

## 29/04/09 - De Programador a Desarrollador

Cuando tenía doce años, comencé a programar en Basic, en un ZX Spectrum de 128K... en esos momentos, y los años siguientes, hacer un programa, para mi, era sentarme delante de mi ordenador, durante unas cuantas horas, incluso echando noches en vela, para al final ver el resultado... un pequeño programa de gestión de contactos, un juego de puzzle, tetris, plataformas 2D... y eso, a lo largo de los años... y de los lenguajes, pasando por Basic, C, C++, Modula-2, Pascal, Java, Perl, PHP, Python, Ruby, Erlang...

La programación se convierte, en mi vida, en mi tiempo libre y, después, en mi trabajo, en una forma de realizar programas de ideas que me surgen, de forma heroica, y sin una fuerte organización que me permita ver el resultado de mi esfuerzo, ni las horas que voy a invertir, ni si merece la pena hacerlo de una cierta forma u otra.

Como programador, disfruto de la programación sin orden, anárquica, programo lo que quiero, como quiero y todo el tiempo que quiero. Me encuentro con los problemas típicos de haber abordado un problema del modo equivocado, y tener que reescribir una y otra vez el mismo código para alcanzar una forma óptima de lograrlo... pero no es óptimo y muchas veces aburre, ¿cuál es el siguiente paso?

Desarrollar es tener una idea en mente, darle forma hasta conseguir una idea completa y organizada, o al menos creíble, de un producto software conseguible mediante algún camino que se pueda tomar. Es decir, que si programo el código de una cierta forma, conseguiré lo que me propongo, ya que, mediante un diseño previo, he podido analizar las posibilidades de que no me ocurra el encontrarme ante un callejón sin salida.

Pero no es solo saber de antemano donde nos metemos al comenzar a programar, sino también, poder distribuir el trabajo en un equipo de desarrollo, un equipo de programadores. El análisis del problema nos permite ver qué hay que desarrollar exactamente, detecta las necesidades para que, después, el diseño nos diga el cómo podemos abarcar el problema para su resolución. Y ese cómo incluye la división del trabajo.

Como desarrollador, y viendo la programación como una parte del conjunto de hacer un proyecto de software, he podido aprender que, para poder realizar de forma más óptima un proyecto, hay dos caminos:

- **predictivo / por procesos:** cuando se ha desarrollado ya bastante software del mismo tipo, es fácil poder estimar y cerrar requisitos, es fácil tener una especificación del problema a resolver, realizar el análisis, el diseño y la programación del mismo. Es fácil predecir cuánto tiempo se puede tardar en cada fase y realizarlo sin problemas. En este sentido, es bastante fácil tomar un modelo de procesos como el CMMI, SPICE o Metrica-3, y seguirlo para conseguir nuestro objetivo, consiguiendo además, la máxima documentación útil sobre el proyecto.
- **ágil / por iteraciones:** si el proyecto que se quiere conseguir es sobre una tecnología que no se domina, plataforma nueva, o requisitos cambiantes, lo mejor es emplear una metodología ágil, que nos permita agregar flexibilidad para el cambio de requisitos, mediante entregas cortas, periódicas y agregando valor cada vez al proyecto a desarrollar, mediante la continua liberación de versiones. Las metodologías más usadas son Scrum, XP, Lean, FDD, DSDM, Crystal, Kanban, ...

En conclusión, el siguiente paso ha sido convertirme en desarrollador. Conocedor de las metodologías predictivas y ágiles y poder dar a cada proyecto la orientación necesaria para poder llevarlo a cabo de la mejor forma posible.

No hay entradas relacionadas