

10 de marzo de 2017

Bogotá D.C.



William Sierra                      william.sierra@exsis.com.co

Carlos Cardona                    carlos.cardona@exsis.com.co

Ervid Molina                        ervid.molina@exsis.com.co

### **MENTORÍA, APRENDIZAJE Y ARTESANÍA**

Constantemente las empresas son decepcionadas por la calidad de graduados que recibe en ciencias de la computación. No es que los graduados no sean brillantes o talentosos, es solo que no se les ha enseñado lo que es el desarrollo en realidad.

#### **GRADOS DE FRACASO**

Es posible obtener una educación excelente en una universidad. Pero también es posible moverse a través del sistema para salir con varios diplomas que certifican su título. Otro gran problema que tienen las universidades, es que los programas difícilmente están enfocados a lo que realmente se necesita en la industria; ya que, casi siempre lo que se ha aprendido en el sistema educativo y lo que se encuentra en el campo profesional son cosas muy diferentes.

#### **MENTORING**

¿Cómo aprendemos a programar?

Una forma de aprendizaje es mediante la observación y el acercamiento a los problemas. Otra puede ser la tutoría de una persona que le muestre el camino, por medio de un manual. Aunque hay otras no tan convencionales como ayudar a un vecino, depurar un programa con un amigo, entre otros.

Todo programador que pretenda convertirse en un desarrollador profesional necesita de un tutor, maestro o mentor, alguien que actúe como un modelo a seguir y le enseñe valores propios; alguien de quien aprender mientras le ayuda a solucionar pequeñas tareas.

#### **APRENDIZAJE**

¿Qué hacen los médicos? ¿Crees que los hospitales contratan a médicos recién graduados y los lanzan a las salas de operaciones para realizar una cirugía a corazón abierto en su primer día de trabajo? Por supuesto que no.

Esta profesión ha desarrollado una disciplina de mentoría intensa. Al graduarse y antes de ser reconocidos como médicos profesionales, deben realizar un año de práctica supervisada (año rural). Con esto se busca capacitar al médico de forma intensa en el puesto de trabajo y estar rodeado de médicos más calificados para seguir sus recomendaciones.

Además, cualquier especialidad médica, requiere de 3 a 5 años más de práctica supervisada (llamada residencia), donde se le asigna responsabilidades cada vez mayores al médico que le ayudan a ganar confianza, rodeado y supervisado por médicos con mayor experiencia. Finalmente, pueden tomar sus exámenes finales y ser evaluados por la junta médica para ser reconocidos públicamente.

Este proceso enfocado a la industria del desarrollo no es ni de cerca semejante, ya que las empresas pierden enormes cantidades de dinero por la inadecuada formación que traen sus desarrolladores de software. De alguna manera la industria del desarrollo de software ha concebido la idea de que los desarrolladores saben programar perfectamente, y que una vez graduados pueden codificar.

De hecho, no es nada raro que las empresas contraten jóvenes recién graduados, los incluyan en "equipos" y les pidan que construyan sistemas más críticos. ¡Es una locura! Los pintores no hacen esto, ni los plomeros, ni los electricistas. Las empresas deberían invertir más en la capacitación de sus nuevos desarrolladores a comparación de los que McDonald's invierte en sus servidores a nivel mundial.

También se sabe que en la actualidad muchas de las cosas que existen están controladas por software, teniendo en cuenta que se le confía a los desarrolladores muchos aspectos de nuestras vidas, que interactúan con la tecnología (software bancario, de vehículos, de telefonía, etc.). Por tanto, es prudente otorgarles a los nuevos desarrolladores un periodo de tiempo considerable para la capacitación y práctica supervisada, con el fin de reforzar su confianza e implementar el conocimiento que la industria necesita poner en producción.

Entonces, ¿Cómo debe la industria de software inducir a los jóvenes graduados en las filas del profesionalismo? ¿Qué pasos deben seguir? ¿Qué desafíos deben cumplir? ¿Qué metas deben alcanzar?

- **Maestros**

Estos son los desarrolladores que han liderado más de un proyecto de software importante. Suelen tener más de 10 años de experiencia y habrán trabajado en muchos tipos diferentes de sistemas, lenguajes y sistemas operativos. Saben cómo dirigir y coordinar múltiples equipos, son eficientes diseñadores y arquitectos y pueden codificar círculos alrededor de todos los demás naturalmente.

Les han ofrecido puestos administrativos; pero los han rechazado, han vuelto a su rol original después de aceptar el cargo o decidieron integrarlo a su rol técnico. Ellos mantienen ese papel técnico al leer, estudiar, practicar, hacer y enseñar. Gracias a este perfil, las empresas les asignan responsabilidades técnicas dentro de un proyecto.

- **“JourneyMen”**

Estos son desarrolladores que están capacitados, son competentes y enérgicos. Durante este periodo de su carrera, aprenderán a trabajar bien dentro de un equipo, y se convertirán en líderes de equipo. Están bien informados sobre las tecnologías actuales, pero por lo general carecen de experiencia con diversos sistemas.

Tienden a tener definida una metodología de trabajo (lenguajes, plataformas y Sistemas Operativos) son supervisados por maestros, u otros desarrolladores con mayor experiencia. Los jóvenes “JourneyMen” rara vez tienen autonomía. Su trabajo es supervisado de cerca. Su código es

escudriñado. A medida que ganan experiencia, su autonomía crece. La supervisión se hace menos directa y más matizada.

- **Aprendices / pasantes**

Los graduados comienzan sus carreras como aprendices. Los aprendices no tienen autonomía. Son supervisados muy de cerca por los “JourneyMen”. Al principio no toman ninguna tarea en absoluto, simplemente proporcionan ayuda a los “JourneyMen”. Este debería ser un tiempo de programación en pares muy intensa. Aquí es cuando las disciplinas se aprenden, los principios se refuerzan y los principios éticos se fundan

Los “JourneyMen” asumen un rol docente. Se aseguran de que los aprendices conocen principios de diseño, patrones de diseño, disciplinas y rituales. Los “JourneyMen” enseñan TDD, refactorización, estimación, etc. Asignan lecturas, ejercicios y prácticas a los aprendices. Ellos revisan su progreso.

El aprendizaje debe durar mínimo un año. En ese momento, si los “JourneyMen” están dispuestos a aceptar al aprendiz en sus filas, harán una recomendación a los jefes. Los maestros deben examinar al aprendiz tanto por entrevista como por el mérito e impacto de sus logros. Si los maestros están de acuerdo, entonces el aprendiz se convierte en “JourneyMen”.

Ése es el derecho de las cosas. Los graduados son supervisados por líderes de equipo jóvenes, que a su vez son supervisados por líderes de proyecto, y así sucesivamente. ¡El problema es que, en la mayoría de los casos, esta supervisión no es técnica! En la mayoría de las empresas no hay supervisión técnica en absoluto. En la mayoría de los casos los desarrolladores son promovidos a un nuevo puesto con un nuevo salario porque, bueno, esa es la realidad.

La diferencia entre lo que ocurre hoy y el programa idealizado de aprendizaje es el énfasis en la enseñanza técnica, la capacitación, la supervisión y la revisión.

La diferencia radica en la noción misma de que los valores profesionales y la perspicacia técnica debe culturizarse siendo enseñados, nutridos y protegidos. Lo que tiene al enfoque actual estéril es la falta de responsabilidad por parte de los desarrolladores experimentados transmitir su conocimiento y experiencia a los más jóvenes.

Ahora, realice la comparación entre los oficios de un desarrollador y la de un artesano. El artesano practica la artesanía, la cual requiere habilidad y calidad. Evoca experiencia y competencia. Además, trabaja rápidamente, sin apresurarse, basado en razones estimadas para siempre cumplir sus promesas. Además, sabe cuándo decir no y cuándo decir sí. Por tanto, un artesano es considerado profesional en su oficio.

¿Y que se requiere para llegar a ese punto? Construir esa mentalidad, a partir de valores, principios, técnicas, actitudes y respuestas. ¿Y cómo un desarrollador logra apropiarse del concepto de “artesanía”? Recibiéndolo de otra persona: Es enseñado por los ancianos a los jóvenes, intercambiando entre pares, observando y aprendiendo como los ancianos analizan a los jóvenes. La artesanía es un contagio, una especie de virus mental. Lo atrapan observando a otros y permitiendo que se apodere de su mente.

Es muy complicado convencer a la gente de ser un “artesano”. No puede convencerlos de que acepten la mentalidad de la artesanía: los argumentos son ineficientes, los datos irrelevantes y los casos de estudio no les significa nada. La aceptación de una mentalidad no es tanto una decisión racional, sino emocional (algo muy humano).

Entonces, ¿cómo puede lograr que la gente adopte la mentalidad de la artesanía? Recuerde que es una mentalidad contagiosa, pero sólo si se puede observar. Así que usted hace la mentalidad observable. Usted actúa como un modelo a seguir, se convierte en un artesano primero y demuestra la artesanía. Finalmente, la mentalidad hace el resto del trabajo.

## **CONCLUSIÓN**

Las Universidades pueden enseñar la teoría de la programación y la informática. Pero no pueden enseñar la disciplina, la práctica y la habilidad para ser “artesano”. Esas cosas son adquiridas a través de años de trabajo personal y tutoría. Es tiempo para que los desarrolladores inmersos en la industria de software, hagan frente al hecho de guiar a la próxima generación de desarrolladores de software a la madurez. Es hora de adoptar un programa de aprendizaje, práctica y orientación profesional a largo plazo.