

# Desarrollo de Software, entre contrastes y debates

*Juan David Chicangana Muñoz*  
*Tecnólogo en Desarrollo de Software*

No es ninguna novedad encontrarnos en un mundo donde la información se ha convertido en un soporte más de la sociedad, desde las costumbres hasta la economía, la información se considera como un patrimonio y la herramienta de primera mano para desempeñar las tareas cotidianas. Si vemos una película de ciencia ficción de hace veinte años, podemos observar “modernos” laboratorios con computadoras gigantescas con luces brillantes manipuladas por palancas mecánicas y botones luminosos.

Suena un poco gracioso observar que ahora esas máquinas existen, pero no ocupan ni la centésima parte de lo que ocupaban en las películas que se rodaron hace veinte años y que la fuerza que se debería emplear en una palanca, ahora se emplea en dar un clic, obteniendo iguales o mejores resultados de los que imaginaron los visionarios de la época.

El desarrollo de Software se ha constituido en una ciencia que ha ido madurando para ofrecer sistemas informáticos que garanticen la disponibilidad, integridad, consistencia y validez de la información que a diario manejamos en un mundo, llamémoslo “Real”.

En el transcurso de los años y gracias a la aparición de nuevas necesidades

relacionadas con el manejo de la información, han ido naciendo y evolucionando técnicas y metodologías en la tarea del desarrollo de Software, estableciéndose a su vez nuevos estándares y patrones para la industria. Esta es la realidad del software, una revolución en el conocimiento que ha transformado las costumbres de nuestra sociedad actual. Cuando hablamos de él, inherentemente encontramos el factor tiempo de por medio, ya que cuando se ofrecen los sistemas informáticos, se necesita que el tiempo de desarrollo sea lo más corto posible.

Debido a que es mayor su demanda que su oferta, han nacido paradigmas y se han creado técnicas con el fin de crear productos software en menor tiempo, de buena calidad y que ofrezcan una arquitectura que soporte el hostil y cambiante ambiente del entorno, la sensación del ingeniero de software cuando entrega su producto se puede comparar a la de un ingeniero civil cuando expone su puente recién construido a los elementos de la naturaleza.

Los tecnicismos y las retóricas se hacen cada vez más densos en la jerga de los ingenieros, muchos conceptos se mezclan para formar un mar de conocimientos, y es casi natural que se adopten realidades con mitos incluidos en los procesos de desarrollo de sistemas informáticos.

Todo empieza desde abajo. Los programas de computadora en su más baja expresión son algoritmos que se traducen en sentencias que dan órdenes a una máquina de propósito general, para no complicarnos, vamos a llamarla computadora. La naturaleza de las órdenes que se le dan a estas computadoras depende del propósito del sistema. Estos aspectos oscuros de la ingeniería del software no son de importancia para los usuarios finales, pero realmente son la realidad diaria para los desarrolladores de software.

Observamos en esta situación dos mundos opuestos y muchas veces excluyentes. Mientras el cliente habla de “Crear Usuarios para el sistema”, los ingenieros piensan en “Insertar un registro en la tabla Usuarios de la base de datos”. Existen paradigmas que los desarrolladores adoptan para disminuir el impacto de este “choque” de mundos, uno de los más utilizados es la Tecnología Orientada a Objetos, la cual modela al mundo “Real” (por lo menos el mundo que corresponde al dominio del problema) en Objetos que se relacionan e interactúan entre ellos, para dar solución a un problema o necesidad planteada.



Este paradigma ayuda a que el problema pueda ser analizado e interpretado por los ingenieros que hablan la lengua del mundo técnico y produzcan la solución en el mundo técnico para el real.

La teoría de los paradigmas se puede conocer, aunque muchas veces no se aplique de la forma correcta. En el mundo real se puede observar que muchos de los conceptos que ofrece el paradigma son imprácticos para ocasiones en que el tiempo es un factor determinante, sumado a la presión por la apariencia de las interfaces de usuario de las aplicaciones por parte del cliente (aspecto concerniente al diseño gráfico y técnicas de usabilidad que también abarca la ingeniería del Software), ésta es una de las etapas que más tiempo demanda.

Son muchos los comentarios que se escuchan acerca de que las teorías y los patrones que se aprenden en la carrera de ingeniería del software son meramente académicos y que del dicho al hecho hay una diferencia muy grande, incluso muchos de estos comentarios provienen de algunas de las empresas desarrolladoras.

Con la evolución de las herramientas de desarrollo y los lenguajes de programación, cada vez más complejos al mismo tiempo que completos, estos comentarios se hacen más fuertes debido a que el tiempo y esfuerzo de aprendizaje de dichas herramientas se hace más grande cada vez, capturando más la atención de los desarrolladores, los cuales suelen adoptar un actitud de afecto hacia los productos que le sirven como herramienta de desarrollo y disminuyen su interés de un enfoque hacia un proceso metodológico.

Este afecto que se adopta por las herramientas de desarrollo incurre en la investigación a fondo de los aspectos técnicos y oscuros para el cliente por parte del desarrollador, debilitando la concepción del paradigma de desarrollo, haciéndolo impráctico y dándole la razón a los comentarios que atacan a un desarrollo de software metodológico y ordenado (ahora, además de que el desarrollador va a pensar en ingresar un registro en la base de datos, va a pensar en la forma en que el lenguaje de programación le va a ayudar a llevar a cabo su tarea).

Una vez más, la solución se convirtió en foco de debate, entre los que apoyan un proceso de desarrollo de software guiado por una metodología y una forma de percibir el mundo del dominio del problema y los que guían su

proceso de desarrollo en el manejo de sus habilidades y herramientas técnicas con una planeación limitada a lo que la herramienta les pueda dar.

Por muy robusta y confiable que sea la plataforma o tecnología de programación, incluso el sistema operativo sobre el que trabaje un equipo de desarrollo de software, un buen diseño y arquitectura flexible estará en función de la percepción que se haga del mundo del dominio del problema, y los fundamentos para lograr esta meta los proporciona un proceso guiado por una metodología que tenga un paradigma para ofrecer.

La ingeniería del software no se limita a programadores escribiendo líneas de código frente a un computador, involucra también un estudio y análisis de un problema, un diseño de una o varias soluciones (incluyendo la escogencia de la mejor) y un estudio y diseño de estrategias para la elaboración de pruebas tanto de verificación como de validación del producto.

Finalmente el proceso de desarrollo no termina cuando se hace entrega del software al cliente, puesto que el ciclo de vida continúa con los diversos tipos de mantenimiento que requieren los sistemas informáticos.