

I.5 NUEVO MÉTODO PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN

*Roberto Mendoza Padilla**

Resumen

En el presente trabajo se describe un nuevo método para el desarrollo de sistemas de información el cual utiliza las herramientas de la Inteligencia Artificial, particularmente se enfoca el problema de los sistemas estáticos y la necesidad de la reprogramación, lo que repercute en un aumento de costos que mediante algunas alternativas propuestas aquí, pueden minimizarse.

INTRODUCCIÓN

Esta y otras ponencias son el resultado de varios años de trabajo en relación a la Informática dentro del Departamento de Computación de la Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas (UPIICSA) del IPN en el cual se ha conformado la Unidad de Investigación y Desarrollo en Computación (UIDC).

Es bien sabido por las personas relacionadas con la informática, y aún por las que no lo están, que el proceso de desarrollo de los Sistemas de Información Administrativos tradicionales para cualquier empresa o institución ha seguido un proceso explosivo en los costos que implican, pasando de la época que los componentes mecánicos y electrónicos jugaban el principal factor económico, al punto en el que los costos de desarrollo y personal lo son, tal como lo muestra la figura 1.

Roberto Mendoza Padilla desarrolló este trabajo en la UPIICSA del IPN en el Departamento de Computación, Unidad de Investigación y Desarrollo. Lo presentó como ponencia en el Instituto Tecnológico y de Estudios Superiores de Monterrey durante el Segundo Encuentro Científico y Cultural de su Proyecto Año 2000 Abril 22, 23 y 24 de 1985

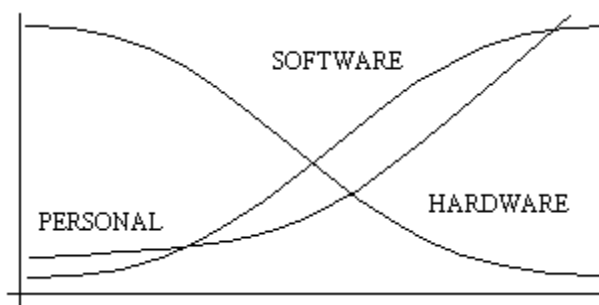


Figura 1. Desarrollo tradicional de sistemas

Este problema es agravado en gran medida por la falta de técnicas apropiadas para el desarrollo de software, o bien, por el caso omiso que hace el personal involucrado en el área de las ya existentes, como por ejemplo la mala documentación o la ausencia de esta para los sistemas desarrollados o en desarrollo, lo que provoca retrasos considerables y un aumento excesivo de costos al estar combinado con la alta rotación de personal, tal como lo muestra la figura 2. En esta figura se tiene que las líneas verticales muestran puntos en los que hay cambio de personal, provocando rearranques de desarrollo por la falta de soporte documental.



Figura 2. Desarrollo de sistemas sin soporte documental y con rotación de personal

Un problema adicional sobre el cual se inician los primeros desarrollos es el que se refiere a la ausencia de un mecanismo que permita modificar al sistema de información sin un esfuerzo considerable, pues se llega a dar el caso en el que hay que rehacer

gran parte del sistema en uso debido a los nuevos requerimientos del medio ambiente o deficiencias en su diseño.

Estas consideraciones acerca del Desarrollo de Sistemas, normalmente se combinan produciendo frustración entre los usuarios que pensaron que una computadora les solucionaría sus problemas en una forma mágica, tan solo por la gran inversión que hicieron al adquirirla.

Estos problemas y otros no mencionados provocan actualmente altos costos y/o una gran pérdida de tiempo, sin embargo, el presente estudio solo se enfocará al tópico que he denominado “Programación en Masa”, la cual es provocada por el enfoque actual de los Sistemas de Información, en el cual se dice implícitamente que los procesos o programas que lo forman deben ser el reflejo de su medio ambiente, es decir, cada rutina o módulo dará solución a una parte específica del problema, llegando posteriormente al punto de que cuando se modifiquen los requerimientos del medio ambiente, se debe hacer una reprogramación, debido a que este se encuentra reflejado en los programas, habiendo tantos como módulos o funciones tenga éste.

Además de lo anterior, este tipo de enfoque considera de alguna forma que su medio ambiente es estático, pues los programas funcionan en base a una fotografía tomada de la realidad en un momento determinado, permitiendo hacer adiciones, cambios o eliminaciones de funciones de acuerdo a la modularidad con que se haya desarrollado el sistema, pero aún en este caso, habrá necesidad de reconstruir el sistema, y en el peor de los casos, si de acuerdo a los nuevos requerimientos del medio ambiente se tuvo que cambiar la descripción de un registro, se deberá rearmar cada uno de los programas para formar nuevamente el sistema.

Existe aún otro caso en el que la empresa o institución tiene nuevos requerimientos, y por la imposibilidad de poder cambiar el sistema, ya sea porque no hay alguien que lo haga o porque se compró hecho, se deberá tomar la decisión de adaptar su medio ambiente a las restricciones estáticas de los programas o bien de tener que reconstruir nuevamente el sistema, ahora con los nuevos

requerimientos y tratando de no cometer los mismos errores de diseño, pero manteniendo esa filosofía fotográfica.

La alternativa que se presenta aquí pretende ser una herramienta de desarrollo con ayuda de la cual se elimine la incomoda relación, entre un medio ambiente dinámico y un sistema computarizado estático, evitando además la costosa Programación en Masa, logrando finalmente llegar a una gráfica de desarrollo tal como se muestra en la figura 3, obteniendo una drástica reducción en los costos de los sistemas, al mantener un software dinámico sin la necesidad de un gran cuerpo de analistas y programadores.

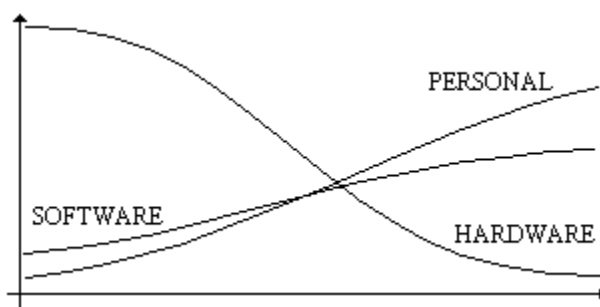


Figura 3. Desarrollo de sistemas "sin programación en masa"

Esta forma de enfocar a los sistemas viene a cumplir históricamente con el papel que le corresponde, pues se ha visto que de acuerdo al diagrama de la figura 4 que representa a un sistema de información, al inicio lo más importante fue definir los procesos involucrados y las entradas y salidas que se obtenían o requerían para estos (figura 5), posteriormente, el enfoque se dirigió hacia una elaboración de procesos que explotaran las estructuras de datos obtenidas de las relaciones y transformaciones entre las entradas y las salidas (figura 6), finalmente, el enfoque dado aquí le da la principal importancia a las estructuras de control que representan a la arquitectura, medio ambiente o flujo de procesos y datos, a partir de la cual emanan las especificaciones y necesidades que deberá cubrir el sistema computarizado.

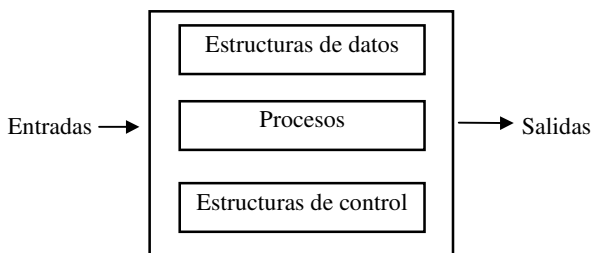


Figura 4. Arquitectura de un Sistema de Información

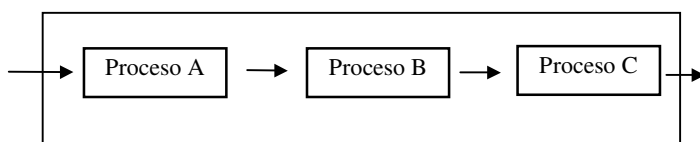


Figura 5. Enfoque inicial del desarrollo de sistemas

La estructura de control a desarrollar tiene la característica de ser independiente de los programas necesarios para emitir los reportes y demás información requerida por los usuarios, así de este manera cuando se deba modificar la representación del sistema, no será necesario modificar los programas que lo interpretan, debido a que estos estarán diseñados de tal forma para que tengan la habilidad de interpretar estructuras de control y flujos de información y no directamente las acciones realizadas en el sistema.

Para lograr lo anterior, es necesario pasar por dos fases de desarrollo, en la primera se dirigen los esfuerzos a representar las operaciones desarrolladas por los usuarios para lograr algún resultado, en una forma tal que no dependa esta de los programas que logran las acciones (figura 7), mientras que en la segunda fase, los esfuerzos se dirigirán a la explotación de los sistemas, mediante el planteamiento de problemas o requerimientos sobre los cuales se deben obtener respuestas concretas (figura 8).

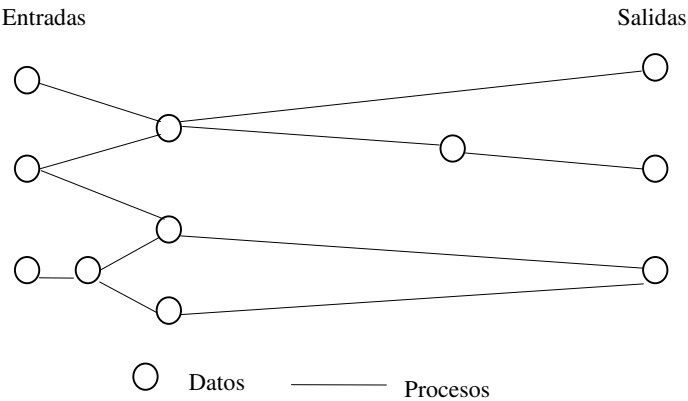


Figura 6. Enfoque actual del desarrollo de sistemas

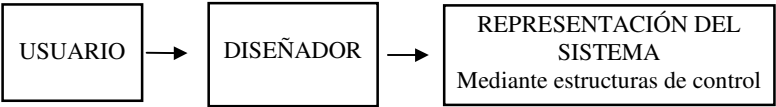


Figura 7. Enfoque propuesto del desarrollo de sistemas

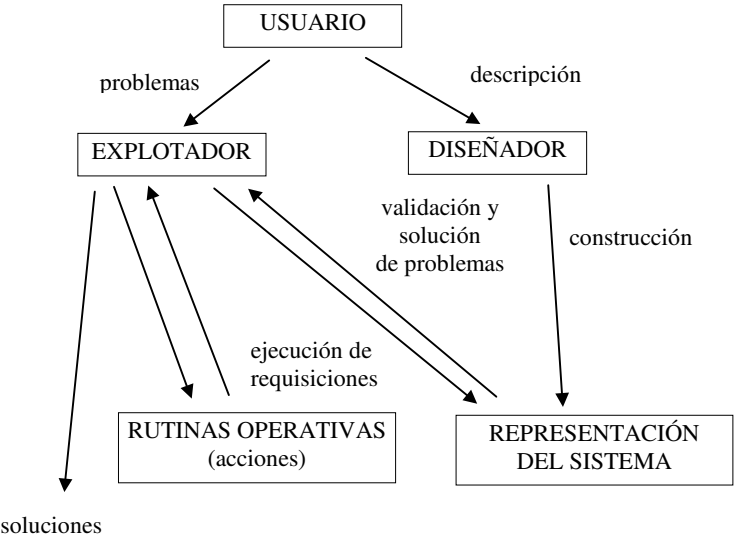


Figura 8. Arquitectura para la definición y explotación de un sistema de información

Tanto el diseñador como el explotador mostrados en las figuras 7 y 8, pueden ser personas o sistemas, dotados con las habilidades necesarias para poder interactuar con el usuario con el fin de obtener una clara definición del sistema a representar y para dar las soluciones adecuadas a los problemas.

En el caso de tener como responsable del Diseño y Explotación a una persona, tendrá que cumplir con el esquema de la figura 9, y en el caso de que el Diseñador sea un sistema, este deberá estar constituido de tal manera para que pueda proponer un modelo en base a un diálogo con el usuario, utilizando la técnica de prototipos, es decir, debido a la imposibilidad de efectuar una observación, el único medio de que dispondría como fuente de información sería el usuario, con lo que se limitaría a una serie de preguntas-respuestas de donde se obtendría una propuesta (prototipo) del modelo de la organización, en base al cual se harían más preguntas-respuestas hasta llegar a la mayor refinación posible, estableciendo así un ciclo de retroalimentación tal como lo muestra la figura 10.

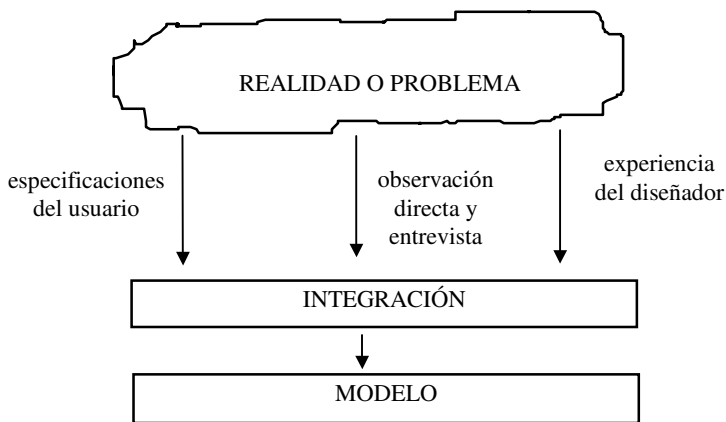


Figura 9. Diseño de sistemas por una persona

El diálogo necesario para que se obtenga el modelo, se puede hacer en base a un cuestionario prefabricado sobre el cual se obtendrían respuestas, seleccionando solo las que sean necesarias hacer.

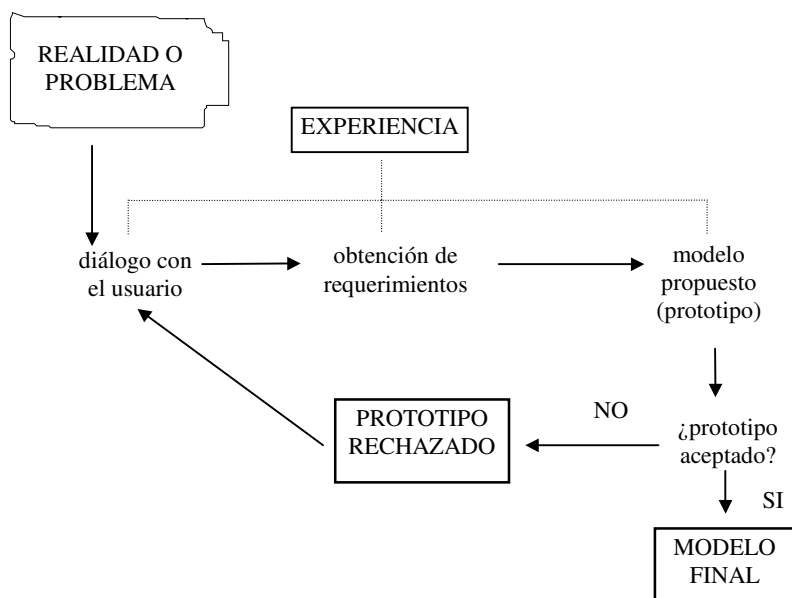


Figura 10. Diseño de sistemas asistido por computadora

En el caso de que se tenga a un sistema para desarrollar la propuesta, deberá tener las herramientas (idealmente) para comunicarse con el usuario en Lenguaje Natural, aunque lo más óptimo sería que este describiera su sistema y/o problema en una forma gráfica sobre una pantalla, utilizando las herramientas del CAD/CAM, esquema a partir del cual se haría la descripción interna de la requisición del usuario.

Después que se ha hecho un análisis de los requerimientos del usuario, en lo que debemos pensar es en la forma de representar internamente esa información de tal manera que refleje la estructura organizacional o la estructura de control del problema, de tal manera que se pueda hacer un programa de computadora que la analice y ejecute las acciones requeridas para dar un resultado de tal manera que si cambiáramos la estructura, el programa no tuviera que modificarse.

La representación de esta estructura se puede llevar a la práctica utilizando diversas herramientas, las cuales estén basadas en el análisis de nodos y líneas de enlace o relaciones entre estos, las cuales denotarían una transformación de un dato o nodo de un elemento en otro. Este tipo de filosofía se puede encontrar en la teoría de las gramáticas, lenguajes formales, bases de conocimiento y otros en los que existan entidades y relaciones entre estas. Para hacer más claros los conceptos aquí manejados, se muestra un ejemplo de su aplicación en las figuras 11 a 13.

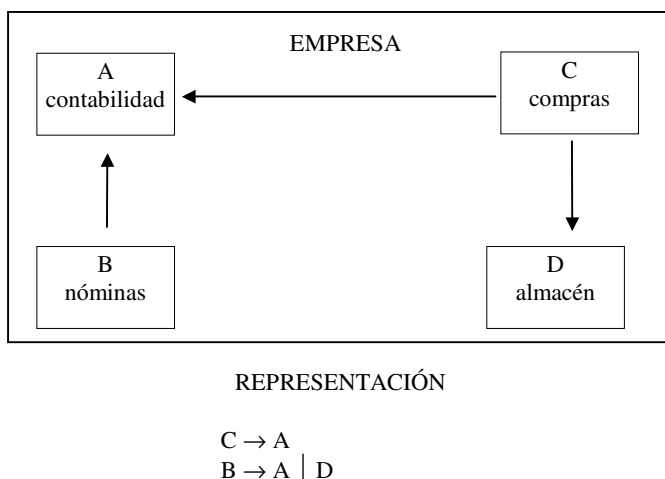


Figura 11. Estructura organizacional y su representación equivalente

CONCLUSIONES

Habiendo analizado brevemente las desventajas que representa el desarrollo tradicional de los sistemas de información, podemos ver que la alternativa presentada aquí, si bien no desarrollada hasta sus últimas consecuencias, ofrece una opción para evitar un gran desperdicio de esfuerzos y recursos por parte de las empresas, que si bien hay algunas que están saliendo adelante a pesar de las crisis por las que hemos atravesado, la mayoría tienen problemas económicos, viendo a los sistemas de computo como algo que les redituara solo gastos o problemas por los grandes volúmenes de recursos a asignarle y que prefieren enfocarlos a otras áreas que les permitan seguir subsistiendo.

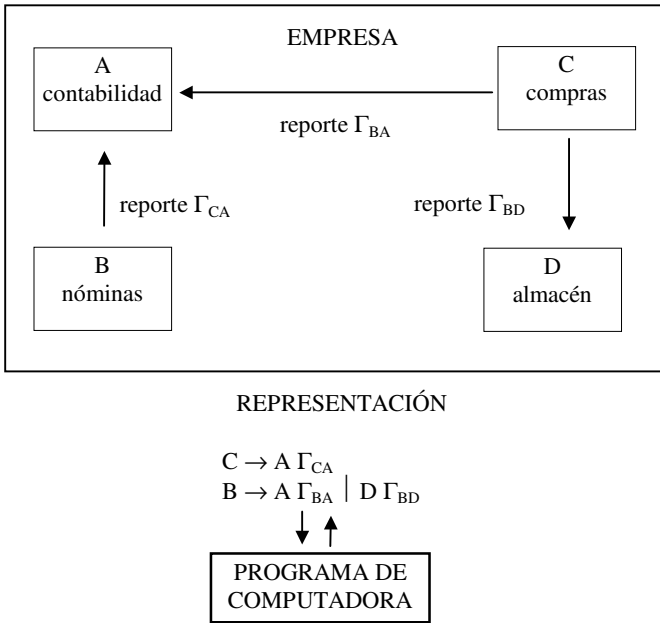


Figura 12. Estructura organizacional con ejecución de acciones

Es obvio además que algunos de los requerimientos planteados como parte del enfoque (p. ej. manejo de lenguaje natural) está siendo apenas desarrollado en México y solo por algunas entidades de investigación, sin embargo se tiene la firme convicción de que los esfuerzos deberían estar encaminados en mayor grado a este tipo de desarrollos y no a seguir desarrollando los sistemas en una forma más eficiente, pero siempre con la misma mentalidad.

REPRESENTACIÓN

$$\begin{array}{l}
 C \rightarrow A \Gamma_{CA} \\
 E \rightarrow A \Gamma_{BA} \\
 B \rightarrow A \Gamma_{BA} \mid E \Gamma_{BE} \mid D \Gamma_{BD}
 \end{array}$$



PROGRAMA DE
COMPUTADORA

Figura 13. Al cambiar las requisiciones del medio ambiente, el programa es el mismo