

Pérez Mena, María.

Investigadora pre-doctoral, UPV/EHU, Facultad de Bellas Artes, Departamento de Dibujo, Grupo de Investigación en Diseño Gráfico y Tipografía —Letraz—.

Lo real y lo virtual en el diseño tipográfico: la fase de producción en la creación tipográfica.

TIPO DE TRABAJO

Comunicación.

PALABRAS CLAVE

Diseño tipográfico, producción tipográfica, creación tipográfica, programación, software.

KEY WORDS

Typeface design, typographic production, programming, software.

RESUMEN

En el ámbito de la creación tipográfica, la fase de producción ha supuesto siempre un auténtico desafío para el hacedor de letras, que ha sido testigo de una imparable evolución e innovación tecnológica, situada, desde hace unas décadas hasta la actualidad, en el entorno informático. Si bien en la fase de ideación formulamos los planteamientos tipográficos a través de la combinación armoniosa de las proporciones espaciales, cuando las digitalizamos, en el proceso de producción, este lenguaje visual es sustituido por uno verbal a través de la sintaxis propia del lenguaje informático. Conocer este lenguaje se hace pues fundamental para el diseñador tipográfico actual, que pasa de trabajar con masas de blancos y negros a trabajar sobre contornos operados por órdenes informáticas o incluso, con meras operaciones informáticas lejos ya de toda forma gráfica. En base a este planteamiento, podemos entender la letra también como *software*.

Este trabajo tiene por objeto observar la escisión que se produce en la creación tipográfica en el proceso de producción con respecto al de ideación, que viene determinada por un cambio de visualización de las formas gráficas en el momento de su producción. En éste, la letra pasa de ser un ente real a uno virtual, cuya fundamentación en el diálogo armonioso de los blancos y los negros deviene ceros y unos.

ABSTRACT

In the area of typeface design, production has always meant a great challenge to the typographer, who has get use to be a witness of an unstoppable technological evolution and innovation up to our digital era. Although along the stage of the typeface's conception we build typographic basis on the harmonic combination of spatial proportions, when these forms are digitalized —in the stage of production— this visual language is replaced by a verbal one, through computer's own language syntax. Therefore, having a good knowledge of this language is essential for the typeface designer of nowadays, who have to deal with a change from working in a language of visual proportions to work over contours operated by digital data or, even, just with solely digital data far from any graphic shape. Based on this proposal, we could understand letterforms, also, as *software*.

The aim of this paper is to observe the split produced in typeface design between the stage of conception and the one of production, which is being determined by a change of the graphics forms that we can see at the moment of production. In this moment we are talking about, letterforms turn from a real being into a virtual one, which basis on the harmonic dialogue of black and white becomes zeros and ones.

INTRODUCCIÓN

La innovación tecnológica supone un salto cualitativo en la rutina diaria del individuo, facilitándole las tareas y dando lugar a nuevos discursos. En el área del diseño tipográfico, la evolución tecnológica, además de facilitar y agilizar las tareas del diseñador en la fase de producción de un proyecto, ha construido nuevas maneras de entender el proceso de configuración tipográfica. En la era digital en la que nos encontramos, la producción de fuentes tipográficas requiere comprender las formas ya no solo como forma gráfica sino como datos informáticos. En este sentido entendemos la letra, también, como *software*.

Con la tecnología de la imprenta, las formas alfabéticas se configuran como elementos de incipiente parametrización¹ y objetos tridimensionales. Como sistema de producción artesanal, y mecanizado desde la era industrial de la mano de la linotipia y la monotipia, esta cualidad tangible de las letras perdura en el tiempo hasta mediados del siglo XX con la fotocomposición². La introducción de esta tecnología como sistema de producción tipográfica supone el comienzo de la dualidad real/virtual que desde entonces marcará el proceso de creación.

El advenimiento de la computación tuvo repercusión, también y evidentemente, en el ámbito de la creación tipográfica. Es en este momento cuando las letras se incorporan a la evanescencia de la virtualidad al convertirse en datos informáticos que son almacenados en la máquina³. Desde este principio, entendemos pues que la posibilidad de la letra de ser bella y eficazmente facturada depende, por un lado, del conocimiento —lejos ya de toda pericia manual— que el diseñador posea del lenguaje de computación que permite la producción y correcta reproducción de una tipografía; y, por otro, de la capacidad de los lenguajes informáticos y del propio sistema que los soporta —programas y ordenador—, de construir y almacenar ordenes específicas de construcción tipográfica.

La democratización tecnológica que ha supuesto la introducción del ordenador en el campo del diseño tipográfico, ha dispersado la especialización y número de los miembros necesarios en el proceso de creación para poder resolver —en la mayoría de los casos— todo el proceso a través de un diseñador-programador que hace las veces de configurador de formas e ingeniero informático. En este estudio no se plantea la influencia que este hecho pudiera suponer en las formas que se generan, sino en los procesos mentales que derivan del proceso de ideación, totalmente visual, y el proceso de producción, en el que intervienen elementos del lenguaje informático cuyo dominio y manipulación puede generar igualmente esas formas que siempre, y en cualquier caso, son testeadas visualmente.

Este trabajo sitúa la letra, en primer lugar, como *software*, en el que se plantea el sistema alfabético como un conjunto de órdenes informáticas almacenadas en un archivo. Esto es, las formas devienen datos. En segundo lugar, se propone entender la letra como programación, ya que el sistema informático no entiende de formas sino de operaciones algebraicas cuya formulación precisa —mediante una serie de operaciones determinadas—, da lugar a la visualización de la configuración del alfabeto. Esto es, los datos devienen formas.

LA LETRA COMO SOFTWARE

La razón de la tipografía⁴ se fundamenta en la sensibilidad hacia el diálogo armónico de los blancos y los negros. Esta fundamentación establece la hoja de ruta de todo proyecto tipográfico, en el que los niveles de legibilidad y expresividad del conjunto de formas alfabéticas se delimita por la tecnología que permite su creación y reproducción. En este sentido, el factor tecnológico es una pieza clave de la creación tipográfica y, aunque toda nueva tecnología intenta emular los resultados de la anterior agilizando el proceso, las nuevas tecnologías pueden influir en los comportamientos de las formas preexistentes, dando lugar a discursos nuevos. Este es el caso de la tipografía digital.

En la década del último cuarto del siglo XX, comienza la era digital con la posibilidad de almacenar imágenes a modo de datos informáticos, que eran representadas en los monitores como mapas de bits. Este hecho, supuso la pérdida de identidad y calidad de las formas tipográficas⁵, cuyos elementos curvos eran difíciles de interpretar y representar por los sistemas de salida —pantalla e impresoras—. Sobre la base del binomio 0 y 1 del lenguaje informático por el que el píxel es o no “pintado”, Donald E. Knuth ideó un sistema por el cual las formas alfabéticas —curvas, rectas y diagonales— son descritas a partir de parámetros matemáticos que el sistema reconoce y almacena⁶. Basándose en los estudios comenzados en el Renacimiento en torno a la construcción de las letras desde una base matemática por medio de la interpolación de arcos, este experto en Ciencias de la Computación, de extraordinaria sensibilidad tipográfica, establece las bases por las que la tipografía pierde su presencia física y se convierte en un ente virtual de composición ya no gráfica sino numérica.

¹ SMEIJERS, Fred: *Counterpunch*. Hypen Press, London, 2011, p. 79.

² En la fotocomposición las matrices son elementos lumínicos.

³ LAWSON, Alexander: *Anatomy of a typeface*. David R. Godine Publisher Inc., Shangai, 2010, p. 402.

⁴ Entendida como alfabeto, no como sistema de impresión.

⁵ KINROSS, Robin. *Tipografía Moderna*. Campgràfic, Valencia, 2008, p. 186.

⁶ KNUTH, Donald E. *Computers Modern Typefaces*. Addison-Wesley, Reading, Massachusetts, 1986.

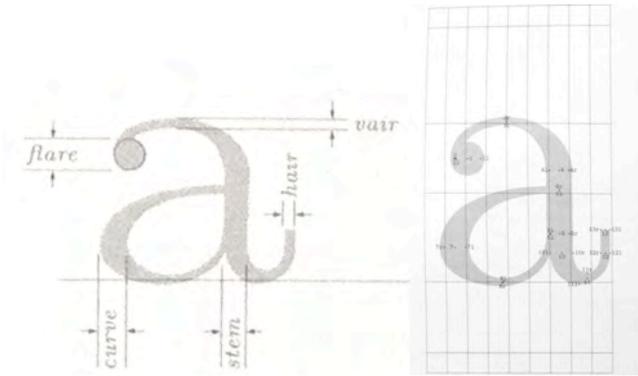


Ilustración 1. Establecimiento de parámetros y descripción de coordenadas de los puntos que construyen la letra 'a' según los planteamientos de Donald E. Knuth

Hasta la introducción de la informática en el ámbito de la creación tipográfica, una fuente era una variante y peso concreto y tridimensional de una tipografía. Hoy en día, el término *f fuente* es definido legalmente como *software*⁷, es decir, como un conjunto de instrucciones que ejecutan ciertas tareas en el entorno informático⁸. La virtualidad del concepto ha supuesto una brecha entre las fases de ideación y de producción de una tipografía, en el que los preceptos de adecuación visual de las formas son rápidamente interpretados en términos informáticos.

En los últimos años se han producido importantes innovaciones en el ámbito de la creación tipográfica. Las herramientas de las que nos valemos hoy para crear alfabetos son más visuales e intuitivas que nunca, lo que nos permite, en cierta manera, olvidarnos del farragoso lenguaje informático. Sin embargo, nuestro modo de comprender la configuración tipográfica se ha visto afectada sustancialmente: ya no trabajamos sobre las formas, sino sobre las instrucciones que generan tales formas, habiendo desplazado la sustancia gráfica a un segundo plano, de comprobación de los resultados.

El proceso de digitalización supone trasladar las formas a datos informáticos. Bien sea a través de una interfaz visual o del lenguaje de programación oportuno, el principio de la creación tipográfica en el entorno digital comienza con la comunicación con la máquina, encriptando la información visual en órdenes informáticas. Estas órdenes de entrada se generan introduciendo la información de la fuente y los valores básicos de construcción tipográfica como son los pertenecientes a sus proporciones vertical y horizontal, la inclinación del eje y el grosor y contraste del trazo. A medida que se avanza en el proceso de digitalización, estos datos son completados con aquellos referentes a los valores de espaciado entre letras y palabras, altura y posición de la acentuación diacrítica, etc.

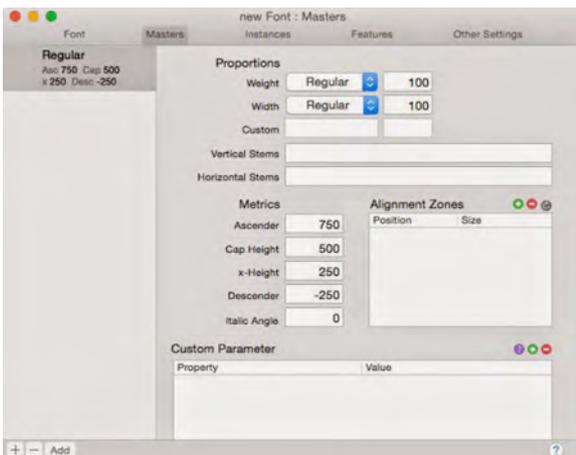


Ilustración 2. Establecimiento de parámetros básicos de construcción de la fuente en la aplicación Glyphs

⁷ LEHNI, Jürg. *Typeface as programme*. Disponible en <https://www.typotheque.com/articles/typeface_as_programme>.

⁸ RAE 22ª Ed. Disponible en <<http://lema.rae.es/drae/?val=software>>

Pensar gráficamente las formas en el medio digital es posible gracias a las tecnologías de curvas llamadas Beziér —propias del lenguaje Postscript— y de las curvas llamadas TrueType —propias del lenguaje TrueType—. Aunque motivadas por cuestiones comerciales, no es objeto de este estudio definir sus características y diferencias. En cualquier caso, lo que nos interesa saber, llegados a este punto, es que ambos tipos permiten visualizar y trazar las curvas a partir de vectores dirigidos por manejadores, los cuales establecen unos puntos de anclaje y definen las coordenadas de la curva que el programa describe y almacena. Cada descripción del conjunto de la fuente es almacenada dentro de un archivo, que será el que se distribuye para su comercialización y que los dispositivos de salida reconocerán e interpretarán una vez sea generada la fuente.

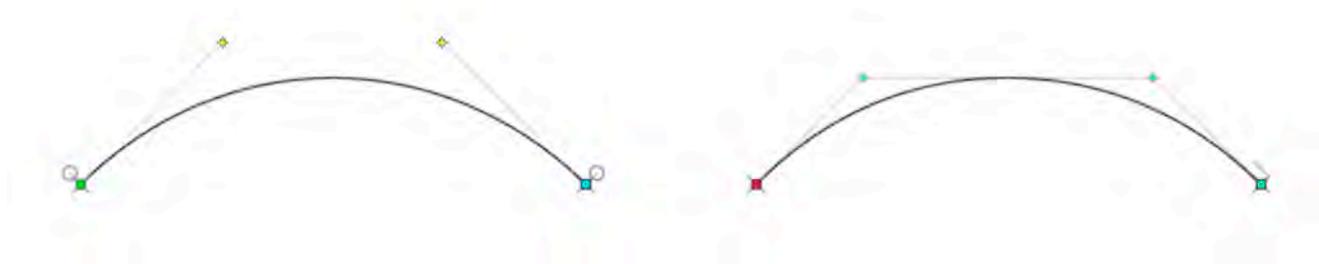


Ilustración 3. Izqda.: curva tipo Postscript; dcha.: curva tipo TrueType

Todo archivo digital es, evidentemente, almacenado en un formato específico, que en el caso de la tipografía puede variar. Aquel que encontramos hoy mayoritariamente en los mercados es el formato OpenType, que de alguna manera combina elementos de las dos tecnologías enunciadas —Postscript y TrueType—. Este formato, además de poder acoger los modos de construcción de formulaciones contrarias, es compatible con el estándar Unicode. Unicode recoge y sistematiza todos los caracteres pertenecientes a miles de escrituras del mundo. El formato Opentype permite así ampliar el mapa de caracteres de nuestra fuente o archivo digital en un gran número de ellos que pueden corresponder a versiones de la misma en grafías distintas que comparten unos mismos parámetros de configuración —proporción, inclinación, contraste, ritmo...—. Pero además OpenType permite trabajar en cuestiones de “preciosismo tipográfico” como la integración de ligaduras, caracteres floreados, versalitas, fracciones, signos históricos, etc.

Establecer la letra como *software* supone alejarnos de todo planteamiento tradicional de configuración tipográfica, ya que la combinación de trazos que hasta entonces generaba se sustancia visual desaparece transformándose en una combinación de operaciones matemáticas descritas a través de la sintaxis del lenguaje informático y almacenadas como elementos virtuales.

LA LETRA COMO PROGRAMACIÓN

Una vez establecido el principio de la letra como conjunto de instrucciones almacenadas en archivos informáticos, el proceso de producción se nos antoja si cabe, más alejado de toda forma tipográfica. La posibilidad que ofrece la tecnología digital de parametrizar los elementos y particularidades de los signos alfabéticos, así como de automatizar las operaciones necesarias para su construcción, ha revolucionado el ámbito de la creación tipográfica y ha abierto el debate sobre la necesidad que el diseñador tipográfico pueda tener de conocer ciertos lenguajes de programación.

Partiendo de la base de que la tipografía digital es construida a partir de la descripción de los datos introducidos en el programa, conocer la sintaxis del lenguaje empleado nos puede ayudar gratamente no sólo a construir herramientas que agilicen nuestra tarea, sino también a comprender la configuración desde principios meramente descriptivos cuya corrección y pertinencia, en términos tipográficos, pasaremos a comprobar con posterioridad.



Ilustración 4. Producción de la fuente desde la sintaxis del lenguaje de programación y comprobación de resultados gráficos en la aplicación Robofont

Esta situación ha conducido a la creación del “diseñador-programador” que siendo conocedor de los lenguajes necesarios es capaz de crearse sus propias herramientas y trabajar la letra meramente desde la descripción de parámetros que la configuran. Este conocimiento otorga una extraordinaria libertad constructiva al diseñador, ya que le permite idear y producir un gran conjunto de formas armónicas a gran velocidad sin mayor detenimiento en la pura forma tipográfica, apenas cambiando un par de parámetros en la información de su fuente.

Sin embargo, hasta hace cerca de diez años tal libertad estaba sujeta por las constricciones de los programas de edición tipográfica, cuyos formatos derivados no permitían la flexibilidad necesaria para trabajar la forma en los términos de hoy. Con el objetivo de acabar con esta situación, en 2004 Erik van Bokland, Tal Lemming y Just van Rossum crean el formato UFO⁹. UFO es un formato de almacenamiento de información de la fuente que puede ser leído por cualquier plataforma y programa de edición tipográfica. Está escrito en el lenguaje de programación Python y basado en XML, ya que tiene por premisa ser fácilmente comprendido por personas sin un profundo conocimiento de este lenguaje, de manera que pueda ser manipulado por diseñadores que tengan unas nociones del mismo sin ser necesariamente programadores.

Si bien Python es el lenguaje en el que se basa UFO, RoboFab es la plataforma que lo acoge y permite exportar los archivos de los programas a este formato abierto. De este modo es posible manipular y transformar “objetos” a través de los plugins o extensiones instaladas en los programas de edición tipográfica como Fontlab, Glyphs o RoboFont. Los denominados “objetos” son en realidad módulos de datos escritos en Python que permiten crear herramientas e instalarlas en el programa de edición tipográfica. RoboFab¹⁰ es la librería que alberga estos módulos. Los principales módulos de RoboFab son el módulo *Fuente* y el módulo *Glifo*. El primero almacena y manipula la información del ajuste del kerning y del espaciado, así como la información de la fuente —nombre, copyright...—. El segundo almacena y manipula aquella información concerniente a cada glifo de la fuente.

⁹ LEMMING, Tal; BOKLAND, Erik van; ROSSUM, Just van. *Unified Font Object*. Disponible en <<http://unifiedfontobject.org/>>

¹⁰ LEMMING, Tal; BOKLAND, Erik van; ROSSUM, Just van. *Robofab*. Disponible en <<http://www.robofab.org/intro.html>>

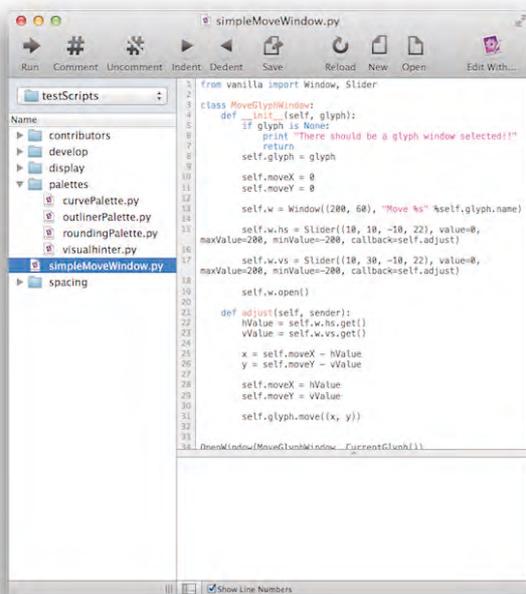


Ilustración 5. Programación de “objetos” en Python

RoboFab no es una herramienta para el diseño de fuentes, sino una plataforma o conjunto de herramientas que permite automatizar las tareas de diseño y de producción de las mismas. Detenernos en esta librería viene razonado por el hecho de que permite modificar los datos de una fuente sin tan siquiera abrir el programa de edición, lo que nos sitúa ante un cambio de paradigma en el proceso de creación tipográfica. Además, la posibilidad de conformar los programas de edición a la medida de las necesidades de nuestros proyectos y flujos de trabajo no solo agiliza el proceso sino que genera nuevos discursos, en los que el diseñador trabaja la tipografía de una manera indirecta, desplazando el tratamiento de las formas a las operaciones informáticas.

Establecer la letra como programación plantea, pues, un nuevo estadio en la configuración de las formas ya que, si bien éste no implica influencia alguna en el resultado gráfico, sí lo hace en los procesos mentales del diseñador entre la fase de ideación y la de producción. Así, el primero se basa en el tratamiento directo de la sustancia gráfica; y el segundo la desvirtúa y dota de virtualidad siendo tan solo real en el momento de su impresión.

CONCLUSIONES

La innovación tecnológica en el ámbito de la creación tipográfica vive hoy un periodo tremendamente fructífero. Aunque la libertad que nos plantea el uso de la programación en el proceso de creación de una fuente se relaciona más con la velocidad y fluidez de los flujos de trabajo del propio proceso, podemos encontrar nuevos discursos tipográficos que han tenido lugar a partir de comprender la letra como producto del lenguaje informático. En este sentido podemos destacar por un lado tendencias que se han venido produciendo en el diseño de tipografías durante los últimos 10 años, así como proyectos concretos.

En cuanto a las tendencias, podemos destacar la formulación de grandes familias tipográficas. La sencillez y agilidad con la que se pueden producir fuentes actualmente ha multiplicado el número de variantes que se incorporan en la familia. Estableciendo valores de peso, anchura y proporción óptica podemos producir una misma tipografía para usos muy distintos, dando lugar a un resultado más versátil que nunca. Por otro lado, las facilidades que aporta el formato OpenType, permite que en una misma fuente o archivo digital puedan combinarse alfabetos distintos regidos por unos mismos criterios de configuración gráfica. Ambos avances suponen, sobretodo, un cambio en la mentalidad del diseñador en el proceso de creación, que aunque siempre tuvo que enfrentarse al diseño pensando en la globalidad del sistema, solo ahora es capaz de trabajar en él simultáneamente, gracias al establecimiento de parámetros y a la posibilidad de control de los datos que los describen.

Así mismo cabe destacar la introducción de las fuentes por capas. Este tipo de fuentes recogen en su mayoría la tradición rotulista del siglo XIX y XX y responde al intento de rescatar la tridimensionalidad del medio bidimensional que antaño se conseguía a través de la combinación cromática. Esta tendencia es resultado del control sobre los lenguajes de programación que determinan la producción tipográfica digital y la libertad creativa derivada de tal control.

Entre los proyectos destacamos aquellos que trabajan sobre la programación como fuente de experimentación en la creación tipográfica, haciendo reflexionar sobre su actual papel y abriendo un campo para la especulación de nuevas posibilidades en el rígido campo de la tipografía. Entre estos proyectos destacamos la tipografía Beowolf¹¹, que en el amanecer de la tipografía digital defendía la letra como datos y código que puede ser modificado a partir de instrucciones precisas; por otro lado, y más cercana en el tiempo, destacamos la tipografía Artl Lanzallamas¹², una tipografía que experimenta sobre la alternancia de las formas en el mismo tiempo de lectura.

Como toda innovación tecnológica, la tipografía digital partió de la intención de imitar aquella que la precedía, mejorando los procesos de producción y las pesquisas tecnológicas que afectaban a la forma. Sin embargo, precisamente al igual que toda innovación tecnológica, la tipografía digital ha puesto sobre la mesa nuevas fórmulas de producción que más allá de facilitar las tareas permiten al diseñador dar rienda a nuevas propuestas, así como a trabajar sobre la forma dentro de una sintaxis verbal y no visual.

La democratización tecnológica ha favorecido la introducción de personas no especializadas en el campo de la tipografía. Sin embargo, pese a que este hecho puede no parecer algo positivo en un primer momento en cuanto a la calidad del resultado se refiere, no da lugar a connotación negativa alguna. Sea cual sea la dificultad o facilidad en el uso de las herramientas que nos permiten producir tipografías, éstas son solo herramientas, que habiendo sido diseñadas ya no por ingenieros informáticos sino por diseñadores tipográficos propiamente, su dominio viene siempre precedido de un profundo conocimiento de la sintaxis tipográfica. Esto supone que, en cualquier caso, estas herramientas proporcionan una inestimable ayuda en el proceso de creación al diseñador especializado y un buen aprendizaje práctico para el no especializado para comprender la letra como software y como programación en una dialéctica entre lo real y lo virtual.

FUENTES REFERENCIALES.

LAWSON, Alexander. "Type making from punch to computer ". En *Anatomy of a typeface*. David R. Godine Publisher Inc., Shangai, 2010, pp. 381-404.

SMEIJERS, Fred: *Counterpunch*. Hyphen Press, London, 2011.

KINROSS, Robin. "La modernidad después del movimiento moderno". En *Tipografía moderna*. Campgráfic, Valencia, 2008, pp. 179-206.

KNUTH, Donald E. "Computer & Typesetting Volume E". En *Computers Modern Typefaces*. Addison-Wesley, Reading, Massachussets, 1986.

LEHNI, Jürg. *Typeface as Programme* [en línea]. 2011. [https://www.typotheque.com/articles/typeface_as_programme]. [20-06-2014].

LEMMING, Tal; BOKLAND, Erik van; ROSSUM, Just van. *Unified Font Object* [en línea]. [21-06-2014]. [<http://unifiedfontobject.org/>].

LEMMING, Tal; BOKLAND, Erik van; ROSSUM, Just van. *Robofab* [en línea]. [<http://www.robofab.org/intro.html>]. [21-06-2014].

BOKLAND, Erik van y ROSSUM, Just van. *Fontfont Beowolf* [en línea]. 2013. [<http://lettererror.com/fontcatalog/fontfont-beowolf/>]. [7-07-2014].

PAMPATYPE. *Artl Lanzallamas* [en línea]. [http://www.pampatype.com/typefaces/arl/05_concept-arl.html]. [7-07-2014].

¹¹ Erik van Bokland y Just van Rossum, 1989 (Letterror). Disponible en <<http://lettererror.com/fontcatalog/fontfont-beowolf/>>

¹² Alejandro Lo Celso, 2008 (PampaType). Disponible en <http://www.pampatype.com/typefaces/arl/05_concept-arl.html>