

Aplicación de un modelo de referencia de aprendizaje en el diseño y construcción de programas de solución de problemas de carácter específico

Implementation of a Reference Model in learning design and construction of programs for solving specific problems

Jorge Aurelio Rodríguez Huerta¹
Jorge Wiener Rodríguez Aliaga,² Paul Karol Zelada Aliaga²

Resumen

El "Modelo de referencia de aprendizaje en el Diseño y Construcción de Programas", constituye un modelo del proceso enseñanza – aprendizaje a tener en consideración en la fase de "programación" en el Ciclo de Desarrollo de Sistemas de Información (tradicional y vigente: Análisis – Diseño – Programación); es decir, construcción de un SOFTWARE de aplicación. El desarrollo de las tecnologías de información, la facilidad de uso y disponibilidad en el mercado de diferentes herramientas de desarrollo (lenguajes de programación), amerita que los procesos de enseñanza – aprendizaje tengan como fundamento y principio el conocimiento, la aplicación de la teoría; es decir, el diseño y aplicación de la teoría con característica fundamental de dosis de innovación, característica base para la generación de nuevos conocimientos.

El modelo que se plantea se sustenta en el conocimiento real y vivencial de los problemas para su solución. Sobre la base de dicho conocimiento se tiene en consideración el uso de las herramientas que proporciona las tecnologías de información y posteriormente su expresión como un modelo para la fase de programación (líneas de código en algún lenguaje de programación) a fin de que cualesquier ordenador (computadora) de la solución respuesta al problema en la oportunidad y exactitud esperada.

Palabras claves

Algoritmo, Análisis, Clases, Construcción, Diagrama de Flujo, Diseño, Hardware, Herramientas de desarrollo, Líneas de código, Métodos, Objetos, Ordenador, Programa, Programación, Seudocódigo, Software, Software de desarrollo

1 Ingeniero Industrial. Magíster en Computación e Informática. Doctorando en Ciencias de la Educación. Segunda Especialidad en Didáctica Universitaria. Docente URP

2 Estudiantes URP

Abstract

The "Reference Model learning in the Design and Construction Programs" is a model of teaching - learning process to take into consideration in the process of "programming" in the Development Cycle Information Systems (traditional and current: Analysis - Design - Programming), ie, construction of a software application. The development of information technologies, ease of use and availability in the market of different development tools (programming languages), warrants that the teaching - learning process have as the basis and principle the knowledge, application of the theory; that is, the design and application of theory with fundamental characteristic of doses of innovation, base feature for the generation of new knowledge.

The model that arises is based on real and experiential knowledge of the problems for solution. Based on that knowledge is taken into consideration the use of the tools provided by information technology and later its expression as a model for the programming phase (lines of code in some programming language) so that any computer (computer) give the solution response to the problem at the time and expected accuracy.

Key words

Algorithm, Analysis, Classes, Construction, Flowchart Design, Hardware, Development Tools, lines of code, methods, objects, computer, software, programming, pseudocode, Software, Software Development

Introducción

En la revista Año IN^o 1 setiembre 2006 PARADIGMAS del Departamento Académico de Ingeniería se publicó el artículo titulado "*Un modelo de cómo aprender a "programar" para la solución de problemas de carácter específico*". El presente trabajo constituye una nueva versión del mismo concepto y finalidad. Los enfoques de aseguramiento de calidad y/o mejora continua son de aplicación en la formulación del presente Artículo.

El desarrollo del presente considera los puntos: Estrategia en el proceso Enseñanza – Aprendizaje, Programación y los lenguajes de programación, Leer y estudiar verbos del conocimiento, Modelo de referencia de aprendizaje en el diseño y construcción de programas, Ejemplo de aplicación y Ejemplo Académico de presentación.

Estrategia en el proceso Enseñanza – Aprendizaje

Las políticas de estado no consideran políticas de innovación, de investigación en aras del crear nuevos conocimientos y/o un benchmarking adecuado y eficiente a las necesidades del desarrollo nacional y/o institucional. El proceso de enseñanza – aprendizaje es la aplicación fría del desarrollo de las tecnologías; y como tal, se convierte en el instrumento de formar profesionales (usuarios expertos) con poca capacidad de investigación; es decir, casi nulo la capacidad de crear, análisis, de análisis crítico; lo cual va significando la despersonalización científico, tecnológico y social de las Instituciones Educativas, fundamentalmente de las Universidades.

El desarrollo de las tecnologías obliga a los nuevos profesionales observar competencias acorde a las tendencias y exigencias del mercado laboral global. El proceso Enseñanza – Aprendizaje, los estamentos mismos que son parte de la formación profesional así como el Estado deberán redefinir estrategias y sentar las bases de una educación que tenga misión – objetivo – meta la innovación, crear conocimiento y su aplicación en pro del desarrollo.

Programación y los lenguajes de programación

Programación es la acción de programar, es construir un programa, un Software; es decir, el diseño y construcción simultánea de líneas de código respetando las reglas y/o sintaxis del lenguaje de programación (LP) bajo uso que de respuesta – solución a los problemas planteados.

Pr ogramación → *Pr ogramar* : *Pr ograma, Código, Software* ← lenguajes de programación

Los lenguajes de programación llamados ahora “herramientas de desarrollo” ó Software de desarrollo y uno de los componentes en el concepto amplio de Tecnologías de Información y de la Comunicación (TIC’s), tienden ha abreviar la capacidad de diagnóstico, análisis y el tiempo de construcción o desarrollo de Sistemas de Información (Software). Las “instrucciones y/o comandos” de los lenguajes de programación encapsulan un conjunto de subprogramas (subinstrucciones, rutinas o subrutinas); lo que podemos definir ahora a los lenguaje de programación como el conjunto de “meta instrucciones”.

La función del programador y/o desarrollador se ve facilitado en la tarea de programación, pero pierde la capacidad de diagnóstico, de análisis; es decir, operar muy mecánicamente las herramientas de desarrollo.

Los lenguajes de programación (LP) conceptualizado por tanto como el conjunto relacional de meta instrucciones, marca del nuevo profesional un nuevo perfil y como tal el proceso enseñanza – aprendizaje deberá responder a tal reto. A nivel de fórmula, un lenguaje de programación y como sinónimo ahora como herramientas de desarrollo estará definido como Meta instrucciones

$$LP \equiv \left\{ \begin{array}{l} \text{Lenguaje de programación} \\ \text{Lenguaje de desarrollo} \end{array} \right\}$$

$$LP \equiv \text{Meta instrucciones}$$

$$LP = f \left(\sum \text{Meta instrucciones}(i) \right)$$

Leer y estudiar verbos del conocimiento

El reto del proceso enseñanza – aprendizaje es marcar en el alumno desde su inicio en asumir el rol / competencia de “analista funcional” y en la formulación de “modelos matemáticos”; así mismo, en el conocimiento de las características, relaciones, implicancias de las tecnologías de información, de los “problemas”, de los problemas que han de tener su solución mediante el uso del computador.

La capacidad de matematizar (diseño de modelos o fórmulas matemáticas) los conceptos y reglas, ayuda al entendimiento de los problemas y ello implica saber leer, saber estudiar. El uso y abuso de la codificación, programación ó el ordenamiento relacional de los meta instrucciones no llevará a la respuesta del “problema” si previamente no se haya entendido el problema, el problema en su real dimensión y contexto; de ahí la siguiente frase:

“El problema del problema, está en entender el problema”

El conocimiento del problema (leer, estudiar) implica identificar y definir alternativas de solución, selección de la alternativa y su expresión como un MODELO de solución del problema (tradicionalmente llamado algoritmo) estará significando la capacidad de solución de problemas utilizando medios tecnológicos; es decir, la computadora.

El no ser solo USUARIOS de la tecnología sino creadores y/o innovadores implica la práctica constante de la lectura y por ende del estudio. La lectura, el estudio, la voluntad y decisión de innovar, de generar nuevos conocimientos nos hará menos dependientes, mas libre de nuestro destino.

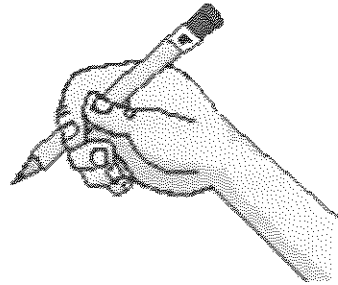
Modelo de referencia de aprendizaje en el diseño y construcción de programas

El proceso enseñanza – aprendizaje en la formación del ingeniero y fundamentalmente en la capacidad de diseño y construcción de PROGRAMAS o SOFTWARE deberá responder en su etapa inicial al reto del entendimiento real del problema; así mismo, a su solución práctica y/o vivencial a la solución mediante el uso de herramientas TIC (Ofimática) para luego diseñar el modelo lógico de solución y finalmente la construcción del programa haciendo uso de algún lenguaje o herramienta (software) de programación.

$$PROBLEMAS(s) \rightarrow (Conocimiento)_{problema} : \{Modelo\ solución\} \rightarrow PROGRAMA$$

El conocimiento del problema plantea como modelo de referencia las fases de:

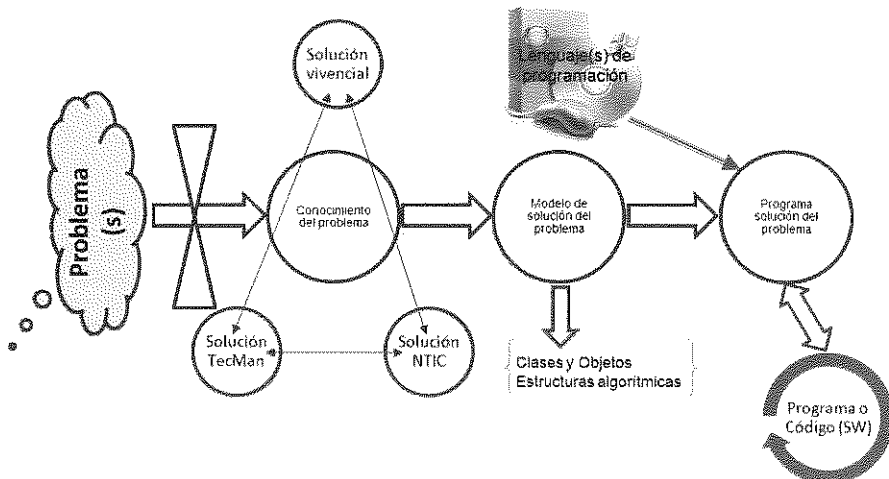
- Modelo solución vivencial (MSVivencial)
- Modelo solución tecnología manual (MStecMan)
- Modelo solución tecnología herramientas TICs – MStecHtic's
- Modelo solución del problema: definición de Estructuras o Sentencias (algoritmo)
- Programa solución del problema: Programa o Líneas de Código



$$PROBLEMA(s) \rightarrow (Conocimiento)_{problema} : \left\{ \begin{array}{l} MSVivencial \\ MStecMan \\ MStechHtic's \end{array} \right\} \Rightarrow \left(\begin{array}{l} Modelo \\ Solución \end{array} \right) \Rightarrow PROGRAMA$$

$$\uparrow Programa = f \left\{ \begin{array}{l} \text{Conocimiento del problema} \\ \left[\begin{array}{l} MSVivencial \\ MStecMan \\ MStechHtic's \end{array} \right] \\ Eficacia(USO)_{PROGRAMACION} \\ Eficacia(USO)_{HERRAMIENTAS} \end{array} \right\} \left\{ \begin{array}{l} \text{Conocimiento del problema} \\ Eficacia(USO)_{PROGRAMACION} \\ Eficacia(USO)_{HERRAMIENTAS} \end{array} \right\}$$

El modelo se expresa mediante la gráfica N° 1:



Gráfica N° 1. Modelo de referencia de aprendizaje en el diseño y construcción de programas
Fuente: Elaboración propia

Ejemplo de aplicación

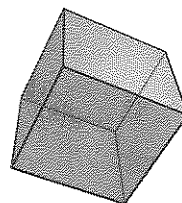
Problema: Construir un programa en Java que calcule el área lateral ó el área total de un cubo de "n" centímetros de arista ("n" dato a ingresar por pantalla – teclado).

Conocimiento del problema:

- Conocimientos previos; los conceptos de los términos: calcular, área, área lateral, área total, cubo, "n", centímetros, arista, lado
- "n" = arista o lado
- Problema: Calcular el área lateral ó el área total de un cubo

Conceptos clave:

- Poliedro regular formado por seis caras cuadradas.
- El cubo es un ortoedro (sus caras son perpendiculares) con todas las aristas iguales.
- El área lateral de un cubo es: $A_l = 4a^2$
- El área total de un cubo de arista a es: $A_t = 6a^2$
- El cubo se llama también hexaedro regular o, simplemente, hexaedro.



Solución TecMan

- Datos: lado = a = L
- Cálculo:

$$AreaLateral = \sum_{i=1}^4 AreaLados(i)$$

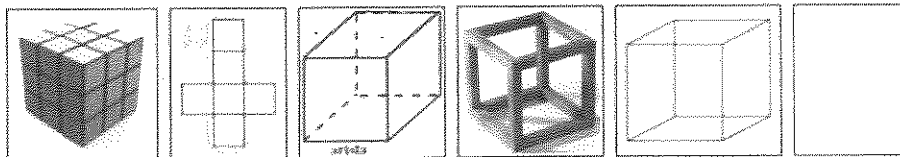
$$AreaLateral = AreaL1 + AreaL2 + AreaL3 + AreaL4$$

$$AreaTotal = \sum_{i=1}^6 AreaLados(i)$$

$$AreaTotal = AreaL1 + AreaL2 + AreaL3 + AreaL4 + AreaL5 + AreaL6$$

Solución vivencial

- Dibujo o construcción de la figura cubo
- Definir una unidad de medida
- Número de lados del cubo
- Calcular el área de cada lado del cubo
- Sumar las áreas de los lados del cubo



Solución Herramientas TIC

- Conocimientos previo; herramientas de productividad; manejo de la Hoja de Cálculo – Excel, Lotus 123, etc.
- Definición de los datos en la hoja de cálculo
- Manejo de CELDAS:

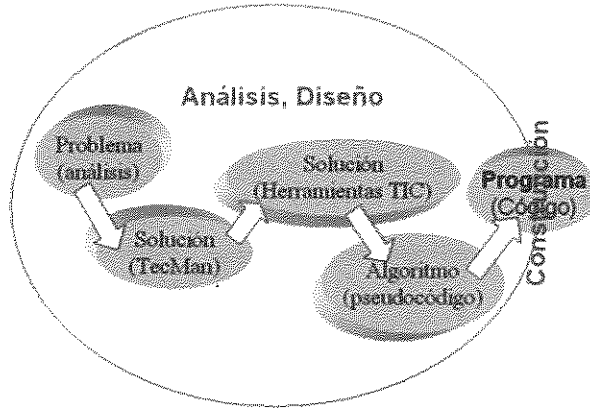
| | A | B | C | D | E | F |
|---|---|---------------------------|--------------|------------------|-------------------------|------------------|
| 1 | | | | | | |
| 2 | | <i>Concepto-variables</i> | <i>Datos</i> | <i>Operación</i> | <i>Unidad de medida</i> | |
| 3 | | | | | | |
| 4 | | Arista: | 5 | dato | cm | datos de entrada |
| 5 | | Número de lados: | 4 | dato | dato - concepto | |
| 6 | | Número de caras: | 6 | dato | dato - concepto | |
| 7 | | Area de cada cara: | 25 | =+c4 * c4 | cm2 | |
| 8 | | Area lateral: | 100 | =+C7 * C5 | cm2 | |
| 9 | | Area Total: | 150 | =+c7 * c6 | cm2 | |

Diseño del Modelo

Definir_Problema → *Análisis_Problema* → *Diseño_Algoritmo* ⇒ *PROGRAMA*

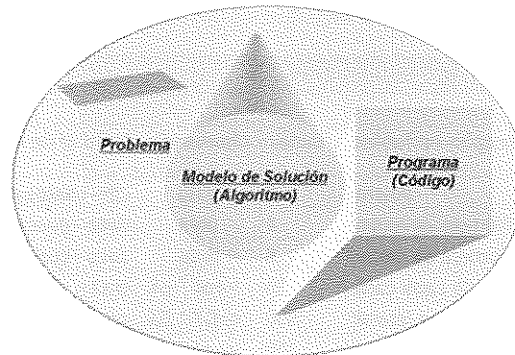
Análisis (problema – QUE) → Diseño (algoritmo – COMO) → programa – HACER

El conocimiento del "problema" permitirá el diseño del MODELO de solución; es decir, la definición e interrelación de los acciones y/o eventos necesarios que permitirá encontrar la respuesta del "problema". Desde luego, el modelo que se plantea constituye una alternativa y es independiente a la herramienta de desarrollo para su implementación.



El proceso enseñanza – aprendizaje deberá centrarse en la definición del MODELO, modelo alternativo de solución del “problema” lo que en el enfoque estructurado se conoce con el nombre de ALGORITMO (pseudocódigo y/o diagramas de flujo)

Construcción del Programa. Ejemplo del código



La expresión del MODELO en base a las reglas y sintaxis de alguna herramienta de desarrollo, constituye el “programa”; es decir, el código como lenguaje intermedio para el entendimiento y ejecución por el ordenador o computadora.

//Definición de variables

```
int l1, opc;// l1 = arista, opc = opción para el cálculo del área lateral o área total
int alateral, atotal;
//
Scanner xLeer = new Scanner(System.in);
//Método leer
public void LeerLado()
{
```



```
System.out.println("Ingresar lado o Arista del Cubo");
l1 = xleer.nextInt();
System.out.println("Ingresar la opcion: 1 = Area lateral, 2 = Area Total");
opc = xleer.nextInt();
}
//Método escoger la opción de cálculo, operación y mostrar
//Condional doble (if - else)
public void CalcularMostrar()
{
    if(opc == 1)
    {
        alateral = 4 * (l1 * l1);
        System.out.println("El area lateral del cubo es: " + alateral);
    }
    else
    {
        atotal = 5 * (l1 * l1);
        System.out.println("El area total del cubo es: " + atotal);
    }
}
```

```
//Método escoger la opción de cálculo, operación y mostrar
//Condional doble (switch() - case)
```

```
public void CalcularMostrarx()
{
    switch(opc)
    {
        case 1:
        {
            alateral = 4 * (l1 * l1);
            System.out.println("El area lateral del cubo es: " + alateral);
            break;
        }
        case 2:
        {
            atotal = 5 * (l1 * l1);
            System.out.println("El area total del cubo es: " + atotal);
            break;
        }
        default:
            System.out.println("Opcion digitada fuera de rango ");
    }
}
```

Modelo académico de presentación

| | | | |
|--|---------------------|----------------------------|-----------------|
| Enunciado del problema: Construir un programa en Java que calcule el área lateral o el área total de un cubo de "n" centímetros de arista ("n" dato a ingresar por pantalla – teclado). | | | |
| Diagrama de Flujo | Pseudocódigo | Variables - Métodos | Programa |
| | | | |

CONCLUSIONES

El desarrollo de las Tecnologías de Información, el modelo mental del desarrollador (alumno – programador) definen su estrategia y/o estilo en la fase de "programación" de considerar el ciclo de desarrollo de Sistemas de Información (Software).

La presencia y participación del docente es la piedra angular en los primeros peldaños en el proceso de entendimiento y aprendizaje. El docente con el enfoque dinámico de vivir, el entendimiento y solución de los problemas mediante medios computacionales incrementa la eficacia en el diseño y construcción de programas o líneas de código que permiten solución de problemas.

El modelo como propuesta está sujeto a observaciones de crítica, mejora y los aportes deberán orientarse a cómo enseñar y aprender mejor a "programar" y por ende ser más competitivos en el diseño y construcción de sistemas de información o Software.

REFERENCIAS BIBLIOGRÁFICAS

- SOTO MEDRANO, VLADIMIRO A. (2003). Organizadores del Conocimiento y su Importancia en el aprendizaje. R & A – Razuwillka Editores. Huancayo – Perú.
- SARAVIA CANALES, LUIS MIGUEL (2004). Una Didáctica para la Pedagogía Histórico-Crítica: Un Enfoque vigotzquiano. Ediciones Fargraf S.R.L. Lima – Perú.
- FLOREZ OCHOA, RAFAEL (1994). Hacia una Pedagogía del Conocimiento. McGraw-Hill INTERAMERICANAS S.A. Colombia.
- RODRÍGUEZ HUERTA, JORGE A. Apuntes de clase (Taller Básico de Programación – URP)
- RODRÍGUEZ HUERTA, JORGE A. Guías de laboratorio (Taller Básico de Programación – URP)