

VBSRIPT EN LA EDUCACIÓN DEL ARQUITECTO:
ESTRATEGIAS Y MÉTODOS DURANTE Y DESPUES DE LA IMPLEMENTACIÓN

Resumen. La implementación de *Rhinoscript* en facultades de arquitectura en Latinoamérica se realizó en Santiago de Chile (2006, 2007) y Lima (2008), estableciendo algunos criterios para explorar formas complejas usando la programación, con estudiantes de pregrado bajo estrategias de postgrado. Después de documentar los resultados de las dos primeras experiencias [1], se hallaron métodos que se usaron durante la implementación. El objetivo es evidenciar como las experiencias llevadas a cabo pueden aplicarse a otros contextos regionales, proponiendo tipologías para empezar y recursos disponibles que permitan a los estudiantes desarrollar por si mismos nuevas exploraciones.

Palabras Claves. Digital Education, End User Programming, Rhinoscripting.

I. INTRODUCCIÓN

Visual Basic (VB) en su modalidad *VBScript* se viene usando para explorar formas complejas con técnicas que permiten personalizar un *software* interactivo a una forma y no como ocurrió en algunos casos, que el resultado de nuestras formas siguió al *software* [2]. Es así que en los últimos años, la programación se ha convertido en una de las nuevas técnicas de representación y construcción en arquitectura [3], aplicadas tanto al diseño [4] y como a la fabricación [5].

Estas experiencias se desarrollaron principalmente en maestría y doctorado en EE.UU. y Europa hacia el 2003 y aunque hubieron experiencias previas en diseño con *Mel* en Autodesk Maya [6], [7], *Rhinoscript* que era usando para resolver problemas de fabricación [8] se difundió rápidamente no sólo en diseño, sino como herramienta para el análisis, visualización y documentación [9]. En estas experiencias, aunque los instructores dedicaron horas en colaborar con los estudiantes la producción de *scripts*, cada estudiante exploró y desarrolló habilidades por si mismo en base a ensayo y error. La referencia [10] evidenció que las tecnologías digitales se propagan con mayor velocidad que la implementación pedagógica, y el autoaprendizaje es una manera de adaptar fácilmente la programación a problemas particulares de diseño.

Estos talleres de diseño usando algoritmos por lo general tienen como requisito solicitar una propuesta de trabajo al candidato antes de empezar, la que finalmente encaja dentro de la temática del taller (por ejemplo las *Independent Activities Period* en MIT y los *SmartGeometry*). La solución a cada propuesta se inicia con el método de descomposición del problema en partes. Temas ampliamente tratados por Christopher Alexander en *Notes on the Synthesis of Form* y Mark D. Gross en *Design as exploring constraints*.

Bajo esta premisa, algunas de las estrategias implementadas en estos talleres se basan en los siguientes métodos: 1) *Rule-Based* en el que se identifican formas y reglas de estructura tomando como base un sistema biológico y han sido usadas en los talleres de Tim Schork (*Royal Melbourne Institute of Technology University*, Australia), Tobias Wallisser (*Stuttgart State Academy of Art and Design*, Alemania) y por John

Snavely (*IAP MIT*, EE.UU). 2) Otra es la que se aplica sobre o entre superficies y es llamada *On/Between Surfaces* que se han usado en los talleres de Carlos de la Barrera (*IaaC*, España). En todos los casos la exploración se sostiene por variables que en conjunto resuelven un mismo tipo de problema con exploraciones basadas en un mismo tema.

Por el contrario, la implementación realizada en Latinoamérica se hizo bajo otras condiciones. Ello porque aún usamos el *software* como instrumento de representación y automatización del dibujo, y el diseño con algoritmos aun se manifiesta en estado embrionario, planteando la exploración libre y que fueran de interés de cada equipo, esperando que en base a los resultados se pueda establecer un grupo de soluciones y temas para futuras implementaciones

II. MATERIALES Y MÉTODOS

Se utilizó *Rhino* por haber sido diseñada para diseñar y fabricar sobre otras tecnologías creadas para animación y visualización en general o como *Autodesk Maya* para la industria del cine y *3DSMax* para videojuegos de forma particular.

Las experiencias se realizaron en la Universidad de Chile (2006 y 2007) y en la Universidad Peruana de Ciencias Aplicadas (2008), evaluando 23 grupos con un total de 66 estudiantes. Una selección de estos trabajos se presenta en la Fig. 1. y están disponibles en <http://www.espaciosdigitales.org/>

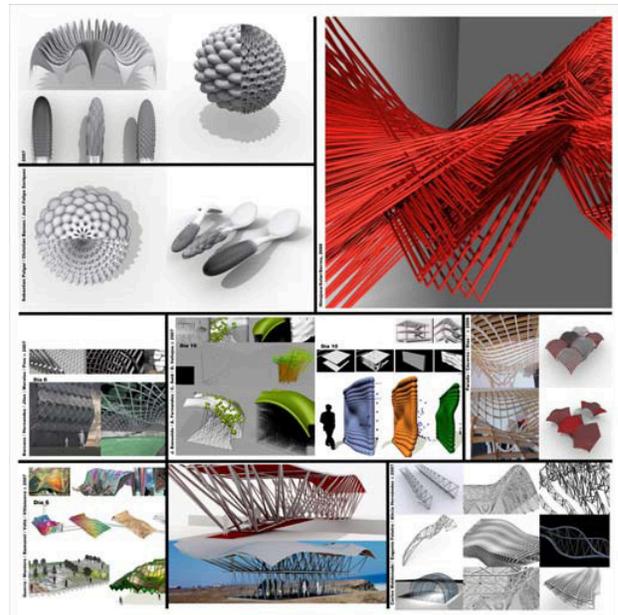


Figura 1. Rhinoscripting Workshops. Talleres en Santiago de Chile y Lima

Para iniciar el proceso, no se propuso ninguna restricción ni orientación previa sobre los temas que elegirían los participantes, así como ninguna bibliografía antes del taller. Como se volvió a comprobar [1], tampoco fue requisito haber

utilizado *Rhino* ni tener conocimientos de programación. Los talleres se implementaron en promedio de cuatro a seis días y es por estas limitaciones de tiempo que la única condición establecida fue solicitarle a los alumnos se agrupen con un mínimo de dos y máximo de cuatro, y presentaran sus propuestas al día siguiente de la primera sesión.

En esta investigación se buscó saber si los alumnos perderían dependencia a la interfaz del *software* utilizando este sólo de forma indirecta y que los resultados se construirían en base a variables que cada equipo consideraba importante para resolver el problema que se proponía trabajar, siendo la interfaz, una plataforma sobre la cual el algoritmo operaría.

Lo que cada equipo tenía claro era que las técnicas que iban a emplear les permitirían generar, controlar y manipular superficies complejas como se les indicó en la primera sesión.

Para evidenciar la validez de las propuestas, cada equipo debía familiarizarse con el referente elegido, esto implicaba entender la geometría y encontrar las reglas que la gobernaban, estructurando el problema en una secuencia de pasos u órdenes llamado pseudocódigo. Para que cada alumno pueda ordenarle a *Rhino* como ejecutar cada tarea, era necesario que los participantes conocieran la sintaxis para que pudieran controlar cada una de las variables involucradas, temas que fueron tratados en la segunda sesión. Cada alumno tenía que utilizar medios analógicos para geometrizar el problema, dibujando y expresando sus ideas con diferentes medios. Pensando como lo resolverían, sin pensar como el *software* lo solucionaría, deconstruyendo las acciones hasta obtener un primer modelo generado algorítmicamente. Cada uno de estos datos tenía que ser ingresado línea por línea con un editor. *Rhino* incluye un editor, sin embargo, para facilitar que la sintaxis de *RhinoScript* se diferencie de las variables y datos en la secuencia escrita, decidimos proporcionarles el editor gratuito *Notepad2* desarrollado por Florian Balmer. Después de terminar el *script* sin errores de sintaxis, lo que siguen son exploraciones sobre diferentes superficies. En la Fig. 2. se presenta una lámina que incluye el pseudocódigo y muestra las exploraciones sobre superficies hechas con el *script* creado.

BITÁCORA

Para evaluar si los resultados de estas experiencias podrían ser utilizados por otros arquitectos o diseñadores y determinar el origen de estos, entre enero y setiembre del 2008 se implementó una bitácora o *blog* con la información detallada sobre arquitectura y programación en <http://arquitecturayprogramacion.blogspot.com/> y que incluía no sólo los talleres descritos (con presentaciones y *scripts*), sino otros que fueron apareciendo durante este año en diferentes partes del mundo. Se implementó luego en julio del mismo año, uno específicamente sobre recursos para *rhinoscript* con enlace en <http://rhinoscriptingresources.blogspot.com/>. En él se incluyó una lista de otros *blogs* que contribuían con experiencias similares incluyendo temas como videos, tesis, ponencias, manuales y métodos, proporcionando información precisa sobre cada tema.

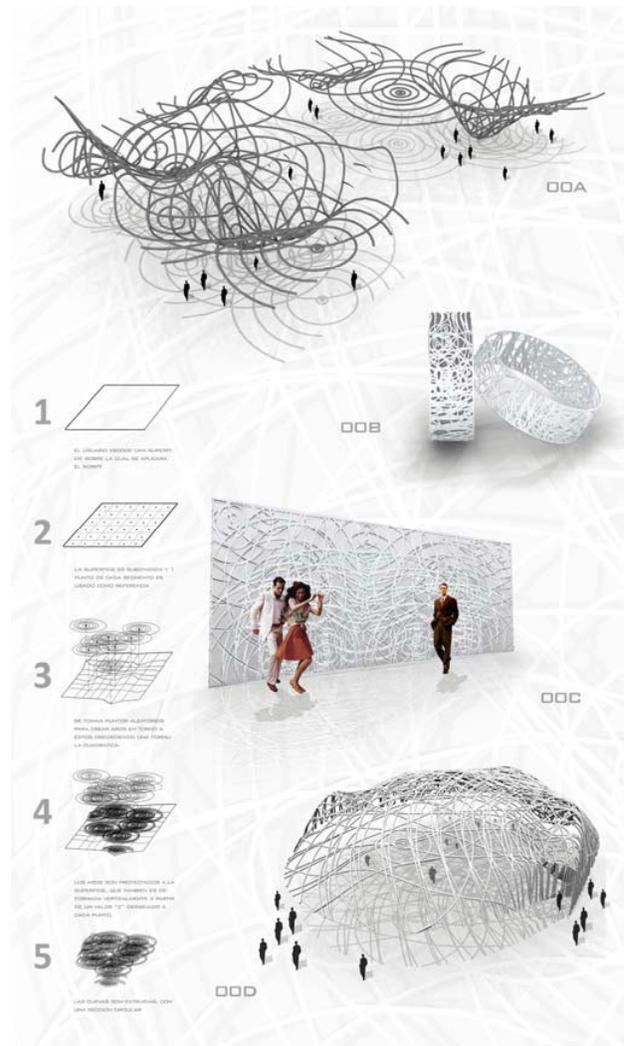


Figura 2. Piedras sobre agua. Pseudocódigo: 1) Tener 2 planos puros paralelos. 2) Definir puntos donde caen las piedras. 3) Características de cada punto (tamaño de piedra, altura, fuerza, masa) y 4) Características de reacción de cada plano. RhinoScripting Workshop Lima. Por Mónica Freundt, Cecilia Antunez de Mayolo, Juan Pablo Acosta y Rodrigo Barreto.

III. RESULTADOS Y DISCUSIONES

Se evidenció que los participantes esperaban que les diéramos algunas sugerencias sobre que tema desarrollar. Muchas de las preguntas se referían a los tipos de herramientas que el *software* utilizaba o que tipo de problemas podrían resolver. Pero en vista que el objetivo era que cada uno trazara su propio camino se decidió no proporcionarles esa información. Otro grupo estaba interesado en conocer que se había hecho en anteriores talleres por lo que se les proporcionó las páginas donde se alojaban las versiones de los talleres. Esta fue tal vez la etapa mas crítica de los talleres, porque definía lo que cada equipo haría en los días siguientes. Al final de la segunda sesión cada equipo presentó sus propuestas y la misma dinámica se realizó en los tres talleres. De los 23 grupos, 9 adoptaron referentes que partían de una

forma biológica (esponjas, gotas de agua, músculos o semillas de girasol) y 14 partieron de una forma arquitectónica (estructuras de soporte, de revestimiento, paneles o divisiones).

Según los resultados y la geometría utilizada para producir la forma, 6 grupos crearon formas geométricas, 9 grupos crearon *scripts* que se aplicaban a superficies mientras que 8 grupos hicieron lo propio cubriendo el vacío entre dos superficies.

Aunque la implementación se realizó con estrategias de postgrado, se comprobó que los problemas se resolvían en la medida que cada alumno lo descomponía y veía en que momento el algoritmo podría ser una ayuda. Algunos alumnos manifestaron que el proceso fue más interesante que el resultado, como también se evidenció en otros talleres [8], pero todos coincidieron en que la exploración de formas era más rápida usando algoritmos.

Como ocurrió en otro taller [9], la manera de abordar el problema de diseño no siguió métodos tradicionales ni de representación ni pensamiento. Las plantas, cortes o elevaciones serán sólo una consecuencia de este proceso porque no fueron punto de partida en esta manera de resolver un problema.

Se comprobó que la dependencia al ícono, trajo consigo que los participantes olviden reglas básicas de geometría (como hallar el punto medio de una recta o hallar el área de un polígono). Haciendo lento el proceso de geometrización del problema.

También se detectó que a un año de cada taller los estudiantes habían perdido la continuidad por incluir esta experiencia en la práctica diaria o regular. A pesar que el material fue publicado, el aprender por sí mismo [9] no tuvo la continuidad que se esperaba. Aunque se implementaron recursos disponibles con información de todas las presentaciones, así como los *scripts* creados para que puedan ser revisados y alterados, el único resultado destacable de esa iniciativa se construyó el 2007. Fue *Structural Hair* de Margaret Dewhurst, Proyecto Ganador del *Architectural Association Student Summer Pavillion* quien basó su *script* en el trabajo de uno de los equipos del primer taller (Santiago 2006). Este año, se ha motivado el autoaprendizaje a través del *blog* creado, motivados además por la difusión de sus trabajos y la participación de los mismos en exhibiciones en Bogotá, Lima y Beijing, obligándose a trabajar en nuevas exploraciones.

Por otro lado, la publicación de los recursos para *RhinoScript* incluidos en el *blog* dejó una estadísticas elaborada entre el 21 de Julio y el 21 de Setiembre del 2008: Con respecto a la procedencia de los usuarios: el 53.1% procedió de Europa (siendo los cinco mas altos porcentajes de usuarios, Italia 9.9%, Reino Unido 9.2%, España 7.9%, Alemania 5.7% y Austria 4.7%). El 19.7% pertenece a EE.UU., el 14% a Latinoamérica (con los cuatro porcentajes mas altos, Perú 5.9%, Chile 2.8%, Colombia 1.8% y México 1.6%), el 9.8% a Asia y el 2.8% a Australia.

Los principales usuarios de estos recursos son en un bajo porcentaje Latinoamericanos, por el contrario Gensler & Associates Architects, Arup Australia, Skidmore, Owings & Merrill, PEI Partnership Architects y centros académicos

como MIT, ETHZ, Yale, Princeton, Delft o Czech Technical University son quienes mas acceden. Así mismo se hizo un recuento de usuarios de RhinoScript que crearon *blogs*. Siendo Europa nuevamente la de más alta concentración con 34 usuarios, EE.UU. con 19 y Latinoamérica con 5.

IV. CONCLUSIONES

Este primer paso de implementación ha dejado muchas sorpresas a aquellos que llevaban siempre la delantera en el uso y exploración de nuevas tecnologías, especialmente en EE.UU. y Europa. Los resultados se basaron en experiencias de estudiantes de primero a quinto año, siendo una experiencia poco usual tener grupos tan heterogéneos. La Universidad de Chile, después de una primera experiencia con estudiantes (2006), hizo la siguiente con asistentes de talleres de diseño (2007). Este año acaban de implementar un laboratorio de fabricación digital, que les permitirá poder cerrar el ciclo del proceso. En el caso de la Universidad Peruana de Ciencias Aplicadas se realizó un solo taller con la experiencia previa de la Universidad de Chile, y en esta vez no solo se incluyeron alumnos, sino profesores de talleres de diseño. La experiencia de un grupo tan heterogéneo, ha permitido que se empiecen a desarrollar proyectos a corto y largo plazo en el que la exploración algorítmica empieza a dar resultados concretos. Finalmente, la compañía McNeel fabricante de Rhino, tomó la iniciativa en Mayo de este año de hacer la traducción a 8 idiomas de la actualización de los recursos que se vienen publicando en el *blog* para llegar a otras regiones donde aun no se exploran los *scripts*.

RECONOCIMIENTOS

A la Facultad de Arquitectura de la Universidad Peruana de Ciencias Aplicadas y la Universidad de Chile quienes han financiado en gran parte esta investigación. A Kenfield Griffith, Daniel Cardoso y John Snavely por su disposición para llevar adelante este proyecto. Y a cada uno de los estudiantes que hacen que este proyecto continúe.

REFERENCES

- [1] P. C. Herrera, "Solución de problemas relacionados al diseño de superficies complejas: Experiencia de programación en la educación del arquitecto.," in *Proc. 11th Iberoamerican Congress of Digital Graphics*, México D.F., 2007, pp. 97–101
- [2] P. Serrano, "Form Follows Software.," in *Proc. 2003 Annual Conference of the Association for Computer Aided Design In Architecture*, Indianapolis., 2003, pp. 185–205.
- [3] M. Silver, "Building Without Drawings: Automason Ver 1.0.," *Architectural Design. Programming Cultures: Art and Architecture in the Age of Software*, vol. 76, no. 4, pp. 46–51, 2006.
- [4] Y. E. Kalay, *Architecture's New Media: Principles, Theories, and Methods of Computer-Aided Design*. Cambridge: MIT Press, 2004.
- [5] M. Silver, "20 Years of Scripted Space," *Architectural Design. Programming Cultures: Art and Architecture in the Age of Software*, vol. 76, no. 4, pp. 46–51, 2006.
- [6] K. Terzidis, *Expressive Form*. UK: Routledge, 2003.
- [7] K. Terzidis, *Algorithmic Architecture*. UK: Architectural Press, 2006.
- [8] Y. Loukissas, "Rulebuilding. Exploring Design Worlds through End-User Programming.," M.Sc. thesis, Dept. Architecture Studies, MIT, Cambridge, USA, 2003.
- [9] T. Schork, "Option Explicit: Scripting as Design Media.," in *Critical Digital Conference Proceedings*, Cambridge., 2008, pp. 41–46.

- [10] N. Senske, "Fear of Code: Approach to integrating computation with architectural design," M. Sc. Thesis, Dept. Architecture Studies, MIT, Cambridge, USA, 2005

Pablo C. Herrera (Lima, 1972). Diseñador Industrial. Profesor a tiempo completo en la Facultad de Arquitectura de la Universidad Peruana de Ciencias Aplicadas y Profesor visitante en la Universidad de Chile. Dirige el proyecto de implementación de tecnologías digitales a través de talleres para facultades de arquitectura en Latinoamérica. <http://www.espaciosdigitales.org/>. El resultado de su trabajo ha sido publicado en México, Santiago, Lima y Bogotá, y recientemente presentado en la exhibición (Im)material Processes: New Digital Techniques for Architecture en la III Beijing Biennial 2008. pablo@espaciosdigitales.org