

Consideraciones y recursos para la concepción de la forma en la arquitectura de la era digital

Considerations and resources for the conception of form in the architecture of the digital age

ERNESTO BUENO

Arquitecto, graduado por la Facultad de Arquitectura y Urbanismo de la Universidad Central de Venezuela, máster en Arquitecturas Genéticas por la Escola Técnica Superior d'Arquitectura de la Universitat Internacional de Catalunya, aspirante a doctor en el mismo Programa de Postgrado, instructor de scripting en el Máster en Diseño y Arquitectura en la ELISAVA, Escola de Disseny, Barcelona, España.

ernestobueno@gmail.com

Resumen

Ensayo que reivindica la generación y representación de la forma en la arquitectura, tomando en consideración basamentos teóricos recientemente revisados en lo que respecta a la topología en la arquitectura y uno de sus recursos de modelado, la NURBS. Se presentan antecedentes a las actuales técnicas generativas, desde la computación manual hasta la programación con lenguajes interpretados como el RhinoScript, del cual se expone un esbozo del estado de la cuestión dentro del área de la proyectación arquitectónica tanto a nivel profesional como académico, especialmente en el ámbito iberoamericano. Esto muestra los recursos con los que cuentan los diseñadores que están poniendo de manifiesto una nueva sensibilidad por la forma en la arquitectura de la era digital.

Palabras clave: topología; NURBS; diseño generativo; scripting.

Abstract

An essay that claims for the generation and representation of form in architecture, taking into consideration theoretical basements recently reviewed in terms of topology in architecture and one of its modeling resources, the NURBS. A background to the current generative techniques is presented, from manual computation to interpreted programming languages, like RhinoScript, from which it's shown a stub on the state-of-the-art within this area in architectural design in both professional and academic levels, especially in the Ibero-american region. These demonstrate what resources have the designers, who are showing a new form of sensitivity in the architecture of the digital age.

Keywords: topology; NURBS; generative design; scripting

Consideraciones y recursos para la concepción de la forma en la arquitectura de la era digital

Introducción

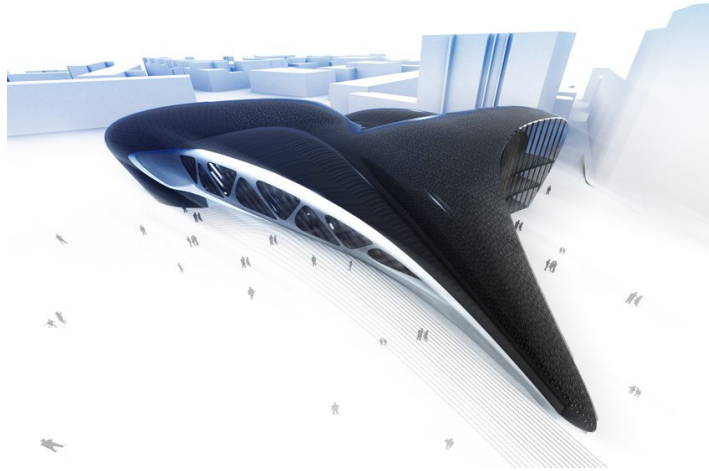


Fig. 1: Biblioteca PAU de Zaha Hadid, concebida con scripts de Marc Fornes.¹

No es ningún misterio para muchos arquitectos como Gabriela Troncoso [2006], el sostener que la era digital está produciendo una tendencia estética de formas redondeadas y complejamente curvas, por encima de las conocidas formas planas y angulosas², más representativas del movimiento moderno y sus descendencias, para muchos, aún en uso. Tan vigente está esta discusión, que se podría ver como un intento de explicar, en el ámbito español, lo que los profesores Medrano y Mikami acaban de mencionar en el editorial del número anterior como la fase heroica de los años 1950-1960 que aun define el acontecer arquitectónico brasileño [MEDRANO; MIKAMI, 2008], y que también podría extenderse al ámbito latinoamericano. En contraposición, la nueva tendencia no es en sí una innovación estética recientemente concebida, sino recientemente lograda gracias a los medios digitales actuales.

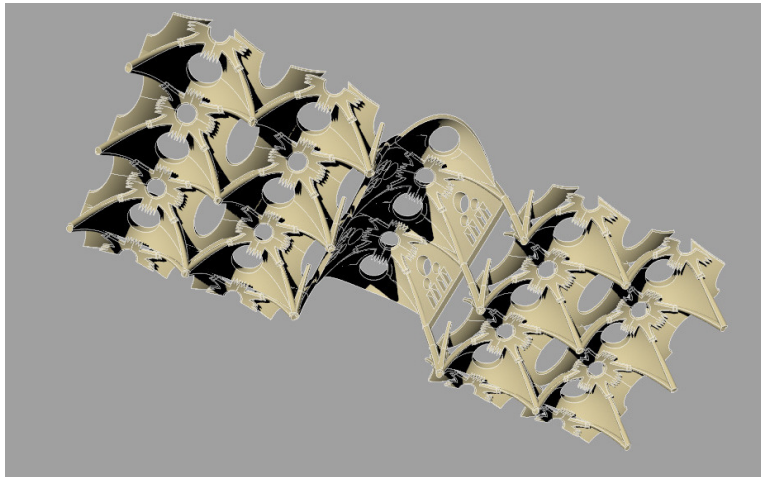


Fig. 2: Fragmento de la Sagrada Familia de Gaudí modelado con NURBS por Manuel Hidalgo [2008].

Si relacionamos estas formas curvas con las biológicas podemos afirmar que han estado con nosotros desde el origen de la arquitectura. Como lo pone de manifiesto Alberto Estévez [2005], las curvas complejas que imitan a la naturaleza se pueden encontrar desde el Antiguo Egipto hasta el modernismo, en la mayoría de los casos como elemento ornamental, y en todos limitado por la tecnología de construcción disponible. Son casos en los que el diseñador ha podido concebir un elemento, posiblemente de mayor complejidad, de lo que la realidad del momento podía fabricar, e incluso a veces ni siquiera representar (como nos suelen decir al comienzo de la carrera, generalmente en alguna asignatura de dibujo arquitectónico: si no lo puedes dibujar, no lo puedes construir). Tan solo unos pocos se atrevieron a empujar estos límites más allá, como Gaudí [FIG. 2] o el posterior Frei Otto o el actual Calatrava, por nombrar los más representativos. El resto de conceptos, precisamente por su aplicación predominantemente ornamental, fueron susceptibles a desaparecer con el funcionalismo del movimiento moderno [ESTÉVEZ, 2005].

Durante la segunda mitad del siglo XX hemos sido testigos de incursiones en la complejidad de la forma arquitectónica que, según Mark Wigley [1995] han sido fallidos intentos deconstructivistas y teorías derrideanas mal interpretadas³. En relación a la forma, críticos como él sostienen, con razón, que tanta complejidad pierde su validez sin un argumento sólido que la justifique (más allá de la fascinación por una estética circunstancial [PORTO, 2006]), o por lo menos que la ordene o describa. Pero poco a poco, la arquitectura ha ido recuperando los conceptos que implican curvaturas y sistemas complejos, en parte por el desarrollo de conceptos más adecuados al fenómeno actual, como algunas teorías arquitectónicas basadas en ideas revisadas de Deleuze, la termodinámica y la topología,

capaces de definir en la forma una lógica interna, generativa o genética, según Manuel DeLanda [2002]; y en parte por el advenimiento de las nuevas tecnologías, que permiten no sólo la concepción, sino también la representación y fabricación de estas reinterpretadas formas [KOLAREVIC, 2003].

Computar la forma

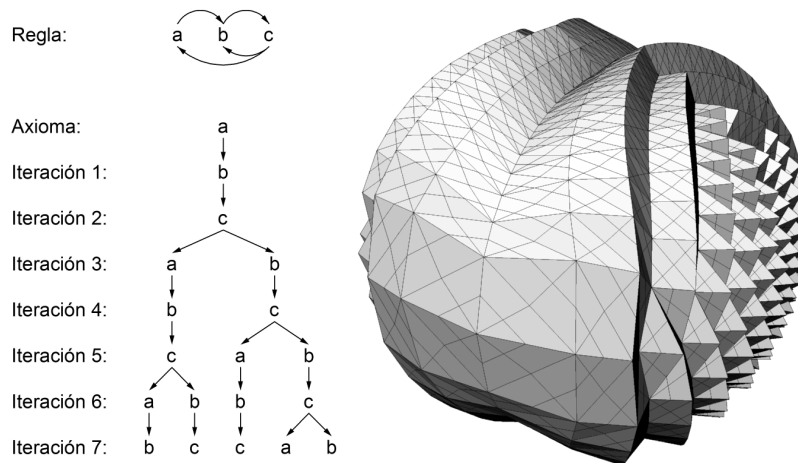


Fig. 2: Ejemplo de D0L-system y superficie mapeada. Desarrollo y modelado: E. Bueno.

Sobre el tema que aquí nos incumbe: la computación en la concepción y representación de la forma arquitectónica, también se ha hablado mucho. Desde el nacimiento del CAD en la década de 1960 se pensaba en el uso de la computadora para asistir tanto la creación como la representación de la forma, aún cuando ha sido sólo lo último el aspecto más conocido y explotado de esta tecnología [CELANI, 2003], principalmente por la complejidad que implica la evaluación, y la toma de dediciones creativas en el proceso de proyecto, en comparación con el problema de la simple representación digital de la información y los procesos de dibujo de precisión. Esto se tradujo en un obstáculo difícil de superar para los desarrolladores de software, quienes apurados por presiones de mercado, se centraron mayormente en el problema de la representación, dando como resultado la actual hegemonía del software basado sólo en dibujo CAD y animación en el ejercicio de la arquitectura (encabezadas por AutoCAD y 3ds Max), muy por encima de cualquier herramienta que verdaderamente asista el proceso creativo, y por encima incluso de otros programas que, aún siendo carentes de ello, son más orientados a la profesión⁴.

Afortunadamente, siempre han existido los que se dedican a la investigación del CAD para un uso más creativo en el proceso de proyecto, como lo demuestran los estudios de



Gabriela Celani [2003] y el LAPAC acerca de la gramática formal ⁵, o los de Kostas Terzidis [2006] acerca de otros procesos algorítmicos. En ambos casos, la idea es computar formas sistemáticamente generadas para componer arquitectura; y la manera de lograrlo es extendiendo la funcionalidad del CAD a través de programación en lenguajes interpretados, o de scripting.

En el caso de Terzidis, además de la técnica de scripting en sí, la ausencia de una única metodología generativa (como la gramática de la forma) es compensada con una respuesta técnica al pensar físico y filosófico previamente mencionado (incluyendo en este caso a Parmenides), sin perder la objetividad que podría dejar en pura ilusión el *“intento de capturar lo elusivo”* en el proceso de proyecto [TERZIDIS, 2006].

Una aproximación similar se puede ver en el trabajo de Karl Chu [2005] que hace referencia a Deleuze, pero que en realidad hunde sus bases en la monadología de Leibniz vista desde la mirada informática de Gregory Chaitin [2003] para crear la *“mónada computacional”* como unidad básica e irreducible para la generación de una arquitectura genética ⁶. Con el concepto de mónada, entendido en su sentido más abstracto como la unidad de información metafísica, se desarrolla un algoritmo en el que un conjunto de símbolos (mónadas) son sometidos a una serie de reglas simples (axiomas) para desarrollar un sistema proto-arquitectónico que en su conjunto puede ser visto como otra unidad computacional capaz de auto replicarse [CHU, 2005].

En términos de programación, esto se lleva a cabo a través de algoritmos recursivos, pero desde un punto de vista más abstracto, se trata de un proceso de generación y ordenación de información que, dependiendo de su complejidad, podría llevarse a cabo a mano, es decir, computación manual ⁷, tan elemental como la estática gráfica desarrollada por Gaudí para obtener por ejemplo, sus arcos catenarios [HUERTA, 2003], sólo que en este caso tratando el problema de la autorreplicación con reglas de sustitución de símbolos como las encontradas en los DOL-systems. Al mapear estos sistemas sobre superficies ⁸, se obtiene este conjunto monádico que exhibe una complejidad emergente, colaborante pero independiente de la complejidad de la superficie en sí.

Y es que la superficie es de por sí responsable de su propio comportamiento, si la tomamos desde el punto de vista topológico. Es decir, en vez de conformarnos con manejar superficies planas como elemento base y superficies de doble curvatura como elementos muy complejos, podríamos entenderlo como transiciones deformadas de una misma variedad topológica. No importa tanto la distinción entre suelo, pared y techo, sino el hecho de que son

superficies delimitadoras de espacios, que si se les da continuidad, se pueden manejar como una sola *2-variedad*⁹, ya no definida por sus dimensiones cartesianas (x, y, z), sino por las direcciones inherentes a su topología, sus parámetros de posición (u, v). De esta manera se puede describir esta superficie que en determinado punto pueda tener una pendiente horizontal, mientras que en otro sea vertical, u horizontal en sentido opuesto.

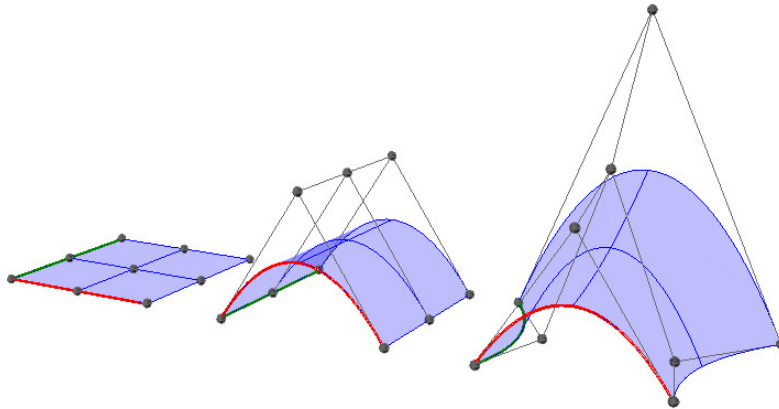


Fig. 4: Variaciones paramétricas de una misma superficie NURBS. E. Bueno.

Para implementar esto en el proyecto arquitectónico, no sólo es necesario un entendimiento básico de topología, sino poseer la herramienta con la cual generarla y controlarla: un modelador de NURBS (*Non Uniform Rational B-Splines*). Las NURBS son un poderoso recurso para representar las curvas (*1-variedad*) y superficies (*2-variedad*) en el medio computacional¹⁰. Se pueden definir y manejar a través de sus puntos de control o sus nodos para aplicar cualquier transformación topológica: estiramiento, torsión, plegamiento, etc. [FIG. 4], con precisión de CAD, y necesitando mucho menos puntos de control que si se trabajara con mallas o con redes de polígonos, haciéndolo práctico y viable.

Esta condición paramétrica inherente a la NURBS es también considerada como una lógica interna de la forma. Si se combina con otras técnicas como el diseño paramétrico-asociativo [KOLAREVIC, 2003] o el scripting se puede desarrollar de manera objetiva y controlada un alto grado de complejidad en el diseño, que si además se aplica para solucionar problemas, digamos, de confort ambiental o de sustentabilidad, puede enriquecer enormemente el proyecto, más allá de la estética.

El Scripting como estrategia de proyecto

```

6 Call FaceSurface()
7 Sub FaceSurface()
8
9 'Variables
10 Dim strSurface
11 Dim arrDomainU, arrDomainV
12 Dim ParamU, ParamV, arrPoint
13 Dim i, j
14 Dim SecU, SecV
15
16 'Cantidad de divisiones
17 SecU = 10
18 SecV = 10
19
20 'Input
21 strSurface = Rhino.GetObject("Selecciona una superficie")
22 If IsNull(strSurface) Then Exit Sub
23
24 arrDomainU = Rhino.SurfaceDomain(strSurface, 0)
25 arrDomainV = Rhino.SurfaceDomain(strSurface, 1)
26
27 'Looping!
28 For i = 0 To SecU
29   For j = 0 To SecV
30
31     'Divisiones
32     ParamU = arrDomainU(0) + i * ((arrDomainU(1) - arrDomainU(0)) / SecU)
33     ParamV = arrDomainV(0) + j * ((arrDomainV(1) - arrDomainV(0)) / SecV)
34
35     arrPoint = Rhino.EvaluateSurface(strSurface, ParamU, ParamV)
36     Call Rhino.AddPoint(arrPoint)
37
38   Next
39 Next
40
41 End Sub

```

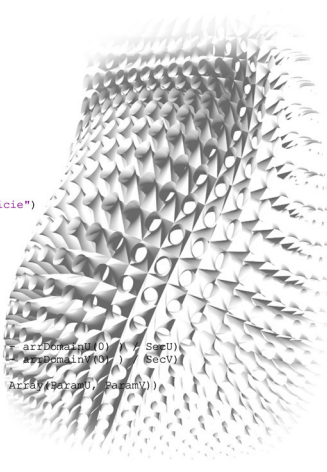


Fig. 5: Fragmento de código RhinoScript y modelo generado. Código: C. de la Barrera. Modelado e imagen: E. Bueno.

Rodrigues [2008] menciona el scripting para “transformar programas de CAD comunes en software paramétrico”. Uno de los programas con mayor crecimiento en el área es el Rhinoceros, un modelador de NURBS con su propio lenguaje interpretado, el RhinoScript. Este es un lenguaje basado en VBScript, que al igual que el VBA (usado en AutoCAD), es muy intuitivo y versátil, dando la oportunidad, no sólo de trabajar con variables, implementar algoritmos generativos, usar funciones de Visual Basic, del Rhino, o crear unas propias, sino también de intervenir en la interfaz del programa, dando la oportunidad de personalizarlo y hasta crear Plug-ins propios. Todo esto con el soporte de NURBS y el manejo de topologías en continua traducción al espacio vectorial cartesiano.

Recientemente se ha visto una explosión en el uso de RhinoScript en proyectos arquitectónicos, tanto a nivel profesional, como académico. Es cada vez mayor la cantidad de diseños, resueltos de esta manera, que figuran en concursos internacionales y nuevos proyectos arquitectónicos publicados en las principales revistas de nuestra disciplina. Empiezan a sonar nombres como Andrew Kudless, Marc Fornes, Skylar Tibbits, Stylianos Dritsas, entre otros, como arquitectos haciendo del scripting su estrategia de diseño. Por ello, oficinas de arquitectura mundialmente conocidas como Zaha Hadid Architecture [FIG. 1], Asymptote Architecture, Future Systems, DECOi, Acconci Studio, han comenzado a incluir en su plantilla a algunos de estos diseñadores/desarrolladores de scripts.

En el ámbito académico, escuelas de arquitectura como la del MIT, la Columbia University, Pratt Institute, SCI-ARCH, Universität Kassel, ya han incorporado la enseñanza de

RhinoScript. En Iberoamérica se empieza a estudiar en workshops como el de Carlos de la Barrera en la ESAP, Porto, junto a Gonzalo de Castro Henriques ¹¹; en el IAAC, Barcelona, junto a Luis Fraguada ¹² y en la ELISAVA, Barcelona, junto a Ernesto Bueno ¹³ [FIG. 5]; o como los de Griffith, Snaveley y Cardoso en la Universidad de Chile ¹⁴, o Griffith, Cardoso y Herrera, en la Universidad Peruana de Ciencias Aplicadas ¹⁵, ambos casos coordinados por Pablo Herrera [2007].

Por todo esto, la Internet se está llenando de recursos para el aprendizaje y desarrollo de scripts, como lo documenta Herrera ¹⁶. Panelización, patrones estocásticos, teselación, diagramas de Voronoi, fractales, autómatas celulares, son los argumentos formales con los que la complejidad de la naturaleza ahora es llevada a la arquitectura, solucionando problemas de diseño por medio de las nuevas técnicas computacionales.

Discusión

Las NURBS son una representación computacional más cercana a la realidad topológica de la naturaleza, y por tanto, su uso en arquitectura permite minimizar las limitaciones de las herramientas computacionales relacionadas con la representación y permitir la formalización de conceptos más adecuados con la realidad actual.

El scripting proporciona una estrategia de diseño más completa al extender el alcance de las herramientas de software. Si ésta técnica se aplica a un modelador de NURBS, como el Rhino, se potencian aún más sus posibilidades, al permitir más control de la complejidad de la forma arquitectónica a través de su lógica interna. Esto permite la implementación de recursos algorítmicos que enriquecen el abanico de soluciones de diseño, poniendo de manifiesto una nueva sensibilidad por los conceptos matemáticos y computacionales que dan pie a la emergencia de una arquitectura representativa del siglo XXI.

No por ello está completamente exento de incursiones infértiles, o de volver a caer en desvíos hacia la fascinación por una estética circunstancial. Como arquitectos, debemos ser autocríticos en muchos aspectos, incluyendo el que se refiere a la experimentación formal. El argumento de la lógica interna puede ser lo suficientemente fuerte para la forma en sí, pero si esta lógica no está directamente relacionada con la razón de ser del espacio habitable que definirá la forma en cuestión (como la percepción, la función o el contexto [PORTO, 2006]), podría quedar tan solo en un estudio estético de la forma en su sentido abstracto, que por interesante que sea, no aportaría mucho a la evolución de una teoría arquitectónica completa.

Atrás ha quedado ahora la discusión (a mi parecer, un problema generacional) sobre la capacidad de las herramientas CAD para mantener enfocada la escala y contexto de la idea cuando se desarrolla completamente en el computador; pero cabría preguntarse si con el scripting, al ser la programación un entorno hasta ahora ajeno a la naturaleza del ejercicio del arquitecto, no se corre este riesgo, que pueda desviarnos de los argumentos originalmente válidos hacia una fascinación de la forma por la forma, o incluso de la forma por su información computacional. En principio, opino que, siendo conscientes de ello, es un riesgo que, hasta cierto punto, vale la pena correr.

Referencias

- BUENO, Ernesto. *Experimentación morfogenética en el diseño arquitectónico*. 2005. 134 f. Disertación (Estudios Avanzados y Suficiencia Investigadora para el Doctorado en Arquitecturas Genéticas) – Escola Técnica Superior d'Arquitectura – ESARQ-UIC. Barcelona, 2005.
- CELANI, Gabriela. *CAD Criativo*. Rio de Janeiro: Campus-Elsevier, 2003.
- CHAITIN, Gregory. "Leibniz, Information, Math and Physics". En: *International Wittgenstein Symposium*, Kirchberg, 2003. Disponible en: <http://www.cs.auckland.ac.nz/CDMTCS/chaitin/kirchberg.pdf>. Acceso en: 15 octubre 2008.
- CHU, Karl. "Metafísica de la Arquitectura Genética y la Computación". En: *Arquitecturas Genéticas II: Medios digitales y formas orgánicas*. Barcelona: SITES Books/ESARQ-UIC, 2005. p. 132-157.
- ESTÉVEZ, Alberto. "Arquitectura Biomórfica". En: *Arquitecturas Genéticas II: Medios digitales y formas orgánicas*. Barcelona: SITES Books/ESARQ-UIC, 2005. p. 18-53.
- DELANDA, Manuel. "Deleuze and the Use of the Genetic Algorithm in Architecture". In: *Designing for a Digital World*. Nueva York: Wiley, 2002. Disponible en: <http://www.cddc.vt.edu/host/delanda/pages/algorithm.htm>. Acceso en: 3 marzo 2008.
- HERRERA, Pablo. "Solución de problemas relacionados al diseño de superficies complejas: Experiencia de programación en la educación del arquitecto". En: *Libro de ponencias del 11º Congreso de SIGraDi*. México, 2007, p. 97-101.
- HIDALGO, Manuel. *Las formas edificatorias de la arquitectura de Antonio Gaudi*. Madrid: ETSAM, 2008. Disponible en: http://download.mcneel.com/s3/mcneel/rhino/4.0/docs/es/Las_formas_edificatorias_de_la_Arquitectura_de_Antonio_Gaudi.pdf. Acceso en: 29 julio 2008.
- HUERTA, Santiago. "El cálculo de estructuras en la obra de Gaudí". *Ingeniería Civil*, Madrid, [s. v.], n. 129, p. 121-133, 2003. Disponible en: <http://oa.upm.es/554/>. Acceso en: 27 abril 2008.
- KOLAREVIC, Branco. *Architecture in the Digital Age: Design and Manufacturing*. Londres: Spon Press, 2003.

MEDRANO, Leandro; MIKAMI, Silvia. "Parc #2: Editorial". *(PARC) Pesquisa em arquitetura e construção*, v. 1, n. 2, Campinas, junio 2008. Disponible en: <<http://www.fec.unicamp.br/~parc/vol1/n2/vol1-n2-editorial.pdf>>. Acceso en: 18 julio 2008.

PORTO, Gentil. "O diagrama e a matemática da arquitetura". *PARC*, v. 1, n.1, Campinas: octubre 2006. Disponible en: <<http://www.fec.unicamp.br/~parc/vol1/n1/parc01porto.pdf>> Acceso en: 18 julio 2008.

RODRIGUES, Gelly. "A geração dos sistemas CAD". *PARC*, v. 1, n. 2, Campinas, junio 2008. Disponible en: <<http://www.fec.unicamp.br/~parc/vol1/n2/vol1-n2-rodrigues.pdf>>. Acceso en: 18 julio 2008.

TERZIDIS, Kostas. *Algorithmic Architecture*. Oxford: Architectural Press, 2006.

TRONCOSO, Gabriela. *Las formas de la arquitectura en la era digital*. En: *Ponencias - Congreso Virtual de Arquitectura - ConVirArq_2006*. <<http://convirarq2006.ath.cx/ponencias/Troncoso.pdf>>. Acceso en: 27 enero 2007.¹⁷

WIGLEY, Mark. *The Architecture of Deconstruction: Derrida's Haunt*. Cambridge: The MIT Press, 1995.

Notas

¹ Imagen: FORNES, Marc. *Theverymany* <<http://www.theverymany.net/>>.

² Troncoso se basa en un estudio psicológico de Wolfgang Köhler en 1927, en el que se inventan las pseudopalabras maluma y takete para establecer una relación entre las formas fonéticas y las formas gráficas al corresponderle una forma curva y angulosa respectivamente. De esta manera, identifica formas *malumianas* y *taketeanas* en la arquitectura [TRONCOSO, 2006].

³ En referencia a las ideas de Eisenman, Wigley afirma que la mala interpretación que él hace de la Deconstrucción de Derrida radica en que se empeña en la relación entre el lenguaje arquitectónico y la filosofía, en vez de la relación entre el lenguaje filosófico y la arquitectura [WIGLEY, 1995].

⁴ Me refiero a programas que implementan formas de modelado más arquitectónicas, ya sea dibujando directamente elementos arquitectónicos y constructivos en vez de líneas y cajas, o usando símbolos y/o bloques AEC. Por ej.: ArchiCAD, VectorWorks Architect o Autodesk Architectural Desktop, del cual se puede ver un análisis en: CELANI, Gabriela; GODOI, Giovana; RODRIGUES, Gelly. "O processo de projeto arquitetônico mediado por computador: Um estudo de caso com o Architectural Desktop". In: *Proceedings of GRAPHICA 2007*. Curitiba, 2007. Disponible en: <<http://www.fec.unicamp.br/~lapac/download/celani-godoi-rodrigues-2007.pdf>>. Acceso en: 21 julio 2007.

⁵ La gramática de la forma es una metodología originalmente aplicada a la arquitectura por W. Mitchell y G. Stiny (entre otros) desde los años 1970 y continuada y difundida por la profesora Celani y su línea de investigación. Ver en: <<http://www.fec.unicamp.br/~lapac/public.htm>>.

⁶ A diferencia de DeLanda, que se basa en Deleuze para formar su idea de arquitectura genética, Chu usa los estudios de Chaitin acerca de Leibniz. Aún cuando existe también un estudio de Deleuze al respecto, Chu no lo deja explícito en su *monadología genética*.

⁷ Con computación manual me refiero al procesamiento manual de información, es decir, computar sin computadora.

⁸ Chu propone la siguiente metodología: una vez hecho el D0L-system, se aplica a una superficie dada, con una técnica de mapeo en la que a cada símbolo se le asignan valores de distancia en la tercera

dirección (normal a la superficie), que definen los puntos de curvas periódicas que se unen para formar el conjunto resultante [BUENO, 2005].

⁹ Ver en: <<http://es.wikipedia.org/wiki/Superficie>>.

¹⁰ Ver en: <<http://en.wikipedia.org/wiki/NURBS>>.

¹¹ RhinoScripting Workshop at Esap, Porto

<<http://www.designemergente.org/rhinoscripting/rhinoscripting-workshop-at-esap-oporto/>>.

¹² Scripting Seminar at IaaC

<<http://www.designemergente.org/rhinoscripting/final-review-seminar-scripting-at-iaac/>>.

¹³ RhinoScripting Workshop at Elisava, Barcelona

<<http://www.designemergente.org/rhinoscripting/rhinoscripting-workshop-at-elisava-barcelona/>>.

¹⁴ Soluciones de Diseño Computacional, Santiago de Chile

<<http://espaciosdigitales.org/santiago/>>.

¹⁵ Soluciones de Diseño Computacional, Lima

<<http://espaciosdigitales.org/lima/>>.

¹⁶ Rhinoscripting Resources

<<http://rhinoscriptingresources.blogspot.com/>>.

¹⁷ Actualmente, el sitio Web del ConVirArq no está en línea. A los interesados en obtener una copia de la ponencia, sugiero ponerse en contacto con la autora Gabriela Troncoso:

<gabriela.troncoso@gmail.com>.