

O PROJETO EMBRYOLOGICAL HOUSE DE GREG LYNN: UMA INVESTIGAÇÃO METODOLÓGICA

GREG LYNN'S PROJECT EMBRYOLOGICAL HOUSE: AN METHODOLOGICAL INVESTIGATION

Carlos Quedas Campoy¹

Fernando Guillermo Vázquez Ramos ²

Resumo

Em consonância com Bird e LaBelle (2010), observa-se que não apenas a arquitetura construída deve ser preservada, mas também a sua metodologia de concepção. O objetivo deste artigo é apresentar uma pesquisa em andamento que procura desenvolver uma metodologia similar à que Greg Lynn aplicou no *Embryological House* (1998), porém, com ferramentas diferentes. Verifica-se a possibilidade de chegar a resultados próximos, do referido projeto, por meio do *visual scripting* do software Grasshopper e de seus *plug-ins* Galapagos e Biomorpher, com visualização gráfica no Rhinoceros. Promove-se uma leitura da concepção do *Embryological House* sob o ponto de vista da "morfogênese digital" (KOLAREVIC, 2003). Entende-se que isso significa posicionar o computador como uma entidade generativa para a busca de soluções e "invenções de formas" (VÁZQUEZ RAMOS, 2009) como um "sócio" (DOLLENS, 2002; ZELLNER, 1999) do arquiteto. Os resultados apontam para a estruturação de uma metodologia de projeto paramétrico e algorítmico generativo, que permite a interação com o arquiteto, aplicável a diferentes contextos.

Palavras-chave: greg lynn; embryological house; visual scripting; algoritmo genético.

Abstract

In line with Bird and LaBelle (2010), it's observed that not only the built architecture should be preserved, but also its design methodology. The objective of this article is to present an ongoing research that seeks to develop a methodology similar to the one that Greg Lynn applied at Embryological House (1998), however, with different tools. The possibility of coming up with close results of the referred project is sought through the visual scripting of the Grasshopper software and its Galapagos and Biomorpher plug-ins, with graphical visualization in Rhinoceros. A reading of the Embryological House conception is promoted from the point of view of "digital morphogenesis" (KOLAREVIC, 2003). It's understood that this means positioning the computer as a generative entity for the search for solutions and "inventions of forms" (VÁZQUEZ RAMOS, 2009) as a "partner" (DOLLENS, 2002; ZELLNER, 1999) of the architect. The results point to the structuring of a parametric and generative algorithmic design methodology, which allows interaction with the architect, applicable to different contexts.

Keywords: Greg Lynn; embryological house; visual scripting; genetic algorithm.

¹ Doutorando em Arquitetura e Urbanismo, Universidade São Judas Tadeu – USJT – São Paulo, SP, Brasil. Professor Mestre na mesma instituição e departamento. prof.carloscampoy@usjt.br; ORCID: 0000-0002-2692-6237.

² Professor Doutor, Universidade São Judas Tadeu – USJT – Departamento de Arquitetura e Urbanismo, São Paulo, SP, Brasil. prof.vazquez@usjt.br; ORCID: 0000-0003-3472-5598.

1. Introdução

Em consonância com Lawrence Bird e Guillaume LaBelle (2010), encara-se o trabalho da arquitetura como um objeto integral, a ser preservado ao longo do tempo. Nesse sentido, intenta-se observar não apenas o resultado, mas especialmente a metodologia e os métodos de concepção. A investigação de um processo de projeto digital existente, como uma engenharia reversa, mostra-se como algo extremamente complexo de ser elaborado e que não configura um percurso único. Parte-se de uma especulação heurística para se tentar uma sistematização possível, ainda que não única.

Este artigo atenta para a concepção do projeto *Embryological House*, de 1998, do arquiteto estadunidense Greg Lynn, sob o ponto de vista da “morfogênese digital”, que pode ser entendida por processos de “derivação da forma e as suas transformações” (KOLAREVIC, 2003, p. 17) auxiliadas pelo computador. Entende-se que esse termo significa posicionar o computador como uma entidade generativa para a busca de soluções, não apenas como uma ferramenta para a representação. Assim, há a possibilidade de potencializar os “desejos do arquiteto permitindo-lhe delegar a ‘invenção’ de formas a uma máquina” (VÁZQUEZ RAMOS, 2009, p. 253). O computador pode colaborar e revolucionar a “estrutura do pensamento” do arquiteto como um “sócio” (DOLLENS, 2002; ZELLNER, 1999), potencialmente ativo e presente.

Dennis Dollens (2002) aponta que Lynn se aprofundou no espaço digital, nem tanto pela busca de um novo conjunto de formas, mas para o encontro de inspiração e de ideias produtivas ao utilizar o computador como colaborador ou sócio dos seus processos de projeto. O que, no final do século passado, era uma atitude ainda carregada de uma condição inovativa e experimental.³ A sua obra desenhada, construída e escrita é considerada de extrema importância para a evolução da arquitetura, desde a década de 1990 (DOLLENS, 2002; KOLAREVIC, 2003; ZELLNER, 1999). Branko Kolarevic (2003) pondera que, pela sua formação também em Filosofia, as reflexões de Lynn são fundamentais para a construção do pensamento sobre a função da tecnologia e os seus respectivos impactos na arquitetura. Lynn revolucionou conceitos, como movimento e permanência, tornando-se uma “contribuição chave para o discurso arquitetônico contemporâneo” (ZELLNER, 1999, p. 138) nas esferas “civil, política e intelectual” (DOLLENS, 2002, p. 96).

Bird e LaBelle (2010) ressaltam que um dos primeiros projetos que partiram de pesquisas sobre o uso exclusivo das tecnologias digitais, desenvolvidos por Lynn, foi *Embryological House*. Tal projeto foi escolhido como um estudo de caso na preservação da arte digital pela Daniel Langlois Foundation, uma aliança internacional entre o Documentation and Conservation of the Media Arts Heritage (DOCAM) e o Canadian Centre for Architecture (CCA). Os autores posicionam *Embryological House* como um projeto seminal naquele campo nascente da arquitetura digital, da década de 1990, que extrapolou o uso do computador, deixando de entendê-lo como uma mera ferramenta – assistente – para desenhar em um ambiente digital, para assumi-lo como uma fundamental peça ativa na concepção de obras – virtuais, inicialmente, mas prontamente levadas à construção graças aos comandos de controle numérico computadorizado, ou ainda as impressoras 3D.

A pesquisa de Bird e LaBelle (2010), iniciada em 2005, documentou o processo de projeto por meio de capturas de telas de algumas fases específicas e fotos dos modelos físicos

³ Apesar dos trabalhos pioneiros dos anos 1970, como os de Christopher Alexander (*A Pattern Language: Towns, Buildings, Construction*, 1977), ou os desenvolvimentos do final dos anos 1980, como os de Marcos Novak (*Cyberspace: First Steps*, 1991), o trabalho de Lynn, no final dos anos 1990, é o mais enfático dos desenvolvidos na defesa de uma associação colaborativa, concepcional, com o computador.

de estudo do arquiteto. O artigo contém um diagrama, que se considera extremamente elucidativo, no qual propõem um fluxo de trabalho que engloba as intercambialidades entre arquivos de softwares diferentes até a prototipagem rápida de algumas soluções eleitas. Contudo, não evidenciam um passo-a-passo, ainda que especulativo, que percorra desde os primeiros esboços, perpassa pelas primeiras modelagens e as suas transformações, e chegue até as renderizações. Esse último percurso é o recorte que se propõe aqui.

O objetivo geral deste artigo é apresentar uma pesquisa em andamento que procura desenvolver uma metodologia para a concepção de um projeto digital similar ao que Lynn aplicou no *Embryological House*, porém, com ferramentas diferentes da abordagem original. O objetivo específico desta pesquisa é verificar se é possível chegar a resultados próximos, do referido projeto, por meio do *visual scripting* do software Grasshopper e seus *plug-ins* Galapagos e Biomorpher, com visualização gráfica no Rhinoceros.

A abordagem conceitual da pesquisa se realiza por meio de uma revisão bibliográfica – para as questões teóricas do projeto arquitetônico digital, para se investigar a obra de Lynn e para se compreender operacionalmente os softwares –, enquanto que a aproximação empírica no desenvolvimento dos artefatos – ainda que neste caso seja o projeto mesmo –, para se chegar às soluções apresentadas, realiza-se pelo *Design Science Research* (DSR). Seguindo Aline Dresch (2015), não se procura chegar a definições, como um encerramento das questões norteadoras, mas a compreender certos aspectos da prática profissional arquitetônica e propor algumas soluções apropriadas condizentes com a tecnologia atual.

Pela natureza proposta – empírica, ainda que sustentada sobre conceitos precisos – neste projeto de pesquisa, elege-se a categoria dos “Métodos”, também seguindo Dresch (2015), pois apresentam um conjunto de passos necessários para desempenhar a determinada tarefa. Podem ser apresentados graficamente ou encapsulados em heurísticas e algoritmos específicos. Há, portanto, um procedimento e um protocolo que têm que ser atendidos e, sobretudo, enunciado – muitas vezes cumprido, acabado, e outras não necessariamente. Podem estar ligados aos modelos, e as suas respectivas etapas podem utilizar partes do modelo como uma entrada que o compõe. Os “Métodos” favorecem, tanto a construção, quanto a representação das necessidades de melhoria de um determinado sistema.

Os resultados apontam para a estruturação de uma linha de raciocínio que utiliza *visual scripting* orientado para um processo de projeto paramétrico e algorítmico generativo, que não prevê apenas uma solução ideal e que não se aplica exclusivamente ao escopo da *Embryological House*. O arquiteto pode interagir com o sistema, em qualquer momento, e estudar as diversas geometrias geradas. Também, é uma tentativa de promover uma aproximação profissional – dos arquitetos – a essa tecnologia de programação, que dispensa a necessidade de aprender códigos.

2. Desenvolvimento

Esta seção procura delinear a estrutura de pensamento de Lynn ao utilizar o computador de uma maneira generativa para o projeto *Embryological House*. Como operacionalizou as premissas dentro dos softwares e quais foram as suas inspirações e os resultados obtidos.

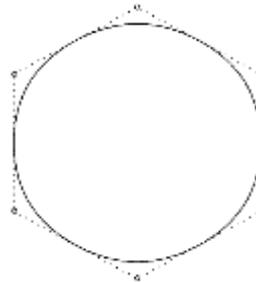
2.1. O processo de projeto do *Embryological House*

Lynn (2007) explica que o projeto teve o seu início marcado por uma investigação sobre a possibilidade de se trabalhar com diferentes inflexões de curvas que mantivessem

dependências operativas entre si. Depois de organizadas espacialmente, essas curvas poderiam formar um objeto tridimensional. Logo o arquiteto percebeu que havia uma infinidade de configurações possíveis para tal geometria. Afirma que foi a primeira vez que trabalhou de maneira paramétrica.

A curva originária, um círculo com 6 vértices de controle, foi desenhada no software Microstation com a ferramenta do tipo *spline*, que gera um “vetor definido com direção (...) que pode acomodar pesos e gravidades direcionados no espaço livre” (LYNN, 1999, p. 20). De maneira diferente de “uma linha ou um raio”, cujas “formas podem ser reduzidas à pontos com coordenadas exatas”, as curvas *spline* são definidas como “fluxos”. Tal objeto passa por entre uma “constelação de vértices de controle”, cada um com um “peso” específico e com “diversas posições” e “graus” de inflexão (LYNN, 1999, p. 22). Aqui, essa curva *spline* originária foi reproduzida no programa Rhinoceros (Figura 1).

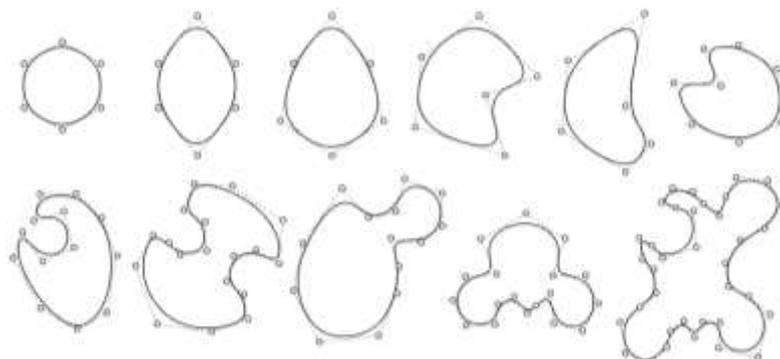
Figura 1: Curva *spline* originária com seis vértices de controle.



Fonte: Elaborada pelos autores. Adaptado de Lynn (2007).

Quaisquer mudanças nos posicionamentos dos vértices de controle tendem a “distribuir inflexões através de regiões dessa entidade” (LYNN, 1999, p. 23). Portanto, observa-se que, para a figura primordial apresentada acima, pela sua simetria radial, há um equilíbrio igualitário entre as forças e pesos dos vértices. Lynn começa a inserir informações que desequilibram essas distribuições de forças e pesos, estabelecendo diferentes tensões ao longo da curva, resultando em algumas variações. Nota-se que, para obter uma quantidade maior de inflexões, o arquiteto inseriu gradualmente mais vértices de controle. A Figura 02 demonstra alguns resultados desse estudo.

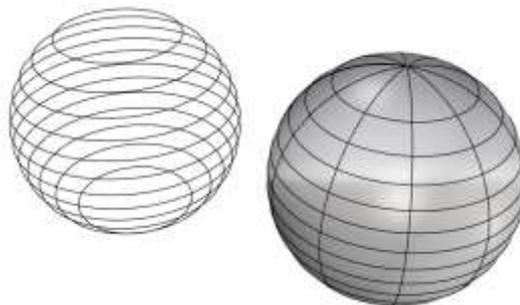
Figura 2: Estudos de diferentes inflexões de curvas *spline*.



Fonte: Elaborada pelos autores. Adaptado de Lynn (2007).

Posteriormente, Lynn (2007) comenta que, a partir da curva originária do círculo, transforma-a em um objeto tridimensional formado por uma superfície associativa. O arquiteto não explicita esse processo. Porém, pode-se assumir que, pelo raciocínio anterior das curvas, ele talvez tenha posicionado algumas delas de maneira estratégica para depois uni-las. Lynn fala sobre o comando *Loft*, que é comum a diversos softwares de modelagem tridimensional. Fato que reforça o processo que se assumiu anteriormente, pois os objetos formados por esse comando dependem de um agrupamento de linhas ou curvas. Como a superfície deve ser responsiva às diferentes aplicações de pesos e forças nos seus vértices de controle, praticamente confirma que esse método presumido é muito próximo ao empregado pelo arquiteto. Apresenta-se, por meio da Figura 3, a evolução aqui descrita.

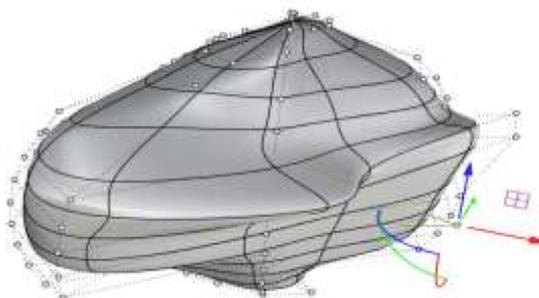
Figura 3: Transformação de curvas em uma superfície por meio do comando Loft do Rhinoceros.



Fonte: Elaborada pelos autores.

O resultado formal se assemelha a uma esfera. Contudo, Lynn adverte que, na realidade, “não se trata de uma simples esfera, mas de um objeto topológico” (2007). Como “entidades topológicas são baseadas em vetores”, elas podem “incorporar forças” que se apresentam sob a forma de “inflexões” ou “curvatura contínua” (1999, p. 23). Na Figura 4 aplicaram-se algumas diferenças de pesos ou forças, por enquanto aleatórias, para se testar a plasticidade da esfera topológica.

Figura 4: Inflexões da esfera topológica.



Fonte: Elaborada pelos autores.

Segundo Peter Zellner (1999), a topologia envolve o estudo de superfícies, resultantes de regras matemáticas específicas, que podem ser transformadas sem se colapsarem ou quebrarem, devido à sua estrutura flexível. O autor correlaciona o conceito da topologia à

definição da anatomia de uma área particular do corpo ou à forma e às qualidades de um organismo. Como D'Arcy Thompson (1860-1948), naturalista, zoologista e biomatemático demonstrou, as funções matemáticas podem ser aplicadas às formações de um determinado organismo para que o transforme em outro. Observa-se que esse é o ponto crucial para se compreender a proposta do projeto *Embryological House*, e de muitos dos processos posteriormente usados por Lynn e outros artistas que trabalharam com este tipo de controle, como, por exemplo, Hani Racshid.

Dollens (2002) e Zellner (1999) reconhecem que parte relevante do rigor da concepção das geometrias, avançadas por Lynn, são desenvolvidas em experimentações inspiradas nos estudos sobre o crescimento e as mutações genéticas de plantas e animais, em contato com os influxos do ambiente, elaboradas primordialmente por D'Arcy Thompson. “Sem a mera intenção de reanimá-las”, Lynn busca fundir esses estudos com as “técnicas adequadas para a produção digital do século XXI” (DOLLENS, 2002, p. 97). O próprio Dollens desenvolveu trabalhos similares inspirados nas pesquisas de Lynn.

Lynn (1999, p. 26) aponta que Thompson “analisou variações na morfologia de animais usando *grids* deformáveis”, o que acabou “produzindo curvas devido às mudanças na forma”. O zoologista “comparou as curvaturas provenientes das deformações com as curvaturas de dados estatísticos, como velocidade, temperatura e peso”. Com isso, “Thompson foi um dos primeiros cientistas a notar a ação gradiente de forças através de deformação, inflexão e curvatura”, conclui o arquiteto.

Portanto, tem-se que a topologia e os conceitos morfológicos desenvolvidos por Thompson são os elementos de lastro para a concepção do projeto *Embryological House*. Assim como outros projetos, Lynn mantém uma estrutura metodológica que é o “resultado de decisões tomadas usando parâmetros” que, por sua vez, simula a “interação gradiente de forças direcionais, gravidades (...) e densidades que afetam os objetos como campos numéricos de força, em detrimento de transformações do objeto” por si só (LYNN, 1999, p. 25).

Lynn (2007) afirma que o seu entendimento das definições dos termos “genérico” e “genético” são provenientes da literatura biológica do final do século XIX. “Esse ‘genérico’ (...) é algo que tem todo o potencial que ainda não foi especificado; ‘genético’ são as especificações”. O arquiteto acrescenta que:

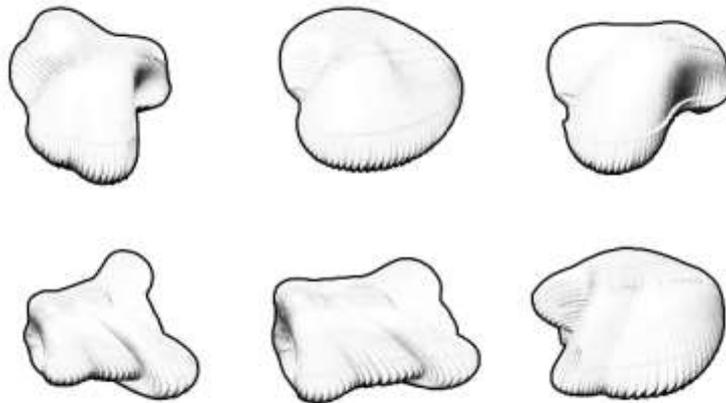
[...] genérico é como um ovo que ainda não se desdobrou, mas está para ser desdobrado. Então, com o projeto *Embryo House*, eu estava muito consciente de começar com uma esfera, como um ovo, uma coisa pura – mesmo sabendo que não é uma esfera, mas um objeto formado por curvas spline. (LYNN, 2007, transcrição e tradução nossa)

Portanto, aquela esfera topológica pode ser entendida como um embrião, que pela sua inerente característica plástica topológica tem o potencial para receber as transformações nos seus vértices de controle. Observa-se a relevância em se esclarecer, neste ponto, o sentido que palavra “transformação” tem para Lynn nesse projeto *Embryological House*.

Em uma aula, Lynn (2009) correlaciona “transformação” com “mutação”. Baseado nos estudos do biólogo inglês William Bateson (1861-1926), ele observou que as mutações ocorrem por inserções de informações externas em um sistema. Toda vez que se perde informação externa em um sistema, esse tende a retornar à simetria. Portanto, conclui que a simetria não era indício de ordem e organização, como geralmente os arquitetos entendem, mas, sob esse ponto de vista da biologia, a simetria era a ausência de informação.

Sendo assim, de maneira semelhante à curva circular primordial, o objeto plano representado na Figura 1, a esfera topológica apresenta simetria, justamente, pela ausência de forças e pesos externos em seus vértices de controle; é o objeto “genérico” ou o “embrião” com seu potencial latente genotípico. A partir do momento em que Lynn começa a inserir informações diferentes – as especificações “genéticas” (LYNN, 2007) fenotípicas –, a superfície começa a se transformar de maneira responsiva. A Figura 5 demonstra algumas instâncias, derivadas da original.

Figura 5: Instâncias derivadas por transformações fenotípicas.



Fonte: Elaborada pelos autores. Adaptada de Zellner (1999, p. 140).

Com a finalidade de estudar as mutações provenientes das interações entre essas entidades e as informações inseridas, Lynn lança uso do software de animação Maya. Segundo o arquiteto, o projeto *Embryological House* foi o primeiro em que empregou a técnica da animação “sem um perfil pseudocientífico” (LYNN, 2007). Assim, a animação foi usada de maneira “generativa para produzir uma coisa genérica para depois inserir mais informações para fazê-la mutar” (LYNN, 2007). Posteriormente, prossegue o arquiteto, esse método continuou a ser aplicado em outros projetos, mesmo com certas adaptações.

Dentro do software Maya, Lynn consegue mudar a forma de cada um dos objetos de uma maneira gradual, que começa com a esfera topológica e vai até o máximo de deformação. Depois, consegue combinar os objetos em um só, em qualquer ponto das transformações individuais, portanto, promove uma interpolação volumétrica entre algumas soluções eleitas. O resultado é um “número infinito de combinações” (LYNN, 2007).

Voltando-se o olhar para o desenvolvimento do método aqui proposto, recorre-se, primeiramente, à explicação de Kolarevic (2003) sobre o conceito da “morfogênese digital”. Segundo o autor, revela-se por um procedimento sistemático distinto, corroborado pela agilidade de cálculos e a demonstração dos resultados em uma interface gráfica, facilitados pelo computador. Para lidar com a complexidade e o processamento da informação, segundo Bueno *et al.* (2019), o seu respectivo desdobramento operativo pode se dar por meio da metodologia dos Sistemas Generativos (SG).

Bueno observa que os SG tendem a proporcionar projetos responsivos ao ambiente, no qual serão inseridos, em uma plataforma que oferece uma parceria entre o arquiteto e o computador, que se entende similar à desenvolvida por Lynn. Prosseguindo, Bueno *et al.*

(2019) ressaltam que, diferentemente de metodologias tradicionais de projeto, o uso dos SG implica em uma relação indireta com o produto, portanto, há um nível de autonomia no funcionamento do sistema em relação ao arquiteto. Para o seu desenvolvimento, deve-se estabelecer regras e procedimentos manifestados por algumas técnicas, das quais se destaca aqui a dos Algoritmos Genéticos (AG), pois não pressupõem formas estáveis e nem uma única solução, além da possibilidade de inserir informações genéticas que podem provocar mutações na geometria genérica, portanto, operações que têm aderência à metodologia empregada por Lynn no *Embryological House*.

Os autores explicam que os desafios estão na dificuldade em se traduzir os problemas relativos às premissas genotípicas e fenotípicas dentro do algoritmo. Nesse ponto, tem-se que a subjetividade do arquiteto é fundamental para a leitura e a interpretação das premissas e as suas respectivas inserções em um processo que, geralmente, estão além do pensamento linear e/ou analógico. A técnica dos AG resulta operativamente diversas alternativas, que conotam certa autonomia funcional da máquina, e não uma autonomia total, pois é o arquiteto quem decide como analisá-las e qual será a geometria ou o conjunto de geometrias estabilizadas.

A partir do entendimento do ponto de vista da morfogênese digital, da metodologia dos Sistemas Generativos e da técnica dos Algoritmos Genéticos, retorna-se à algumas definições de termos, agora, do campo da matemática e da programação, cuja compreensão prévia é central para o processo de projeto aqui proposto e desenvolvido no programa Grasshopper.

Segundo Scott Davidson e Robert McNell (2015, p. 37-38), o Grasshopper (GH) é um software que permite criar algoritmos por meio de programações visuais denominadas por “definições”, que, por sua vez, são elaboradas por “blocos funcionais” interconectados por cabos, que podem resultar em geometrias apresentadas graficamente no Rhinoceros. Existem dois tipos de blocos funcionais, por eles entendidos como: “parâmetros” – que armazenam dados, como números, cores e geometria –, e “componentes” – que realizam ações, como operações matemáticas, por exemplo. Ressalta-se que o uso da palavra “parâmetro”, nesse caso, pode provocar alguma confusão, que se procurará elucidar.

A arquiteta e professora Gabriela Celani (2003, p. 21-22) observa que, na matemática, “parâmetros são valores que podem ser atribuídos a uma determinada variável, permitindo o cálculo de diferentes soluções para um problema”. A autora utiliza, como exemplo, a equação reduzida da reta ($y=mx+b$), onde “m e b são parâmetros constantes (...), x e y são as variáveis que representam as coordenadas de cada ponto da reta”. Portanto, a partir dessa definição operativa matemática, no GH, segundo Davidson e McNell (2015), verifica-se que os parâmetros englobam mais do que apenas valores.

Celani (2001) aponta que, em uma época anterior à invasão dos computadores nos escritórios de arquitetura, a partir da década de 1990, o arquiteto William Mitchell pesquisava sobre como os SG poderiam promover uma integração entre a arquitetura e a computação não apenas gráfica, oferecendo infinitas soluções. Mitchell (1975) ressaltava que os SG podem ser agrupados em três categorias, não exclusivas ao mundo dos computadores: “análogo” – como as experimentações físicas com cabos e pesos, que se transformavam em catenárias, promovidas por Gaudí, no final século XIX –, “icônico” – desenvolvido por intermédio de plantas, elevações e modelos em escala –, e “simbólico” – elaborados por meio de símbolos, como palavras, números e operações matemáticas. O autor conclui que, para promover transformações no estado de um SG simbólico, as operações devem ser lógicas e aritméticas.

Observa-se que a operacionalização do Grasshopper, e a sua interface com o usuário,

pode ser aproximada à categoria simbólica definida por Mitchell. Nesse sentido, acredita-se na importância em se compreender como o autor identifica cada elemento constituinte do SG. Para Mitchell (1975), “variável de projeto” é pensada como uma “conceitual ‘caixa vazia’” nomeada, na qual diferentes valores podem ser atribuídos. É possível modelar um sistema simbolicamente ao permitir que “variáveis de projeto” específicas representem “propriedades específicas do sistema”. O autor exemplifica isso ao imaginar a modelagem de uma sala retangular, que possui um comprimento (C) e uma largura (L). Nesse caso, C e L são as variáveis que, por sua vez, possuem valores numéricos atrelados. Pode-se criar uma “coleção de variáveis” associadas entre si e referenciadas como uma “estrutura de dados”.

Portanto, seguindo Celani (2003) e Mitchell (1975), entende-se que, do ponto de vista da matemática e da programação computadorizada, parâmetros e valores são o mesmo elemento, de um SG, que atribui uma dimensão quantitativa à uma variável, seja para se desenhar uma reta, um ambiente quadrangular ou até uma geometria de característica topológica. Dentro da “definição” do Grasshopper, o “bloco funcional” denominado por “parâmetro”, empregado por Davidson e McNell (2015), engloba parâmetro e variável ao mesmo tempo, contudo, ficam dispostos em painéis separados. Esses, por sua vez, são localizados em abas que agrupam características específicas, como entradas de dados, funções e operações matemáticas, conjuntos e sequências de listas, vetores, curvas, superfícies, objetos do tipo *mesh*, intersecções, transformações e apresentações gráficas. O usuário/arquiteto não precisa ser um programador, mas deve compreender muito bem as disciplinas da matemática e da geometria, além de atentar para possíveis erros de sintaxe que inviabilizam o funcionamento do algoritmo.

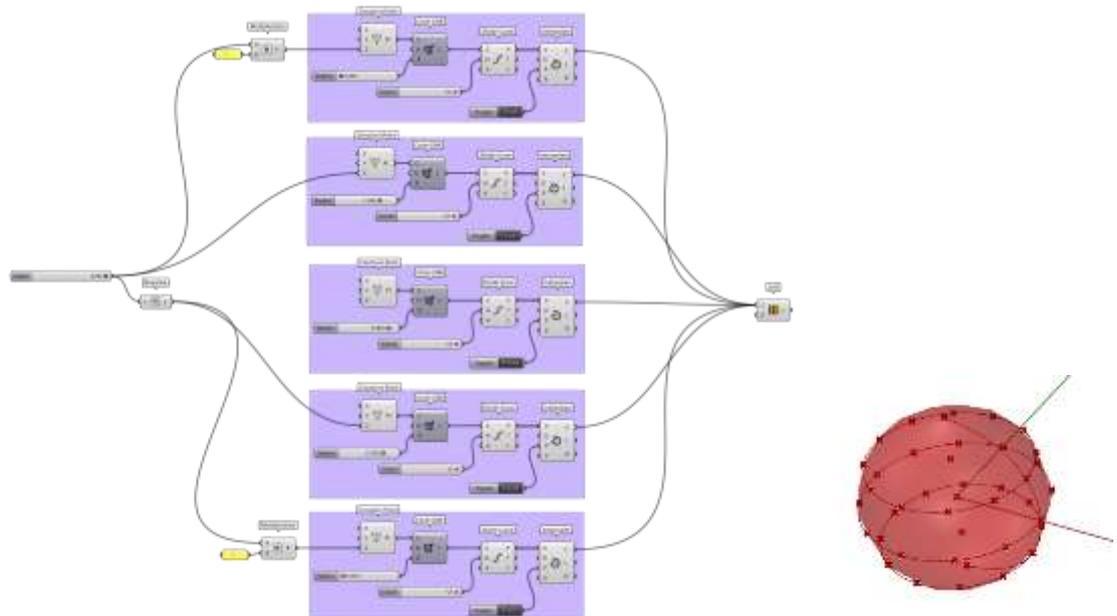
É comum, atualmente, ouvir o termo “parametrização” empregado erroneamente para se referir a softwares de modelagem tridimensional, dirigidos à arquitetura, que não o Sketchup. Kolarevic (2003, p. 25) explica que o “projeto paramétrico” é estruturado para “satisfazer certas condições”. Assim, são declaradas as relações ou as “dependências operativas” entre as variáveis; não a forma gerada no final do processo. Tampouco são cruciais as diferenças entre valores atribuídos às variáveis – salvo algumas programações que não permitem a inserção do numeral zero. De um modo geral e resumido, a parametrização de um projeto está nas regras. Então, a partir das alterações dos valores, ou dos parâmetros, ocorrem transformações nos “comportamentos dos objetos”, que resultam em uma, ou mais, “geometria associativa”, a princípio, não estável devido à grande quantidade de soluções – que podem ser até infinitas. No GH, é possível estabelecer a parametrização do projeto ao se inserir os “componentes”, que realizam ações parametrizadas, em conjunto com os “parâmetros” (DAVIDSON; MCNEEL, 2015), desde que estejam dentro de um mesmo algoritmo.

Após a compreensão dessas nomenclaturas e das suas respectivas funções no universo da matemática computacional e, especificamente, como são empregadas no software GH, torna-se o olhar para a programação alvo deste estudo. Na tentativa de uma construção algorítmica digital, que possa resultar em um estudo similar ao *Embryological House*, procurou-se acompanhar em todo o processo – não somente nos primeiros passos –, a mesma cadência de raciocínio de Lynn (2007), porém, algumas adaptações e estudos paralelos tiveram que ser realizados.

3. Implementação do Método Proposto

Inicia-se aqui com a “definição” (DAVIDSON; MCNEEL, 2015) da esfera topológica primitiva. Lynn (2007) começa a explicação, da fase que utiliza o Maya para a modelagem do projeto, referindo-se às curvas *spline* elaboradas por meio dos seus respectivos vértices de controle. Entendeu-se que o arquiteto utiliza o comando *Loft* para unir todas as *splines* e, assim, criar o objeto tridimensional. Passando-se para a implementação da programação aqui proposta, a Figura 6 é o conjunto formado pela captura de tela da programação realizada no Grasshopper (GH), no lado esquerdo, e o resultado gráfico – a esfera –, no lado direito.

Figura 6: programação no GH e a geometria associativa gerada.

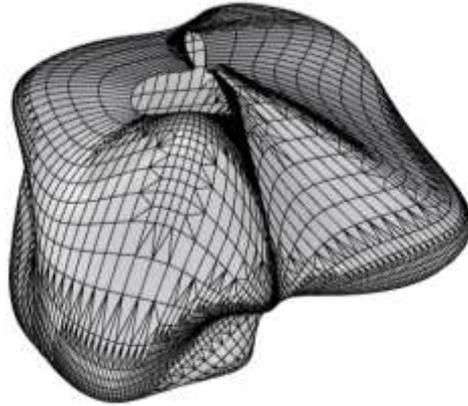


Fonte: Elaborada pelos autores.

A lógica da programação acima contou com o posicionamento de quatro pontos, distribuídos igualmente ao longo de um diâmetro de 18m, no eixo Z. Círculos foram criados com seus respectivos centros referenciados nesses pontos anteriores. Promoveu-se a divisão dos círculos em doze partes iguais, o que resultou nos pontos de controle. Os componentes de interpolação criaram as *splines*. O *Loft* construiu a geometria por meio dessas curvas *spline*.

Contudo, ao se promover alguns testes de movimentações desses pontos – vértices de controle – das tais curvas, identificou-se que a geometria apresentava algumas nervuras que não estavam de acordo com as demonstrações de Lynn (2007). Além do fato de que a parte superior e inferior da forma tendia a ficar aberta, dependendo da amplitude das movimentações (Figura 7).

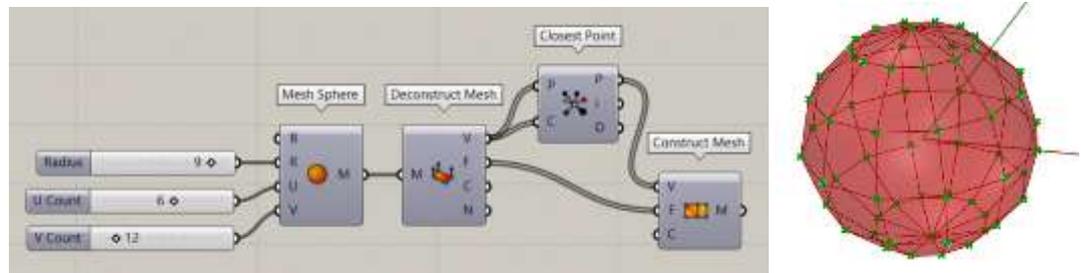
Figura 7: resultado ao se aplicar transformações na geometria formada pelo comando *Loft*.



Fonte: Elaborada pelos autores.

Partiu-se, portanto, para a pesquisa de outros blocos funcionais que pudessem resolver esses problemas e proporcionar um comportamento transformativo similar ao *Embryological House*. Encontrou-se uma boa solução ao se trabalhar com objetos do tipo *mesh* – uma concepção diferente da anterior, que foi formulada pelo tipo superfície. A Figura 8 mostra essa nova programação da esfera topológica e o seu resultado gráfico.

Figura 8: programação no GH para a esfera *mesh* e o seu resultado no Rhinoceros.

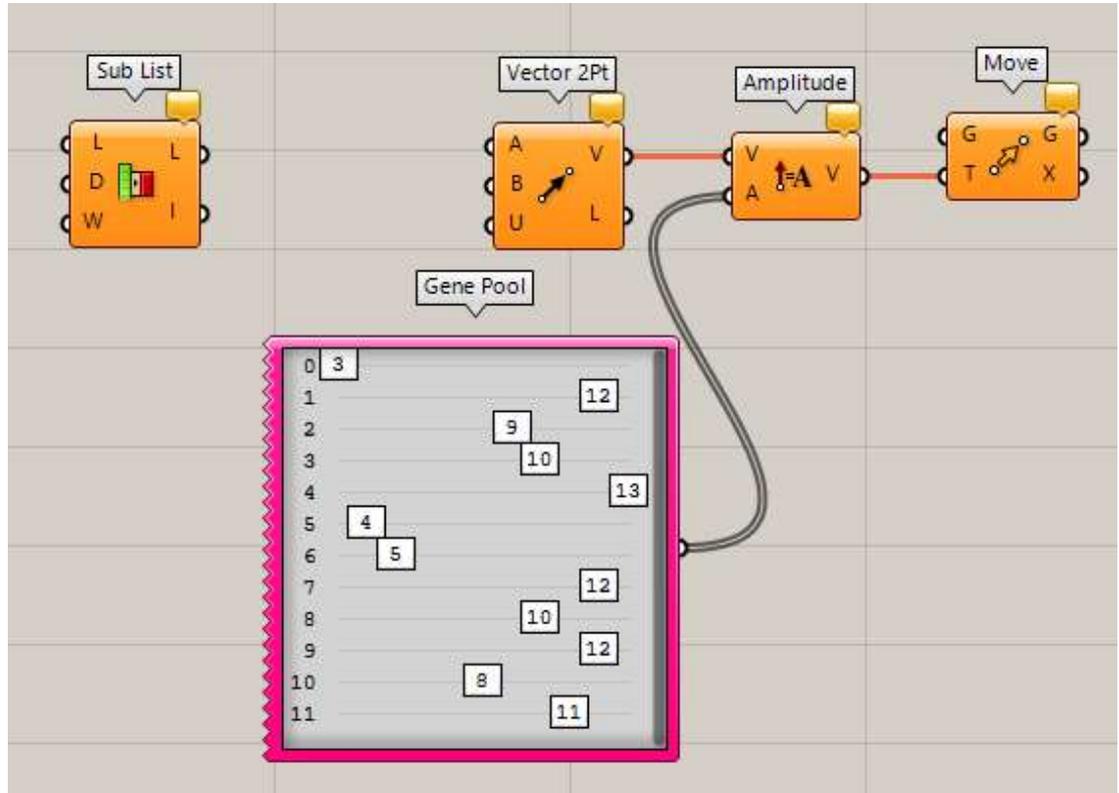


Fonte: Elaborada pelos autores.

Essa programação, até mais enxuta que a anterior, contou com a criação de uma esfera *mesh* com o mesmo raio de 9m. Seis divisões no sentido vertical e doze de maneira radial se mostraram como o mínimo para promover a maleabilidade desejada. Tem-se um total de 62 vértices de controle, identificados pelo componente *Deconstruct Mesh*. O *Closest Point* adquire de volta esses pontos e o *Construct Mesh* recria a geometria. Esses três últimos componentes são necessários para que, posteriormente, seja possível alterar os posicionamentos de cada vértice e, assim, provocar transformações no objeto. Ressalta-se que o aspecto facetado da geometria não é um problema, pois existem componentes que suavizam essas faces, de tal nível, que uma esfera é formulada.

Na sequência, tem-se a Figura 9 para apresentar os componentes: *Sub List* – que promove o agrupamento dos vértices de controle conforme as suas determinadas alturas no eixo Z –, os que definem os vetores radiais de movimentação dos pontos de controle ao longo do plano XY –, e o parâmetro *Gene Pool* (reservatório genético) – que restringe a amplitude de cada movimentação.

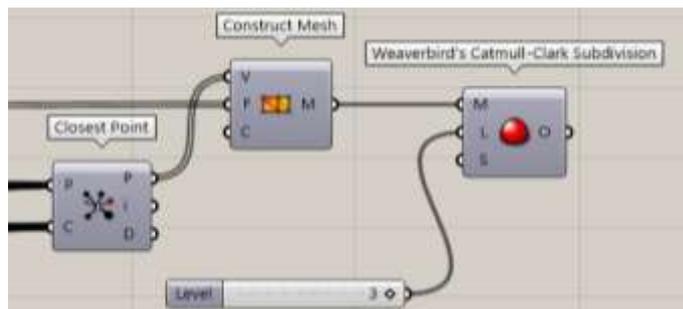
Figura 9: programação no Grasshopper para captura dos vértices de controle.



Fonte: Elaborada pelos autores.

A última parte da programação (Figura10) conta com a construção da esfera topológica por intermédio dos componentes *Closest Point* e *Construct Mesh* – conforme demonstrado na programação inicial na Figura 8. Para suavizar a forma, convertendo-a em uma esfera, utiliza-se o *Weaverbird's Catmull-Clark Subdivision*, com um terceiro nível de força ou de atuação, que não faz parte da instalação original do Grasshopper. Trata-se de um *plug-in*, desenvolvido por Giulio Piacentino, que contém vários operadores para a modelagem topológica. Segundo o autor, a ferramenta *Catmull-Clark* calcula o tipo de subdivisão recursiva descrita pelos cientistas da computação Edwin Catmull e Jim Clark, em 1978. O resultado está localizado no lado direito da Figura 10.

Figura 10: programação no GH para conclusão da geometria e o seu respectivo resultado no Rhinoceros.

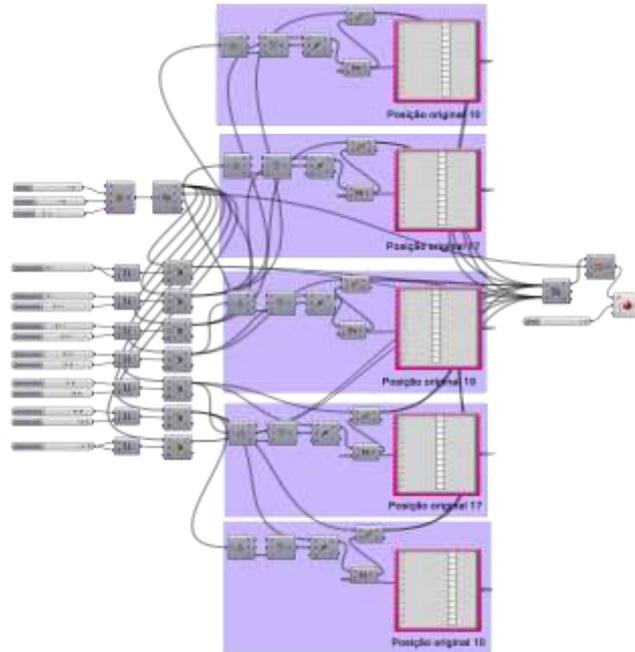


Fonte: Elaborada pelos autores.



A Figura 11 é uma captura de tela da programação completa.

Figura 11: programação no GH para conclusão da geometria - embrião.

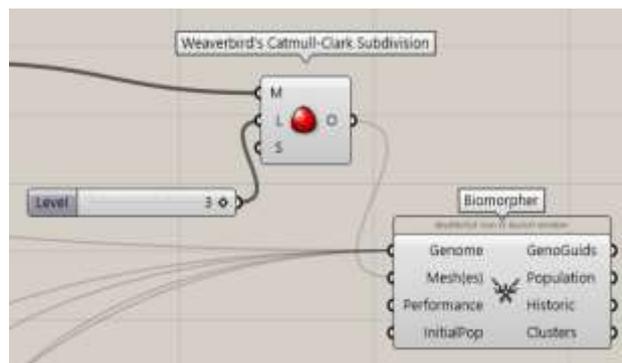


Fonte: Elaborada pelos autores.

Para o estudo das diversas soluções provenientes das combinações possíveis entre as diferentes formas – geradas pelo parâmetro *Gene Pool* –, elegeram-se os *plug-ins* Biomorpher e Galapagos. Tem-se, na sequência, uma breve explicação sobre a função, a operacionalização, e os resultados, para cada um deles.

O Biomorpher (Figura 12), desenvolvido por John Harding, trabalha com princípios evolucionários aplicados a soluções de problemas. É conectado com a esfera topológica do tipo *mesh* – o que reforça a substituição da definição que criava esse objeto por intermédio do componente *Loft*, pelo atual *Construct Mesh* –, e com os parâmetros disponíveis no *Gene Pool*. Esse último define, geneticamente, o máximo e o mínimo de alterações em cada vértice de controle do embrião, ainda simétrico.

Figura 12: o componente Biomorpher e as suas conexões no Grasshopper.



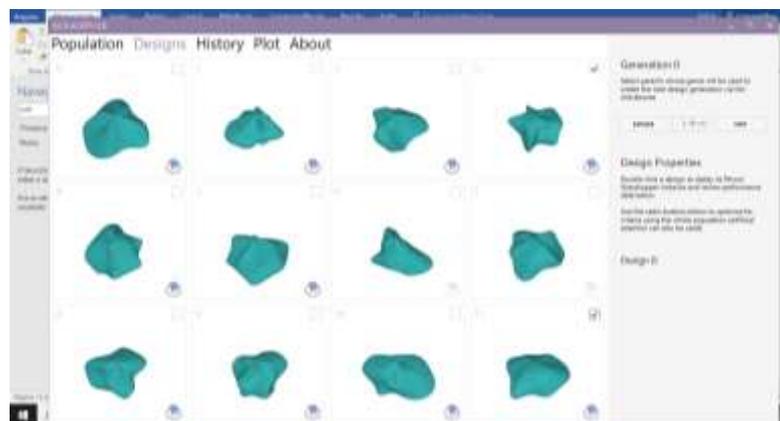
Fonte: Elaborada pelos autores.

Há a possibilidade de se definir uma performance específica como, por exemplo, atingir um volume máximo e, ainda, pode-se ajustar uma população inicial para as combinações entre os indivíduos gerados. Como a intenção desse estudo é analisar as diferentes saídas de geometrias responsivas aos limites genéticos, e não a adaptação dessas a um ambiente específico, escolhe-se não trabalhar com essas entradas do componente.

Ao se iniciar o Biomorpher, uma tela com configurações iniciais é apresentada. Há controles sobre a população inicial, a taxa de cruzamento entre as entidades e intensidade das mutações entre gerações. Opta-se pela operação aleatória, disponibilizada em *Go Random*, pois entende-se que, dessa maneira, o arquiteto pode delegar ao computador uma liberdade maior para a busca e a apresentação de soluções. Sugere-se que a interferência do arquiteto deva ocorrer em etapa posterior.

Na sequência, após esses primeiros cálculos serem concluídos, o programa retorna outra tela com algumas soluções para a geometria – a geração 0 –, onde é possível escolher algumas variações para serem cruzadas e evoluídas para uma próxima geração. A Figura 13 demonstra duas opções que foram selecionadas, a de número 3 e a 11, para se cruzarem e evoluírem para uma próxima geração 1. Esse procedimento pode ser repetido indefinidamente. Dependendo das escolhas do arquiteto, um certo tipo de solução, cada vez mais pontiaguda ou suave, por exemplo, pode ser condicionada.

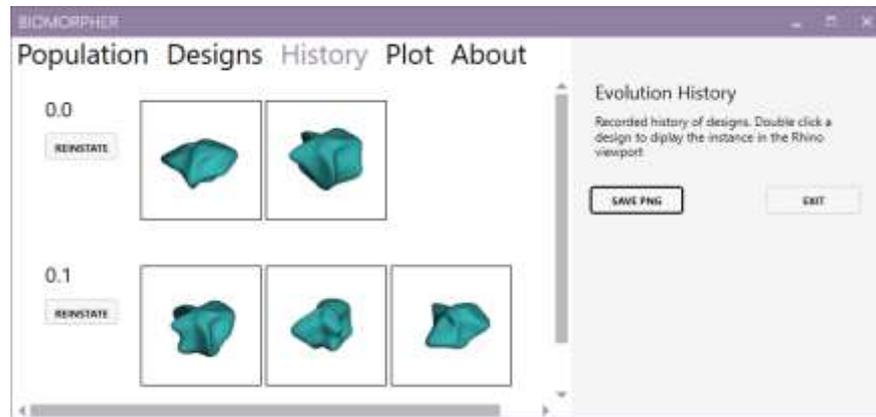
Figura 13: primeiras geometrias criadas pelo Biomorpher.



Fonte: Elaborada pelos autores.

A Figura 14 é o resultado de duas gerações, que contém cinco soluções diferentes.

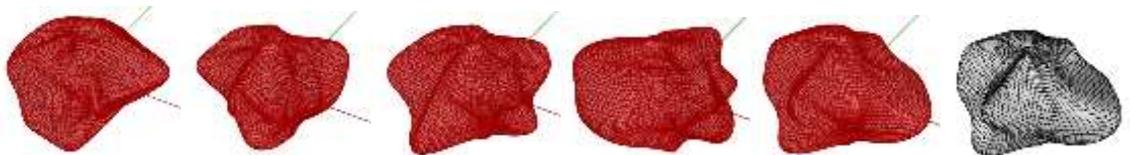
Figura 14: soluções geométricas no Biomorpher.



Fonte: Elaborada pelos autores.

A Figura 15 é um compilado dessas cinco geometrias, agora, no ambiente do Grasshopper e mais uma, à direita, construída do Rhinoceros. Ressalta-se que, a partir do momento em que o objeto é criado no Rhinoceros, torna-se possível intercambiá-lo com outros softwares como, por exemplo, o Revit.

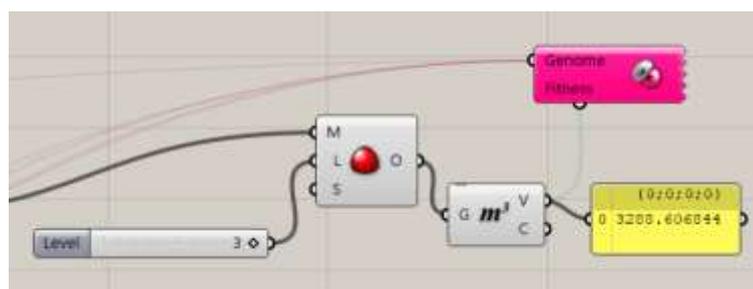
Figura 15: geometrias no Grasshopper e no Rhinoceros provenientes do Biomorpher.



Fonte: Elaborada pelos autores.

Percorre-se, a seguir, um método similar ao demonstrado no Biomorpher, por meio do *plug-in* Galapagos. Os dois possuem as mesmas bases conceituais, porém são operados com uma diferença crucial relativa às restrições do ambiente – fenotípicas. O Galapagos, desenvolvido por David Rutten, tem a sua respectiva entrada *Genome* conectada com as definições do parâmetro *Gene Pool* – nesse ponto, igual ao efetuado com o Biomorpher. Contudo, para que o componente seja ativado, é necessário ligar a entrada *Fitness* (aptidão) com algum condicionante de adaptabilidade proveniente do ambiente. Nesse caso, apresentado na Figura 16, elegeram-se a condição de um volume para o cálculo.

Figura 16: o componente Galapagos e as suas conexões no Grasshopper.



Fonte: Elaborada pelos autores.

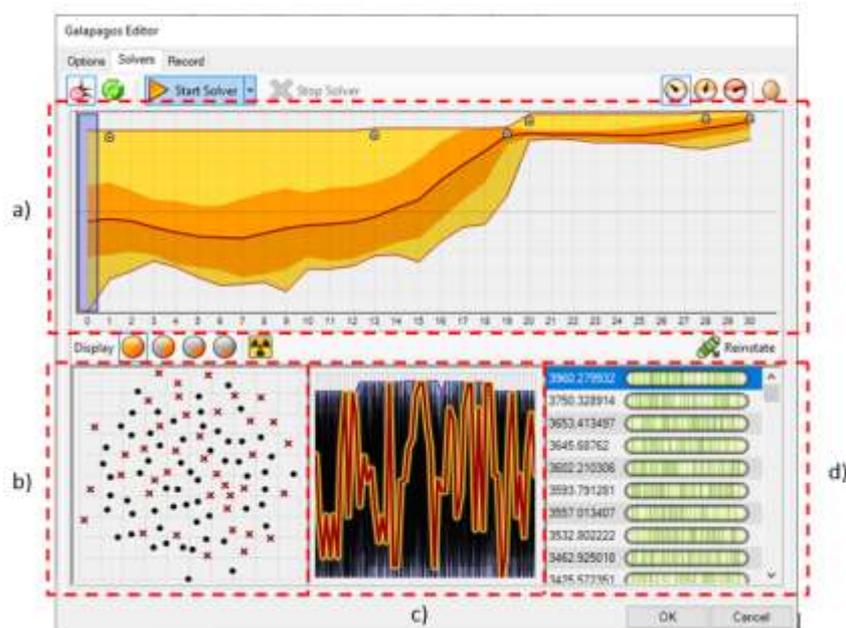
Ao se inicializar o Galapagos, uma primeira tela é visualizada, onde é possível definir se o volume a ser buscado será o máximo ou o mínimo – no campo *Fitness* –, a população máxima inicial, a manutenção de uma porcentagem específica dos melhores indivíduos gerados e algumas características do algoritmo solucionador *Annealing*. Optou-se pela condição de volume máximo, para esse experimento, e a sobrevivência de 5% dos melhores indivíduos.

A próxima aba “Solucionadores” começa a apresentar, em quatro áreas distintas, a evolução do algoritmo (Figura 17). A área a) mostra trinta gerações calculadas e a quantidade de indivíduos em cada uma delas – no gráfico de cores amarela e laranja. Nota-se como a geração 0 possui uma quantidade grande de indivíduos que foram cruzados e selecionados para gerar as próximas, que seguem a tendência de uma diminuição da quantidade dessa população, pois apenas os 5% mais aptos a sobreviver nas condições ambientais são passados para frente.

A área b) apresenta, por meio de pontos, os indivíduos que foram selecionados, e por meio de um “x” os que foram eliminados pelo algoritmo. Rutten comenta que a escolha dos indivíduos que vão se acasalar segue uma lógica que analisa uma distância média entre seus respectivos posicionamentos nesse plano cartesiano. O Galapagos procura uma distância não muito próxima, para não gerar uma relação incestuosa, e nem uma muito distante, para não promover um possível acasalamento entre espécies diferentes geneticamente, o que, em ambos os casos, tenderia a produzir descendentes defeituosos, portanto, com poucas chances de perdurarem no sistema.

A área c) é uma representação da rede neural, cerne do algoritmo. A área d) dispõe, por intermédio das barras horizontais na cor verde, os indivíduos. As suas respectivas partes internas, em verde e branco, representam as combinações genéticas de cada controle do *Gene Pool*. Ainda é possível inserir uma mutação, em qualquer momento da evolução algorítmica, ao se acionar o botão, cujo ícone é o de radiação nuclear.

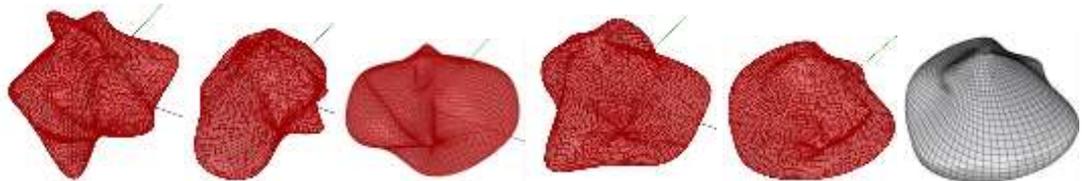
Figura 17: Tela do plug-in Galapagos.



Fonte: Elaborada pelos autores.

Observa-se que a geração 0 é a que mais apresenta variações entre as geometrias criadas. A Figura 18 demonstra algumas dessas opções no ambiente do Grasshopper – as formas na cor vermelha – e a última construída no Rhinoceros, à direita.

Figura 18: geometrias no GH e no Rhinoceros provenientes do Galapagos.



Fonte: Elaborada pelos autores.

Quanto mais se avança para as próximas gerações, mais se obtém objetos com formato semelhante à última opção, no lado direito, da Figura 19 anterior. Portanto, do ponto de vista do estudo da diversidade de indivíduos, a geração 0 é a mais interessante.

Ao se comparar os dois algoritmos generativos, o Biomorpher e o Galapagos, visualiza-se que o Biomorpher proporciona maior interação com o arquiteto, pois é ele quem define quais são os indivíduos que serão combinados para se transformarem em uma geração seguinte, e assim por diante. O Galapagos se mostra mais independente dessa interação com o arquiteto, pois leva em consideração apenas os parâmetros genotípicos e os fenotípicos definidos no momento pré-cálculo. De qualquer modo, as geometrias criadas pelos dois *plugins* podem ser diferentes quanto às suas formas, porém, possuem a mesma característica topológica e a mesma fundamentação genética evolucionária. A Figura 19 é uma renderização de uma solução elaborada pelo Biomorpher, no lado esquerdo, e uma pelo Galapagos, no lado direito.

Figura 19: renderizações dos estudos finais elaborados no Biomorpher e no Galapagos.



Fonte: Elaborada pelos autores.

4. Considerações Finais

Acredita-se que as duas ferramentas atendem ao objetivo aqui proposto. São instrumentos flexíveis que oferecem uma plataforma genérica para a aplicação de algoritmos evolucionários que podem ser utilizados para solucionar diversos tipos de problemas sem a necessidade de o arquiteto ser um programador. Ambos oferecem grandes conjuntos de soluções e não

somente uma final – condizente com a metodologia de Lynn (2007). Entende-se que essa abordagem facilita, pelo alto grau de tolerância de entradas de dados e ações, o desenvolvimento de projetos que lidem com “problemas mal definidos” (MITCHELL, 1975). Como a execução dos *plug-ins* é progressiva, as respostas intermediárias podem ser resgatadas a qualquer momento, fato que não condiciona, necessariamente, a escolha de soluções que estejam no final do processo – que poderiam ser consideradas mais evoluídas, do ponto de vista da biologia, e não do arquitetônico.

Observa-se que esse arcabouço pesquisado está em consonância com a metodologia desenvolvida por Lynn no *Embryological House*. Trabalhou-se com diferentes inflexões que mantiveram suas interdependências topológicas e paramétricas. Pela sistematização aqui proposta, pode-se manter o mesmo embasamento da morfogênese digital, explanada por Kolarevic (2003) e aplicada por Lynn, ao posicionar o computador como seu “sócio” (DOLLENS, 2002; ZELLNER, 1999), permitindo assim “delegar a ‘invenção’ de formas a uma máquina” (VÁZQUEZ RAMOS, 2009). Conjuntamente, esse artigo pode complementar a pesquisa de Bird e LaBelle (2010) no sentido de promover uma preservação de um processo de projeto digital, semelhante ao *Embryological House*, que utiliza e demarca uma tecnologia atual e ainda pouco difundida e empregada no Brasil.

Entende-se que a possibilidade de soluções diversas termina gerando uma atuação do arquiteto – ou do parceiro humano, na equação homem-máquina – que deixa a decisão nas mãos do homem, mas essa resolução não é, necessariamente, a única possível, pois novos algoritmos poderiam ser construídos para que a própria máquina tomasse a decisão final. De fato, a ideia de que é o homem, e não a máquina, quem decide, é anterior ao desenvolvimento da Inteligência Artificial (IA). Essa é uma ideia falaciosa, também, no sentido de que muitas das decisões humanas, hoje, já estão sendo tomadas por máquinas e esse processo tende a se impor a partir da disseminação da tecnologia 5G.

Referências

- BIRD, L.; LABELLE, G. Re-Animating Greg Lynn’s Embryological House: A Case Study in Digital Design Preservation. *Leonardo*, v. 43, n. 3, p. 243-249, 2010.
- BUENO, E. et al. Generative Systems: intertwining physical, digital and biological processes, a case study. *eCAADe SIGraDI 2019*, Porto, Portugal, v. 1, n. 23, p. 25-34, 2019.
- CELANI, G. CAD em arquitetura: Indo além da representação. *Revista UniVap*, São José dos Campos, v. 8, n. 14, p. 46-53, 2001.
- CELANI, G. *Cad criativo*. Rio de Janeiro: Campus, 2003.
- DAVIDSON, S.; MCNEEL, R. *The Grasshopper Primer Project*. MODELAB, [s. l.], 2015. Disponível em: <<http://grasshopperprimer.com/en/>>. Acesso em: 09 ago. 2020.
- DOLLENS, D. *De lo digital a lo analógico*. Barcelona: Gustavo Gili, 2002.
- DRESCH, A. *Design Science Research: método de pesquisa para avanço da ciência e tecnologia*. Porto Alegre: Bookman, 2015.

HARDING, J. Biomorpher. Disponível em: <<https://www.food4rhino.com/app/biomorpher>>. Acesso em: 10 maio 2020.

KOLAREVIC, B. **Architecture in the digital age**: design and manufacturing. New York: Spon Press, 2003.

LYNN, G. **Animate form**. New York: Princeton Architectural Press, 1999.

LYNN, G. Excerpts from a working session with Greg Lynn and CCA curator Howard Shubert recorded in Lynn's, 2007. Disponível em: <<https://www.youtube.com/watch?v=2R6LdAnjS3E&t=1202s>>. Acesso em: 30 agosto 2020.

LYNN, G. Greg Lynn: How calculus is changing architecture, 2009. Disponível em: <<https://www.youtube.com/watch?v=DeyzUysMLy0>>. Acesso em: 12 setembro 2020.

MITCHELL, W. J. The theoretical foundation of computer-aided architectural design. **Environment and Planning B**, v. 2, 1975. 127-150.

PIACENTINO, G. Weaverbird. Disponível em: <<http://www.giuliopiacentino.com/weaverbird>>. Acesso em: 21 setembro 2020.

RUTTEN, D. I eat bugs for breakfast. Disponível em: <<https://ieatbugsforbreakfast.wordpress.com/>>. Acesso em: 15 maio 2020.

VÁZQUEZ RAMOS, F. G. Do analógico ao digital? **XIII Congresso Anual da Sociedade Iberoamericana de Gráfica Digital–SIGRADI**, São Paulo, 2009.

ZELLNER, P. **Hybrid space**: new forms in digital architecture. London: Thames & Hudson, 1999.