



XXIV International Conference  
of the Iberoamerican Society  
of Digital Graphics  
Medellin | Colombia

## Form and urban life in Christopher Alexander's work: translation of patterns for parametric code

Isabella Eloy Cavalcanti

Universidade Federal de Pernambuco | Brasil | isabellaeloy@hotmail.com

Leticia Teixeira Mendes

Universidade Federal de Pernambuco | Brasil | leticiamendes.edu@gmail.com

### Abstract

Computational design, specifically parametric modeling, has played important role in reaching complex forms, optimizations and automations of design processes. In addition to using parametric technology as a tool to generate form, this article aims to discuss the potential of parametric design as a connection between theory and design activity, both in practice and in the teaching activity. To illustrate that, this paper will present results of a bigger research that used the work of the architect Christopher Alexander as a basis for the development of decision-making instruments that deal with the complexity between form and urban life.

**Keywords:** Urban design; Parametric modeling; Computational design; Christopher Alexander.

Para Terzidis (2006), pela primeira vez o *design* arquitetônico e, adicione, urbanístico deve estar alinhado não a um formalismo ou racionalismo definido, mas a uma forma inteligente e de criatividade rastreável. A noção de que algum estilo específico ou medida fixa é capaz de fornecer soluções prontas para os problemas projetuais está gradativamente menos presente nos processos de planejamento e de tomada de decisão.

Jones [1966(1962)] aponta que temos apenas duas opções na tentativa de alcançar os padrões estabelecidos pelos projetos urbanos de sucesso do passado: tornar as nossas cidades simples e descomplicadas como eram há mil anos – o que é quase impossível – ou aceitarmos as complexidades do mundo ao nosso redor e buscar sistemas de decisões mais poderosos para basear a tomada de decisão em projetos urbanos.

Diante dessa indiscutível complexidade, Batty e Longley (1997) afirmam que a capacidade limitada de lidar com a dinâmica não linear de crescimento espacial também foi concebida como a crise de planejamento racional, que usa apenas análises newtonianas e probabilidades estatísticas lineares. O *modus operandi*, que antes era baseado na tentativa de adequar realidades à soluções quantitativamente elaboradas, torna-se cada vez mais a busca de solução para as especificidades e variáveis de cada contexto.

Tratando especificamente do *design* urbano, além da complexidade, Beirão (2012) destaca que é diferente de processos de convencionais de *design* arquitetônico e de produto uma vez que o objeto do projeto não é um elemento único, mas um conjunto de partes envolvidas em relações econômicas e simbólicas complexas. Enquanto um projeto de edificação tem um escopo de influência visível e razoavelmente limitado, as decisões na escala urbana podem gerar consequências desastrosas adiante, mesmo que inicialmente não apresente seus efeitos (Kou & Zausinger, 2010).

Posta a seriedade e a dimensão dos processos decisórios na escala urbana, buscar modos dinâmicos, mais poderosos de processamento, para além do cérebro humano, e mais abertos à contribuição multidisciplinar faz-se cada vez mais necessário. Nesse sentido, Chaillou (2019) aponta que a máquina - que em um momento foi extensão do lápis -, atualmente pode ser utilizada para mapear conhecimento e ser treinada para auxiliar o processo de criação de opções de *design* que considerem cada vez mais a complexidade e grande quantidade de variáveis que envolve os problemas projetuais.

Ao utilizar tecnologias computacionais, tal como a paramétrica, os problemas de cognição e computação de variáveis podem ser amenizados uma vez que cria-se um sistema de multivariáveis rapidamente computáveis. A tecnologia paramétrica mostra-se, assim, capaz de alterar não apenas a instrumentalidade do processo, otimizando cálculos e visualizações, como contribuir para a mudança no fluxo projetual, tornando-o mais aberto e claro.

Celani (2002), assinala que a computação tem sido usada na arquitetura na síntese de projetos (geração), em sua modelagem (representação) e análise (avaliação). Além dessas importantes inserções, defende-se e investiga-se neste artigo que a tecnologia computacional também é capaz de tornar prático um saber, ou seja, ser utilizada como meio para tradução e visualização de conhecimento. Conhecimento esse aqui utilizado como instrumento para colaborar na tomada de decisão no que tange a forma e vida urbana.

Ao traduzir algum conhecimento para uma linguagem computacional, outro nível de interação é estabelecida entre a mente humana e a computacional, uma relação a nível de pensamento. Como aponta Wing (2008), um pensamento computacional mais profundo nos ajudará não apenas a lidar com sistemas mais complexos, mas também analisar a enorme quantidade de dados e variáveis disponíveis.



Diante da possibilidade de tradução de conhecimento para o meio digital, ampliando a visualização e atualizando a plataforma de trabalho, este artigo apresenta parte dos resultados de uma pesquisa que propõe a união entre princípios da obra "A Pattern Language" (Alexander *et al.* 1977) e a tecnologia computacional paramétrica, buscando desenvolver modos de experimentação que tomam uma base teórica projetual consagrada e a traduzi-la de uma linguagem natural (escrita) para uma linguagem formal (computacional).

A escolha dessa obra fez-se primeiramente pela relevância e consagração na historiografia arquitetônica e urbanística. Como apontam Kuo e Zausinger (2010), paralelamente ao desenvolvimento na teoria da complexidade e na tecnologia computacional, *A Pattern Language* (1977), mostrava uma mudança de paradigma no campo teórico ao defender que as organizações urbanas não são como uma árvore de progressão escalar linear, mas que envolve diferentes escalas de *design*, interligadas ao mesmo tempo que detêm autonomia.

Além da importância teórica, a produção científica de Christopher Alexander influenciou grandemente o desenvolvimento da área de ciências da computação (Dawes e Ostwald, 2017). O caráter sistêmico e aproximado a teorias base para a Era Digital como a Teoria dos Sistemas, Cibernética e Pesquisa Operacional (P.O.) também justificam a escolha dessa obra para o início de um instrumento computacional.

A estrutura da obra e a característica mutável e adaptável de seus padrões também contribui para o início da investigação de tradução de saberes teóricos consagrados. Seus princípios claramente apresentados tem o potencial de ser base no estabelecimento de parâmetros e a geração de códigos (Figura 1).

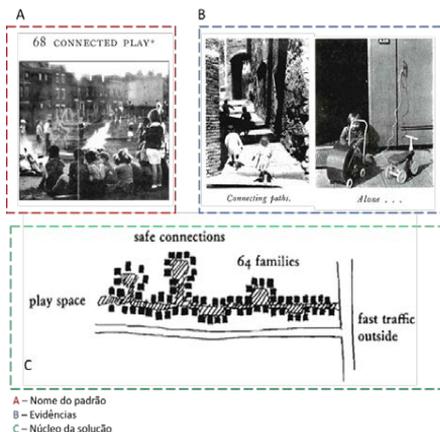


Figura 1: Estrutura da *pattern* '#68 Connected play' (Espaços para brincar conectados). Fonte: Adaptado de Alexander *et al.* (1977).

Cientes da dimensão e complexidade dos 253 padrões apresentados por Alexander *et al.* (1977), para iniciar as investigações sobre a tradução de saberes teóricos, a presente pesquisa fez um recorte temático na obra. Foram

estudados padrões que abordam a forma urbana edificada relacionados a aspectos da vida urbana.

Apesar de defender um ideal de filosofia de vida almejado em meados do século passado, a de se concordar com Barros (2008) que o livro oferece uma visão muito interessante de princípios que favorecem a conexão do espaço construído com os seres humanos. Princípios estes que na contemporaneidade ainda estão sendo incorporados, ou desejados, em projetos atuais.

Assim como em Barros (2008) uma série de outros estudos apresentam resultados interessantes alcançados a partir da união entre elementos (estruturas metodológicas ou diretrizes projetuais) dos escritos de Christopher Alexander e o processo projetual (também referido neste *paper* como processo de *design*). Dentre eles, Vaz (2011), Kühn e Herzog (1991), Leibovich *et al.* (2020) e Ozel (2007) exemplificam diferentes maneiras de inserir as obras do autor com o projeto e o ensino da arquitetura e do urbanismo.

Além da obra de Alexander, questões de proposição de formas edilícias que consideram a vivência urbana têm sido trabalhadas teoricamente por outros consagrados autores, tais como Jacobs [1993 (1961)], Teller (2001), Gehl (2013), Dobbins (2011), mas por estar entre tantos outros problemas urbanos, muitas vezes não são discutidas ou levadas em consideração na construção da cidade.

A fim de apresentar os resultados da mencionada pesquisa e o processo de criação do código paramétrico, este artigo resumirá a metodologia em três pontos principais: (i) critérios de escolha dos *patterns* que tratam da forma edificada relacionada à vida na cidade; (iii) conexão e criação de um sistema de relação entre *patterns*; e (iv) tradução para linguagem de programação visual e desenvolvimento do código. Já na seção de resultados será apresentado o código em funcionamento, expondo a tradução e testes de um *pattern* que constitui parte do código.

## METODOLOGIA

A estrutura metodológica a ser apresentada, foi acompanhada por uma profunda leitura e estudo da trilogia *A Pattern Language* (1975, 1977 e 1979), e da leitura de materiais escritos com base nessas obras e na temática recortada, forma e vida urbana, tais como Lang (2006), Jackson (1972), Moughtin (2007), Kuchpil (2008), entre outros.

### (I) CRITÉRIOS E SELEÇÃO DE PATTERNS

Segundo Dino (2012), a modelagem paramétrica requer intencionalidade e impõe ao *designer* muita explicitude antes da exploração dos parâmetros. Esse passo anterior à programação pode ser relacionado ao que Jones (1970) refere como identificação de variáveis. As soluções resultantes da modelagem paramétrica só existem dentro de uma definição do problema, que é traduzido em uma sequência de componentes e regras compositivas. As possibilidades são infinitas dentro de um sistema finito.

Concomitante a isso, Wing (2008) aponta que a abstração é a essência do pensamento computacional, necessário

para a tradução proposta nesse artigo. Essa abstração envolve trabalhar com *layers* de informação e relação, ou seja, para que um código seja desenvolvido, é fundamental existir um pensamento, uma imaginação do que se deseja fazer e como será feito. Caso contrário, o processo de construção do código pode se tornar muito confuso, demorado e sem um objetivo claro.

Alexander *et al.* (1977) deixam claro que a aplicação dos *patterns* variam de acordo com o contexto e de quem os utilizam, semelhantemente, a maneira de traduzir esses princípios para uma linguagem computacional também é variável. O desenvolvimento de qualquer código depende dos conhecimentos de quem está programando. Uma mesma função pode ser executada de diversas maneiras

Assim, o primeiro passo para a tradução dos princípios relacionados à forma e vida urbana foi a escolha desses padrões. A primeira seleção foi feita de acordo com o tema abordado e quais seriam mais adequados no *design* urbano. Além da forma urbana em relação à temática da vida na cidade, vários outros *patterns* abordavam a morfologia urbana, mas partiam de princípios voltados ao sistema viário, detalhamento de elementos construtivos, entre outros. Uma vez que a complexidade dos temas principais da pesquisa já é considerável, todos aqueles que não se encaixavam diretamente, não foram selecionados.

Importante mencionar que desde o primeiro momento de leitura do livro, percebeu-se que enquanto alguns *patterns* têm um potencial claro de se tornarem algoritmos (como parâmetros em um código), outros são essencialmente abstratos e aplicáveis como princípios de gestão. Assim, após a delimitação temática, os *patterns* que eram relacionados às ações administrativas e subjetivas foram retirados.

Após os critérios mencionados, os 253 *patterns* foram reduzidos a 26. Apesar da significativa diminuição da quantidade ainda se fez necessário analisar a viabilidade de utilizá-los em conjunto. Essa análise foi feita a partir da construção de um quadro resumo (Figura 2).

Nesse quadro, os *patterns* foram classificados de acordo com o tema principal abordado por cada um, morfologia, forma urbana edificada ou vida urbana, a aplicabilidade, considerando a clareza e relação com o design urbano e finalmente o nível de clareza pelo qual havia sido descrito, apontando assim a possibilidade maior ou menor de tradução algorítmica.

Através do quadro análise foi possível ter uma visão ampla de todos os *patterns* e facilitar a posterior conexão entre eles. Além dos assuntos em comum, foi importante fazer mais uma leitura aprofundada de cada um a fim de identificar essas ligações.

Assim como a utilização dos *patterns* e o desenvolvimento de qualquer código, os critérios e os passos seguidos para a seleção dos elementos foi feita a partir das leituras já mencionadas e não objetivam definir um caminho único ou mais adequado para esse processo de tradução computacional. Sua descrição visa o compartilhamento metodológico para uma utilização mais profunda das tecnologias computacionais no processo de projeto digital.

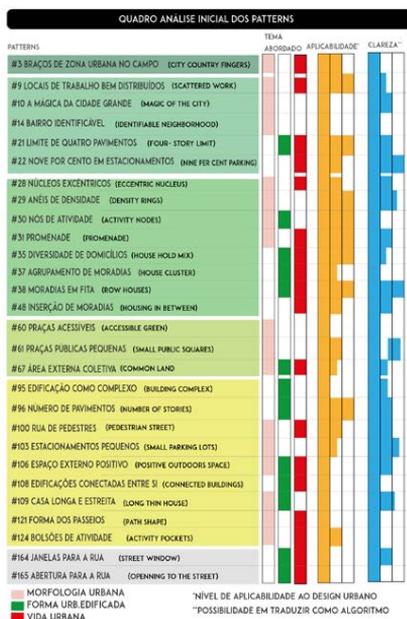


Figura 2: Quadro análise dos *patterns*. Fonte: das autoras, 2020.

## (II) CONEXÕES ENTRE PATTERNS E CRIAÇÃO DO SISTEMA

Após a seleção e a compilação em um quadro resumo, foi feita outra leitura aprofundada de Alexander *et al.* (1977) com o objetivo de identificar conexões indicadas pelos próprios autores. Apesar de dividir o livro em escalas distintas, o texto em vários momentos relaciona *patterns* de diferentes níveis, escalas e assuntos:

“Esse padrão ajuda a evolução gradual do #8 Mosaico de subculturas (*Mosaic of subcultures*), colocando famílias e trabalhando juntos, intensificando o surgimento de subculturas altamente diferenciadas, cada uma com seu caráter individual” (Alexander *et al.* 1977, p. 52 – tradução livre das autoras).

A fim de melhor visualizar as conexões e identificar os *patterns* que possuem a maior quantidade de relações e, consequentemente, maior influência no código, foi feito um esquema de teia de relações como base para o sistema em construção (Figura 3).

Os *patterns* com maior quantidade de ligações e aqueles com maior afinidade ao tema foram mais uma vez analisados e os primeiros, identificando que poderiam ter maior impacto no desenvolvimento de um sistema que relaciona a forma urbana edificada, foram então chamados âncoras.

CONEXÕES ENTRE OS PATTERNS  
A PARTIR DA ANÁLISE DOS PRINCÍPIOS

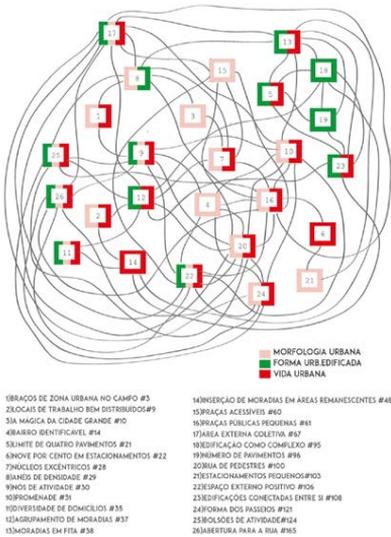


Figura 3: Esquema de teia de relação entre *patterns*. Fonte: das autoras, 2020

Percebeu-se que os *patterns* âncora influenciaram a forma urbana edificada em diferentes escalas, logo, o código deveria seguir o que a obra em questão defende: a multiescalaridade do *design* urbano e suas conexões. Estabeleceu-se então que a escala 1, de menor dimensão, altera a edificação, mudando a forma urbana edificada a nível do lote. A escala 2 trata da edificação a nível de disposição das edificações em um recorte urbano de quadra. Por último, a escala 3 trabalha com elementos a nível de recortes urbanos médios e grandes, manipulando densidades construtivas e configuração desses recortes (Figura 4).

A partir da definição das escalas, dos *patterns* âncora e dos objetivos de cada uma, os 26 *patterns* anteriormente selecionados foram, mais uma vez, analisados e aqueles que apresentavam maior conexão e coerência com o sistema e com as escalas propostas foram escolhidos (Figura 4).

A partir da clareza do sistema na linguagem natural, a pesquisa partiu para a tradução na linguagem de programação visual. Para tal tradução utilizou-se o *software Rhinoceros* e *plug-in Grasshopper*, *softwares* que possibilitam uma programação do tipo visual, não necessitando conhecimentos das linguagens de computação e do desenvolvimento de *scripts*. Assim, cada vez mais a apropriação da programação tem sido feita por arquitetos e urbanistas.

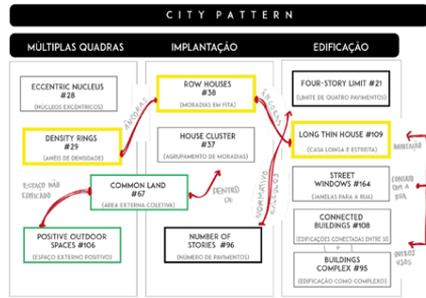


Figura 4: *Patterns* âncora e os demais organizados nas escalas a serem trabalhadas pelo código. Fonte: das autoras, 2020.

### (III) TRADUÇÃO E DESENVOLVIMENTO DO CÓDIGO

A partir da temática trabalhada (relação entre forma e vida urbana) e a criação do sistema apresentado anteriormente, alguns objetivos para o código paramétrico foram elaborados: (a) utilizar três escalas dimensionais de manipulação de parâmetros da forma urbana edificada; (b) traduzir para a linguagem computacional os fundamentos dos seguintes *patterns*: #28 Núcleo excêntricos, #29 Anéis de densidade, #67 Área externa coletiva, #106 Espaço externo positivo, #38 Moradas em fita, #37 Agrupamento de moradias, #96 Número de pavimentos, #21 Limite de quatro pavimentos, #109 Casa longa e estreita, #164 Janelas para rua, #108 Edificações conectadas entre si e #95 Edificação como complexo; (c) possibilitar não apenas a parametrização como a otimização dos cálculos de características relativas ao espaço urbano edificado e não edificado, tais como: áreas, alturas médias, distâncias.

Para alcançar os objetivos pretendidos, o processo de tradução partiu do estudo aprofundado dos *patterns* âncora, identificando as variáveis e elementos necessários para a parametrização de seus princípios. A identificação dessas variáveis levou ao estabelecimento de diferentes tipos de dados. Enquanto alguns elementos poderiam ser traduzidos em números e poderiam ser variáveis, outros eram predominantemente geométricos ou que possibilitavam escolhas. Por exemplo: A forma base para a edificação poderia ser gerada a partir de um polígono de lado e dimensão variável ou partir de um quadrilátero com cada dimensão estabelecida pelo *designer*. Essa possibilidade de escolha, polígono ou quadrilátero gerava automaticamente um tipo de dado de seleção, uma bandeja de escolha.

Esse tipo de análise e caracterização das variáveis e elementos de cada *pattern*, levou à uma classificação de variáveis a serem manipuladas, facilitando a organização dos tipos de dados e da construção do código.

Os dados parametrizáveis são aqueles que variam com uma simples mudança de valor, normalmente são alterados por um *number slider* (componente existente no *grasshopper* semelhante a uma barra deslizante, que possibilita a mudanças de valores apenas com um clique do *mouse*). O resultantes da parametrização são dados gerados a partir das mudanças do parametrizável e sua função principal é a otimização de cálculos e comparações, por exemplo: As dimensões de um

quadrilátero são dados parametrizáveis e sua área é um resultado da parametrização. Por fim, a bandeja de escolha, como já mencionado, permite a escolha, normalmente inicial, de determinadas características da forma, também usada para desativar ou ativar partes do código.

Para que um código seja considerado útil e funcional, espera-se que se tenha uma maior quantidade e/ou complexidade de *outputs* do que *inputs*. A partir da análise, criação de relações, o código desenvolvido, recebe um total de 4 dados e fornece 33 novos elementos para a manipulação da forma urbana (Figura 5).



Figura 5: Quantidade e tipos de dados do código desenvolvido. Fonte: das autoras, 2020

## RESULTADOS

Apesar de ter sido construído seguindo uma ordem específica de escalas e parâmetros, é importante enfatizar que não há uma maneira certa ou errada de se programar qualquer que seja o código. Nesse sentido, a de se concordar com a professora Dra. Ana Bell (2016) no curso "Introduction to Computer Science and Programming in Python", do Massachusetts Institute of Technology – MIT, ao apontar que avalia-se um bom programador pela quantidade de funcionalidade do código. Assim, não será aprofundada a estrutura nem os componentes escolhidos na construção do código, mas será colocado em destaque as capacidades da tradução feita.

Mesmo que o foco do presente artigo não seja esmiuçar a estrutura do código, a forma final foi organizada para facilitar o entendimento e a manipulação dos parâmetros (Figura 6). A separação criada entre os componentes a serem manipulados e os componentes que geram esses

resultados, zoneando a estrutura, teve a finalidade de deixar mais claro para o usuário os componentes parametrizáveis, assim como os dados gerados pelo funcionamento do sistema. Na área de manipulação, os componentes foram agrupados de acordo com seu tipo (Figura 7) e com as escalas de atuação já mencionadas nesse artigo.

A tradução dos *patterns* encontra-se dissolvida nos parâmetros e nos diferentes tipos de *outputs*. Alguns princípios podem ser verificados na área de dados resultantes da parametrização, como é o caso do *pattern* "#109 Casa longa e estreita" que propõe o aumento da distância entre os extremos de uma edificação, aumentando a sensação espacial, pode ser verificado em um componente que faz o cálculo automático dessa distância na medida que as dimensões da forma da edificação são alteradas, possibilitando que o usuário identifique qual(is) formas aumentam esse valor. Já outros *patterns* estão inseridos nos dados parametrizáveis e nas bandejas de escolha.

Devido a impossibilidade de apresentar cada tradução, sua relação com os demais elementos e o seu rebatimento no código desenvolvido, será apresentado um *pattern* âncora, exemplificando assim uma tradução e apontando algumas contribuições.

### TRADUÇÃO DO PATTERN #29 ANÉIS DE DENSIDADE

Escolhido como o âncora da maior escala de atuação do código, o *pattern* #29 Anéis de densidade, descrito em Alexander *et al.* (1977, p.156) trata de um princípio de distribuição de densidade em recortes urbanos. Bastante relacionado com o anterior #28 Núcleos excêntricos, ele possui uma abordagem conciliatória ao propor diferentes distribuições de habitação em uma vizinhança a fim de atender a diferentes modos e desejos de se viver:

"As pessoas querem estar perto de lojas e serviços, ou da excitação e da conveniência. E querem ficar longe dos serviços, por um tempo e para ir o verde. O equilíbrio exato desses dois desejos varia de pessoa para pessoa, mas no

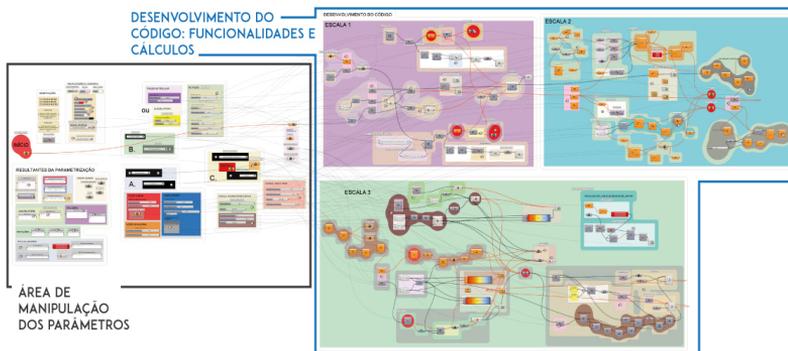


Figura 6: Código desenvolvido, zoneado por funcionalidade. Fonte: das autoras, 2020.

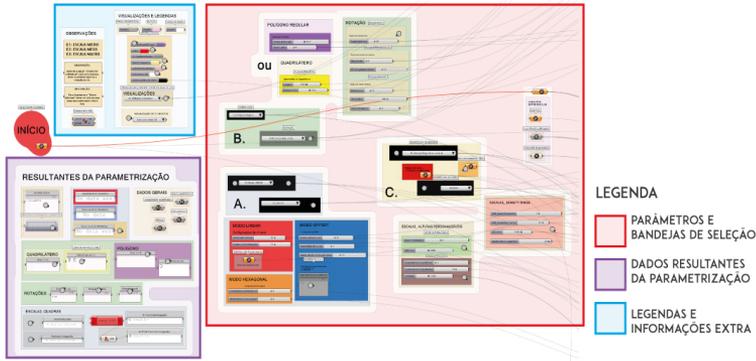
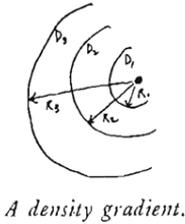


Figura 7: Área de manipulação do código zoneado segundo a funcionalidade. Fonte: das autoras, 2020.

fim, é o equilíbrio desses dois desejos, que determina o gradiente de densidades habitacionais em uma vizinhança" (Alexander *et al.* 1977, p.156).

A partir dessa afirmação inicial, o *pattern* #29 fornece modos de cálculo do chamado *gradiente de densidade*. Eles podem ser especificados através de anéis de diferentes distâncias do centro de atividade principal, recebendo atribuições de densidades diferentes a cada anel (Figura 9). Assim como todos os *patterns*, Alexander *et al.* (1977) destacam que os gradientes devem variar de comunidade para comunidade, de acordo com a posição e com o contexto cultural.



A density gradient.

Figura 8: Ilustração do *pattern* #29 Anéis de densidade. Fonte: Alexander *et al.* (1977, p.157)

A partir das recomendações e descrição dos princípios presentes nos anéis de densidade, os autores resumem o modo de aplicar esse *pattern* da seguinte maneira:

"Uma vez que o núcleo de uma comunidade esteja claramente colocado - defina anéis de diminuição da densidade local da habitação em torno desse núcleo. [...] obtenha essas densidades a partir das intuições das próprias pessoas que vão morar na comunidade" (Alexander *et al.* 1977, p.162).

Diante dessa declaração e considerando os demais parâmetros necessários para a tradução dos outros *patterns* presentes no código, escolheu-se a manipulação dos valores de verticalização, aumentando e diminuindo a quantidade de habitações, para a criação os anéis de diferentes densidades.

Considerando que um dos principais elementos do *pattern* #29 é a indicação da direção do decrescimento da densidade, nomeado por eles como o centro de atividade, criou-se a seleção desse elemento de atração de densidade no código. Ao modificar a localização desse elemento, automaticamente o gradiente de densidade se adequa à nova configuração.

Os parâmetros principais para alteração desse gradiente de densidade e o controle da verticalização foram: quantidade mínima e máxima de pavimentos, e consequentemente quantidade de habitações, a altura desses pavimentos (pé direito) e a representação dessas características na geometria gerada (Figura 10).

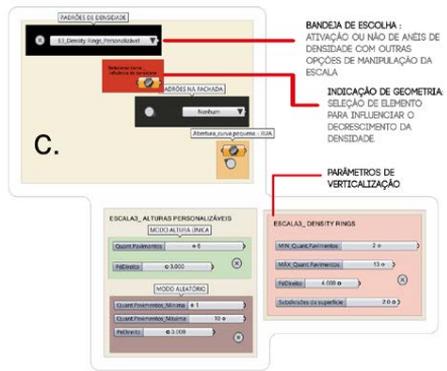


Figura 9: Parâmetros da escala 3. Fonte: das autoras, 2020.

Uma vez que o *pattern* anéis de densidade é apenas um dos vários traduzidos para a construção desse código paramétrico, escolheu-se criar outras opções de visualização dessa escala de recorte urbano, no qual o usuário tem a possibilidade de ativar ou não esse princípio ao selecioná-lo na bandeja de escolha (Figura 10).

Diante da dimensão da escala trabalhada por esse *pattern*, o ponto inicial para visualização do anéis de densidade é a seleção do recorte urbano a ser simulado. A fim de atender toda e qualquer morfologia de quadra ou de tecido urbano, esse recorte pode ter qualquer forma, regular ou

não. Ao ser selecionado, o código gera automaticamente uma implantação específica e uma forma de edificação decorrentes da relação entre os parâmetros das outras escalas já mencionadas.

Após a seleção do recorte urbano, para a visualização da influência do gradiente de densidade, faz-se necessária a indicação do elemento de atração, apontado em 'A Pattern Language' (1977) como um centro de maior interação (Figura 11).

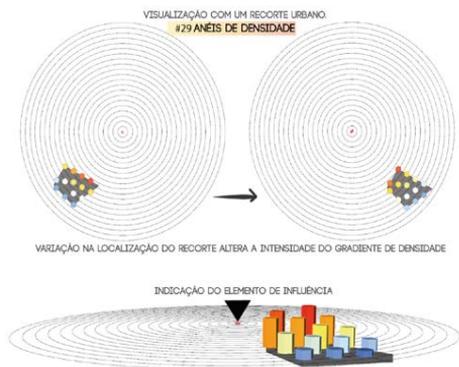


Figura 10: Tradução do *pattern* #29 anéis de densidade, recorte urbano único. Fonte: das autoras, 2020.

Além da seleção de um recorte urbano (Figura 11), a tradução do princípio possibilita também a seleção de múltiplas áreas para simulações simultâneas. Dessa maneira, passa a existir a possibilidade de criação de cenários mais amplos a nível de cidade, como também ilustra-se mais claramente a diferença que a distância até o ponto de influência (Figura 12).

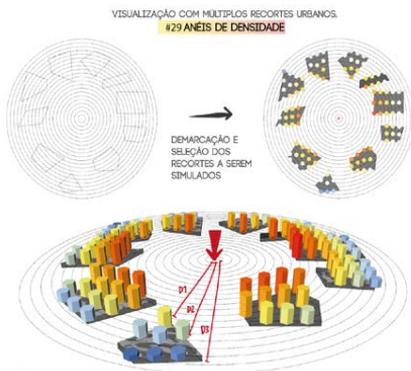


Figura 11: Tradução do *pattern* #29 Anéis de densidade, múltiplos recortes urbanos. Fonte: das autoras, 2020.

Somado à visualização do *pattern* e às experimentações possíveis pela variação dos parâmetros, o código fornece dados resultantes da parametrização nessa escala tais como: quantidade de elementos, altura média das

edificações e um gráfico de variação desses gabaritos (Figura 13).



Figura 12: Dado resultante da parametrização, variação do gabarito. Fonte: das autoras, 2020.

A partir da tradução do *pattern* #29 Anéis de densidade na linguagem computacional paramétrica (Figura 14) ilustrou-se parte do funcionamento do sistema desenvolvido e a potencialidade de inserir tecnologias computacionais na experimentação de cenários e criação de propostas embasadas em princípios e teorias reconhecidas.

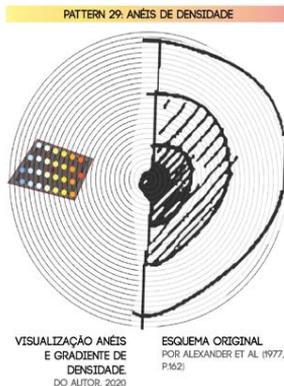


Figura 13: Paralelo entre a tradução computacional e o esquema original do *pattern* #29 Anéis de densidade (Alexander *et al.* 1977). Fonte: das autoras, 2020.

## CONCLUSÃO

Partindo de uma pesquisa mais ampla que investiga a inserção de tecnologias computacionais na tomada de decisão em projetos urbanos, esse *paper* apresentou parte dos resultados do desenvolvimento de um código paramétrico para a tradução e aplicação computacional de princípios presentes de maneira teórica nas obras de Alexander *et al.* (1975, 1977 e 1979).

Defende-se que a utilização de tecnologias computacionais no processo de projeto, conhecido também como projeto digital, contribui para o maior entendimento e clareza nos processos de tomada de decisão. Além de otimizar cálculos e simulações, gerando uma maior base experimental, ao expor as intencionalidades de *design*, o modo de fazer projeto torna-se mais aberto para contribuições multidisciplinares.

Mais do que utilizar técnicas e modos de trabalho computacionais, esse artigo aponta que ao traduzir princípios teóricos reconhecidos, a tecnologia computacional constrói uma ponte entre o passado (existente nas teorias) e a instrumentalidade e modo de pensamento do futuro.

Ao desenvolver um código para a visualização de princípios e diretrizes, esse artigo também defende a maior inserção do pensamento computacional nas tomadas de decisão. Compreende-se que o nível de abstração mental necessária e presente nas ciências da computação em muito pode contribuir no planejamento e gestão de variáveis cada vez mais complexas do espaço urbano.

A partir da tradução computacional do *pattern* #29 Anéis de densidade ilustrou-se uma parte do sistema desenvolvido para visualização e otimização de variáveis que relacionam forma e vida urbana presentes na obra *A Pattern Language* (1977). Nota-se no entanto, que esse é um estudo em uma obra específica, mas com a intenção de estimular e apresentar possibilidades de conectar o ensino e a produção de projeto no modo tradicional com aquele defendido pelo projeto digital.

Para investigações futuras, esse artigo indica a ampliação temática da tradução dos princípios das obras de Christopher Alexander, assim como a utilização de meios computacionais para o estudo de outras teorias e diretrizes consagradas, aumentando a conexão mental do homem com a máquina e as possibilidades de trabalho em conjunto.

## AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) do Brasil pelo financiamento da pesquisa aqui apresentada.

## REFERÊNCIAS

- Alexander, C. *et al.* (1977). *A pattern language: towns, buildings, construction*. Oxford university press.
- Alexander, C., Silverstein, M., Angel, S., Ishikawa, S., & Abrams, D. (1975). *The Oregon experiment* (Vol. 3). *Center for Environmental Struc.*
- Alexander, C. (1979). *The Timeless Way Of Building* (Vol. 1). New York: Oxford University Press.
- Barros, R. R. M. P. (2008). *Habitação coletiva: a inclusão de conceitos humanizadores no processo de projeto*. São Paulo: Annablume, 2011.
- Batty, M. & Longley, P. (1997) *The fractal city, AD Architectural Design*, 129, pp. 46- 49.
- Beirão, J. (2012). *CityMaker: designing grammars for urban design*. TU Delft
- Bell, A. Grimson, E & Guttag, J. (2016) 6.0001 *Introduction to Computer Science and Programming in Python*. *Massachusetts Institute of Technology: MIT OpenCourseWare*, <https://ocw.mit.edu>. License: Creative Commons BY-NC-SA.
- Celani, M. G. C. (2002). *Beyond analysis and representation in CAD: a new computational approach to design education* (Doctoral dissertation, Massachusetts Institute of Technology, Department of Architecture).
- Chaillou, S.(2019). *AI + Architecture towards a new approach*. Tese de doutorado. *Harvard University Graduate School of Design*, Estados Unidos.
- Dawes, M. J., & Ostwald, M. J. (2017). *Christopher Alexander's A Pattern Language: analysing, mapping and classifying the critical response*. City, Territory and Architecture, 4(1), 17.
- Dino, I. G. (2012). *Creative design exploration by parametric generative systems in architecture*. METU Journal of Faculty of Architecture, 29(1), 207-224.
- Dobbins, M. (2011). *Urban design and people*. John Wiley & Sons.
- Gehl, J. (2013) *Cities for people*. Island press.
- Kuchpil, E. (2008). *Edifício vertical e a cidade: imagens da modernidade sob o olhar do espaço público*. Tese de doutorado, Universidade de São Paulo.
- Kuo, Jeannette; Zausinger, Dominik (2010) *Scale and Complexity: Multi-layered, multi-scalar agent networks in time-based urban design*. FUTURE CITIES [28th eCAADe Conference Proceedings / ISBN 978-0-9541183-9-6] ETH Zurich (Switzerland) 15-18 September 2010, pp.651-657
- Lang, J. (2006). *Urban design*. Routledge.
- Jackson, J N. (1972). *The Urban Future: A Choice between Alternatives*. University of Birmingham.
- Jacobs, J. (1993) *The death and life of great American cities*. New York: The modern library, 3ed (1961)
- Jones, C. (1970). *Design method: seeds of human future*. New York: John Wiley & Sons.
- Jones, B. (1966) *Design from Knowledge, Not Belief*. In: Whiffen, M (Ed.). *The Architect and the City: Papers from the AIA-ACSA Teacher Seminar, Cranbrook Academy of Art, June 11-22, 1962*. (pp.139-152). Cambridge, Mass., MIT Press.
- Kühn, C., & Herzog, M. (1991). *A language game approach to architectural typology*. Proceedings of the ECAADE, 91.
- Leibovich, L., Nitzan-Shifan, A., & Sprecher, A. (2020) *Cybernetic Methodologies for Flexible and Generative Architectural Systems*. eCAADe 2020.
- Moughtin, C. (2007) *Urban design: street and square*. Routledge.
- Ozel, F. (2007). *Pattern language and embedded knowledge in Building Information Modeling*. eCAADe 2007.
- Teller, J. (2001). *La régulation morphologique dans le cadre du projet urbain. Spécification d'instruments informatiques destinés à supporter les modes de régulation performantiels* (Doctoral dissertation, Université de Liège, Liège, Belgique).
- Terzidis, K. (2006). *Algorithmic Architecture*. Oxford: Architectural Press
- Vaz, C. E. V. (2011). *Um sistema de ensino de projeto baseado no conhecimento= sistemas generativos e ontologias aplicados no ensino de arquitetura paisagística*. Tese de doutorado, Universidade Estadual de Campinas (UNICAMP).
- Wing, J. M. (2008). *Computational thinking and thinking about computing*. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-37