo del sistema, de la misma manera se les indicó que el sistema contaba con un módulo de ayuda que los llevaría paso a paso en la realización del material.

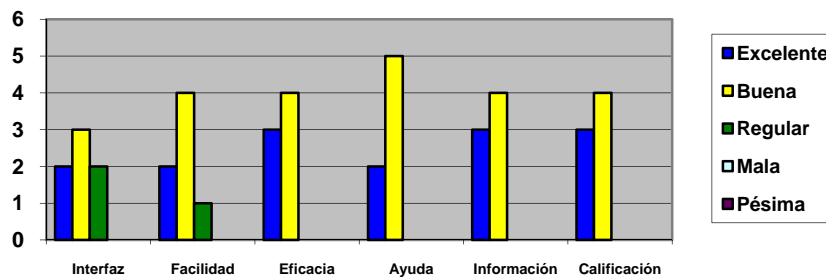
Se aplicó un cuestionario en donde se tomaron en cuenta los siguientes aspectos:

- a) La facilidad con la que se adaptó al sistema (Facilidad)
- b) Cumple con las expectativas y necesidades para el desarrollo de materiales educativos (Eficacia)
- c) El módulo de ayuda del sistema proporciona el servicio para el que fue desarrollado (Ayuda)
- d) La información de introducción para utilizar el sistema (Información)
- e) Calificación del sistema en general (Calificación)
- f) La Interfaz presentada es amigable (Interfaz)

El cuestionario se entregó a cada profesor al final de la prueba, y se le indicó que fuera honesto en sus respuestas para poder obtener información real acerca del uso del sistema.

Las unidades de aprendizaje que se utilizaron para la evaluación fueron: Expresión Oral y Escrita, unidad que se imparte en todas las carreras, y Teoría de Sistemas.

Los resultados arrojados por la aplicación de esta prueba se presentan en la siguiente gráfica. El eje X, corresponde a los aspectos evaluados, interfaz, facilidad de uso, eficacia, ayuda, información y calificación general del sistema, el eje Y, el número de profesores que califica cada uno de los aspectos con los adjetivos representados en colores por ejemplo, en el aspecto de calificación del sistema, 3 profesores le asignaron excelente y 4 buena.



Gráfica No.1 Resultados arrojados en la Prueba Alfa

En esta prueba únicamente se detectó un problema:

- Se necesitaba que el profesor trajera organizada su información.

Las aportaciones más importantes de los profesores al sistema.

- El color del sistema les pareció muy formal.
- Manejar una lista con todas las carreras que oferta la UAEM, para facilitar la selección.
- Hacerlo más amigable, por ejemplo incluir música o sonidos.

En la práctica, cuando el software es utilizado, pueden surgir más defectos y con ello inicia otro ciclo de comprobación y depuración, en esta fase de pruebas resulta demasiado costosa, además lleva mucho tiempo, por lo que es necesario que desde las fases de análisis y diseño se ataque cualquier inconsistencia con los requerimientos.

Los resultados importantes que arrojó esta prueba son: la calidad de los materiales está condicionada a la información, si los profesores tienen más información acerca de las unidades de aprendizaje, sus materiales serán de mayor calidad, asimismo, cuando cuentan con conocimientos acerca de las teorías del aprendizaje, se emplea una mayor variedad de recursos para presentar su unidad.

Al finalizar se obtuvo un sistema que asiste a los profesores en la elaboración de materiales educativos, y que transforma el proceso de construcción que anteriormente era manual, a uno automatizado, facilitando, por una parte, la estructuración adecuada de los materiales presentados y por otra la adquisición del conocimiento por parte de los estudiantes, permitiendo con ello construir nociones básicas, forjar sus ideas y formar conceptos propios.

Conclusiones

La educación mediante el uso de software educativo, apoya el proceso de enseñanza-aprendizaje facilitando la enseñanza de conceptos, ayuda a resolver problemas y contribuye en el desarrollo de habilidades cognitivas del individuo. Asimismo permite que el aprendizaje se realice en menos tiempo, sin sacrificar la profundidad y la calidad de la enseñanza y además es adaptable a las necesidades y ritmo de aprendizaje del estudiante. Las teorías de enseñanza-aprendizaje aportaron elementos fundamentales en el procedimiento de elaboración de materiales y que fueron tomados como requisitos para el diseño del software de elaboración del software educativo.

La metodología RUP proporciona una respuesta a la tarea de buscar una metodología que permita la elaboración de software con calidad e identifique las necesidades del usuario desde el inicio hasta el final ya que además se adapta a las nuevas tecnologías. El sistema desarrollado es un software amigable, adaptable, sencillo, que transforma el proceso manual de construcción de software educativo en otro proceso donde el profesor es asistido por el mismo sistema en la elaboración de materiales educativos que incorporan las aportaciones de las teorías psicológicas de la educación, facilitando por una parte la estructuración de los materiales auto contenidos a ser presentados y por otra la adquisición del conocimiento por parte de los estudiantes de forma personalizada.

Las pruebas realizadas al software se basaron en las estrategias de prueba para arquitecturas convencionales y se pudo comprobar que el sistema puede generar software educativo utilizando el proceso descrito.

Es importante continuar investigando sobre el diseño de los contextos de aprendizaje con computadora que se adapten a las necesidades del estudiante e incorporen técnicas de Inteligencia Artificial y lenguaje natural.

Referencias

- [1] A. Cuevas, Sistema Tutorial Inteligente LIREC. Tesis de Maestría, México, CINVESTAV-IPN, 1994.
- [2] T. O'Shea, J. Self, Teaching and Learning with Computers: Artificial Intelligence in and Education, USA, Prentice-Hall. (1983)
- [3] Olds, H. Shwartz, J. Willie: People and Computers: Who teaches whom? Newton, MA: Educational Development Center, (1980)
- [4] J. Kaput.: Technology and Mathematics Education: Handbook of Research on Mathematics Teaching and Learning; A Project of the National Council of Teachers of Mathematics., Editor D. A. Grows, USA, MacMillan Publishing Company (1992)
- [5] D. P. Ausubel, J. D. Novak, H. Hanesian: Psicología cognitiva. Un punto de vista cognoscitivo, México. Trillas, (1989)
- [6] F. Díaz-Barriga, M. Lule: Estrategias Metodológicas de la Enseñanza. México. Material de Apoyo Didáctico, UAEM (2004)
- [7] J. Figueroa-Nazuno, M. Guadarrama-Tapia: Polilibros 1.7. Magno Congreso Internacional de Computación 2006. CIC-IPN México (2006)
- [8] M. G. Díaz-Anton, M. A. Pérez, A. C. Grimmán y L. Mendoza: Propuesta de una Metodología de Desarrollo de Software Educativo Bajo enfoque de Calidad Sistemática. Universidad Simón Bolívar, Caracas, Venezuela (2002)
- [9] G. Booch, J. Rumbaugh, Jacobson: El lenguaje Unificado de Modelado UML, México Addison Wesley (2000)
- [10] R. S. Pressman, Ingeniería del Software, Un enfoque práctico, México, Mc Graw Hill (2005)

Saturnino Job Morales Escobar: **Licenciado en Computación egresado de la Benemérita Universidad Autónoma de Puebla donde también obtuvo la Maestría en Ciencias de la Computación. Actualmente es profesor investigador en el Centro Universitario Valle de México de la Universidad Autónoma del Estado de México.**

Dora María Martínez Magaña: Licenciada en Informática egresada del Instituto Tecnológico de Tlalnepantla, tiene la Maestría en Informática de la Universidad Autónoma del Estado de México, es profesor investigador en el Centro Universitario Valle de México de la UAEM.

Ivonne Rodríguez Pérez: Licenciada en Matemáticas Aplicadas y Computación egresada de la Universidad Autónoma de México, tiene la Maestría en Ciencias de la Educación en la Universidad del Valle de México, es profesor investigador en el Centro Universitario Valle de México de la UAEM.

Sara Lilia García Pérez: Licenciada en Contaduría egresada de la Universidad Autónoma de México, tiene la Maestría en Auditoría en la Universidad Autónoma de México, es profesor investigador en el Centro Universitario Valle de México de la UAEM.

Desarrollo de un sonómetro digital

Pablo Roberto Lizana Paulin¹, José de Jesús Negrete Redondo², Sergio Beristain³

¹ESIME Zacatenco, email plizana98@gmail.com, ²ESIME Zacatenco, email josejnegreter@yahoo.com.mx, ³ESIME Zacatenco, sberista@hotmail.com.

Resumen.-

En este trabajo se diseñó un sonómetro digital utilizando el sistema de programación de Labview 7.0, basado en las normas nacionales e internacionales. La señal de sonido es capturada en un micrófono, y a través de la tarjeta de audio de la misma computadora se convierte la señal en digital, después se desarrolló un algoritmo matemático para la curva de ponderación “A” y la señal de entrada se convoluciona con la del algoritmo de la curva A para transformarla a una señal de acorde a la respuesta del oído humano.

Abstract.-

This paper presents a digital sound level meter design by Labview 7.0, it is based on national and international standards. The sound signal is captured by a microphone, and converted to digital by the audio card of the computer, then by a mathematical algorithm designed to conform the weighting curve “A”, the input signal is convoluted with the A curve in order to transform it to a signal that complies with the human ear response.

Palabras clave: Algoritmo matemático, Curvas de ponderación, lenguaje de programación Labview, sonómetro.

Keywords: Mathematical algorithm, Weighting curve, Labview programming language, Sound level meter.

I. Introducción

Los sonómetros son instrumentos diseñados para responder de manera semejante al oído humano, dando mediciones objetivas de nivel de presión sonora; hay diferentes sistemas de medición disponibles y cada sistema consiste en un micrófono, una sección de procesamiento y una unidad de lectura de salida en decibeles (dB). El decibel se puede definir así:

$$NPS=20\log_{10}P^2/P_0^2 \dots(1)$$

NPS = Nivel de Presión sonora en decibeles (dB).

P = Presión acústica en Pascales (Pa).

P₀ = Presión acústica de referencia que es 2 X 10⁻⁵ Pa.

El factor de 0.00002 Pa. Es la presión mínima que el oído humano alcanza a percibir a una frecuencia de 1 KHz.

El micrófono convierte la señal y produce una señal eléctrica equivalente. La señal generada en el micrófono es muy pequeña por lo que requiere de un preamplificador antes de ser procesada.

En la Ecuación 1 se puede notar que si se dobla la presión acústica se incrementara el NPS en aproximadamente 6 dB; si se cuadruplica la presión acústica se aumentara el NPS en 12 dB, y si la presión se incrementa ocho veces aumentara 18 dB, además de que la respuesta del oído no es uniforme con la frecuencia. Por lo que se debe de diseñar un algoritmo matemático que responda de manera muy parecida a la respuesta subjetiva del oído humano, como se muestra en la Figura 1 conocida como curvas de igual sonoridad (nivel percibido por el oído humano).

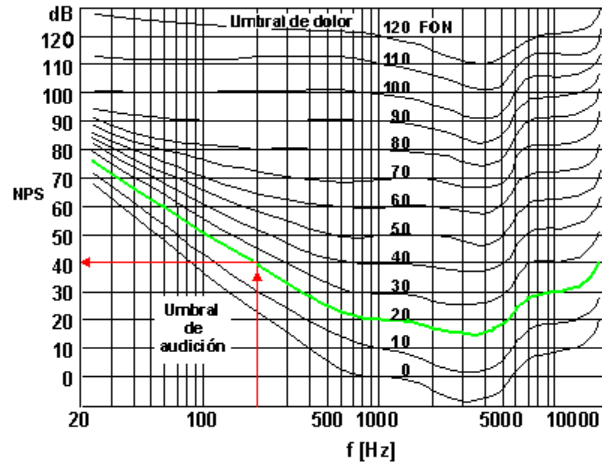


Figura 1.- Curvas de igual sonoridad de Fletcher y Munson.

En la curva de la Figura 1 se puede observar que el oído humano no es igual de sensitivo a todas las frecuencias, por ejemplo si se considera una frecuencia de 60 Hz. Con un NPS de 70 dB, este sonido tendrá la misma sonoridad en fonos que un tono de 1 KHz a 40 dB, para diseñar un algoritmo matemático cuyas características varíen de forma semejante que las curvas de percepción subjetiva del oído humano es difícil. Esto ha dado como resultado que se diseñaran cuatro tipos de curvas para los sonómetros, que son conocidas como curvas de ponderación A, B, C y D. en la actualidad, solo se utilizan las curvas de ponderación A y C. Pero cuando se realizan mediciones de aviones a reacción se les aumentan 10 dB a las mediciones hechas por el sonómetro, Figura 2.

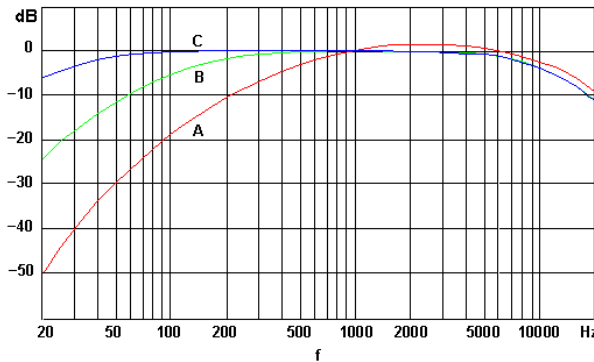


Figura 2.- Curvas de ponderación A, B y C por convención internacional.

El principal objetivo para el desarrollo de los sonómetros se basa en la idea principal de diseñar un algoritmo matemático que responda de forma igual o parecida al oído humano.

Las curvas de ponderación son una convención internacional para la manufactura de sonómetros. En la escala A la respuesta disminuye mucho a bajas frecuencias. Por ejemplo, si se tiene un tono de 50 Hz en la escala de ponderación A, este valor será menor en 30 dB que uno de 1 KHz. En la curva de ponderación C solamente se tienen pequeñas diferencias menores de 1 dB entre las frecuencias de 63 y 4 KHz.

II.- Metodología

Para el desarrollo de este sonómetro se elaboró un algoritmo matemático programado en el lenguaje de LABVIEW con el fin de sustituir hardware por software, empleando también la tarjeta de audio que ya las computadoras traen integrada y así aprovechar la capacidad de una computadora portátil y económica para contar con un instrumento versátil y que normalmente tiene un costo mucho mayor.

III.- Desarrollo del algoritmo.

Una señal real está formada por una sumatoria de señales de diferentes frecuencias y ruido, por lo tanto, para su simulación basta con crear varias señales puras con diferentes frecuencias y amplitudes para después sumarse y crear una “señal real simulada” (proceso inverso al análisis).

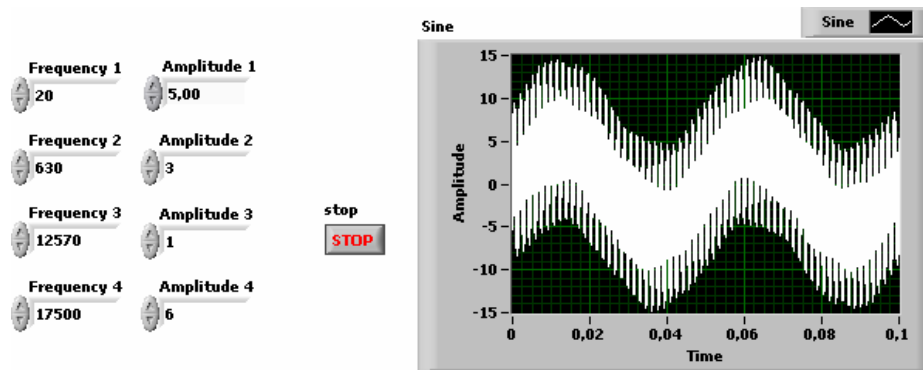


Figura 3. Suma de frecuencias para obtener de una Señal Audible en forma simulada.

Esta parte se concentra en la obtención los componentes de la señal de entrada. Es decir, una vez simulada la señal de entrada, se hace pasar a través de filtros para obtener las diversas señales que la conforman. Los centros de banda de los filtros simulados se han fijado conforme lo marca la norma NMX-C-102-1976.

Una vez que se ha simulado la señal real, se procede a aplicar la transformada Rápida de Fourier para obtener la respuesta espectral de la señal en el dominio de la frecuencia, es decir, en esta parte se ha pasado del dominio del tiempo al dominio de la frecuencia. Lab View 7.0 permite realizar este paso sin complicaciones, de manera rápida y eficaz.

La simulación en Lab View 7.0 de la curva de ponderación “A” se ha llevado a cabo a partir de la Ecuación 2:

$$A(f) = 12200^2 f^2 / (f^2 + 20.6^2)(f^2 + 12200^2)(f^2 + 107.7^2)^{0.5}(f^2 + 737.9^2)^{0.5} \dots (2)$$

Esta ecuación es producto de un trabajo de investigación realizado anteriormente en el que se utilizaron métodos de interpolación de datos (método de Newton), para describir el comportamiento del circuito físico de la curva de ponderación “A”.

Es importante considerar que esta fórmula dará valores lineales, no en decibeles.

En la simulación de esta curva se dan valores de frecuencias y el programa calcula el valor que le corresponde según la Ecuación 2.

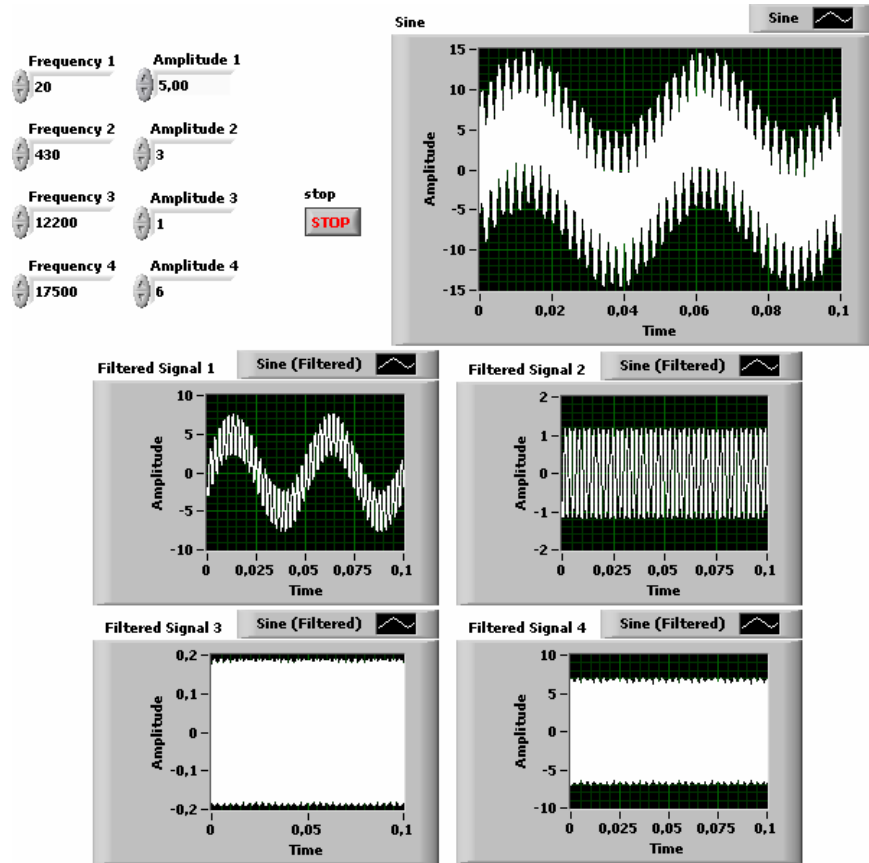


Figura 4. Aplicación y Obtención de la Transformada Rápida de Fourier

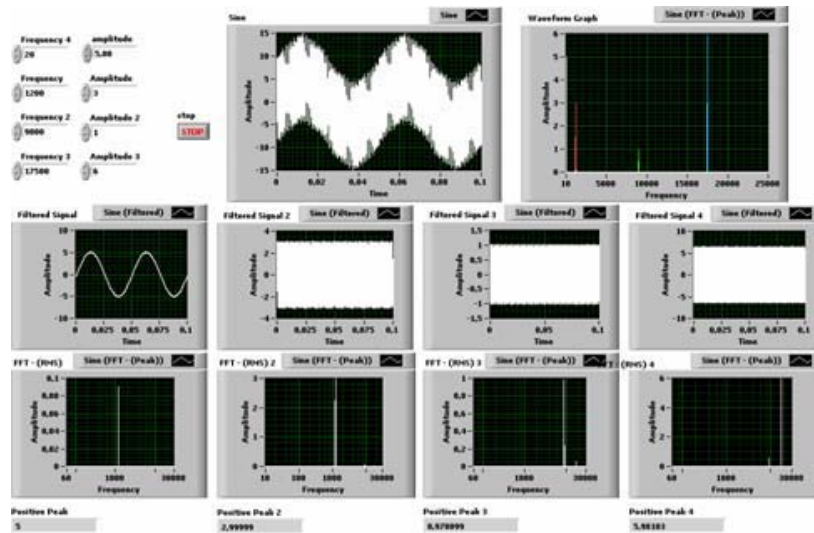


Figura 5. Simulación del Algoritmo que describe la curva de Ponderación "A".

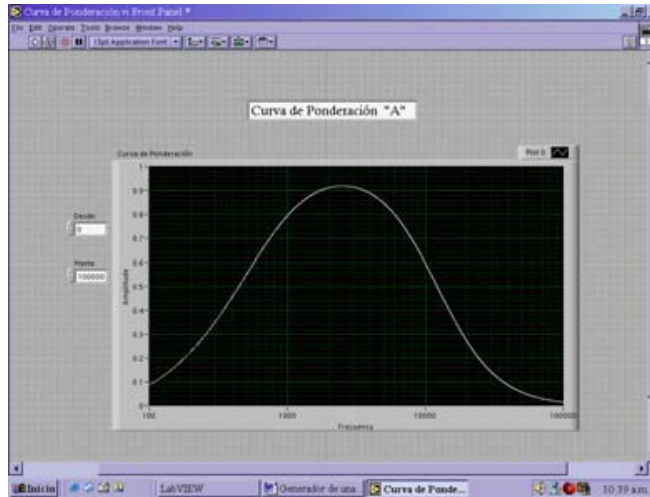


Figura 5. Obtención de la Curva de Ponderación "A".

Hasta este momento, como se señaló anteriormente, los resultados que da la ecuación empleada son lineales. Para proseguir se requiere corregir dos aspectos: en primer lugar, las unidades que se tienen son volts y se necesita tener valores en dB's; y segundo, la curva de ponderación debe dar un resultado de 0 dB cuando se le introduce un valor de frecuencia de 1,000 Hz y en este caso el valor obtenido en esta frecuencia es de 0.7943463957 V.

Así pues se pueden resolver los dos problemas en un solo paso, esto es, se convierten los valores que se han obtenido a decibeles mediante la Ecuación 3:

$$dBA = 20 \log_{10} A(f)/A(\text{ref})... (3)$$

En la ecuación A(f) hace referencia a cierto valor "X" de frecuencia, y A (Ref) se refiere al valor de referencia. Es con este valor de referencia con el cual se puede ajustar la curva que se ha obtenido: el valor de referencia a utilizar es por lo tanto 0.7943463957 V.

La Ecuación 3 quedaría:

$$dBA = 20 \log_{10} A(f)/A(1000)... (4)$$

Ahora los valores que contiene la curva de ponderación simulada están dados en dB y corresponden a los determinados por la norma NMX-C-102-1976 que muestra las respuestas relativas y tolerancias asociadas para condiciones de campo libre de la curva de ponderación "A".

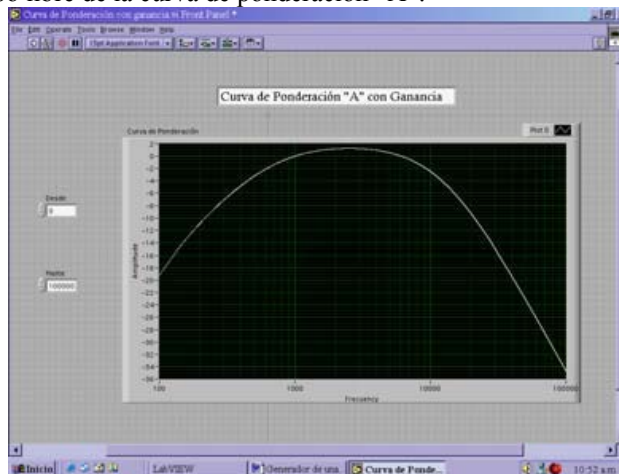


Figura 7. Cálculo del Nivel de Presión Sonora con la curva A.

Cómo último paso hay que calcular el Nivel de Presión Sonora en la señal que se ha filtrado para lo cual se procede a calcular una amplitud ponderada de la señal, es decir, se convolucionan las señales que se han generado y analizado con el algoritmo de la ponderación "A".

Una vez que se ha obtenido la amplitud ponderada de la señal, se está en posibilidades de calcular el Nivel de Presión Sonora mediante la Ecuación 5:

$$NPS = 20 \log_{10} \text{AmpPond}/V_{\text{ref}}... (5)$$

Tabla I
Atenuaciones y tolerancias de la curva A
NORMA NMX-C-102-1976

Centro de la banda de frecuencia	Corrección a los niveles de presión acústica (en dB)	Tolerancia	
31.5	-39.4	3	-3
63	-26.2	3	-3
125	-18.1	1	-1
250	-8.6	1	-1
500	-3.2	1	-1
1000	0	1	-1
2000	1.2	1	-1
4000	1.0	1	-1
8000	-1.1	1.5	-3
10000	-2.5	2	-4
16000	-6.6	3	-0
20000	-5.3	3	-0

En este caso el V_{Ref} es de 20×10^{-5} Pa.

Falta dibujo completo

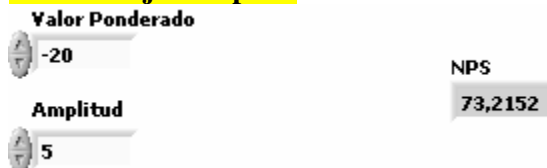


Figura 8.- Cálculo del valor en dB.

De esta manera, uniendo todos los pasos anteriores en un solo programa, se obtiene el diagrama a bloques mostrado en la Figura 9 y en la Figura 10 (ver anexo I), se muestra el panel frontal en el cual el usuario puede introducir diferentes componentes en frecuencia para la simulación de una señal de entrada, con esos datos el programa muestra la señal resultante con los valores introducidos, los valores obtenidos una vez que se ha aplicado la Transformada Rápida de Fourier y el algoritmo de ponderación, valores necesarios para realizar el cálculo correspondiente y desplegar los valores de Amplitud Ponderada y el Nivel de Presión Sonora.

IV.- Resultados experimentales

Estas pruebas se realizaron con la finalidad de saber si los valores obtenidos eran semejantes a los que nos da a conocer la norma anterior. Para esto se utilizó un micrófono de bobina móvil, conectado a la entrada de la tarjeta de audio de la PC, una sonómetro tipo 1 Bruel & Kjaer modelo 2230, un altavoz dinámico conectado a un generador de audio. Las pruebas se realizaron en una cámara anecoica o sonoamortiguada,

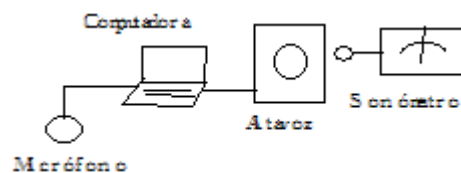


Figura 7. Experimento realizado en la cámara sonoamortiguada del laboratorio de acústica.

Se genera una señal de audio por medio del generador conectado al altavoz dinámico. Se pone el sonómetro Bruel & Kjaer a un lado del micrófono que está conectado a la PC, los valores obtenidos fueron los siguientes:

Tabla 2
Tabla de resultados obtenidos

Frecuencia (Hz)	B&K (dBA)	Labview (dBA)	Diferencia
80	55.80	54.90	0.90
100	56.00	54.50	1.50
125	54.50	54.00	0.50
160	60.00	59.90	0.10
200	72.20	72.80	-0.60
250	70.60	72.50	-1.90
315	67.00	66.80	0.20
400	72.10	71.60	0.50
500	76.60	76.80	-0.20
630	77.40	79.50	-2.10
800	82.60	84.00	-1.40
1000	81.00	81.50	-0.50
1250	81.40	83.20	-1.80
1600	84.00	86.00	-2.00
2000	77.40	79.10	-1.70
2500	78.50	79.10	-1.70
3150	82.60	86.40	-3.80
4000	83.80	90.00	-6.20
5000	81.20	88.10	-6.90
6300	74.50	84.10	-9.60

IV. Conclusión

Como se podrán ver los resultados obtenidos son muy semejantes a los dados por la norma. Por lo que se puede decir que a través del procedimiento descrito se puede obtener un sonómetro que cumpla con la legislación vigente con la ayuda de una PC, y que también pueda ser modificado de acuerdo a los cambios tecnológicos y normativos

El principal objetivo, al diseñar este sonómetro, fue que pudiera ser llevado a la práctica por alumnos de cualquier escuela o facultad de ingeniería. Por esta razón, la implementación de este medidor es un recurso didáctico muy importante, además de proporcionar a los alumnos la posibilidad de contribuir en su diseño y mejoras.

Este proyecto ha dado base para el desarrollo de una serie de instrumentos de medición acústica que sirven como apoyo de instrumentación en los laboratorios de acústica.

Referencias

- [1] **Recuero López, Manuel. “Estudios y controles para grabación sonora”. Editorial Instituto Politécnico Nacional. México, 1991. 356 pp**
- [2] **Kinsler, Lawrence E., et al. “Fundamentos de Acústica”. Editorial Limusa. México, 1990.**
- [3] **Cooghin, Robert F., y Driscoll, Frederich F. “Amplificadores operacionales y circuitos integrados lineales”. Editorial Prentice Hall. México, 1993.**

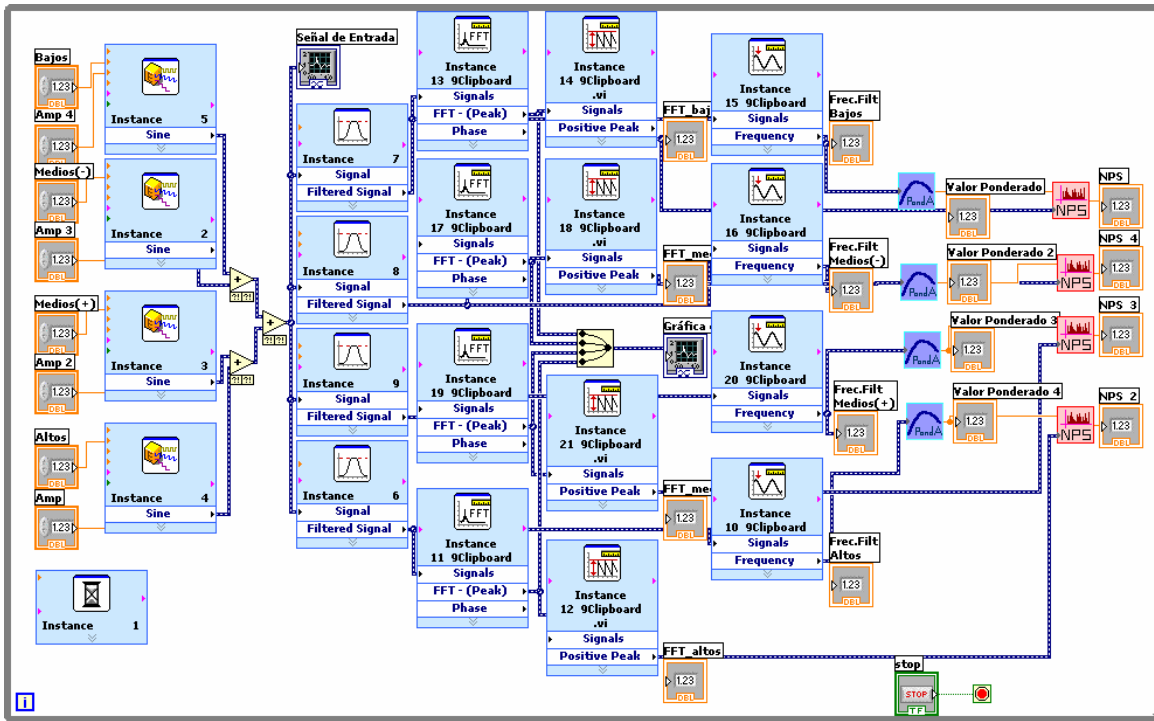


Figura 9.- Diseño del sistema digital del simulador de sonómetro en lenguaje de programación en Labview 7.0

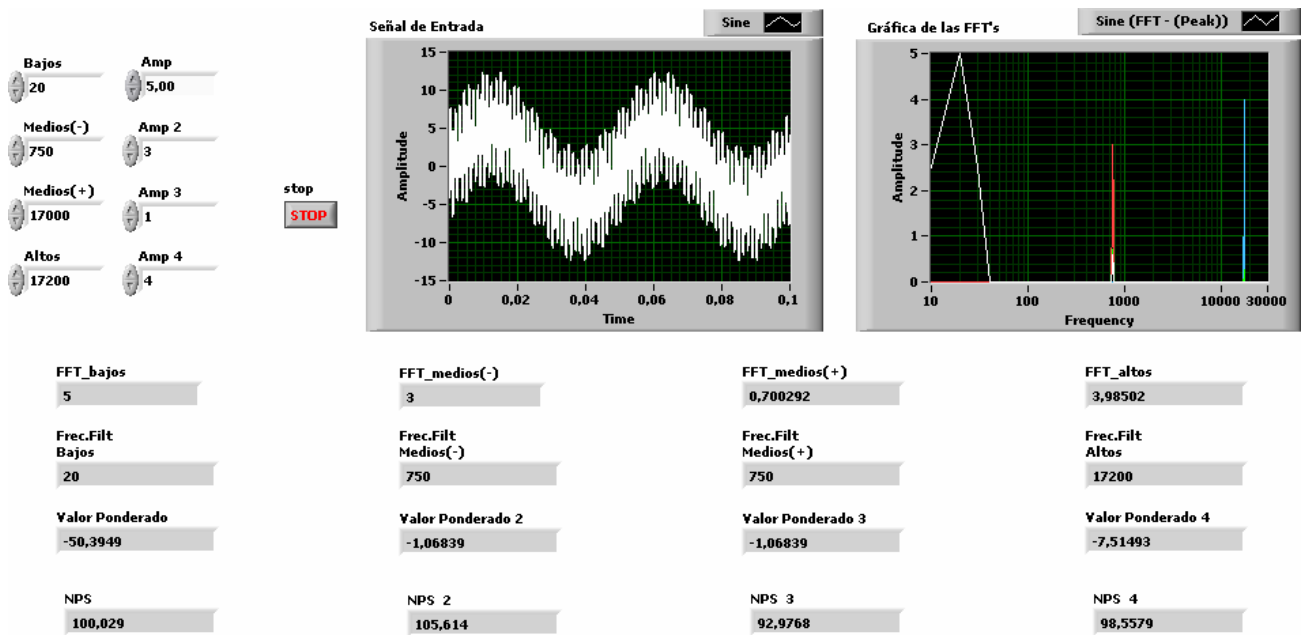


Figura 10.- Señal de entrada en el simulador del sonómetro y la señal de salida en el sonómetro.

Sistema de detección del nivel de glucosa en sangre para la administración de insulina

Rubén Ortega González, José Sánchez Juárez, Fabiola Ocampo Botello, Roberto de Luna Caballero

Escuela Superior de Cómputo, ESCOM-IPN. Av. Juan de Dios Bátiz Esq. Miguel Othón de Mendizábal S/N
Unidad Profesional Adolfo López Mateos. Col. San Pedro Zacatenco. CP. 07738. México D.F.
focampob@ipn.mx, Tel. 5729.6000 ext. 52028

Resumen.- En este artículo se presenta el análisis y diseño de un sistema automatizado y computarizado que utiliza un biosensor de glucosa para medir y controlar el nivel de glucosa en la sangre de una persona diabética, con la finalidad de inyectar la dosis de insulina que el paciente necesita. Debido a que la cantidad de personas que padecen esta enfermedad crece cada día más, este tipo de sistema proporciona la agilidad que se necesita en los hospitales para la atención inmediata de estos pacientes, ya que en una base de datos se almacena el historial de cada paciente [1], lo que permite realizar comparaciones del comportamiento de la glucosa y con ello llevar un mejor control de las dosis a aplicar. Este sistema ayuda a minimizar la cantidad de personal técnico de laboratorio encargado de detectar el nivel de glucosa en la sangre.

Palabras clave: Biosensor, dosis, glucosa, insulina, nivel.

I. Introducción

La insulina es una hormona producida en el páncreas que regula la cantidad de glucosa en la sangre. En personas con diabetes, el páncreas no produce nada de insulina, o produce insulina insuficiente, inclusive, insulina defectuosa.

La glucosa entra a las células a través de receptores, estos son sitios en las células que dejan entrar a la insulina. Una vez adentro, la glucosa puede ser usada como combustible; pero, sin insulina, es difícil para la glucosa poder entrar a las células. La función de la insulina es, de manera semejante, el embudo que permite a la glucosa (azúcar) pasar a través de los receptores hasta el interior de las células, tratamiento que se aplica a los diabéticos, debido al problema que presentan en la generación de insulina.

Los objetivos del tratamiento de diabetes por medio de suministro de insulina son principalmente:

Obtener una adecuada compensación metabólica

Prevenir las complicaciones agudas (hipoglucemias)

Evitar, retrasar o minimizar las complicaciones tardías

Disfrutar de sensación subjetiva de bienestar físico y psíquico

Conseguir una expectativa de vida similar a la de un individuo no diabético.

Las diferentes pautas de tratamiento se consiguen mezclando los diversos tipos de insulina según su acción para conseguir un control satisfactorio. Su administración es siempre inyectada de forma subcutánea. Existen dos tratamientos que son utilizados comúnmente en pacientes diabéticos:

- Tratamiento convencional: dos inyecciones de insulina nph o lenta, uno antes del desayuno y otro antes de la cena.

- Tratamiento intensivo: consiste en administrar dosis de regular antes de las comidas y además 1, 2 ó 3 de nph o lenta para cubrir las necesidades basales.

II. Procedimiento metodológico

Dado que en la actualidad los problemas de diabetes se presentan con mucha frecuencia en los diferentes niveles de la sociedad, resulta de suma importancia su estudio, considerando que en la mayoría de los casos el control de esta enfermedad se lleva a cabo a partir de la evaluación periódica por parte de un médico a un paciente y del correspondiente suministro de cierta cantidad de insulina que comúnmente resulta ser imprecisa, de lo cual resulta justificable el desarrollo de un sistema de control por computadora que permita la aplicación adecuada de esta sustancia con la finalidad de prolongar y mejorar la vida de estos pacientes.

El diagrama de contexto del sistema desarrollado se presenta en la figura 1.

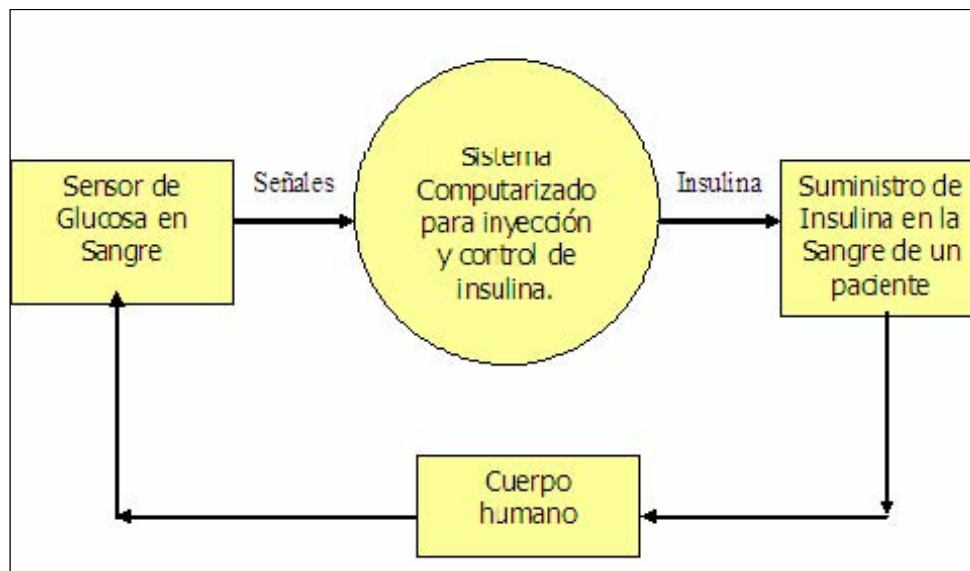


Figura 1 Prototipo de detección de glucosa y suministro de insulina.

Como puede apreciarse en el diagrama de contexto, este es un sistema realimentado que permite medir el nivel de azúcar de un paciente a través de un sensor de glucosa, para luego ser procesado y analizado por medio de la computadora, la cual nos dará la información necesaria que indique cual será la cantidad adecuada de insulina que se le debe inyectar al paciente de manera automática.

Los módulos desarrollados en este proyecto son los siguientes:

- Biosensor de Glucosa.
- Acondicionamiento de la señal.
- Digitalización de la señal analógica.
- Base de Datos y Control de Glucosa.
- Suministro de Insulina en Sangre.

Cuya función se explica a continuación, en la figura 2 se muestran los componentes que integran el prototipo desarrollado [2], el cual funciona de la siguiente manera, el biosensor de glucosa se introduce en la sangre, el cual reacciona y produce una corriente que depende de la cantidad de glucosa. La corriente que entrega el biosensor se debe acondicionar y digitalizar, para ser utilizada en la comparación de los datos que están almacenados en una base de datos relacional [3][4][5].

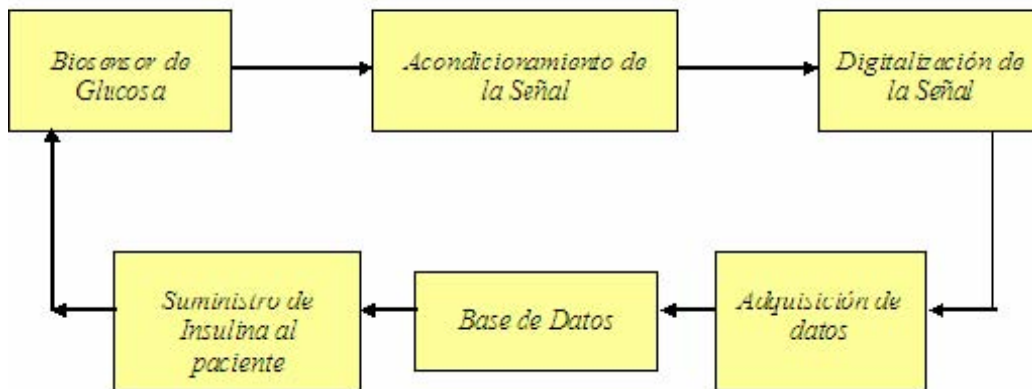


Figura 2 Diagrama de contexto.

III. Exposición de resultados

En el sistema desarrollado el biosensor de glucosa (sensor bioenzimático) es de suma importancia, por lo que debe operar con un excelente funcionamiento y una exactitud preponderante, para brindar un eficaz desempeño del sistema de control de insulina.

El sensor de glucosa está constituido con materiales inmovilizados biológicamente en una composición simple: cuando el sustrato específico para una enzima es adherido, la reacción enzimática toma lugar y el producto de esta reacción es un sustrato para la segunda enzima [7] [8]. En los últimos años, el estudio de los biosensores enzimáticos se ha incrementado hasta el punto que han hecho reaccionar a más de una enzima que puede incrementar la sensibilidad. En los biosensores bioenzimáticos, la segunda enzima es responsable de una señal analógica; de ahí entonces, que la segunda enzima determina los límites de detección.

El biosensor se muestra en la figura 3:

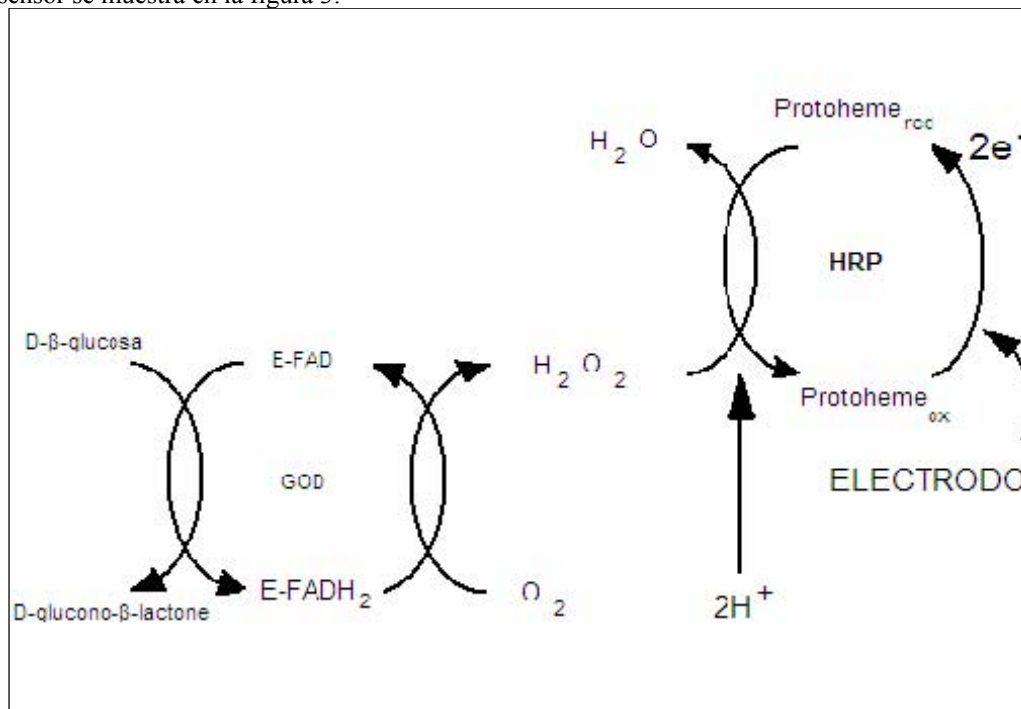


Figura 3 Mecanismo de la bioenzima basado en el biosensor para glucosa.

La detección de la glucosa está basada en una respuesta de corriente que se da a partir una reacción electroquímica [6]. El biosensor de glucosa está formado de un electrodo de prueba de platino.

La temperatura es uno de los factores que más afectan a la respuesta del sensor debido a que se ha reportado que la máxima actividad enzimática se presenta aproximadamente a 310° K. Entonces, los estudios conducen a establecer que entre 298° K y 321° K se da la máxima respuesta del sensor.

IV. Discusión de resultados

El sensor de glucosa es muy importante, y uno de los aspectos que se debe mantener con mucho cuidado es la temperatura de operación, por lo que en seguida se presenta un estudio del efecto de la temperatura en el biosensor.

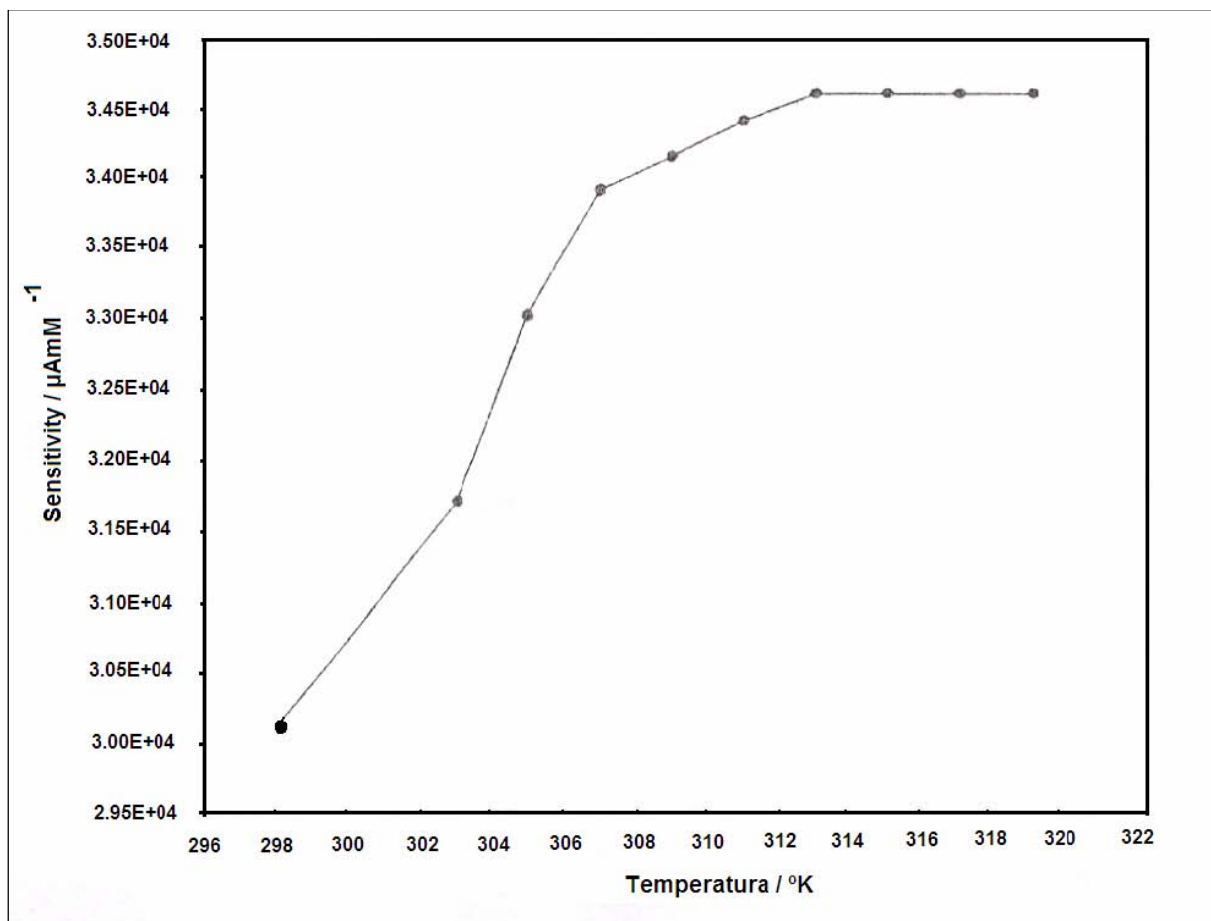


Figura 4 Curva de temperatura del biosensor de glucosa.

Como se puede observar en la figura 4, la sensibilidad incrementa al aumentar la temperatura, aproximadamente después de 313° K la respuesta del biosensor es independiente de los cambios en la temperatura.

pH y Temperatura:

La actividad enzimática está fuertemente relacionada al pH. Considerando lo anterior, el pH es una variable que necesita ser eficientemente controlada ya que este afectará la respuesta de la enzima que modifica la respuesta del sensor. De esta forma, el pH seleccionado tiene que conformarse por dos requerimientos: a) la

respuesta del biosensor tiene que ser la máxima posible y b) que la biocomposición no se destruya fácilmente.

S.E. (M)	pH = 6	pH = 6.5	pH = 7	pH = 7.5	pH = 8.0	pH = 8.5
0.01	28576.1	27887.7	28092.6	27276.6	26441.8	25336.9
0.1	31023.2	34739.0	33216.2	31150.4	28662.6	25980.0
0.5	31278.2	34705.4	32630.3	32218.2	28849.8	25008.0

Tabla 1 La sensibilidad promedio.

De la tabla 1 se observa que en todos los casos el electrolítico suministrado (S. E.) es NaH_2PO_4 / Na_2HPO_4 para diferentes concentraciones y diferentes pH's.

Se puede observar también que cuando el pH oscila en un valor de 6.5 y la concentración electrolítica de apoyo es de 0.1 o 0.5 M, se registra la mayor sensibilidad.

V. Conclusiones

Una vez que en la base de datos ya se ha determinado la cantidad necesaria de insulina que se le tendrá que inyectar al paciente así como el tipo, el suministro podrá llevar a cabo de manera manual o automática. Manualmente: el médico tendrá la posibilidad de conocer de manera precisa la cantidad necesaria y el tipo de insulina que se tendrá que inyectar al paciente.

Automáticamente: de manera automática la información que arroje la base de datos se tendrá que procesar a través de un controlador proporcional que se programó en un PIC, el cual a su vez enviará los datos necesarios para que se active una bomba neumática que acoplada a una jeringa que permite inyectar la cantidad necesaria de insulina al paciente. Otro aspecto importante es el tiempo de vida del sensor, esto se toma de las características del fabricante.

Referencias

- [1] D. Anick, D. Mitra, and M. M. Sondhi, Stochastic theory of a data handling system with multiple sources, Bell Syst. tech. J., 61, pp. 1871-1984, (1982).
- [2] S. Boyd and C. Barratt, Linear controller design: Limits of performance, Prentice-Hall, (1991).
- [3] David M. Kroenke, "Procesamiento de bases de datos", 8a Edición, Ed. Pearson Educación, (2003).
- [4] Edgard Yourdon, "Análisis Estructurado Moderno", Ed. Prentice Hall, (1989).
- [5] Ralph M. Stair, George W. Reynolds, "Principios de Sistemas de Información", 4a Edición, Ed. International Thomson Editores, (2000).
- [6] Kulys J., Schmid R., Biosensors and Bioelectron, 6, 1, 43, (1991).
- [7] Rivera B., Morales-Pérez A., Glucose determination in commercial serums by means of an amperometric bienzymatic sensor with a rigid matrix, Sensors and Chemometrics, pp. 47-54, (2001).
- [8] Céspedes Mulero F., Desarrollo de materiales compuestos graffito-epoxy de matriz modificada química y biológicamente, y su utilización en la construcción de biosensores amperométricos, Tesis Doctoral de la Universidad Autónoma de Barcelona, España, (1993).

Instrucciones para los autores

Los artículos que se someten a **RISCE** deben contener resultados inéditos y originales, no haber sido publicados con anterioridad ni haber sido sometidos simultáneamente a otra revista científica. Si el artículo ha sido presentado, sometido o publicado en alguna otra parte, deberá informarse al coordinador editorial. Los artículos deben ajustarse a las siguientes especificaciones:

- Idioma Inglés (anexar un resumen y palabras clave en español)
- Idioma Español (anexar un resumen y palabras clave en Inglés)
- Procesador de texto admitido: MS-Word.
- Tamaño de página: carta, utilizar un solo lado de la hoja. Máximo 10 páginas.
- Márgenes: izquierdo 2.5 cm y derecho 2 cm., superior 2.5 cm e inferior 2.5 cm.
- Autores: primer nombre seguido de los dos apellidos (sin abreviaturas), abajo: afiliación y e-mail.
- Tipo de letra del texto regular: Times o Times New Roman de 10 pt (título original 22 pt; secciones 11.5 pt, subsecciones 11.5 pt, en negritas).
- Texto: a una columna y con espaciado sencillo (renglón seguido).
- Resumen/Abstract: entre 70 y 150 palabras, colocado al principio del texto, seguido del de Español o inglés según sea el caso.
- Palabras clave/Keywords: colocadas después del resumen en negritas, y no más de 10.
- Imágenes y fotografías: deben ser de alta calidad, con colores bien definidos y contrastantes, en mapa de bits (no sectorizadas) en formato JPG e incrustadas en el texto de forma que se puedan manipular independiente.
- Fórmulas: Deberán de presentarse en formato de tabla sin bordes, centradas y la numeración de c/u justificada a la derecha con negritas en mapa de bits, no vectorizadas.
- Pies de figura. Deben mencionarse dentro del texto y numerarse de manera consecutiva con un tipo de letra Times New Roman 9 puntos
- Cabecera de tabla. Deberá presentarse en la parte superior de la tabla un numeración consecutiva y descripción con tipo de letra Times New Roman 9
- Referencias:

En cualquier caso el nombre del autor del artículo o publicación web deberá mostrarse al principio. Deberán ordenarse conforme aparezcan dentro del texto encerradas entre paréntesis cuadrado —[]—. A continuación algunos ejemplos:

- [1]. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. *Int. J. Digit. Libr.* 1 (1997) 108–121
- [2+]. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.): *Theoretical Aspects of Computer Software. Lecture Notes in Computer Science*, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (1997) 415–438
- [3]. van Leeuwen, J. (ed.): *Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science*, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)
- [4]. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)

Instrucciones:

Enviar el archivo en extenso a la siguiente dirección electrónica: ebustosf@gmail.com

Los revisores técnicos le harán llegar sus observaciones y modificaciones, las cuales deberá realizar y reenviar el archivo corregido al correo arriba mencionado.

El comité editorial se comunicara mediante correo electrónico indicándole la aceptación o rechazo del artículo.

Se le solicitará autorización para publicación; en caso de aceptar se le indica la cuenta donde debe hacer el depósito por cobro de publicación y el costo, el cual no debe exceder de \$1000.00 pesos mexicanos.

Reserva de Derechos 04-2008-062613190500-203