



Artículo invitado

Todavía no sabemos enseñar programación

Agustín Cernuda del Río
guti@agustincernuda.info

Resumen

A lo largo de años de profesión, un docente (y sobre todo un docente interesado en la mejora de la docencia) va conociendo y compartiendo técnicas nuevas. Va precisando el propósito y el contexto en el que funcionan mejor las viejas. Va enriqueciendo y diversificando su repertorio de instrumentos para facilitar la formación de sus alumnos o hacerla más significativa. En la última década, la docencia en Informática ha evolucionado mucho; muchas áreas de la formación que hace tiempo quedaban totalmente descuidadas están hoy en día asumidas e integradas en las clases, muchas prácticas que parecían extravagantes o arriesgadas son hoy de uso corriente. Muchos debates se han ido superando. Sin embargo, después de un viaje apasionante conociendo mundo, en algunos aspectos da la impresión de que seguimos como al principio.

Palabras clave: Programación, fundamentos, métodos docentes

Recientemente he releído algunos clásicos de Isaac Asimov, escritos en los años 1950. Asimov anticipaba un futuro cientos, miles de años posterior. En él había avances tecnológicos impensables hoy en día: planetas colonizados, robots pensantes, análisis de ondas cerebrales, viajes interestelares. Y al lado... tarjetas perforadas, necesidad de libros físicos o grabaciones sonoras para cargar información en ordenadores, comunicaciones entre ordenadores muy trabajosas y mediante voz hablada, necesidad de buscar «el intercomunicador más cercano» para poder hablar con otra persona. Está claro que Asimov, aun con toda su formación y capacidad, no podía entender del todo cómo funcionarían los ordenadores unos pocos años después (no se le puede culpar de ello, ni mucho menos).

Cuando elegí la Informática, los ordenadores no estaban en las casas. Eran aparatos fascinantes, y apenas conseguía entender su propósito y posibilidades; creo que menos aún que Asimov. Pero sabía que quería dedicarme a ellos. Poco antes de empezar a estudiar, tuve uno de aquellos primeros ordenadores domésticos, y empecé a hacer programas muy sencillos. Cuando llegué a la Universidad, ignoraba casi todo lo que se puede ignorar sobre programación (*arrays*, ficheros, funciones, algoritmos) pero aun así las materias de programación (tan terribles, tan monstruosas) fueron coser y cantar.

Aun siendo estudiante, me asombraba aquel fenómeno. Yo había entendido solamente unas pocas cosas, pero eran su-

ficientes para que el resto fuera un simple corolario. Podía dedicarme por entero a absorber conocimientos nuevos. Sin embargo, mis compañeros se atascaban una y otra vez en la misma zona; aquella que yo ya había dejado atrás en casa. Intentaba explicárselo a mis compañeros, pero a diferencia de lo que ocurría en otras materias, por alguna razón no conseguía hacerme entender.

Creo que todo el que estudia programación conoce el fenómeno y seguramente ha pasado por esa misma situación. Cuesta un tiempo entender los programas más simples: la iteración, los condicionales, o el simple intercambio de valor entre dos variables. Con un poco de suerte, un buen día algo hace *clic* y de repente todo empieza a cobrar sentido. El alumno por fin es capaz, efectivamente, de ponerse en el lugar del ordenador y saber qué ocurrirá con el programa que ha escrito. No significa que no haya trabajo que hacer, pero en algún momento se supera una etapa básica de comprensión, en la que se abre el camino para empezar a avanzar.

Ese clic que ya me resultaba fascinante como alumno, me siguió interesando como profesional, y no digamos como docente. Estoy seguro de que los resultados académicos de miles de alumnos mejorarían si fuésemos capaces de transmitir esa habilidad elemental. Pero muchos alumnos llegan a ello (si es que llegan) por un camino largo, aparentemente ineficiente. ¿Realmente no hay una buena forma de contarlos?

Muchos años después, he ido recibiendo muchas horas de formación como profesor, he acumulado experiencia, he interactuado con los mejores en el campo de la docencia, he investigado. Y aquel problema que me sorprendía como estudiante hablando con mis compañeros sigue en pie, inatacable. Después de leer cientos de artículos e intentar muchos enfoques, he encontrado muchas cosas que funcionan bien, que mejoran diversos aspectos de la docencia. Pero no he encontrado ninguna manera de transmitir un conocimiento tan sencillo, ese algo fundamental que subyace en la programación. Se ha hablado de potenciar las habilidades básicas de resolución de problemas, y algo de eso hay; pero, sinceramente, por prometedora que sea la idea no veo que a partir de ella estemos mucho más cerca de entender y dominar el *clic*.

En esa búsqueda del santo grial sigo igual que el primer día. Intentamos organizar actividades que pongan al alumno en situación de encontrar ese mecanismo, pero a partir de ahí básicamente esperamos a que lo encuentre por sí mismo.

No sé por qué es tan difícil enseñar algo tan sencillo. Y creo que la música puede enseñarnos algo al respecto (no la solución, lamentablemente).

Estudiar música, más allá de ser un placer, es verdaderamente interesante para alguien que ama la docencia. Saber hacer música es, en muchos sentidos, como saber hablar; quien sabe hacerlo lo encuentra (relativamente) fácil, quien está aprendiendo lo encuentra increíblemente difícil y desalentador. A la vez, la teoría que hay detrás de la música es una construcción muy formal, sistemática. La relación entre la música y las matemáticas es un lugar común, y no creo necesario extenderme en ello.

El caso es que, pese a esa formalidad, en todos los momentos del aprendizaje musical uno se encuentra con enfoques docentes prácticamente personales. La nomenclatura y la notación son, a veces, un tanto particulares, y a un científico le cuesta entender cómo es posible, cómo se permite tal cosa en una disciplina tan asentada y con una tradición (los conservatorios) tan estricta. Pero eso es solo un síntoma; la diferencia significativa viene cuando se examinan los métodos de enseñanza musical. Para llegar al mismo resultado, cada profesor y cada músico usan un abordaje muy distinto.

Si no entiendes un resultado de armonía, o si pretendes aprender a hacer algo concreto con tu instrumento, tu profesor te sugerirá que utilices determinado recurso. Un amigo te dirá que él lo piensa de otra forma. Y un tercero te dará una ruta distinta. Parecerá que quieren volverte loco; y al final elegirás «lo que a ti te funciona». Con el tiempo lo habrás mecanizado, ya no tendrás que pensar, y será menos relevante cómo llegaste allí. Pero tiempo después, cuando alguien te pregunte, les darás tu receta particular.

Cuando empecé a estudiar música me parecía incomprendible que hubiera tal variedad de recetas. No es raro que los elementos de la música estén conectados entre sí por infinidad de relaciones de diversos tipos; ocurre en cualquier sistema coherente derivado a partir de unas reglas sencillas. En matemáticas suele haber muchas formas diferentes de demostrar lo mismo. Pero después de siglos enseñando música, ¿es que

no se ha asentado un plan de estudios, unas pautas, una hoja de ruta, que permita a alguien explorar toda esa nube de relaciones de una forma ordenada, prevista y eficaz?

Parece ser que no. Hay tantos caminos porque todos los caminos son, de alguna manera, deficientes (con d). Pero todos ellos, con más o menos rapidez, y si el alumno practica miles de horas, producirán un músico solvente, que domina el lenguaje y ha encontrado sus soluciones (un artista es otra cosa, evidentemente, y queda fuera de este ensayo).

En ese sentido, y si estoy en lo cierto, todos los profesores de música son profesores *amateurs*, todos ellos colocan al alumno ante un programa de actividades de aprendizaje... y es el cerebro del alumno el que tiene que hacer clic. Exactamente como hacemos los profesores de programación.

¿Por qué el proceso es tan poco sistemático en la música? No sé si los años de historia van añadiendo confusión, pero hay al menos otro factor inevitable y, quizás (lo sé), decepcionantemente obvio: porque los alumnos son diferentes. Han llegado aquí por diferentes caminos; han entendido unos conceptos mejor que otros; han trabajado más unas cosas que otras. Y fuera del conservatorio su vida ha discurrido de manera diferente. Y cuando afrontan un problema nuevo, recurren a metáforas, a comparaciones y elaboran sobre lo que ya saben. Y en ese preciso momento unos caminos les funcionan y otros no.

Frente a esta variabilidad individual, las taxonomías, los estilos de aprendizaje o cualquier otra simplificación me parecen casi inútiles. Que un alumno sea más visual o más auditivo o que responda a diferentes perfiles de personalidad para trabajo en grupo resulta muy interesante y permite, claro está, incorporar mejoras en la docencia, algunas de ellas de gran importancia. Pero si vamos a algo tan concreto como el clic, si realmente pretendemos transmitir un conocimiento difícil, hay que mirar esa mente mucho más de cerca. En música, para saber cuál es el acorde dominante de uno dado, unos alumnos recurren al ciclo de quintas; otros, al ciclo de cuartas (si por puro azar se estudiaron más intensamente uno de los dos); otros han practicado mucho los intervalos y no necesitan pensar en los ciclos; un guitarrista puede que tienda a recordar visualmente su instrumento, si aprendió a tocarlo antes de aprender la teoría que hay detrás... No tengo ninguna noción de cómo se toca la trompeta, así que soy incapaz de suponer si los trompetistas abordan esto (mentalmente) de otra manera.

El lector puede estar casi enfadado en este punto. Que todos los alumnos son distintos es una verdadera perogrullada. Pero creo que merece la pena pensar en esto con más calma.

Todos los alumnos son distintos, claro, pero hay muchos conocimientos que podemos transmitir (o cuyo hallazgo podemos provocar, si se me permite la expresión) de manera bastante homogénea. Cuando un alumno ha hecho el avance básico en programación no cuesta demasiado trabajar ciertos temas. Hay un entendimiento mutuo, unas rutinas mejor establecidas. No parece haber barreras infranqueables; que siga el itinerario planteado depende en gran medida de sus circunstancias y del trabajo que pueda dedicarle, pero no tanto de su

planteamiento mental de partida. En circunstancias ideales en las que siguiera el itinerario propuesto, es muy probable que consiguiera los objetivos.

Sin embargo, los fundamentos de la programación son una barrera peculiar. Afecta a alumnos más o menos brillantes, más o menos trabajadores. Algunos la superan dedicando mucho tiempo, otros poco. Un profesor piensa muchas veces que si supiera cómo explicar exactamente determinada cosa a determinado alumno, este la entendería al instante. Esto es indemostrable, pero siempre he tenido esa sensación, de forma muy acusada, cuando intentaba aclarar un fundamento de programación a alguien; la sensación no era tan apremiante explicando otras cosas.

Además, intuyo que resulta crítico elegir el camino adecuado a la primera. Que si uno va probando diversas recetas, cuanto más tiempo pase y más se acumulen las recetas fallidas, mayor es la confusión en el alumno. Que si uno fuese capaz de elegir desde el principio la receta adecuada para ese alumno, con toda claridad, sin contaminaciones, daría el paso de manera firme.

Todo esto es muy vago y ocurre en cualquier tipo de enseñanza, lo sé. Pero es muy acusado en el caso de la programación. Quizás es un sentimiento personal, pero enseñando programación básica he sentido una frustración muy especial, muy concreta, muy propia de esa materia. Es una cima sorprendentemente difícil de alcanzar para lo cerca que está. Tiene algo y creo sinceramente que no tenemos idea de qué es.

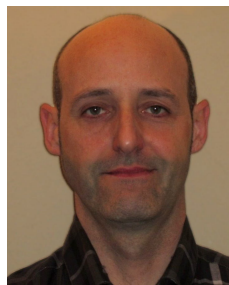
El caso es que después de años de ejercicio docente he conseguido superar diversos retos. He conseguido motivar a alumnos aparentemente inmotivables. He conseguido ayudar a alumnos con determinadas barreras o discapacidades (barreras o discapacidades que, más que nada, ponían de manifiesto las mías propias). Pero hay dos objetivos que nunca he conseguido alcanzar, y que creo que son quizás los «teoremas de Fermat» de nuestra docencia.

Uno es conseguir que un alumno de proyecto final de carrera planifique, maneje la planificación y la cumpla. Creo que esto es factible, pero yo no lo he conseguido nunca.

Y el otro es, pues, tener un papel decisivo como profesor al explicar los fundamentos de programación. He conseguido motivar, allanar el camino, ofrecer salidas. Pero a la hora de la verdad, no he podido hacer otra cosa que esperar a que el

alumno resolviera la situación.

Siendo positivos, que un alumno resuelva sus propios problemas no parece mala idea, por otra parte. Y seguir fascinado años después por un problema que no has conseguido resolver... tampoco. Pero me encantaría ver un día publicado algo que arroje luz sobre este asunto.



Dr. Agustín Cernuda del Río (Mieres, 1970). Primer Ingeniero en Informática por la Universidad de Oviedo (1993), y Doctor por la misma Universidad (2002). De 1994 a 2000 trabajó como técnico de sistemas/investigador en el departamento de I+D de Seresco S.A., encuadrado en su división de desarrollo de software.

Desde 2000 hasta 2015 fue profesor asociado en el Departamento de Informática de la Universidad de Oviedo, tarea que desde 2005 simultaneó con la de jefe de informática en el Parlamento asturiano, que ejerce en la actualidad. Como profesor ha impartido docencia, entre otras, en asignaturas como Metodología de la Programación o Servicios Web. Es miembro desde su fundación del Grupo de Estudio para la Innovación Docente en Informática (GEIDI), trabajando en la innovación de la docencia de asignaturas informáticas.

Ha ocupado diversos cargos de gestión en la Escuela de Ingeniería Técnica en Informática de Oviedo, de la que ha sido Subdirector y Secretario, y también ejerció como Vicedecano en el Colegio Oficial de Ingenieros en Informática del Principado de Asturias.



2017 A. Cernuda del Río. Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional que permite copiar, distribuir y comunicar públicamente la obra en cualquier medio, sólido o electrónico, siempre que se acrediten a los autores y fuentes originales y no se haga un uso comercial.