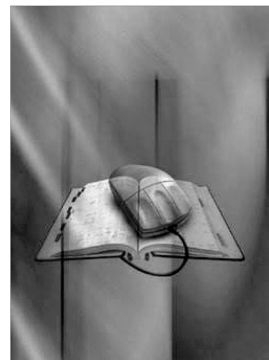


Ciencia y Luz



Universidad Veracruzana
Dirección General de Difusión Cultural
Dirección de Comunicación de la Ciencia

La poesía al servicio de la programación de



computadoras

WULFRANO ARTURO LUNA RAMÍREZ *

En literatura, y en particular en poesía, no hay países atrasados o en vías de desarrollo. Una mejor manera de decir lo anterior, una manera poética, la encontramos en el poema o mejor dicho poemínimo *Ay Poeta* como lo nombró su creador, el escritor mexicano Efraín Huerta: *Primero / Que nada: / Me complace / Enormísimamente / Ser / Un buen / Poeta / De segunda / Del Tercer / Mundo.*

La creación poética, expresión motivada desde la reflexión y el sentimiento, evitando el debate sobre si la poesía tiene más de uno que de otro, engendra un proceso cognitivo común a todas las personas. En el esfuerzo poético se dan cita abstracciones, ideas, concepciones y circunstancias, entre otras cosas, abordadas en extenso en los tratados de poesía y los debates literarios. Dichas abstracciones obtienen su concreción en palabras, frases, versos y poemas. Esto manifiesta un paralelismo con la actividad de programación de computadoras (en lenguajes de alto nivel), orientada a resolver un problema, partiendo de su entendimiento y posterior traducción a una representación rigurosa que evita ambigüedades, misma que finalmente será convertida en una representación ejecutable en una máquina, es decir, un programa de computadora. La generación y uso de las abstracciones y metáforas propios de la creación poética, funciona también como entrenamiento para desarrollar la capacidad de resolver problemas mediante algoritmos y programas de computadora (sin soslayar las indispensables herramientas otorgadas por el pensamiento lógico-matemático).

Obviamente, no hay que caer en el exceso de comparar directamente poemas con programas. Éstos son distintos en tanto que en los programas el sentido de funcionalidad y desempeño son cruciales, existen métricas y métodos de evaluación objetiva; mientras que en los poemas (y las artes en general) la valoración está dada en buena medida por el sentido estético y la comunicación lograda con el lector, de un carácter eminentemente más subjetivo (hasta donde sé, no existe una metodología de evaluación de poesía que trascienda la valoración de los críticos literarios y los colegas y el grado de aceptación entre los lectores).

Algunos autores ya se han ocupado del uso del quehacer artístico o literario como herramienta didáctica para generación de conocimiento, tal es el caso de las obras *Poesía para aprender a aprender* de José Negrete, *What has literature to offer Computer Science* de Mark Dougherty, o *Hackers and Painters* de Paul Graham. Además de algunas iniciativas que ya las vinculan en la práctica, en particular la Competencia de Poesía en Código Fuente (<http://www.sourcecodepoetry.com/>). Amparada en uno de sus lemas "Los verdaderos programadores pueden codificar poesía" (Real programmers can code poetry), se enfoca a generar código en un lenguaje de programación de alto nivel de corte imperativo y

orientado a objetos (Java, C++ y C#), que rimen, tengan significado y sean legibles. Cabe señalar, que en nuestro caso, se impone la barrera del idioma dado que la gran mayoría de los lenguajes de programación de alto nivel usan una base de palabras reservadas en inglés.

Sin embargo, este esfuerzo de programación poética o poesía programable/programada/codificada (aún no se establece un nombre), no incluye la resolución de problemas, es decir, deja fuera el objetivo principal de un programa. Si bien pudiera argüirse que el problema es justamente la creación de un poema en código fuente, los programas resultantes no son aplicables a algún dominio en particular, además, excluyen también a los lenguajes inspirados por otros paradigmas —funcionales, lógicos, etcétera—, como puede verse en los ejemplos al final de este artículo. Ya sea que se trate de un poema de autoría propia o no, estamos hablando de esfuerzos lúdicos al alcance de quienes ya saben programar y conocen bien el lenguaje a utilizar, y el desafío se pue-

de complicar aún más, preguntémonos por ejemplo, ¿cómo poner en términos de un programa algún haikú o algún poemínimo de Huerta?

Las iniciativas de poesía codificada o programación poética tienen una utilidad no sólo lúdica, sino eminentemente pedagógica y didáctica que puede y debe ser explotada en la formación de profesionales de la computación. Concretamente, incentivar la lectura y la creación literaria en los aprendices de programación de computadoras, especialmente me refiero a la creación poética por la síntesis que exige y por el uso de metáforas y su poder expresivo, traería algunos beneficios, entre los que se encuentran:

1. El uso cuidadoso del lenguaje. Usar adecuadamente las palabras y la ortografía desarrolla un ojo crítico necesario en la programación de computadoras. Ayuda a reducir el número de errores, no sólo ortográficos (cualquiera que sea su origen, éstos serían fácilmente identificables y corregibles por el ambiente de desarrollo vía un mensaje de error de "compilación/in-

terpretación"), sino otros más graves, como los errores en "tiempo de ejecución" que derivan en: a) un mal funcionamiento (hacer mal la tarea para la que fue diseñado, que es una forma de decir que no la hace) pese a que el programa parezca ejecutarse sin problemas; y b) detención abrupta del programa (conocido como petición de interrupción) por insuficiencia de memoria, tipos de datos incongruentes o fallas de otra índole. Si cambiando una palabra por otra se puede alterar el sentido de una frase, arruinando la métrica y distorsionando la emotividad y el mensaje; en un código, esto puede originar literalmente otro programa.

2. Incentivar el trabajo original. El ya clásico "copiar y pegar", frecuente entre los estudiantes principiantes para reutilizar códigos enteros o fragmentos, ya sea por ignorancia metodológica o intención deliberada (plagio), además de implicar cuestiones éticas y legales, el hecho es que lesiona el mismo proceso de aprendizaje. Presentar una copia en donde debiera ir el resultado de un esfuerzo creativo (un programa) es quizá peor que copiar en un examen. Cosa distinta es tomar código y estudiarlo, modificarlo y hasta compartirlo, ideología en que se sustenta el código bajo licencias de software libre.
3. Mejorar el estilo de programación. El uso de las metáforas y la oportuna disposición de frases para transmitir una idea, puede ser útil para la programación, la depuración y optimización de programas, y la reestructuración de código, denominada refactorización (del inglés re-factoring).

Y ya entrados en los interregnos de la poesía programada, qué tal sería hacer que los programas-poemas trasciendan la mera creación de rimas, es decir, aprovechar la semántica del lenguaje en cuestión para construir el discus-

so poético. Esto exigiría mayor dominio del lenguaje de programación y de sus abstracciones. ¿Cómo suena generar poemas en código fuente con la expresividad de un haikú o de un poemínimo huertiano? No es trivial desarrollar esta expresividad. Sin embargo, al margen de generar tales programas-poemas, lo verdaderamente importante es aprovechar las herramientas cognitivas que este esfuerzo permitirá desarrollar.

Finalmente, creo que el desarrollo de software en México es uno de los potenciales nacionales que deberían ser explotados. Requiere incluso inversiones menores que en otros campos de la tecnología, el equipo de cómputo y herramientas de desarrollo (muchas de ellas bajo licencias de software libre), de no muy difícil adquisición. La inversión real está en la obtención del conocimiento necesario, la preparación y el esfuerzo constante para generar software de calidad y desarrollar proyectos, útiles, eficientes y robustos, que resuelvan problemas del mundo real. Esto puede situarnos a la vanguardia en esta industria, además de atender al mercado nacional necesitado de opciones viables y funcionales.

El poema ganador del concurso de poesía en código fuente de 2014 puede consultarse en: <http://instacod.es/90608>.

*Licenciado en Informática por la UNAM y maestro en Inteligencia Artificial por la UV.

Actualmente labora en el Departamento de Tecnologías de la Información, División de Ciencias de la Comunicación y Diseño, UAM-Unidad Cuajimalpa y cursa el Doctorado en Ciencias de la Computación en la Escuela de Ciencias de la Computación e Ingeniería Electrónica, de la Universidad de Essex, Reino Unido.

