

A GRAMÁTICA DA FORMA E AS PRÁTICAS DE PROJETO NA ERA DIGITAL

Denise Capalbo de Oliveira (IC) e Luiz Alberto Fresl Backheuser (Orientador)

Apoio: PIBIC Mackenzie

RESUMO

Nas diversas etapas de projeto de arquitetura, os arquitetos muitas vezes executam ações previsíveis e repetitivas sendo que atualmente temos à nossa disposição tecnologias que são capazes de otimizar determinados processos em busca de uma maior economia de tempo e dinheiro. A pesquisa sobre a gramática da forma tem como objetivo se estudar o desenvolvimento de processos de projeto na atualidade e para o desenvolvimento dos estudos, foi escolhido o modelo de projeto da FDE que possui ambientes escolares já projetados, podendo ser adotados como um vocabulário formal com a intenção de propor uma otimização em suas manipulações e combinações para a concepção de novos projetos. Partindo inicialmente de um programa de necessidades de um edifício escolar da FDE, foi identificado um recorte urbano no bairro da Santa Ifigênia (região central da cidade de São Paulo) onde seria necessária a introdução do programa em questão. Depois de desenvolvido um projeto de escola por métodos convencionais, os elementos apresentados que definem o partido arquitetônico foram extraídos e inseridos no desenvolvimento de um algoritmo através do software de modelagem 3D Rhinoceros e seu plug-in de edição gráfica de algoritmos, o Grasshopper. A parametrização desses elementos resultou em novas propostas de projeto que partiram da proposta inicial. A introdução de programação no processo criativo de projeto potencializou os resultados de maneira que a ação do arquiteto não foi diminuída, mas sim otimizada. Além dos resultados apresentados buscou-se também que através da metodologia desenvolvida possa-se futuramente adaptá-la e desenvolvê-la para novas situações de projeto.

Palavras-chave: Gramática da forma. FDE. Parametrização.

ABSTRACT

In the steps of the architectural project there are still many predictable and repetitive actions even though we have at our disposal technology capable of optimizing certain processes saving a lot of money and time. The research of shape grammar has as an objective to study the making of present project processes, and for that research it has been chosen an FDE project model which has already school environments, allowing it to be used as a formal vocabulary offering an improvement in the manipulation and combination of those models forming new project conceptions. Starting from a

program of necessities of a FDE scholar building, it was made an urban cut of the Santa Ifigênia district (downtown São Paulo) where it'll be needed an introduction to the program in question. After it was made the school project by conventional methods the elements of the architectural party shown was taken and inserted via algorithm in a 3D modeling software Rhinoceros and your graphic algorithms edition plugin Grasshoper. The parametrization of those elements resulted in new project proposals originated from the original one. The insertion of the programming in the creative process potentiated new results in a way the role of the architect was not diminished but enhanced. Beside the results showed, it was intended to show new ways to adapt and develop new projects through the methodology enunciated.

Keywords: Shape grammar. FDE. Parameterization.

1. INTRODUÇÃO

Mesmo com a tecnologia à nossa disposição, muitas vezes arquitetos trabalham através de métodos previsíveis e repetitivos em diversas etapas de projeto. Essa metodologia pode demandar um certo tempo de trabalho, o que ocasiona em despesas desnecessárias para as empresas de arquitetura, sendo que alguns métodos poderiam ser otimizados.

É importante estabelecer uma interação ativa com as ferramentas computacionais onde o arquiteto informa dados e o computador os processa podendo sugerir novas propostas afim de agilizar e potencializar os processos criativos que muitas vezes são feitos de maneira manual. Com a possibilidade de manipulação de dados em uma etapa de conceituação de projeto de arquitetura é possível se obter mais opções de projeto de forma que o arquiteto possa avaliar melhor quais opções seriam mais interessantes para determinada situação.

Tendo como referência estudos como o de Stiny e Mitchell (1978) por exemplo, onde foi pesquisada a gramática da forma de projetos de villas de Andrea Palladio, o presente trabalho busca um desenvolvimento de estudos que se baseiam na gramática da forma da FDE (Fundação para o Desenvolvimento da Educação) de maneira que seus ambientes escolares já projetados sirvam como vocabulário formal para uma experimentação sobre possibilidades de combinações dos ambientes entre si através da manipulação de dados.

2. REFERENCIAL TEÓRICO

Definição

Os primeiros a desenvolverem o conceito de gramática da forma aplicada à arte foram George Stiny e James Gips na década de 1970, quando se referenciaram nas teorias do linguista Chomsky a respeito das estruturas gramaticas das composições de palavras. As palavras são compostas por letras do alfabeto (símbolos) que são capazes de formar infinitas sentenças a partir de suas combinações. Sendo assim a gramática linguística dá origem a gramática da forma no sentido lógico de sua caracterização (STINY; GIPS, 1971).

Baseado nesse conceito, Stiny e Gips fizeram experimentos afim de implementar esse tipo de lógica na prática da concepção de obras de arte, mais especificamente para a composição de pinturas e esculturas com a finalidade de otimizar os processos criativos de forma que essas regras lógicas de processo poderiam ser introduzidas em algoritmos para que possam processar passos já previstos pelos autores (STINY; GIPS,

1971). No final desse processo automatizado o que se esperava era que o autor da obra tivesse a opção de poder selecionar dentre esses resultados apresentados quais seriam as melhores propostas que resultaram do mesmo conjunto de regras que foi estabelecido.

Trata-se de um processo generativo, que se inicia com regras bem definidas, mas mesmo assim são capazes de apresentar resultados muito diferentes uns dos outros apesar de suas origens idênticas. Para o experimento em questão, foram estabelecidas regras para a composição das formas e da aplicação dos materiais.

Um método de geração de formas que utiliza gramáticas da forma, seleciona suas formas primitivas e as apresenta através de regras específicas. Um formalismo para ser completo, possui uma especificação generativa de classes não-representacionais, como pinturas geométricas ou esculturas que são compostas por componentes estruturais primários, a gramática da forma. (...) No projeto com base nas especificações generativas, o artista escolhe relações estruturais e materiais e, em seguida, determina através de algoritmos os objetos de arte resultantes. (...) O nosso objetivo subjacente é a utilização de técnicas formais, geradores para produzir bons objetos de arte e desenvolver a compreensão do que faz bons objetos de arte. (STINY G; GIPS J, 1971, p. 125) ¹

¹ A method of shape generation using shape grammars which take shape as primitive and have shape specific rules is presented. A formalism for the complete, generative specification of a class of nonrepresentational, geometric paintings or sculptures is defined, which has shape grammars as its primary structural component (...) In design based on generative specifications, the artist chooses structural and material relationships and then determines algorithmically the resulting art objects. (...) Our underlying aim is to use formal, generative techniques to produce good art objects and to develop understanding of what makes good art objects. (STINY G; GIPS J, 1971, p. 125)

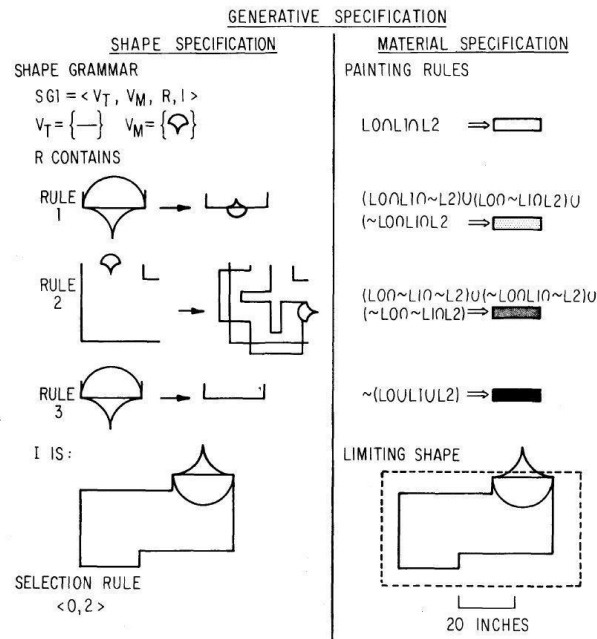


Figura 1: Esquema explicativo de uma *shape grammar*, sendo do lado esquerdo a lógica da gramática da forma em si e do lado direito a lógica da aplicação de material
 Fonte: STINY, G.; GIPS J., 1971, p. 127

Uma vez que as decisões são tomadas quanto às relações que vão estar na base de uma classe de pinturas ou esculturas, uma especificação generativa está definida e as consequências estruturais e materiais das relações são determinadas através de algoritmos. Isto permite que o artista para obter objetos de arte com simplicidade, originalidade e complexidade visual que são fiéis a essas relações e que seria difícil conceber por outros meios. (STINY G; GIPS J, 1971, p. 134) ²

Vocabulário, regras e sentenças

As gramáticas são compostas basicamente por **vocabulário**, **sentenças** e **regras** e dependendo de como essas classificações são desenvolvidas e aplicadas, determinarão o tipo de gramática da forma que será apresentado.

Vocabulário = conjunto finito de símbolos que irão compor as sentenças;

Sentenças = possíveis relações resultantes entre os símbolos do vocabulário;

² Once the decisions are made as to the relationships that are to underly a class of paintings or sculptures, a generative specification is defined and the structural and material consequences of the relationships are determined algorithmically. This enables the artist to obtain art objects with specification simplicity and visual complexity which are faithful to these relationships and which would be difficult to design by other means. (STINY G; GIPS J, 1971, p. 134)

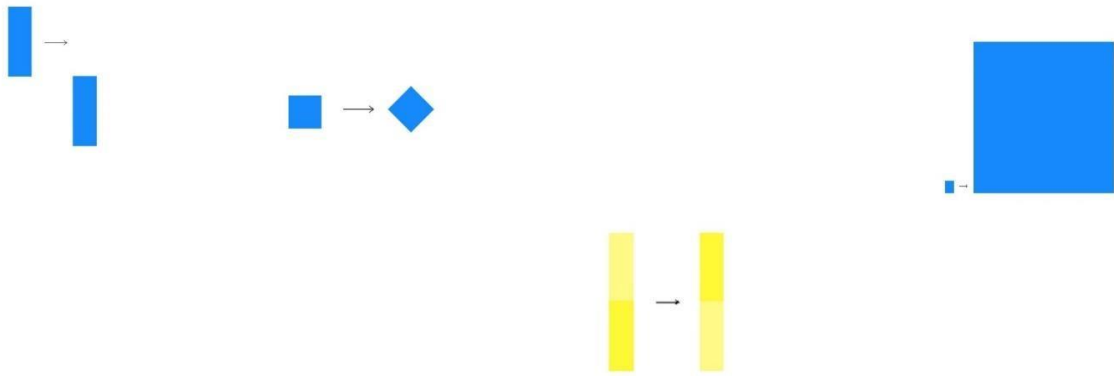


Figura 7: Ilustração de operações euclidianas, sendo 1 - translação, 2 - rotação, 3 - reflexão e 4 - escalamento Fonte: MORAES, 2016, p. 25

A gramática de Palladio

Partindo da **Gramática Analítica**, que é o tipo de gramática que estuda a estrutura dos desenhos explorando seus processos até o resultado final, é possível entender a geração de outras sentenças partindo das mesmas regras, simulando as possíveis variações desse estilo. George Stiny e William Mitchell estudaram a lógica de projeto das Villas de Andrea Palladio (1978), afim de traduzir essa lógica em regras práticas e aplicáveis para a concepção de novas Villas que partiriam de regras já definidas e limitadas, mas aplicadas de uma forma que resultasse em opções diversas de projeto, mesmo partindo de regras iguais.

O recorte de estudo de Stiny e Mitchell foi o projeto da Villa Malcontenta, onde a obra foi analisada e foram extraídos os raciocínios por trás de sua concepção geométrica, que foram explicitadas em formas de regras, já baseadas em regras que o próprio Palladio publicou em seu livro “I quattro libri dell’architettura” (1570).

Foram criadas regras lógicas para cada uma dessas etapas de projeto, de forma que poderiam ser aplicadas de diversas maneiras, partindo de uma mesma lógica. Esse experimento foi possível pois eles enxergaram uma semelhança formal nas diversas construções de Palladio, que é o que caracteriza a existência de uma gramática nesses projetos (STINY; MITCHELL, 1978). Sendo assim eles buscaram definir essas semelhanças, criando uma gramática parcial generativa do estilo Palladiano (sem levar em consideração o desenvolvimento de fachadas e outros detalhamentos).

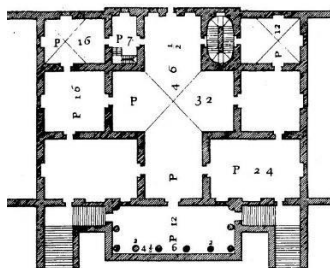


Figura 8: Planta da Vila Malcontenta Fonte: STINY G.; MITCHELL W. J., 1978, p. 6.

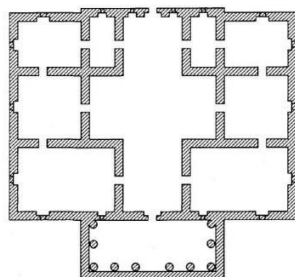


Figura 9: Planta gerada através de *shape grammar* Fonte: STINY G.; MITCHELL W. J., 1978, p. 16.

A gramática fornece a base para a classificação das Villas em termos de propriedades das sequências de regras aplicadas a elas para gerar seus desenhos e até mesmo para contar o número de possíveis Villas de um certo tipo (STINY; MITCHELL, 1978). Ou seja, com esses parâmetros é possível através de uma análise combinatória das possíveis aplicações das regras, chegarmos a um cálculo do número de todos os possíveis projetos de Villas dentro do formato já estabelecido por Palladio em um grid de 5 x 3. A gramática da forma pode também ser explorada como um método de entendimento, desenvolvimento ou definição de um estilo arquitetônico e no caso aqui mostrado, funcionou como um mecanismo de definição de um sistema estético baseado na teoria desenvolvida por Stiny e Gips em 1978.

Outra pesquisa importante foi o trabalho de José Pinto Duarte que efetuou uma teorização geral a respeito da gramática de projeto de Álvaro Siza, com um recorte na obra da Quinta da Malagueira (2001). Partindo do princípio de que para a concepção do projeto da Quinta da Malagueira, Álvaro Siza preocupou-se em criar variações nas tipologias (total de 1.200 habitações sendo mais de 35 layouts diferentes), José Pinto Duarte estudou essas variações nas plantas das habitações e as introduziu em um algoritmo de maneira que esses elementos pudessem ser combinados de diversas maneiras afim de se atingir resultados mais adequados para o futuro morador de cada habitação (DUARTE, 2001).

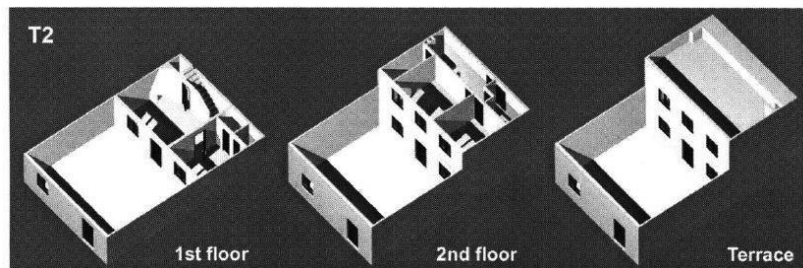


Figura 10: Esquema de casas da Quinta da Malagueira Fonte: DUARTE, 2001, p. 123

Gramática na Era Digital

Existe uma estreita relação entre a lógica computacional e a lógica das gramáticas da forma, devido ao modo que os computadores trabalham, a base de algoritmos e de interpretação de símbolos seriam as ferramentas ideais para trabalhar com a gramática da forma e explorá-las mais a fundo.

Gips (1999) coloca que existem 4 tipos de implementações computacionais para as gramáticas da forma, seriam elas:

Reconhecedor (interpreter) – o usuário dá as regras para o computador gerar as sentenças; **Analisador sintático** (parsing) – é apresentada uma gramática e uma forma, se a forma dada for reconhecida como parte da gramática, o programa dará possíveis sequências de regras que poderiam tê-la produzido, como se fosse uma decomposição; **Inferência** (inference) – é dado um conjunto de formas para ser construída uma gramática partindo dele afim de extrair outras composições semelhantes; **Assistência** (CAD shape grammar) – criação de gramáticas da forma auxiliado por determinadas ferramentas ou plug-ins (GIPS, 1999).

Além da representação

Gabriela Celani (2002) desenvolve sua pesquisa com o objetivo de demonstrar possíveis aplicações das ferramentas CAD para desenho de arquitetura além de uma simples representação. Baseia-se nos estudos de Mitchell (1975), que de acordo com ele nós devemos explorar as ferramentas computacionais de uma maneira mais ambiciosa, buscando que ela seja auxiliar no sentido de geração automatizada de alternativas de projeto, tudo isso com base em uma lógica bem definida feita pelo arquiteto. Outra opção também seria a utilização desse sistema de forma a gerar e avaliar o resultado como sendo adequado para aquela determinada situação ou não.

Para isso é necessário que antes de serem feitos os experimentos deve-se aprender a descrever um procedimento de projeto arquitetônico através de algoritmos. Não sendo essencial a compreensão de linguagem de programação, mas sim de lógica de programação, podendo ter como exercício prático o entendimento de conceitos abstratos sendo explicados de maneira concreta e lógica.

Os exercícios onde se utilizam gramática da forma costumam ser eficientes para a aplicação desses conceitos, pois esse método por si só (independente da utilização de ferramentas CAD) já são resultado de sequências lógicas de ações, tornando a lógica da arquitetura e a lógica de programação, estratégias complementares (CELANI, 2002).

Conceitos:

1 - Simetria = para determinar o número de formas diferentes combinadas;
2 - Recursão = operações ilimitadas de repetição, são definidas ao longo do processo;

3 - Regras de design = regras que o arquiteto determina partindo de sua análise da situação;

4 - Formas parametrizadas = formas que se modificam através de variáveis;

5 - Sistemas generativos = gera novas formas partindo de uma família específica;

6 - Algoritmização = algoritmos atuando na criação de espaços;

7 - Formas emergentes = surgimentos de novas formas a partir de operações matemáticas.

A algoritmização no processo de projeto deverá partir de critérios já estabelecidos pelo arquiteto, sendo o princípio de tudo as regras de Design. Partindo delas se estabelecem os critérios e limites de onde o algoritmo poderá atuar, sendo as diversas combinações possíveis definidas matematicamente, tornando-se inúteis caso não respeitem as regras de Design. O arquiteto tem que definir mentalmente a sua intenção e explicitá-la logicamente em algoritmos.

Para conseguirmos chegar nessa lógica, devemos tentar descrever a nossa intenção de projeto, identificando quais são as variáveis do problema, as restrições para cada variável e as condições iniciais da questão, sempre tendo como conclusão desse raciocínio um valor numérico. O ideal é a aplicação de um algoritmo em situações onde o arquiteto enfrenta problemas onde é aplicado o mesmo tipo de procedimento mas que

variam de situações, de forma que esses problemas sejam traduzidos em fórmulas matemáticas tendo suas constantes e variáveis (CELANI, 2002).

A sequência de passos para gerar o desenho tem que ser descrita incorporando, se necessário, condicionais e repetições estruturais (como SE/ENTÃO e FAÇA/ENQUANTO). Todas as possibilidades de soluções têm que resultar em números. (CELANI, 2002, p. 79) ³

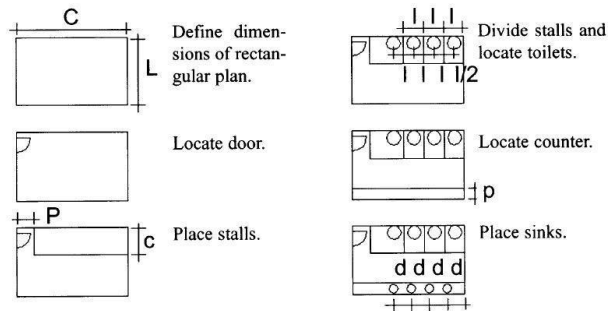


Figura 11: Passo a passo da construção de um layout Fonte: CELANI, Gabriela, 2002

3. METODOLOGIA

Vocabulário – Proposta de projeto

Sendo responsáveis pela qualidade técnica das construções escolares, a FDE disponibiliza para os arquitetos que desenvolverão os projetos, um Catálogo Técnico de Ambientes, com as especificações técnicas já definidas. Devido essa característica, justifica-se a escolha de se trabalhar com o vocabulário já apresentado pela instituição ao invés de se projetar um novo vocabulário para serem feitos os experimentos.

Para o desenvolvimento do trabalho, foi escolhido o programa para Escola de Ensino Fundamental Ciclo II e Ensino Médio, tendo já definido então qual será a biblioteca de ambientes (vocabulário) que será utilizada no projeto. Partindo de uma biblioteca de ambientes já estabelecida, foi identificado dentro do Catálogo Técnico de Ambientes da FDE o vocabulário formal a ser utilizado no projeto concebido por métodos convencionais e também no experimento.



Figura 12: Projeto da escola Fonte: Denise Capalbo

Tendo primeiro um programa de necessidades sem uma proposta de projeto, foi então selecionado um recorte urbano onde o programa em questão fosse necessário para o desenvolvimento de uma proposta de projeto de escola.

Com base nas principais decisões para a concepção do partido arquitetônico desenvolvido, foram extraídos elementos que possivelmente seriam constantes e variáveis formais significativas para a caracterização do projeto de forma que esses elementos fossem parametrizados a fim de serem feitos os experimentos. Sendo assim a proposta é que através da manipulação de um vocabulário já definido, seja possível a apresentação de novas propostas de projeto, dentro de um mesmo partido arquitetônico já conhecido.

Com a utilização do software de modelagem 3D Rhinoceros e seu plug-in de edição gráfica de algoritmos (Grasshopper), o primeiro passo do experimento foi introduzir uma malha estrutural (sugerido pela FDE) de 7.20 x 10.80m tanto para o pavimento térreo quanto para o pavimento superior, sendo projetados sobre o modelo 3D do recorte urbano.

Sendo o grid composto por um total de 36 módulos de 7.20 x 10.80m (6 linhas e 6 colunas), no qual foram inseridos os ambientes cuja as dimensões já se encaixam dentro dessa modulação, foi adotado como primeiro parâmetro de manipulação de dados a posição do ponto de origem da geometria base do ambiente na lista de dados do grid. Ou seja, as definições que dão origem a geometria de cada ambiente foram feitas pensando na possibilidade de posteriormente ser estabelecida uma relação de posicionamento entre elas. Sendo assim a primeira possibilidade de posicionamento dos ambientes é por módulo da malha estrutural que é manipulado por um slider (barra numérica deslizante) com variação de 0 a 35.

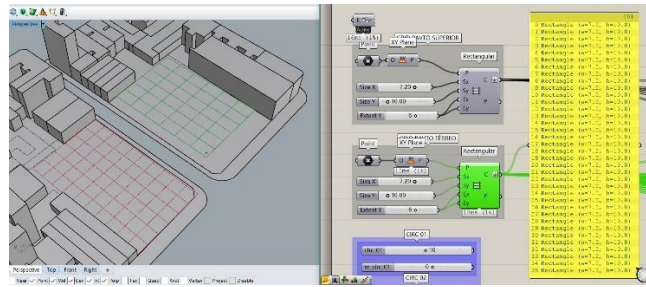


Figura 13: Trecho da definição para inserção de malha estrutural Fonte: Denise Capalbo

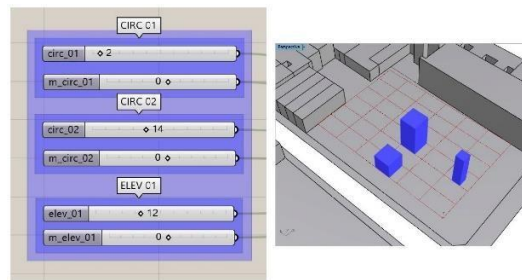


Figura 14: Figura 41 – Exemplo de parâmetros de posicionamento dos pontos de origem da geometria de cada ambiente na malha estrutural e de posicionamento das respectivas geometrias no modelo 3D Fonte: Denise Capalbo Fonte: Denise Capalbo

Os parâmetros “circ_01”, “circ_02” e “elev_01” apresentados são responsáveis pelo posicionamento dos pontos de origem da geometria de seus respectivos ambientes (circulação 01, circulação 02 e elevador 01) tendo como referência sempre a malha estrutural.

Os parâmetros “m_circ_01”, “m_circ_02” e “m_elev_01” também apresentados na imagem são responsáveis por acrescentar um módulo estrutural a mais no posicionamento quando houver necessidade. Seus valores deverão ser 0, 1, 2 ou 3 e sempre serão multiplicados pelo módulo de 3.60m, afim de explorar novas possibilidades de posicionamento dos ambientes no módulo além do já determinado pela definição do ambiente em questão.

As definições que dão origem a forma dos ambientes foram feitas de maneira que suas propriedades básicas possam ser modificadas quando necessário, tais como **largura, altura, comprimento e posicionamento em relação ao ponto de origem de sua geometria**. Todos os ambientes apresentados para o experimento foram concebidos de maneira muito semelhante com poucas alterações em alguns casos.

Exceto nas definições das salas de aula que demandam uma maior liberdade de posicionamento devido sua necessidade de repetição em maior número.

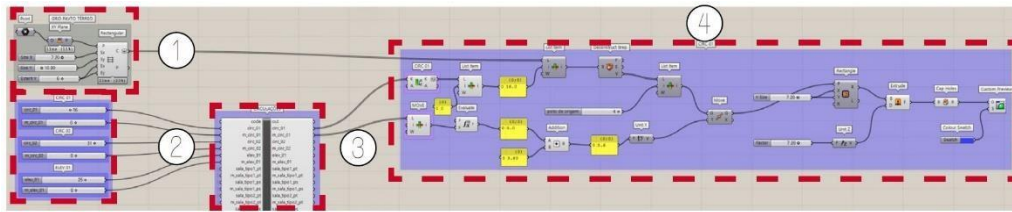


Figura 15: Trecho da definição que exemplifica como foram inseridos os ambientes que compõe o vocabulário. Fonte: Denise Capalbo

1) Definição que insere a malha estrutural, além dessa função também serve de lista base para o “list item” da definição 4; **2)** Parâmetros de posicionamento dos ambientes no grid que deverão servir de inputs para o componente de código em Python; **3)** Componente com o código escrito em Python que deverá estabelecer relações de domínio entre os ambientes (qual ambiente terá prioridade para ocupar determinada posição no grid); **4)** Definição que compõe a geometria base do ambiente “circulação 01”.

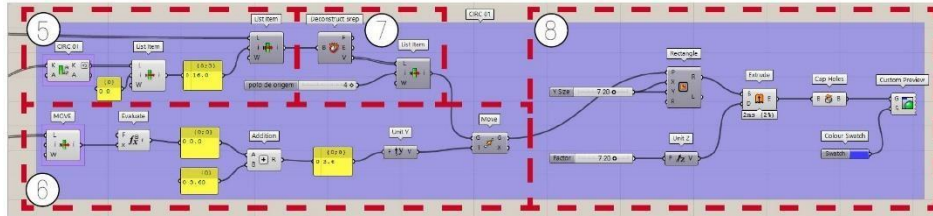


Figura 16: Definição do ambiente “circulação 01” que também serve de exemplo para explicar como foram feitos os demais ambientes. Fonte: Denise Capalbo

5) Trecho da definição responsável por captar o item (retângulo no grid) mais adequado para se começar o desenvolvimento da geometria; **6)** Trecho da definição responsável por atribuir um movimento dentro do módulo de 3.60m quando houver necessidade. O eixo onde deverá ocorrer esse movimento poderá ser alterado também conforme necessidade; **7)** Trecho da definição responsável por criar o ponto de origem da geometria (que ainda deverá sofrer influência do parâmetro de movimento do item 3); **8)** Definição que cria efetivamente a geometria do ambiente em questão.

Regras – Parâmetros e manipulações

Depois de inseridos os ambientes, apresentou-se um problema onde em vários momentos os ambientes tinham a possibilidade de ocupar a mesma posição na malha estrutural, se sobrepondo. Sendo assim foram introduzidos códigos utilizando a linguagem de programação Python que mantém os movimentos dos ambientes relacionados entre si com base em uma regra.

Quando determinado ambiente for posicionado no mesmo módulo onde já se encontra outro, o valor do segundo será **[valor numérico do módulo] + 1**. De forma que o ambiente modificado mantenha o domínio sobre o outro que deverá ser modificado em função do primeiro.

```

Grasshopper Python Script Editor
file Help
1 val_circ_02 = ()
2
3 if circ_02 == circ_01:
4     val_circ_02 = circ_02 + 1
5     val_m_circ_02 = m_circ_02
6 else:
7     val_circ_02 = circ_02
8     val_m_circ_02 = m_circ_02
9
10 circ_02 = val_circ_02
11 m_circ_02 = val_m_circ_02
12 circ_01 = circ_01
13
14 val_elev_01 = ()
15
16 if elev_01 == circ_01:
17     val_elev_01 = elev_01 + 1
18     val_m_elev_01 = m_elev_01

```

Figura 17: Trecho do código em Python que mantém as relações de posicionamento entre os ambientes. Fonte: Denise Capalbo

if = condicional “**se**” que apresenta a situação; **else** = condicional “**então**” que executa uma ação; **circ_02** = ambiente que terá seu valor modificado (posição); **m_circ_02** = condicional de movimento que irá variar de 0 a 3 e sempre será multiplicada por 3.60m; **circ_01** = ambiente que manterá sua posição já determinada (domínio);

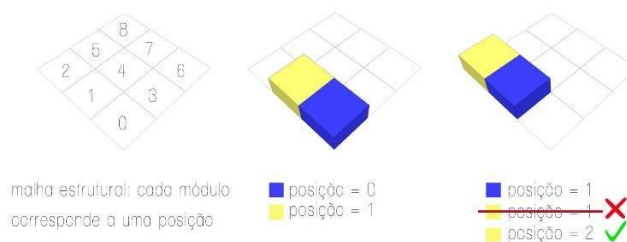


Figura 17: Esquema explicativo da lógica de posicionamento dos ambientes. Fonte: Denise Capalbo

A condicional de movimento será sempre 0 e quando houver necessidade de algum deslocamento a mais no modelo (além da mudança de módulo na malha estrutural), seu valor poderá ser 1, 2 ou 3. Eles representam “um passo”, 1 significa 1 passo de 3.60m, 2 significa 2 passos de 3.60m e 3 significa 3 passos de 3.60m. Esses dados serão transmitidos para o componente de movimento do ambiente. Isso é possível através da introdução da expressão “multiplicar por 3.60” no componente de movimento que sempre irá multiplicar o dado recebido por 3.60.

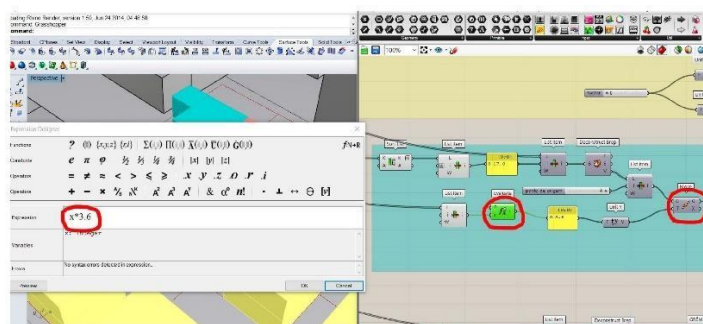


Figura 18: Trecho da definição onde os parâmetros de movimento chegam no componente *move* e através da expressão “x*3.60” são convertidos em passos a cada 3.60m. Fonte: Denise Capalbo

As salas de aula devido o fato de serem os únicos ambientes com a possibilidade de se repetirem várias vezes, tem seu sistema de formação diferente dos demais, sendo feitas através de séries numéricas, que permitem a manipulação de dados quanto ao “**ponto de início**” da contagem na lista de dados da malha onde começarão a serem posicionadas as salas; “**espaçamento entre módulos**”, que seriam os espaçamentos entre cada sala contando em módulos estruturais de 7.20 x 10.80m e “**quantidade de salas**”, que é a quantidade de salas que se apresentará no modelo. Quando os dados do ambiente que está em posição de domínio se encontram com os dados de alguma das salas, acontecia uma sobreposição de dados entre as próprias salas de aula de maneira que quando o valor da sala de aula em questão for modificado para “[valor numérico do módulo] + 1”, uma sala passava para o módulo seguinte e se sobrepunha com outra sala de aula pertencente a mesma série de dados.

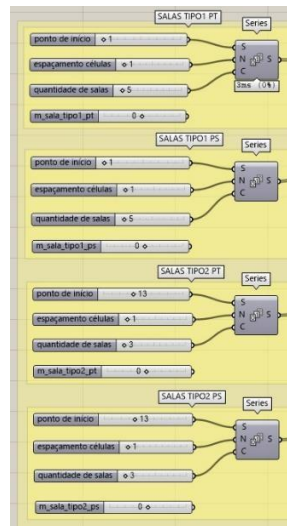


Figura 19: Séries numéricas que dão origem as salas de aula. Fonte: Denise Capalbo

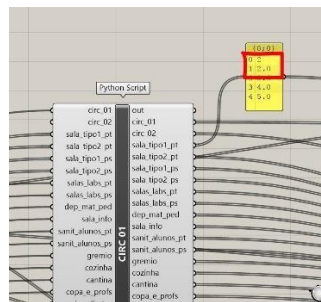


Figura 20: Dados resultantes da combinação de “circ_01” com “sala_tipo1_pt” que tendo como domínio “circ_01”, comprova que existe uma sobreposição de dados entre as salas de aula. Fonte: Denise Capalbo

Para a resolução desse problema os dados que o código gerava em relação as salas de aula que tinham seus valores modificados, foram convertidos em um mesmo tipo de informação para serem manipulados. Com as informações organizadas é possível que o dado repetido (que gera a sobreposição), seja excluído da lista criando uma nova lista com um dado a menos (uma sala a menos). O valor numérico excluído (que sempre será = 1) deverá ser somado ao último número da lista para completar a sequência, compensando o dado que foi excluído e gerando assim uma lista completa com a quantidade de salas de aula correta e sem sobreposições. Ou seja, a lista final sempre terá o número correto de dados (número de salas de aula) e a sequência das posições delas nos módulos estruturais, mas pulando a posição que está ocupada pelo ambiente de domínio.

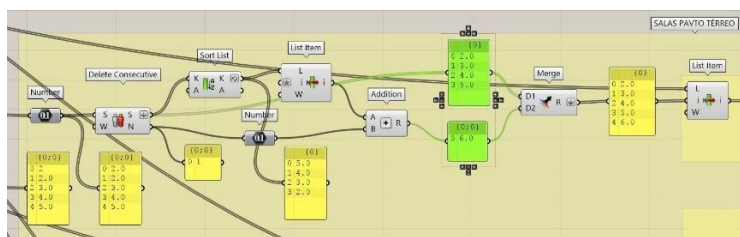


Figura 21: Trecho da definição que cria a relação de posicionamento das salas de aula com os demais ambientes de forma que uma sequência numérica seja composta sem nenhum elemento repetido que resultaria em uma sobreposição no modelo. Fonte: Denise Capalbo

A lógica desenvolvida para a manipulação de dados se mantém a mesma para os dois terrenos que compõem o projeto (para as duas quadras), apenas modificando-se o vocabulário (conjunto de ambientes) e as dimensões das malhas estruturais, mantendo as manipulações independentes entre si e dependentes de suas malhas estruturais.

4. RESULTADO E DISCUSSÃO

Sentenças – Resultados obtidos

Os ambientes podem ser posicionados da mesma maneira que já haviam sido apresentados no projeto inicial, mas também de diversas outras maneiras que se fossem estudadas através de métodos convencionais de se projetar, demandariam muito mais tempo para serem feitas.

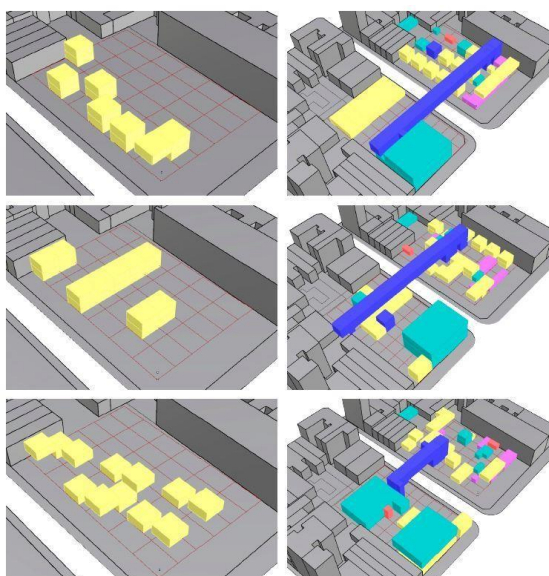


Figura 22: Esquerda - exemplo de sentenças compostas apenas por salas de aula. Direita - Exemplo de sentenças compostas por todos os símbolos (ambientes) da escola. Fonte: Denise Capalbo

As sentenças obtidas não apresentaram completamente os resultados que eram esperados para o experimento sendo que a maneira como foi desenvolvido o código em Python não permitiu que se mantivesse uma relação de domínio de **todos os ambientes** com **todos os demais ambientes**, de maneira que em vários momentos o código se opunha a ele mesmo, com trechos executando uma ação e outros executando a ação contrária. Por exemplo, quando “circ_01” encontrar “circ_02”, “circ_02” deverá mudar de lugar, tendo “circ_01” como domínio. Quando “circ_02” encontrar “circ_01”, “circ_01” deverá mudar de lugar, tendo “circ_02” como domínio. Tendo essas informações opostas não foi desenvolvido um método que entende quando é o momento em que deverá ser executada determinada ação, o código desenvolvido executa apenas a primeira ação que lhe é apresentada.

A condicional de movimento foi apresentada como uma solução para os encaixes entre os ambientes que passam dos limites do módulo estrutural. Entretanto essa solução se mostrou eficiente apenas quando se trata da relação de ambientes sequenciais (na sequência da lista de dados), quando a sobreposição acontece pelas laterais dos ambientes, a solução torna-se ineficiente. Ou seja, a maneira como foi pensada a manipulação de dados com base no ponto de origem da geometria base dos ambientes, não é suficiente para se estabelecer uma completa relação de posicionamento, sendo necessária uma identificação completa da geometria base dos ambientes para estabelecer uma relação por completo.

Entretanto, os resultados obtidos foram positivos dentro do que se esperava para a pesquisa, sendo que o sistema desenvolvido tem muitas condições de ser melhorado e adaptado para novas situações de projeto futuramente.

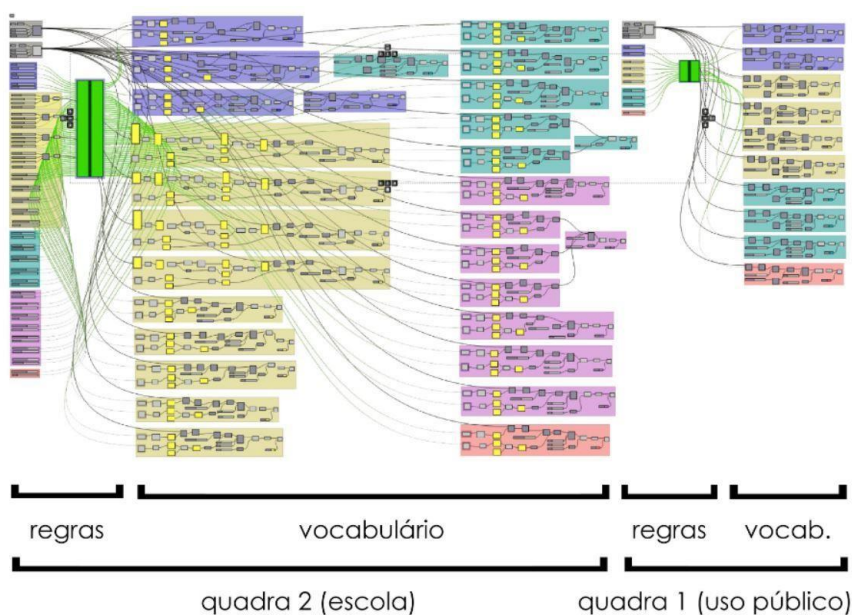


Figura 23: Definição completa do experimento sendo a quadra 2 correspondente ao projeto da escola e a quadra 1 correspondente ao programa de uso público. Fonte: Denise Capalbo

5. CONSIDERAÇÕES FINAIS

Observa-se que a introdução da programação em meio a um processo criativo não exclui a ação do arquiteto, sendo que o processo foi otimizado pelo computador, mas as definições das regras que dão origem às sentenças foram definidas por um arquiteto.

A lógica aplicada para a concepção do projeto inicial (feito por métodos convencionais), conseguiu ser traduzida para o algoritmo que trabalha de maneira ativa sendo que ele responde com sugestões de novas combinações que não seriam tão rapidamente previsíveis dentro de um método convencional de se projetar.

A metodologia utilizada poderia ser desenvolvida e adaptada em diversas situações de projeto futuramente, sendo que se trata de um trabalho que não procurou direcionar-se efetivamente a um resultado certo, mas sim através de um estímulo à pesquisa e reflexão crítica sobre as práticas de projeto na atualidade, valorizando o processo em si, afim de propor um estímulo a experimentações de novas metodologias de trabalho.

6. REFERÊNCIAS

CELANI, Gabriela. **Beyond analysis and representation in CAD: a new computational approach to design education.** PhD - Massachusetts Institute of Technology – Cambridge, MA – 2002.

DUARTE, José Pinto. **Customizing Mass Housing: A Discursive Grammar for Siza's Malagueira Houses.** PhD - Massachusetts Institute of Technology – Cambridge, 2001.

GIPS, James. **Computer Implementation of Shape Grammars.** Boston College, Chestnut Hill, MA - 1999.

GIPS, James. **Shape Grammars and their Uses. Artificial Perception, Shape Generation and Computer Aesthetics.** Suíça, 1975.

MORAES, Rodrigo Agostini de. **ABC da arquitetura. Fundamentos conceituais para uma gramática de projeto.** Dissertação (mestrado), Faculdade de Arquitetura e Urbanismo da Universidade de São Paulo – São Paulo, SP – 2016.

STINY, George. **Shape: Talking about Seeing and Doing.** Massachusetts Institute of Technology - Cambridge, MA - 2006.

STINY, G.; GIPS J., **Best Computer Papers of 1971: Shape Grammars and the Generative Specification of Painting and Sculpture.** Iugoslávia, 1971.

STINY, G.; MITCHELL, W. J. **Environment and Planning B: The Palladian grammar.** Inglaterra, v. 5, páginas 5-18, 1978.

TEDESCHI, Arturo. **AAD Algorithms-Aided Design - Parametric strategies using Grasshopper.** Le Penseur, Brienza - 2014.

Contatos: denise.capalbo@yahoo.com.br e luiz.backheuser@mackenzie.br