ReVisión
Revista de Investigación en Docencia Universitaria de la Informática
AENUI Asociación de Enseñantes Universitarios de la Informática

Docencia 2.0

Juan Julián Merelo, Fernando Tricas

El miedo a fallar

En general, en la enseñanza universitaria de la programación y tareas aledañas, como administración de sistemas, se enseña a acertar: a que tu código compile correctamente, pase todos los *tests* y finalmente haga lo que tiene que hacer. Los contenidos, ejercicios y prácticas se encaminan a que no haya ningún fallo en ese proceso.

Pero en ingeniería las cosas no siempre van como a uno le gustaría que fueran. En cualquier paso en el proceso de creación de un sistema informático, un espacio mal colocado o un punto y coma omitido y el ordenador expulsa una parrafada, muchas veces en inglés, expresando su impotencia y su falta de comprensión por lo que el ser humano está intentando hacer. Y el problema es que, cada vez más, el ser humano matriculado en un grado de ingeniería, ante esa parrafada, sólo sabe expresar su impotencia y su falta de comprensión al profesor de teoría o prácticas más cercano.

«*Use the force, Luke*» decía el maestro al aprendiz en aquella saga que se ha puesto de moda otra vez durante las últimas semanas. Y a pesar de que también aquél otro decía aquello de «*The force is strong with this one!*», lo cierto es que el protagonista no dejaba de utilizar por ello las herramientas que tenía disponibles: la espada láser, las naves y los robots, por ejemplo.

¡Incluso se entrenaba para aprender a manejarlas adecuadadamente!

Así que será nuestra tarea intentar convencer a nuestros estudiantes del valor de los mensajes de error: nuestro compilador favorito (o para el caso, de alguien) nos lanza mensajes y notificaciones que varían mucho en calidad y claridad cuando hemos cometido algún errorcillo. En muchos casos nos ayudarán a hacernos una idea de lo que está pasando y en otros nos darán por lo menos alguna pista de por dónde tenemos que empezar a mirar.

Por eso, pedimos a los jóvenes *padawanes* que practiquen las siguientes virtudes teologales de un desarrollador cuando aparecen los errores:

- Fe: el compilador sabe (o, más bien, no sabe otra cosa). Si se queja es que algo hemos hecho mal (no encuentra lo que esperaba encontrar). Y nos avisa.
- Esperanza: todo tiene remedio. Revisemos la línea donde nos señalan, desde el principio de la lista de errores, y sus alrededores. A veces ayuda dejarlo un rato y volver más tarde. No ayuda volver a hacer exactamente lo mismo a ver si pillamos despistado al compilador y se le olvida darnos el mismo error.
- Caridad: siempre encontraremos quien nos ayude. O a quién ayudar. El código fuente tiene la rara *virtud* de confundirnos cuando lo miramos fijamente. Nuevas miradas y ojos nuevos son inmunes a esta confusión, a veces.

Y que, una vez practicadas y alcanzado el zen con ellas

JJ Merelo es catedrático de Universidad en el área de Arquitectura y Tecnología de Computadores, y actualmente director de la Oficina de Software Libre de la UGR. Mantiene un blog desde el año 2002, y lo ha utilizado en clase desde el año 2004; también wikis y, ultimamente, agregadores y otras herramientas TIC. Últimamente le ha dado por el *flipped learning*, de lo que se informará debidamente en esta columna.

Fernando Tricas García es profesor titular de Lenguajes y Sistemas Informáticos del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza. Empezó a estudiar la blogosfera casi cuando aún no existía (allá por el año 2002) y a tratar de integrarla en los cursos y tareas docentes un poco después. Ha impartido numerosas charlas relacionadas con el tema de la Web 2.0. Es actualmente Director de su departamento.

(«*The force awakens*»), se pueden complementar con las virtudes cardinales.

- Prudencia. Una copia de seguridad, o un "commit" a tiempo y una nueva rama, para probar cambios sin destrozar ninguna funcionalidad. A ver si arreglando algo se nos va a estropear lo demás.

- Justicia. El compilador no tiene la culpa. El computador tampoco, ni el que nos encargó la práctica. Especialmente el que nos encarga la práctica.

- Fortaleza. El que resiste gana. Y las herramientas son muy robustas, pero… ¿Dejaremos que nos venzan unas pocas líneas de código?

- Templanza. Añadir otra opción a nuestro programa no lo hace mejor. Incluir las últimas características de la última versión del lenguaje, tampoco. El objetivo es resolver la funcionalidad que sea necesaria y, de paso, aprender cosas interesantes. A veces tenemos la tentación de modificar código, añadir líneas y duplicar funciones para resolver errores.

Lo importante es hacerlo. Como dijo Yoda, «*Do. Or do not. There is no try.*» Si lo intentas y no funciona, vuelve a intentarlo hasta que funcione.

Para terminar, «*And may the Force be with you!*». La fuerza es, sin duda, StackOverflow, donde preguntar y contestar ayudará a tu karma. No hay duda que StackOverflow no resuelva, si ya ha fallado el grupo de WhatsApp de clase, el profesor de prácticas que sabe siempre más que el de teoría y el que sacó más de un ocho el año anterior en la misma asignatura. Raro es el error que no le ha ocurrido a alguien, en algún lugar, alguna vez. Y esa persona, muy probablemente, lo haya resuelto y lo haya puesto en algún lado. Simplemente busca y encontrarás, joven padawan.

Todas las columnas de la serie Docencia 2.0 pueden descargarse en formato LaTeX desde `https://github.com/ReVision-Docencia-20/Columnas`