

# APROXIMACIÓN A LA POESÍA ESCRITA EN LENGUAJES DE PROGRAMACIÓN (SOBRE BELÉN GARCÍA NIETO)<sup>1</sup>

AN APPROACH TO POETRY WRITTEN IN PROGRAMMING LANGUAGES (ON BELÉN GARCÍA NIETO)

**Encarna ALONSO VALERO**

Universidad de Granada

enalonso@ugr.es

**Resumen:** Este artículo intenta realizar una aproximación a las prácticas poéticas escritas en lenguajes de programación o en cualquier tipo de lenguaje informático, es decir, la llamada “poesía código”. Haremos un recorrido por la génesis y los principios fundamentales de esta práctica poética y analizaremos los textos de una de las poetas españolas más destacadas de la poesía código, Belén García Nieto.

**Palabras clave:** Poesía código. Lenguajes de programación. Belén García Nieto.

**Abstract:** This article tries to be an approximation to the poetic practices written in programming languages or in any type of computer language, that is, the “code poetry”. We will take a tour through the origins and the

---

<sup>1</sup>Este trabajo es resultado del proyecto “La configuración del patrón poético español tras la instauración de la democracia: relaciones literarias, culturales y sociales” (FFI2016-80552-P), financiado por el Ministerio de Economía y Competitividad / FEDER.

fundamental ideas of this poetic practice and analyze the texts of one of the most representative Spanish poet of the code poetry, Belén García Nieto.

**Key Words:** Code Poetry. Programming Languages. Belén García Nieto

## 1. INFORMÁTICA Y POESÍA

En la actualidad, la revolución que han supuesto los recursos digitales ha motivado un cambio radical en la manera de concebir la poesía, así como de escribirla y de leerla. La poesía es, tras la irrupción de los medios digitales y de manera particular de las redes sociales, un espacio en transformación que ha vivido en los últimos años una alteración profunda y los consiguientes movimientos sucesivos de reordenación. Si, como nos enseña Pierre Bourdieu, “comprender significa comprender primero el campo con el cual y contra el cual uno se ha ido haciendo” (Bourdieu, 2006: 17), el análisis tanto de los discursos poéticos como de las trayectorias de los y las poetas están marcados en los últimos años por la aparición de los recursos digitales, que han contribuido además de manera práctica a estructurar el campo literario (y, dentro de él, el subcampo relativo a la poesía).

El concepto de campo desarrollado por Pierre Bourdieu (2000) hace referencia a un espacio social relativamente autónomo y dotado de una lógica específica, irreductible a la de los otros campos. En cada campo hay intereses propios y, al mismo tiempo, una lucha por cambiar el peso de los distintos capitales dentro del campo y por conseguir aumentar el capital social, siempre distribuido desigualmente, lo que tendrá profundas implicaciones (por ejemplo, de género: las mujeres tienen menos y mucho más inestable).

En esas luchas no participan solamente los agentes que producen directamente las obras (poetas, en nuestro caso) sino todos aquellos que

se relacionan con la edición, producción, publicación, distribución, etc., de esas obras: revistas, editoriales, suplementos de revistas, revistas especializadas... De nuevo, resulta fundamental detener la mirada en los cambios que la irrupción de los recursos digitales y las redes sociales han traído consigo, de manera que podamos reflexionar sobre cuestiones como la anulación (total o parcial) de la jerarquización tradicional del mundo literario (con figuras como la de editor o de crítico) gracias a la posibilidad de relación directa con el público lector que ofrecen los medios digitales.

Un campo es, por tanto, una configuración de relaciones objetivas entre posiciones ocupadas por individuos o instituciones. Cada posición es definida por el capital específico detentado y por el volumen y la estructura de diferentes especies de capitales que se poseen. La irrupción de las redes sociales en el campo literario ha ayudado a cuestionar (y a disminuir dentro del campo) las jerarquías verticales entre los seres y los productos simbólicos o, por lo menos, ha potenciado la interrogación acerca de cómo se producen y, de manera inevitable, a cuestionar las clasificaciones canónicas.

Del mismo modo, el uso de las redes sociales en el campo literario y en particular en el ámbito de la poesía nos señala la necesidad de pluralizar el concepto de consagración dentro de la nueva situación y la nueva estructura del campo, por una parte, y de revisar el concepto de escuela (entendida como un conjunto de personas que se definen en relación con la figura de un poeta o una poeta) en el nuevo marco. Estudiar cómo ciertos factores asociados a la nueva era digital operan sobre la producción poética permite potenciar su reflexividad y aportar herramientas adicionales a las que la tradición de la historia literaria maneja para pensar la relación de la poesía y del campo literario con el nuevo medio en el que tiene lugar.

Este trabajo se centra en el desarrollo de la poesía ligado a medios digitales, y en concreto las prácticas poéticas escritas en lenguaje de programación o en cualquier tipo de lenguaje informático, es decir, la llamada “code poetry”, nombre adaptado en España como “poesía código”. Belén García Nieto, la poeta española más representativa de esta línea,

define en una entrevista (Durán Rodríguez, 2017: s/p) la poesía código como:

*Poemas escritos en lenguaje de programación y cualquier tipo de lenguaje informático. Es utilizar todos los recursos que tiene este lenguaje para una creación artística: todos los símbolos, silencios, variables... Todo el conjunto de elementos que forman el código de la programación.*

En este tipo de textos, estamos ante código que en ocasiones puede compilar (es decir, que está de acuerdo con la sintaxis y la semántica del lenguaje en el que está escrito), pero otras veces no compila (esto es, tiene errores semánticos y/o sintácticos, por lo que el compilador no va a tener la posibilidad de generar código máquina válido).

Así, en ciertos desarrollos de poesía código, el autor o autora construye estos poemas de manera que el código funcione y, por tanto, ese código pueda ejecutarse y verse algo en la pantalla (en algunos casos, realmente se ejecuta y el resultado forma parte de la obra); otras veces, el objetivo no es que el código funcione y pueda ejecutarse, que sea correcto según el lenguaje en el que está construido, sino que se utiliza el conjunto de símbolos más allá de su aplicación formal para expresar un contenido estético y, en ocasiones, ético. En realidad, como señala Belén García Nieto en la entrevista antes mencionada, “no hay reglas. Todo lo que utilices de lenguaje informático se considera poemas en código” (Durán Rodríguez, 2017).

El debate teórico sobre la literatura hasta ahora se ha ocupado poco de los desarrollos poéticos ligados al uso del conjunto de elementos que forman el código de programación, una de las prácticas más innovadoras surgidas de la relación de la literatura con la informática. Sin embargo, la actualidad de estas propuestas muestra la necesidad de desarrollar herramientas de análisis adecuadas a la diversidad de textos y formatos utilizados por este tipo de poesía.

La relación entre informática y creación poética es reciente y supone un auténtico rompecabezas taxonómico si atendemos a los escasos acercamientos teóricos que se han producido, ya que incluye manifestaciones muy diversas (en cuanto a soporte, grado de interacción, concepción del lenguaje, proyecto estético, finalidad ética...) y que, en muchos casos, no tienen en común más que el propio hecho de utilizar en su composición algún tipo de lenguaje informático.

Hacer un recorrido de los antecedentes, historia y desarrollo de la poesía ligada a la informática excede las dimensiones y los propósitos de este artículo, pero haremos un brevísimo repaso de los hitos en el camino hacia la poesía código.

Si la historia de la creación poética por ordenador es breve, la de la poesía código es brevísima. Este tipo de poesía es un caso particular dentro de la literatura asociada a los medios digitales puesto que, en gran parte de sus desarrollos (y es el caso de los que vamos a estudiar aquí) mezcla nociones y formas de la poesía convencional con código de programación. A diferencia de la *e-poetry*, que usa computadoras físicas de manera prominente, los poemas código, como antes señalábamos, pueden o no ejecutarse, y pueden no estar escritos en código correcto. Un poema código puede ser o no interactivo, y puede presentarse como digital o analógico.

En ese mapa, nos fijaremos en este trabajo en aquellos desarrollos que se centran en la cuestión del lenguaje y, en esa vía, en aquellos que se proponen sacar el lenguaje informático y el código del ámbito meramente técnico, frente a otras opciones que ejecutan el código y están más unidos al soporte digital, lo multimedia, etc.

La primera manifestación de poesía código fue *Poèmes ALGOL*, de Noël Arnaud, uno de los miembros fundadores del OULIPO, y se publicó en 1968. Es un poemario escrito en lenguaje informático ALGOL (Algorithmic Language), hoy obsoleto, pero precursor de C, uno de los lenguajes más importantes en la actualidad. Arnaud utilizó las veinticuatro palabras predefinidas que constituían las palabras clave de ALGOL para, traducidas al francés, crear sus textos. Los poemas no eran código válido.

Es necesario, por tanto, en esta breve historia hacia la poesía código, hacer referencia al OULIPO (Ouvroir de Littérature Potentielle) y al posterior nacimiento del ALAMO (Atelier de Littérature Assistée para la Mathématique et les Ordinateurs), la rama tecnológica del grupo, sin la que no pueden entenderse algunas derivas de la literatura digital en los años 80 (Vuillemin, 2005).

ALAMO fue fundado por varios miembros pertenecientes al propio OULIPO. De hecho, algunos críticos, como Braffrot, han hablado del ALAMO como un estadio más de evolución del grupo OULIPO, al hablar de tres etapas en la historia de este (hasta 1988, fecha del análisis de Braffrot): una primera etapa gobernada por la efervescencia creativa de la vieja guardia oulipiana (entre ellos, Arnaud); una época de transición marcada por la obra de Georges Perec y en la que empiezan a distinguirse dos grandes corrientes oulipianas, la combinatoria y la estructural; y una tercera época caracterizada por un interés cada vez mayor por las nuevas tecnologías (Braffrot, 1988: 108-109), momento en el que el ALAMO adquiere un papel preponderante. De este modo, el ALAMO aparece en 1981 como un proyecto cuyo objetivo era utilizar, de todas las maneras posibles, los ordenadores y la informática al servicio de la literatura, sin restricción alguna.

Si el OULIPO se conformó como un grupo que reunía escritores y matemáticos, el ALAMO surge con la intención de aunar escritores e informáticos, un interés que está presente en el OULIPO desde el principio. Así, Braffrot señala la importancia en el OULIPO de “la formalisation. Avant même que l’éternelle brigade eût quitté l’immanence, les pères fondateurs caressaient le projet d’une *laboratoire* où le langage mathématique, puis l’outil informatique devraient fournir une contribution essentielle” (Braffrot, 1988: 108).

Naturalmente, la utilización que los autores del OULIPO y el ALAMO han hecho de la informática y los ordenadores ha sido diversa (Martín Sánchez, 2012: 94-96), e incluye, entre otros procedimientos posibles, textos en los que los ordenadores solamente intervienen en el

proceso de producción de la obra, otros en los que el ordenador estaría presente tanto en el momento de composición del poema como en el de lectura, y otros casos (mayoritarios, en el caso del ALAMO), en los que la creación de la obra incluye al lector que, a través de la interacción con el ordenador, participa en ese proceso.

Como antes señalábamos, en el caso de la poesía código tenemos obras que, dentro de las vías anteriormente descritas, crean un código que puede ejecutarse en el ordenador, pero tenemos otras (es el caso de la producción en código de Belén García Nieto, la autora cuya producción vamos a analizar), en las que lo fundamental es el uso mismo del lenguaje informático, sin que sea necesaria la presencia del ordenador ni la corrección sintáctica o semántica del código que se produce. En ese sentido, se trata de textos más cercanos a los experimentos de OULIPO sobre hacer poesía con un lenguaje limitado y una estructura prefijada, en la línea de las célebres constricciones oulipianas (<http://oulipo.net/fr/contraintes> [28/04/2018]), que a los del ALAMO. Un ejemplo claro y que resulta fundamental para la poesía código es el de los textos estructurados a partir de bifurcaciones sucesivas, uno de los resultados más emblemáticos del OULIPO, cuyo mecanismo está basado en el lenguaje de programación informático y se presenta en forma de grafo bifurcante.

## 2. CODE {POEMS}: HÁGASE EL CÓDIGO

La poesía código ha encontrado hasta ahora poco desarrollo en España, aunque sí en otros países europeos y en América. Existe una variedad de eventos y sitios web que permiten a los autores presentar o publicar poesía codificada. Entre ellos hay que mencionar el Poetry Slam de la Universidad de Stanford (<https://web.archive.org/web/20131231091141/http://stanford.edu/~mkagen/codepoetryslam/> [28/04/2018]), un concurso que a día de hoy es probablemente el acontecimiento con más visibilidad de estas prácticas poéticas. También promueve este tipo de escritura la página PerlMonks (<http://www.perlmonks.org/?node=Perl%20Poetry>

[28/04/2018]), que se especializa en desarrollos poéticos en lenguaje de programación Perl.

Aunque no intenta escribir poesía, cabe destacar el influyente concurso *The International Obfuscated C Code Contest* (<http://www.ioccc.org/index.html> [28/04/2018]), muy conocido en el mundo de la informática, y cuyas normas son las siguientes, según aparece en su web:

- To write the most Obscure/Obfuscated C program within the rules.*
- To show the importance of programming style, in an ironic way.*
- To stress C compilers with unusual code.*
- To illustrate some of the subtleties of the C language.*
- To provide a safe forum for poor C code.*

Como puede verse, además de la obligación de escribir en lenguaje C, el concurso plantea como criterio fundamental un valor estético, en este caso en negativo: hay que crear un código oscuro y feo, inusual, aunque en este caso se pide también que resulte operativo y pueda ejecutarse<sup>2</sup>.

En ese sentido, la aplicación de criterios estéticos al código de programación es habitual, unidos a otros como la eficacia o la legibilidad. Así lo muestran las definiciones sobre el código limpio (es decir, la programación eficaz y bella) que ofrecen conocidos programadores. Bjarne Stroustrup, inventor del lenguaje C++, asegura: “Me gusta que mi código sea elegante y eficaz” (Martin, 2012: 34). Como sigue diciendo Martin, “aparentemente Bjarne piensa que el código limpio es un placer a la hora de leerlo” (34).

Otro famoso programador, Grady Booch, asegura que “el código

---

<sup>2</sup>Aunque no se trata exactamente de literatura, al menos en el objetivo, William Poundstone cuenta en *¿Es lo bastante inteligente para trabajar en Google?* un ejercicio exigido por los entrevistadores de Google que bien podría considerarse en la línea de la experimentación oulipiana: utilizar un lenguaje de programación para describir una gallina. Según dice el autor, “describir una gallina utilizando ALGOL o C++ es un ejercicio quijotesco” (Poundstone, 2012: s/p).

limpio se lee como un texto bien escrito” (35). Podríamos seguir porque, con frecuencia, en las definiciones del código, sobre todo del código limpio, están presentes las metáforas literarias.

El primer proyecto importante en poesía código, con dimensión internacional y que dio a este tipo de poesía una gran proyección fue *Code {poems}*, un proyecto de varios desarrolladores informáticos que escriben poemas con lenguajes de programación. El formato del volumen es el clásico de un libro y los autores que aparecen en él son de distintas nacionalidades, aunque no aparece ningún español.

El volumen muestra una de las características fundamentales de los lenguajes informáticos y, por tanto, de buena parte de la producción poética escrita con ellos: se trata de lenguajes transnacionales, que funcionan como una especie de lengua franca, aunque tienen una amplia influencia del inglés.

El prólogo de *Code {poems}*, firmado por Jamie Allen, es una de las primeras aproximaciones teóricas a este tipo de poesía y pone en el centro de su reflexión la relación entre el lenguaje natural y el del código de computación. Asegura que estamos ante un experimento con un proceso similar al de la poesía en lenguaje natural (o lenguaje humano, como se dice en este prólogo; Allen, 2012: xi) “but in the realm of computer software languages” (x).

El proyecto pone en primer plano el objetivo de sacar el lenguaje informático de su trabajo puramente funcional y utilizarlo como un idioma (en este caso, como decíamos, transnacional). Además, se impugna el carácter supuestamente neutral de los lenguajes técnicos: “Machinic languages are never merely indicative or functional, but a written condensation of the mood, the personality, and the world-view of its autor” (xi).

Aparecen también conceptos y reflexiones fundamentales en ciertas vías de la poesía código, sobre todo en aquellas que producen un código válido que puede ejecutarse, y que probablemente estarán en el centro de los desarrollos teóricos que vayan apareciendo en el futuro. Por ejemplo, la

cuestión de la lectura y los lectores de este tipo de textos: “With all computer code though, there *might* be an intention toward a secondary audience of human readers, but there is *always* deference toward the primary electronic audience” (xii). ¿Puede la máquina considerarse primera audiencia lectora o es mera ejecutora? Estaríamos ante una problemática que, en sus últimas consecuencias, incluiría conceptos propios de la inteligencia artificial.

En los poemas que vamos a estudiar, parece difícil que el ordenador pueda considerarse primera audiencia lectora. De hecho, con frecuencia en ellos la máquina no tiene papel alguno. En la poesía código de Belén García Nieto, uno de los objetivos primordiales es sacar ese lenguaje del ámbito técnico, pero no se hace necesariamente ligado al uso de un ordenador.

Sigue preguntándose Allen: “Does the writer deal elegantly with RAM memory use, or allow the program to leave messy traces of itself all over the motherboard?” (xii). Como vemos, nos aparecen a cada paso consideraciones estéticas asociadas al código. El código, en este caso sacado de su ámbito técnico y despojado de su exigencia de funcionalidad, sería una parte de la cultura:

*Removed from prying gaze of the linker, compiler and CPU – written into a physical book like the one you now hold in your hands –code enters culture, history and imagination. It speaks uncharacteristically, and directly, to people: human readers, and our worlds (xii).*

### 3. LA POESÍA CÓDIGO DE BELÉN GARCÍA NIETO

Como hemos señalado con anterioridad, una de las pocas personas que escribe poesía código en España es Belén García Nieto, poeta y programadora informática. Su producción poética en código incluye, junto con poemas escritos íntegramente en código informático, otros en los que mezcla lenguaje informático con lenguaje natural. Belén García

Nieto apunta, al hablar de sus poemas código (Durán Rodríguez, 2017) a la cuestión central del lenguaje:

*Sacar al lenguaje de su trabajo puramente funcional, utilizarlo como un idioma. Para mí, tiene mucho sentido hacer esto no solo por lo estético sino por lo ético. Es un lenguaje que, al dotarlo de carga semántica y ponerlo en un contexto, dice muchas más cosas que las que comunica su mera funcionalidad. Es un lenguaje técnico que, como todo lenguaje, no es neutral, está sesgado por la persona —hay alguien que lo escribe detrás—, y eso implica una responsabilidad. Por ello en poesía código hay más cosas que el hecho estético, un poema visualmente atractivo que utiliza símbolos raros. Es un lenguaje que aparenta neutralidad pero ningún lenguaje lo es.*

Así, Belén García Nieto construye sus poemas en torno a los pivotes centrales de la reflexión sobre el lenguaje y la ética frente a funcionalidad, que con frecuencia no se da en el código que produce en sus poemas. De este modo, el lenguaje se convierte aquí en instrumento privilegiado de la reflexividad crítica, condición imperativa de la lucidez colectiva.

La cuestión de la ética y de lo colectivo se entrelaza, en este tipo de poemas, con el problema del lenguaje especializado y su público potencial, al tratarse de lenguajes limitados, al menos en teoría, a los especialistas que pueden escribirlo, leerlo y comprenderlo. Para paliar en lo posible ese problema Belén García Nieto utiliza una serie de recursos:

*Escribo poesía código y poesía en lenguaje natural. También utilizo el pseudocódigo, que está a medio camino de los dos y puede llegar a más personas. Hay autores que hacen poesía especialmente para informáticos, que requiere conocer el lenguaje y entenderlo, e incluso hay poemas código que necesitan ser compilados, hay que escribir el código en una consola para*

*que te devuelva algo. Pero también hay otros poemas que no necesitas saber nada, puedes entenderlos porque el código de programación comparte mucho con el inglés.*

Para sacar el lenguaje informático de su ámbito técnico y funcional e insertarlo en la vida, uno de los recursos fundamentales es el de la ironía:

*En muchos poemas se juega con la ironía del lenguaje, en qué se parece a la realidad. En Java se pueden hacer clases, que son como plantillas, y se juega con ironías propias del lenguaje. En html, que no es lenguaje de programación sino de etiquetas y es bastante comprensible, se puede comentar todo salvo ciertas palabras que son las que forman el poema al ponerlo en el navegador. Es el juego entre ese lenguaje y la vida real.*

Como hemos señalado, en el caso de la producción poética de esta autora, tanto en lenguaje natural como en lenguaje de programación, en el centro hay que poner la noción de compromiso e incluso la de militancia. Ejemplo de ello y del uso de pseudocódigo, que no funcionaría si se ejecutase en un ordenador pero que consigue ser poéticamente efectivo y llegar a un público más amplio, es un poema que se publicó en la web Yo Sí Sanidad Universal (<http://yosisanidaduniversal.net/noticias.php/http-belengnieto-com-yo-programo> [28/04/2018]) y que llevaba por título “Yo programa”.

El punto de partida de la composición del poema fue la noticia de que seis mil menores iban a ser dados de baja de la Seguridad Social por la aplicación de una serie de requisitos que se traducían en que se dejaría sin cobertura sanitaria a seis mil niños y niñas inmigrantes. El poema nos muestra una *query* (es decir, una consulta a una base de datos), de ocho líneas en pseudocódigo (mitad lenguaje natural y mitad código), en la que se nos explica cuánto tardaría el programa informático en hacerlo. En el poema, en los versos en los que se dice “Resultado de la consulta: 6.000

columnas afectadas en 0.03 segundos / Query OK, 6.000 rows affected (0.03 sec)”, tenemos la simulación de la ejecución:

*El resultado de la query fue que tardaría 0,03 segundos en eliminar a 6.000 menores de una base de datos. Y eso se hace, un informático tuvo que hacerlo. El poema es simplemente eso, una query a una base de datos y devuelve 6.000 registros eliminados. Está hecho para que se entienda, como denuncia de la responsabilidad que tiene este lenguaje, que no es solo técnico (Durán Rodríguez, 2017)*

Se trata, por tanto, de poner de manifiesto que el lenguaje informático no es un lenguaje neutral y que puede llevar consigo una importante responsabilidad para las personas que lo manejan (como la del caso que se nos cuenta en el poema, que es real y alguien tuvo que hacer y ejecutar):

*6.000 menores hijos de inmigrantes no tienen acceso a la sanidad madrileña  
He creado una base de datos, la llamo COBERTURA\_ SANITARIA  
CREATE DATABASE COBERTURA\_ SANITARIA;  
En ella genero una tabla, la llamo CITAS y agrego los campos donde almacenar los datos.  
CREATE TABLA\_ CITAS  
(  
NOMBRE,  
APELLIDOS,  
FECHA\_ NACIMIENTO,  
REGULARIZADOS (verdadero o falso),  
TARJETA\_ SANITARIA (verdadero o falso)  
);*

*Genero una query para acceder a la tabla. Hago una consulta que me permita actualizar la base de datos e igualar el valor del campo TARJETA\_SANITARIA a falso siempre que se cumplan dos condiciones: la FECHA\_NACIMIENTO sea mayor a 90 días y el valor del campo REGULARIZADOS sea falso.*

```
UPDATE TABLA_CITAS  
SET TARJETA_SANITARIA = false
```

*where*

```
(FECHA_ACTUAL - FECHA_NACIMIENTO) > 90 &&  
REGULARIZADOS == false
```

*Resultado de la consulta: 6.000 columnas afectadas en 0.03 segundos.*

*Query OK, 6.000 rows affected (0.03 sec)*

*Es viernes, 10.15 de la mañana. Anoto como realizada la primera tarea del día.*

*Es hora de un café, me dirijo a la máquina.*

*Hoy parece un día tranquilo.*

En una línea parecida se sitúa “Estado del bienestar”, uno de los poemas código que pueden leerse en la página web de la autora (<http://compilado.net/category/poesia-y-codigo/> [28/04/2018], de donde tomamos los poemas que veremos a continuación). En él tenemos una lista de elementos, a la que se da el nombre “desahucio”, y “suma” es una variable de acumulación. Para cada elemento de la lista “desahucio”, se elimina el nombre correspondiente a la persona desahuciada (o, tal como lo expresa el poema, “suma=0”); ese procedimiento se acumularía y, como final del proceso, la pantalla nos mostraría un mensaje notificando que el nombre de la persona afectada por el desahucio ha sido eliminado (“alert (nombre + “: eliminado”). De este modo, la autora vuelve a defender la no neutralidad del lenguaje informático y la responsabilidad de las personas que lo utilizan:

```
var desahucio=new Array(),
suma = 0;
desahucio.forEach(
    function eliminar(nombre) {
        suma += nombre;
        alert(nombre + ": eliminado");
    }
);
```

Un poema muy significativo en cuanto al compromiso ideológico que nutre los versos en código de Belén García Nieto es “Prototipo super clase”, a la vez que una muestra clara del juego irónico con el que la autora utiliza en ocasiones este tipo de lenguajes. Está escrito en JavaScript, un lenguaje orientado a objetos, es decir, que describe un mundo formado por objetos que se comunican entre sí. En estos lenguajes existe el concepto de herencia, vinculado a jerarquías de conceptos.

Los objetos son instancias de clases que definen sus propiedades y comportamientos, y esas propiedades y comportamientos se heredan desde la superclase a las subclases que son casos particulares de ella.

El poema hace un juego relacionando la idea de clase social y las jerarquías sociales con las jerarquías de clases en este tipo de lenguajes informáticos:

```
function superClase() {

    this.rol      = “Acumulación”;
    this.medios   = “Privados”;
    this.imprimir = “CAPITALISMO”;
    this.metodo = function() {
        CrearCrisis()
        .then(CrearExpansion())
        .then(CrearShock());
    }
}
```

```

        .then(CrearDesorientacion())
        .then(CrearAislamiento())
        .then(function(){
            return new superClase();
        });
    };
}

```

Como antes hemos señalado, en estos lenguajes los objetos son instancias de clases que definen sus propiedades y comportamientos. Así, la primera función, denominada “superclase”, define la creación de un objeto con una serie de propiedades (rol, medios, que son iguales a “acumulación” y “privados”, respectivamente), y el mensaje que ello imprime es “capitalismo”.

En cuanto a los comportamientos de esa superclase, son las funciones que comienzan en “CrearCrisis” y continúan, en una secuencia, con “CrearExpansion”, “CrearShock”, “CrearDesorientacion” y “CrearAislamiento”.

En el poema “Colapso”, encontramos una secuencia de condiciones lógicas, encabezadas por “if”. Si son ciertas (es decir, si se van cumpliendo), devuelve el valor “colapso”:

```

if(this.habits){
    if(temperature.rise){
        if(thaw){
            if(species.extinction){
                if(deforestation){
                    if(desertification){
                        return colapso;
                    }
                }
            }
        }
    }
}

```

```
    }  
  }  
}
```

El poema “Invisibles” no está escrito en lenguaje de programación sino en html (Hiper Text Markup Language), el lenguaje que se utiliza en las páginas web para describir la estructura del contenido web y la forma de presentarlo. La autora juega con ambas cosas y las define en el poema.

La manera de presentar el contenido (es decir, cómo lo presentará el navegador web) es lo que tenemos entre las etiquetas “style” y fin de “style” (</style>). Como podemos ver en el poema, estaría todo vacío (hidden, 0, “”, 0). Si se tratase de una web real, al abrirlo en el navegador no se vería nada.

El contenido, por su parte, sería lo que correspondería a la sección “body” hasta fin de “body” (</body>). Aunque en la manera de presentarlo aparezca todo vacío, el contenido está (o, como dice el poema, “Las ideas / Y el deseo / Siguen en pie”):

```
<html>  
  <style>  
    body{visibility:hidden}  
    header{opacity:0}  
    section{content:””}  
    footer{margin:0}  
  </style>  
  <body>  
    <header>Las ideas</header>  
    <section>Y el deseo</section>  
    <footer>Siguen en pie</footer>  
  </body>  
</html>
```

En todos estos poemas queda clara la impugnación del estatuto de neutralidad de los lenguajes informáticos y el radical compromiso que, en la misma línea de su poesía escrita en lenguaje natural, asume la práctica de la poesía código de Belén García Nieto. Son versos llaman siempre a lo colectivo y en ellos el código, como los lenguajes en los que se escribe, se convierten en herramienta para la reflexión y la crítica. Ambas se realizan, como explica la propia autora en otra entrevista ([http://www.libreconfiguracion.org/proyecto\\_genoma\\_poetico/entrevistas-belen-garcia-nieto/](http://www.libreconfiguracion.org/proyecto_genoma_poetico/entrevistas-belen-garcia-nieto/) [28/04/2018]):

```
while(true){  
    always();  
}
```

## REFERENCIAS BIBLIOGRÁFICAS

- ALLEN, J. (2012). “Foreword”. En *Code {poems}*, Ishac Bertran (ed.), v-x. Barcelona: The Authors.
- BOURDIEU, P. (2000). *Propos sur le Champ Politique*. Lyon: Presses Universitaires de Lyon.
- \_\_\_\_ (2006). *Autoanálisis de un sociólogo*. Barcelona: Anagrama.
- BRAFFORT, P. (1988). “F.A.S.T.L.: Formalismes por l’analyses et la synthèse de textes littéraires”. En *OULIPO: Atlas de Littérature Potentielle*, Oulipo, 108-137. París: Gallimard.
- DURÁN RODRÍGUEZ, J. (2017). “La poesía código de Belén García Nieto: versos éticos en lenguaje de programación”. *El Salto*, 24 de octubre. <https://elsaltodiario.com/poesia/codigo-entrevista-belen-nieto-rimas-eticas-lenguaje-programacion> [28/04/2018].
- GARCÍA NIETO, B. (2015). “Yo programo”. En *Yo Sí Sanidad Universal*, <http://yosisanidaduniversal.net/noticias.php/http-belengnieto-com-yo-programo> [25/04/2018].

- \_\_\_\_ (2017). “Entrevistas # Belén García Nieto”. En *Proyecto Genoma Poético*, 11 de abril, [http://www.libreconfiguracion.org/proyecto\\_genoma\\_poetico/entrevistas-belen-garcia-nieto/](http://www.libreconfiguracion.org/proyecto_genoma_poetico/entrevistas-belen-garcia-nieto/) [23/04/2018].  
<http://compilado.net/category/poesia-y-codigo/> [23/04/2018].
- MARTÍN, R. C. (2012). *Código limpio. Manual de estilo para el desarrollo ágil de software*. Madrid: Anaya Multimedia.
- MARTÍN SÁNCHEZ, P. (2012). *El arte de combinar fragmentos. Prácticas hipertextuales en la literatura oulipiana (Raymond Queneau, Georges Perec, Italo Calvino, Jacques Roubaud)*. Tesis de doctorado. Granada: Universidad de Granada.
- OULIPO. <http://oulipo.net/fr/contraintes> [23/04/2018].
- POUNDSTONE, W. (2012). *¿Es lo bastante inteligente para trabajar en Google?* Barcelona: Conecta.
- VUILLEMÍN, A. (2005). “Poesía, informática y creación: las nuevas aproximaciones”. En *Textualidades electrónicas. Nuevos escenarios para la literatura*, Laura Borrás Castanyer (ed.), 163-175. Barcelona: Editorial UOC.

Recibido el 15 de mayo de 2018.

Aceptado el 28 de junio de 2018.

