

## **METODOLOGÍAS PARA EL DESARROLLO DE SOFTWARE EDUCATIVO**

Jorge Calderón (jorgelcs@ula.ve), William Díaz, Zulix Angulo, Neila Márquez

La construcción de un sistema computacional o software implica la toma de decisiones sobre la arquitectura del sistema (definir los componentes del sistema de software y sus interacciones). Estas decisiones pueden ser cruciales para el éxito o fracaso del sistema resultante, por lo que se requiere seleccionar un proceso de desarrollo de software con el fin de obtener la calidad del sistema de software deseada y cumplir con los requerimientos establecidos.

Para hacer más probable que un determinado producto sea de buena calidad, se han desarrollado diversas metodologías de ingeniería de software que atienden muy bien estos requerimientos y permiten al equipo multidisciplinario encargado de dicha labor, asumir con propiedad sus funciones.

Diversos autores han utilizado la ingeniería de software para la elaboración de material multimedia interactivo, logrando de esta manera que el proceso de desarrollo y mantenimiento del software educativo sea una actividad que dependa de pautas establecidas, con modelos conceptuales y herramientas de trabajo, y no del arte de aquellos que tengan la experiencia exclusivamente.

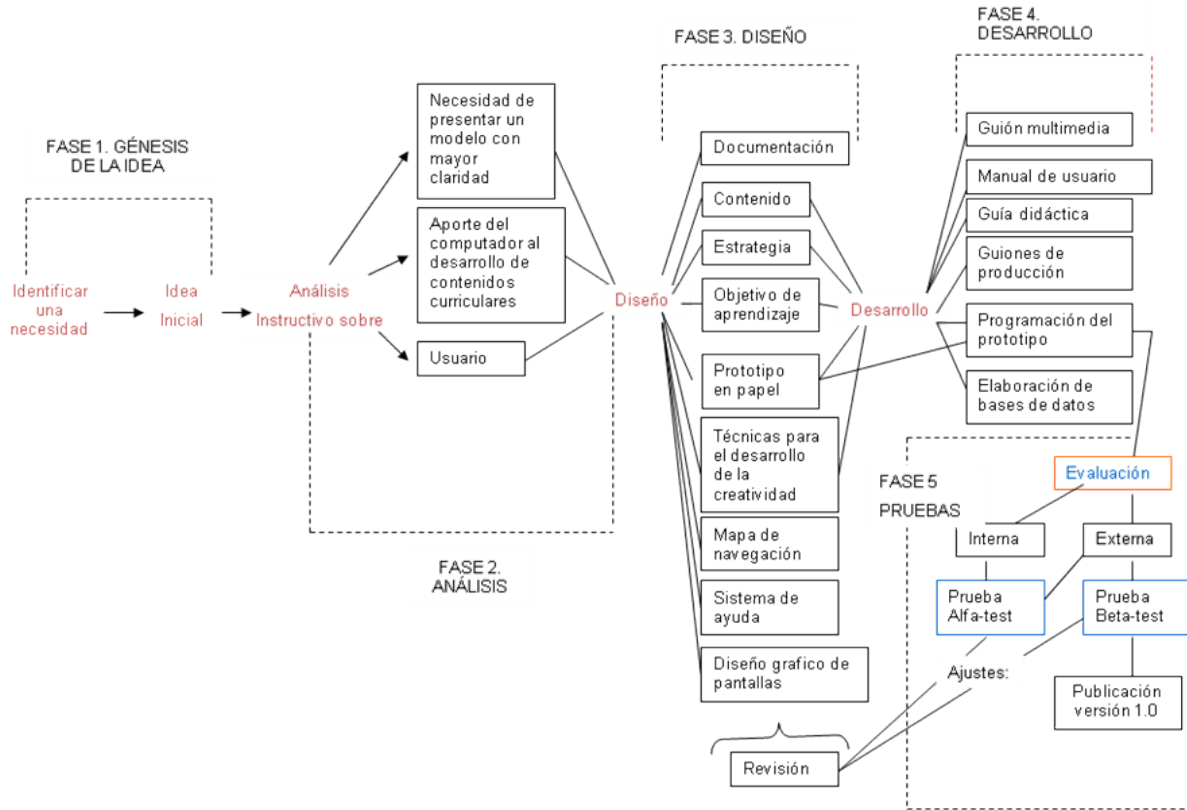
Se sabe además, que para lograr software educativo con las condiciones deseadas, se deben incorporar dentro de las fases de análisis y diseño, aspectos didácticos y pedagógicos, es decir, el diseño instruccional, a fin de que faciliten y garanticen la satisfacción de las necesidades educativas del usuario al cual va dirigido el software. Se deben involucrar también a estos usuarios, para conseguir identificar necesidades y problemas puntuales, para luego establecer mecanismos de resolución adecuados y apoyar cada una de las fases en sólidos principios educativos, comunicativos y computacionales (Galvis 2000).

Las diferentes metodologías conservan los grandes pasos o etapas de un proceso sistemático para desarrollo de software: análisis, diseño, desarrollo, implementación y evaluación (ADDIE). Sin embargo, hacen – unas en mayor y otras en menor medida - especial énfasis a los siguientes aspectos: la solidez del análisis, como punto de partida; el dominio de teorías sustantivas sobre el aprendizaje y la comunicación humanas, como fundamento para el diseño de los ambientes educativos computarizados; la evaluación (bajo criterios predefinidos) a lo largo de todas las etapas del proceso, la documentación adecuada y suficiente de lo que se realiza en cada etapa, como base para el mantenimiento que requerirá el material a lo largo de su vida útil.

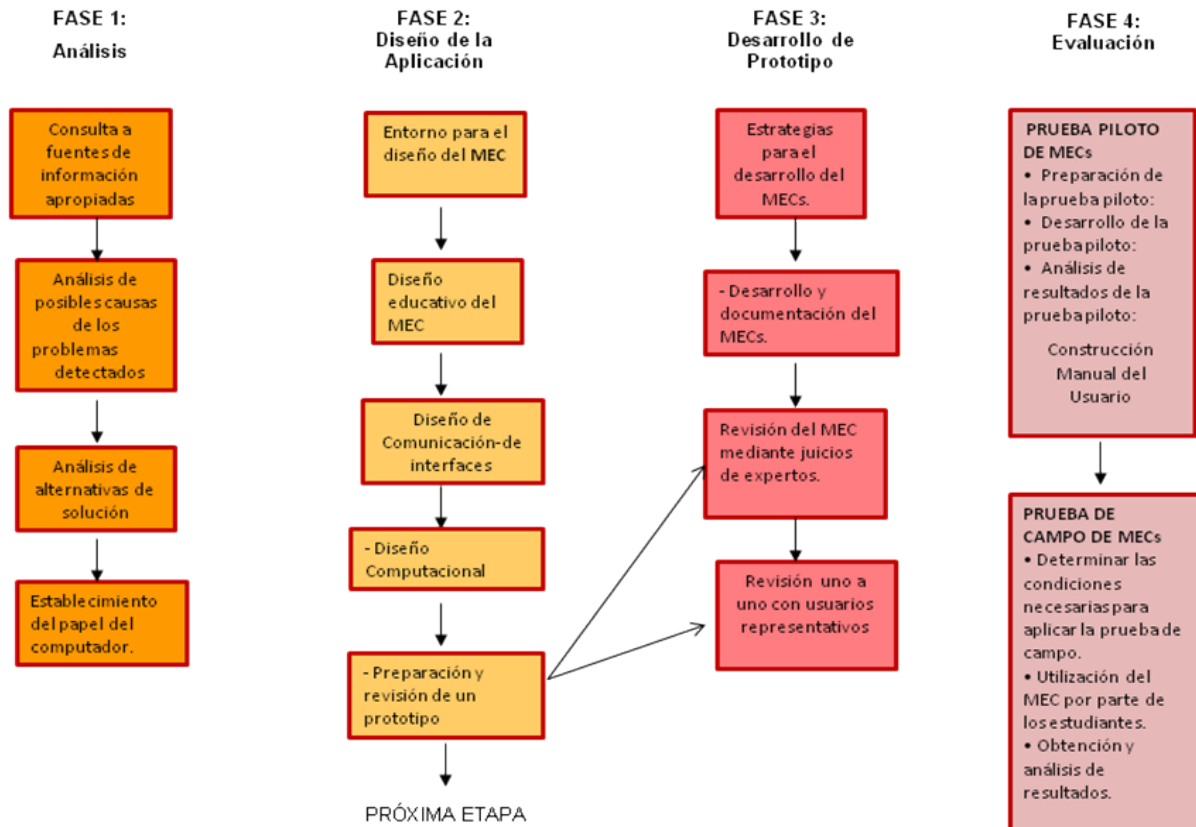
En fin, cada autor intenta conciliar una perspectiva computacional, que garantice un producto de calidad desde ese punto de vista, con la perspectiva de los aspectos pedagógicos para las buenas prácticas educativas.

A continuación se presentan, de forma esquemática, cuatro de las metodologías para el desarrollo de software educativo tratadas en los cursos Desarrollo y Evaluación de Software Educativo I y II por la V Cohorte de MEIDI.

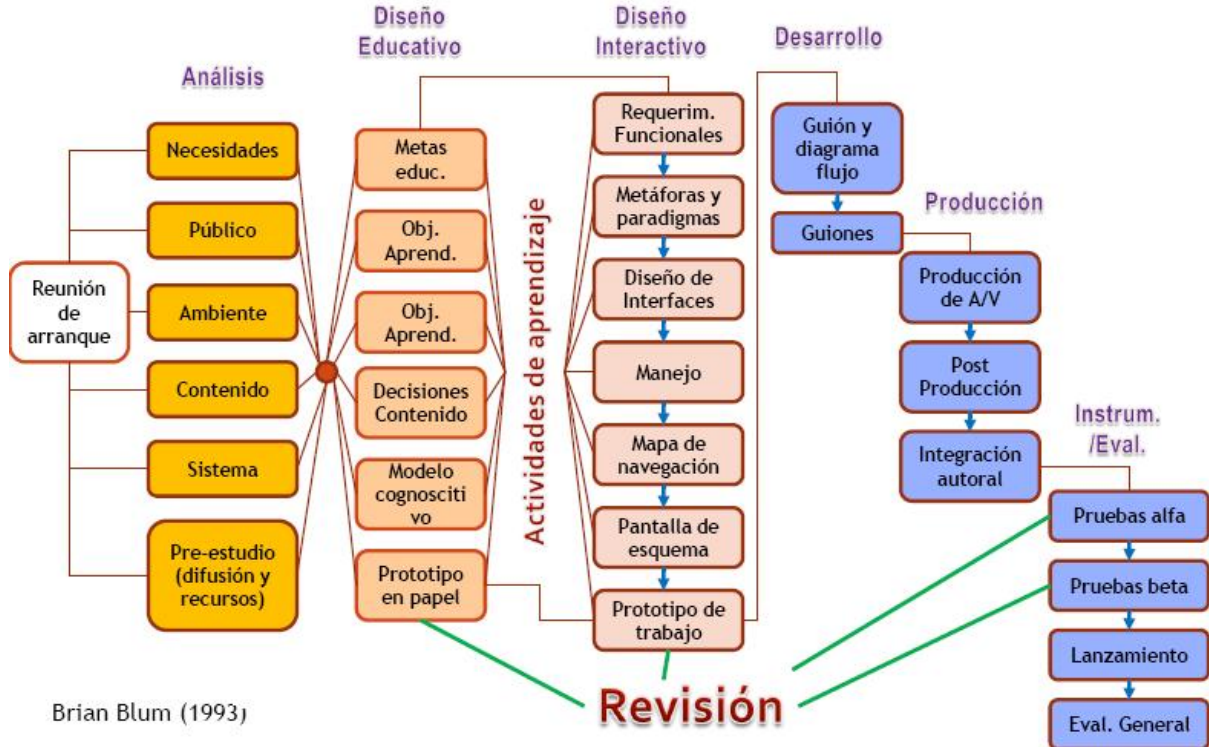
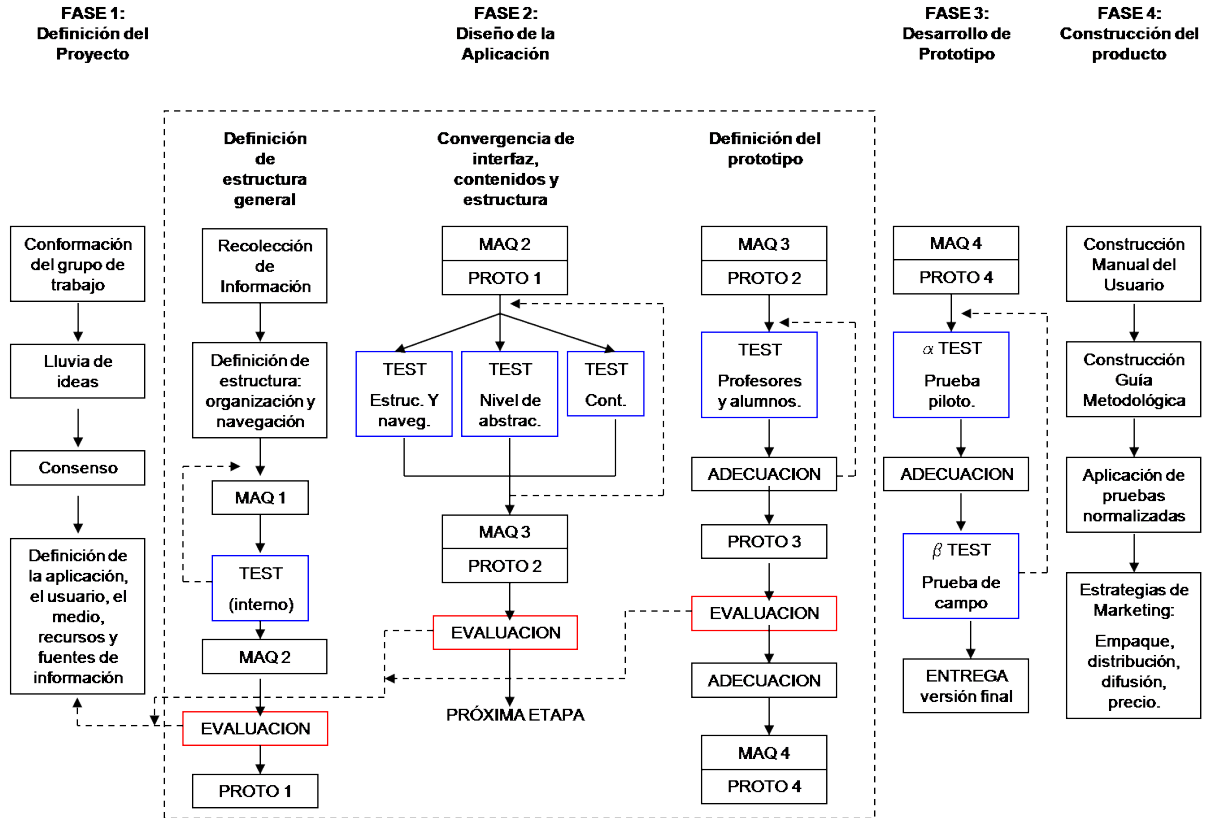
## METODOLOGIA PERE MARQUÉS



## METODOLOGIA ALVARO GALVIS



## METODOLOGIA JAIME SANCHEZ



## COMPARACIÓN DE LAS METODOLOGÍAS

<b>AUTOR</b>	<b>FORTALEZAS</b>	<b>DEBILIDADES</b>
<b>PERE MARQUÉS</b>	<ul style="list-style-type: none"> <li>• Ofrece pautas para elaboración de evaluaciones, pruebas piloto y pruebas de campo.</li> <li>• El análisis es profundo en cada uno de los aspectos del producto: curricular, instruccional, necesidades de usuarios e instituciones, entre otros.</li> <li>• Propone la incorporación de grupos de trabajo multidisciplinario para el desarrollo del proyecto, incluyendo al cliente.</li> <li>• Prevé documentación del proceso en cada una de sus fases. Contempla la ficha resumen que recoge las principales características del programa, el manual de usuario y una guía didáctica.</li> </ul>	<ul style="list-style-type: none"> <li>• Las fases de su metodología de desarrollo de software educativo se entremezclan con las fases de diseño instruccional del material.</li> <li>• Al tener un nivel de detalle muy alto, se tiende a confundir las actividades presentes en cada fase, ocasionando dificultades para su aplicación</li> <li>• Peso muy alto en el aspecto instruccional y de contenido, disminuyendo la importancia de los aspectos computacionales, lo cual podría generar productos de baja calidad de acuerdo con estándares internacionales.</li> <li>• No prevé evaluaciones al final de cada etapa del desarrollo.</li> </ul>
<b>ÁLVARO GALVIS</b>	<ul style="list-style-type: none"> <li>• Posee un fuerte basamento empírico (experiencias previas).</li> <li>• Propone la incorporación de grupos de trabajo multidisciplinario para el desarrollo del proyecto</li> <li>• Fácil aplicación y desarrollo de la Metodología de trabajo</li> <li>• Ofrece pautas precisas para elaboración de evaluaciones por expertos, pruebas piloto y pruebas de campo.</li> </ul>	<ul style="list-style-type: none"> <li>• Desde la perspectiva computacional no ha avanzado, por lo cual es necesario enriquecerla tomando en cuenta los avances tecnológicos en el diseño y desarrollo computacional que se han logrado en los últimos años.</li> </ul>
<b>JAIME SÁNCHEZ</b>	<ul style="list-style-type: none"> <li>• Posee un fuerte basamento empírico (experiencias previas).</li> <li>• Cada fase incorpora dimensiones de desarrollo (gráfica, arquitectura, contenidos, funcionabilidad)</li> <li>• Incorpora evaluaciones tempranas al proyecto favoreciendo la conveniencia y factibilidad del producto</li> <li>• Emplea técnicas de representación (maquetas, bocetos y prototipos) disminuyendo los costos del proyecto</li> <li>• Propone la incorporación de grupos de trabajo multidisciplinario para el desarrollo del proyecto</li> </ul>	<ul style="list-style-type: none"> <li>• Riesgo de generar retrasos en el proceso, dada la gran cantidad de actividades y la obligación de lograr consensos entre los integrantes del grupo de desarrollo.</li> <li>• No ofrece pautas precisas para elaboración de evaluaciones, pruebas piloto y pruebas de campo. Sólo señala como está compuesto el grupo evaluador y qué aspectos se evalúan.</li> <li>• Falta de detalle en la fase de análisis, ya que deja muchos de los aspectos a criterio del grupo de desarrollo.</li> </ul>

	<ul style="list-style-type: none"> <li>• Fácil aplicación y desarrollo de la Metodología de trabajo</li> <li>• Emplea herramientas de documentación, control y organización en cada una de las fases del proceso.</li> <li>• Separación de roles</li> </ul>	
<b>BRIAN BLUM</b>	<ul style="list-style-type: none"> <li>• Las actividades de aprendizaje constituyen una relación conexas entre el diseño educativo y el diseño interactivo.</li> <li>• El análisis es profundo en cada uno de los aspectos del producto</li> <li>• Realización de prototipos en papel.</li> <li>• Emplea herramientas de documentación, control y organización en cada una de las fases del proceso</li> </ul>	<ul style="list-style-type: none"> <li>• No se contempla una secuencia estricta para la realización de las actividades que constituye cada una de las fases de análisis y diseño.</li> <li>• No prevé evaluaciones al final de la fase de análisis.</li> <li>• Se limita al uso del modelo de aprendizaje cognoscitivo para el diseño de los materiales.</li> <li>• No se especifica el modelo de prueba final (usuarios, aspectos a evaluar, etc.)</li> </ul>

### **Fuentes consultadas:**

GALVIS, A., (2000) *Ingeniería de software educativo*. 2da. reimpresión. Universidad de Los Andes. Ediciones UNIANDES. Colombia.

MARQUÈS GRAELLS, Pere (1995). *Software Educativo: guía de uso, metodología de diseño*. Barcelona: Editorial ESTEL.

SÁNCHEZ, Jaime. (1993). *Informática Educativa*. Santiago de Chile: Ed. Universitaria.

<http://www.javeriana.edu.co/cursos/ntae/metodologia.htm>