sigradi 2021 designing possibilities ubiquitous conference

Automated Generation of Layouts and Budgets: Development and Application from Written Programming and Parametric Modeling

Pedro Oscar Pizzetti Marino[1], Mirian Aparecida Maia[2], Bruna Pacheco Campos[1], Carlos Eduardo Verzola Vaz[1]

[1] Universidade Federal de Santa Catariana - UFSC, Brasil
pedro.pm@hotmail.com
arquiteturabrunacampos@gmail.com
cevv00@gmail.com
[2] Universidade Federal de Ouro Preto – UFOP, Brasil
mirian.maia@aluno.ufop.edu.br

Abstract. The article aims to demonstrate the applicability of the use of programming in the formation of floor plans in the initial design of the architectural layouts of houses of social interest and to evaluate the possible cost of building these proposals. The development of the algorithm follows known programming rules and structures written in a visual modeling and programming software, in addition to using the methods of stochastic search, shape grammar, CUG (geometric unit cost) and CUB (basic unit cost). As a result, it was possible to develop space layouts that resemble the rooms of a single-family home and calculate their possible construction cost. The use of the complete process also brings the discussion of the use of technologies and automated processes within current design practices, demonstrating how the use of these technologies can enhance the imagination and present compositions and quantitative results not envisioned by the designers.

Keywords: Layouts; written programming; parametric modeling; budget.

1 Introduction

Stangl (2013) describes those technologies have ubiquitously infiltrated almost every aspect of our lives, and within the field of architecture and urbanism they are also present as digital tools that help the practice of designing, integrating the different stages of the architectural process and changing the way of designing significantly.

With the global expansion of Industry 4.0, various changes affected the development and construction processes of architectural projects. Computational tools, resulting from this new period, cause new fundamental questions about the project development phases to be raised (KOVACS; SZOBOSZLAI; CSUSZ, 2019).

Emerging technologies are being reorganizing and changing the functionality of different economic and productive sectors around the world. The growing use and development of computational and autonomous tools for production and development is already noticeable in the creation of jobs that involve routine aspects, in which even the trades that require cognitive actions are seeing fewer job offers (CREW, 2019; DVORKIN ,2016).

With the advancement of computational tools aimed specifically at architects and engineers, design problems can now be solved more quickly and accurately. The union of different and new disciplines leads to the formation and resolution of spatial design problems, making computer systems adapt to solve different problems collectively, solving different and specific goals. With the advancement and introduction of disciplines and the need for an increasing performance of constructions, the increase in the demand for multi-criteria resolutions only tends to grow (ALFARIS and MERELLO, 2008; CALIXTO, 2016).

Thus, this article aims to demonstrate the applicability of the use of written and visual programming in the formation of floor plans in the early stages of the architectural design of houses of social interest and choose the most appropriate solution in terms of form and cost. The use of this programming aims to demonstrate that it is possible to expand the formal composition options, being able to create patterns that are not in the designer's, making it possible to envision the solution of design problems from a set of formal resolutions. The article also investigates the possibility that the developed floor plan solutions can have their construction cost evaluated through the two methods. The first is the CUG - Geometric Unit Cost -, which according to Lima et al., 2016, is a method that calculates the cost of the square footage of a building from formal and spatial characteristics of a layout, and, the second, CUB - Basic Unit Cost - which according to with Mascaró (1985) and Lima et al., (2016) is a type of simplified budget commonly used in Brazil.

As a result, it is possible to develop different compositions similar to floor plans in an abstract space delimited by parametrically adjustable setbacks. The number of rooms, their dimensions and organization are coordinated through rules described in the schedule, which provide a series of random floor plans, but which follow a certain pattern. The schedule also divides the composition into quantities of partitions, areas and environments, formulating an estimated construction cost.

2 Theoretical Review

In this section, the main ideas that guided the development of the program will be presented. Initially, the computational tools used will be described, then the methods that help in the formation of the compositions, and finally the two methods used to calculate the cost of the developed compositions - CUB and CUG.

2.1 Computational tools

For the construction of the process, four computational tools were used, a base three-dimensional modeling software with a visual programming component and two other plugins. The modeling software was Rhinoceros 3D 6® (MCNEE, 2014), its Grasshopper component (RUTTEN AND MCNEEL, 2004) and the plugins Wasp (ROSSI, 2017) and Colibri from the TT toolbox (TOMASETTI, 2017).

The computational modeling tool Rhinoceros® was used to visualize the programming results, in addition to being the base software for visual programming (MCNEE, 2014). Grasshopper was used both for the development of visual programming and for writing (RUTTEN AND MCNEEL, 2004). The plugin that enabled the use of stochastic search and the application of shape grammar was Wasp, developed in Python, offering combinatorial tools to design elements from a topological graph (ROSSI, 2017).

Finally, the Colibri component, from the TT tool box plugin, was used to collect the output quantitative data, directing the results to Excel tables. Colibri allows Grasshopper's results to be data compatible with other computer programs (TOMASETTI, 2017).

2.2 Shape Grammar, Stochastic Search and Dynamic Programming

Using methods from the field of computing and optimization, the production processes of various results were merged, such as shape grammar, stochastic search and dynamic backpack. Shape grammar is a system molded from rules that define the type of geometry generated, respecting a pre-established logic and the various types of combinations (CELANI et al. 2006).

Stochastic searches are models elaborated from concepts of probability and random events that are plausible to occur at a certain time and according to certain parameters (KNILL 2009).

The last process used was the dynamic backpack from computational complexity theory, an algorithm that aims to solve combinatorial optimization problems in a finite set with various particularities (YANASSE; SOMA; MACULAN, 2000).

2.3 Basic Unit Cost and Geometric Unit Cost

The production of an architectural project requires various steps that are evaluated and validated by architects and engineers throughout the process. (MORAES; PICCHI; GRANJA, 2016). However, in the initial stages of the project, little information is passed on in the models, which present a low level of detail, resulting in inaccurate budgets when associated with models that use the built unit. (MASCARÓ; 1985). Every day when we refer to the cost of construction for budgeting in Brazil, one commonly used is the basic unit cost (CUB), a value that is multiplied by the calculation of the equivalent construction area (LIMA et al., 2016).

According to Gonçalves and Ceotto (2014), CUB has only one variable for calculating the unit price, and it can be a tool without variations and unreliable. Lima et al. (2013) state that different configurations and characteristics present in the project can change the construction price estimates. An alternative, to measure the construction cost seeking greater precision of possible values during the design or study phase, is the geometric unit cost (CUG). To arrive at the independent variables of the formula, four mathematical processes are necessary, as follows: X1 is equal to the division between the total external wall area by the total built area; X2 is the total interior wall area divided by the built-up area; X3 is the wet area divided by the constructed area; and X7 may vary depending on the construction standard, being 0.1624050 for high standards, 0 for normal standards, and -0.437 75848 for low standards. These values are arranged in equation 1, which, in turn, makes it possible to identify an estimated cost of a square meter (LIMA 2013; LIMA et al., 2013).

The default value of the formula is 481,60837 and was used by Lima et al. (2013) and Lima (2013) based on the CUB of Rio de Janeiro in April 2012. In works such as LIMA et al. (2016) the default value of the CUG is used in a commercial Excel table, but the method is not described. For this adjustment, the INCC (FGV IBRE, 2020) National Civil Construction Index was used, with the purpose of evaluating the monthly readjustments of the cost of construction. The result used in the programming was a variation index of 1.679462 which multiplied by 481,60837 resulted in 808,84296. From this new value, the equation of Lima (2013) was adapted to equation 1,

$$Y = 808.84296 * e^{(0,97266795 * X1)} * e^{(0,13635413 * X2)} * e^{(-0,003251045 * 1/X3)} * e^{(0,99294145 * X7)} \qquad (1)$$

3 Method and construction of the programmimg

For the construction of the parametrically changeable process, knowledge and computational tools described above were used, the complete process was developed in four steps, using a cascade structure, and to initiate each step, each previous one should be completed. Fig. 1 presents the logical structure used to build the parametric process.

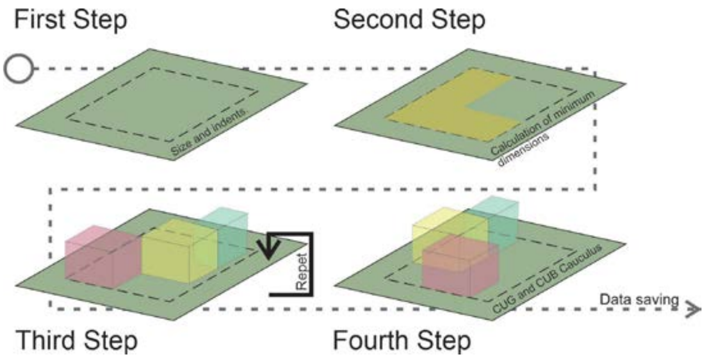


Figure 1. The four steps that structure the schedule. Source: The authors, 2021

The first part of the program, developed in written language, aims to parameterize the urban constraints in order to graphically display and define the dimensions and the useful area of the land. For this purpose, 6 input variables were established: (1) width and (2) depth of the terrain, setbacks (3) front, (4) lateral and (5) back, all of the real type; and a (6) Boolean button for choosing between entering standard or manual offsets.

The final stage of the parameterization of urban constraints is dedicated to the representation of the terrain in the virtual environment of Rhinoceros 3D. Initially. A plane an axis is generated on which the project elements will be launched. The terrain boundaries and the setback lines, in turn, are represented in Rhinoceros 3D, with lines that use the coordinates calculated previously. Finally, two output parameters are generated to inform the successive steps: dimensions, width and depth, useful for the terrain.

The second step of programming, developed in script, collects information from the first stage and combines it with the number of rooms that will be used in the composition of spaces. From this a number of values are generated providing an ideal area for the combined spaces to fit within the terrain. For the construction of this step, fundamental aspects of Brazilian legislation in relation to housing and land were used, based on Law 10257, of July 2001 - Statute of Cities.

In the definition of this step greater attention was paid to access to housing, such as the Minha Casa Minha Vida program (PROGRAMA CASA VERDE E AMARELA, 2021) and areas dedicated to housing for low-income people and irregular occupations, called Zones of Social Interest - ZEIS. Such zoning provides for lots of between 60m² and 300m², with a national average of 180 m² (SECRETARIA NACIONAL DE HABITAÇÃO, 2009), but according to the federal law for land subdivision, the minimum standard lot must be 125 m² (FEDERAL SENATE ,2001). A study carried out by Amore, Shimbo and Rufino in six Brazilian states, found the average size of Minha Casa Minha Vida housing units to be 42 m², consisting of 2 bedrooms, a living room, a kitchen and a bathroom. (AMORE; SHIMBO; RUFINO, 2015).

The third step of programming starts from the use or analysis of the previous results, supported by the defined minimum limits and sizes. Here it is possible to run a formal grammar along with a stochastic search capable of generating space arrangements that resemble the rooms of a single-family home. For the development of this step, the programming was built using Grasshopper's graphical components, but with a plugin that helps in the formation of compositions through the stochastic search: Wasp.

The rules that make up the shape's grammar work in a sequence defined from a set of tests executed at the end of the process, in which a group of one type of room is arranged for the next to be installed. The logic follows a sequence of five sets of rules:

1. The main room is defined from a parametrically adjustable point at the limit of the frontal distance, and then the rooms are inserted, always in juxtaposition with one another.
2. The circulation is located next to the main room, which has its length added to that of the bedrooms.
3. The bedrooms are placed next to the circulation.
4. The kitchen is located next to the first room, and if there are two kitchens, the second is considered as a service area, located next to the first.
5. finally, the bathrooms are defined, the first always next to one of the rooms, and the second next to the corridor or bedroom.

Stochastic search is used in the implementation of each of the grammar rules so that inserted spaces never invade already-located spaces, preventing different rooms from occupying the same position. And so that spaces are always positioned logically (following structural alignments and seeking to avoid narrow spaces or external bottlenecks) the rule that governs spaces makes them always connect 0.60 m away from their edges. Fig. 2 shows the functioning of the rules used and Fig. 3 shows a sample of the organization of rules to avoid unoccupied spaces.

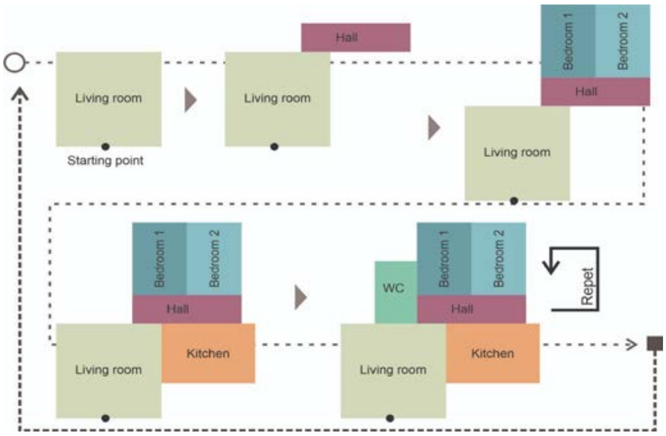


Figure 2. Example of the logical sequence of the arrangement of rooms in the formation of the composition organized by stochastic search. Source: The authors, 2021

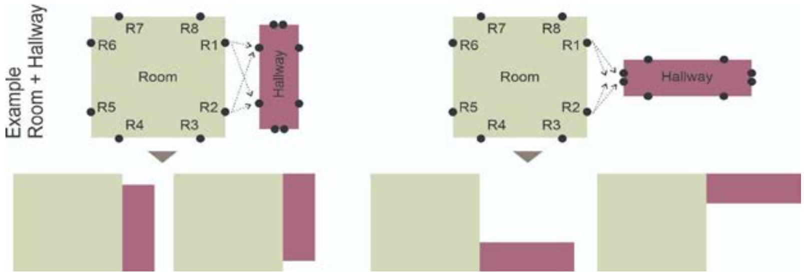


Figure 3. Example of the functioning of the rules for setting up the environment, always connected at 0.60m from the corners of the geometry. Source: The authors, 2021

This step is recalculated and readjusted using a Boolean controller. When restarting, each of the rules is reconnected, thus allowing a new composition of spaces to be formed. At the end, it is already possible to create numerous room layouts that already resemble the layout of a single-family residence.

The fourth and last step uses the layout of spaces to recreate something similar to a floor plan capable of passing on sensitive information to the CUG and CUB calculation, such as total square footage, square footage of wet areas, perimeter of internal walls and of external walls. These collected characteristics are transferred to a mathematical formula built with Grasshopper components. Here it is possible to change the build pattern for both CUB and CUB. These values were referenced to the Santa Catarina Sinduscon table (2020) for the month of November and December 2020, and are R$2674.88 for high standard, R$2221.20 for medium standard and R$1826.60 for low standard.

4 Algorithm validation and results

To validate the process, it was subjected to a set of tests in which it was adjusted and analyzed. The validation process took place in three cycles, seeking to improve the third step of programming. In each round, three of the research members observed a set of results of 20 random compositions in a terrain of 12m x 30m - starting the composition in the central point, then another 20 compositions in a terrain with the same dimensions, but with the point starting at 1/3 of the left side boundary, and finally another sequence of 20 compositions in a terrain measuring 18m x 35m - starting the composition at the central point. The tests took place at three different times, the first two rounds with the characteristics described above and the third taking into account the minimum lot size established as 125m² according to the City Statute of the Federal Senate (2001), with a 12m x 10m terrain – with the composition start point in the center of the front boundary. In total, for the refinement of the third step of the programming, 140 compositions were generated. In the sequence, Fig. 4 shows the test structure used for the refinement of the third step of programming.

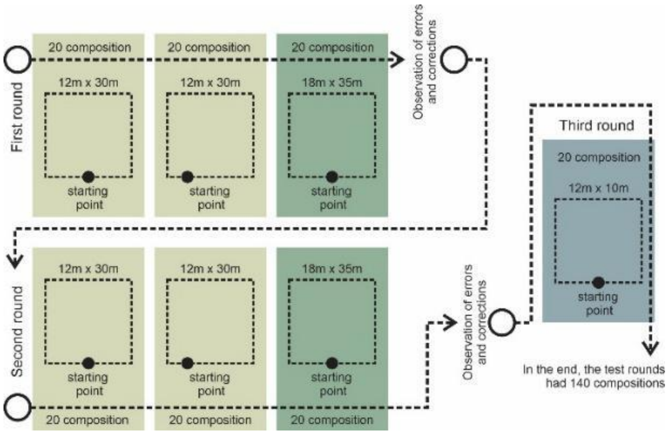
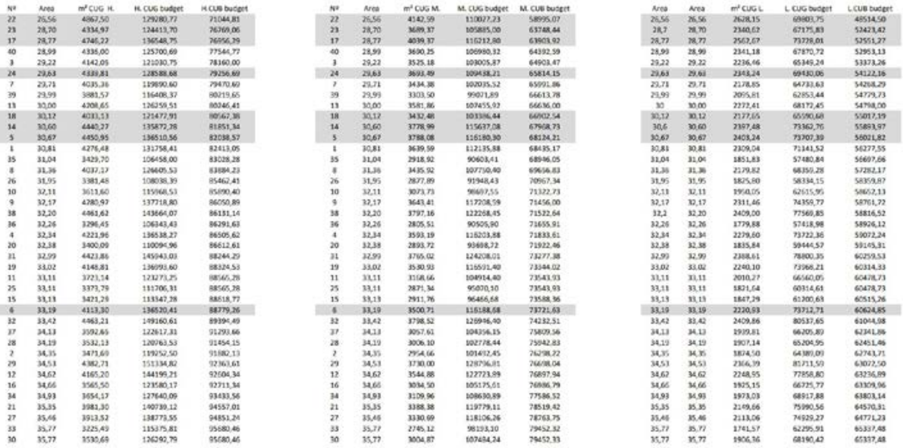


Figure 4. Logical sequence of programming tests. Source: The authors, 2021

The three tests had various issues that were adjusted and refined. The main negative characteristic of the first round of tests was the difficulty of positioning the rooms and the generation of unusual shapes. The problem was fixed by changing the shape's grammatical order, in the second round of testing the lack of bedrooms remained along with the lack of a bathroom in some of the compositions. This step was again adjusted, improving the organization of grammar and allowing the bathrooms to also connect to circulation. The third round of testing returned to show results with unusual shapes, and the error was corrected with new adjustments in the grammar of the shape and the distances that organized them.

As a result, the improved programming is able to: delimit the dimensions and its legal variables; dimension the minimum sizes for the implementation of rooms; organize spaces whose composition resembles the structure of a floor plan – layout; and finally calculate a possible cost of these constructions through two methods: CUG and CUB.

To use the programming and generate data to analyze, 40 compositions that resemble single-family layouts were made and developed on a 12.5m x 10m lot that corresponds to 125m². From these 40 compositions, possible construction costs are analyzed with CUG and CUB. As a result, it was possible to develop a comparative table that is able to visualize the difference in the budget of the 40 compositions. Fig 5 presents the 40 calculated budget results and Fig 6 their budget results. It is possible to associate the two figures from the number in the upper left corner of each housing configuration with the similar enumeration in the first column of the table



Nº	Area	m² CUG H.	H. CUG budget	H CUB budget
22	26,56	4867,50	129280,77	71044,81
23	28,70	4334,97	124413,70	76769,06
17	28,77	4746,22	136548,75	76956,39
40	28,99	4336,00	125700,69	77544,77
3	29,22	4142,05	121030,75	78160,00
24	29,63	4339,81	128588,68	79256,69
7	29,71	4035,36	119890,60	79470,69
39	29,99	3881,57	116408,37	80219,65
13	30,00	4208,65	126259,51	80246,41
18	30,12	4033,13	121477,91	80567,38
14	30,60	4440,27	135872,28	81851,34
5	30,67	4450,95	136510,56	82038,57
1	30,81	4276,48	131758,41	82413,05
35	31,04	3429,70	106458,00	83028,28
8	31,36	4037,17	126605,53	83884,23
26	31,95	3381,48	108038,39	85462,41
10	32,11	3611,60	115968,53	85890,40
9	32,17	4280,97	137718,80	86050,89
38	32,20	4461,62	143664,07	86131,14
36	32,26	3296,45	106343,43	86291,63
4	32,34	4221,96	136538,27	86505,62
20	32,38	3400,09	110094,96	86612,61
31	32,99	4423,86	145943,03	88244,29
19	33,02	4148,81	136993,60	88324,53
11	33,11	3721,14	123273,25	88565,28
25	33,11	3373,79	111706,31	88565,28
15	33,13	3421,29	113347,28	88618,77
6	33,19	4113,30	136520,41	88779,26
32	33,42	4463,21	149160,61	89394,49
37	34,13	3592,65	122617,31	91293,66
28	34,19	3532,13	120763,53	91454,15
2	34,35	3471,69	119252,50	91882,13
29	34,53	4382,71	151334,82	92363,61
12	34,62	4165,20	144199,21	92604,34
16	34,66	3565,50	123580,17	92711,34
34	34,93	3654,17	127640,09	93433,56
21	35,35	3981,30	140739,12	94557,01
27	35,46	3913,52	138773,55	94851,24
33	35,77	3225,49	115375,81	95680,46
30	35,77	3530,69	126292,79	95680,46

Nº	Area	m² CUG M.	M. CUG budget	M. CUB budget
22	26,56	4142,59	110027,23	58995,07
23	28,70	3689,37	105885,00	63748,44
17	28,77	4039,37	116212,80	63903,92
40	28,99	3690,25	106980,32	64392,59
3	29,22	3525,18	103005,87	64903,47
24	29,63	3693,49	109438,21	65814,15
7	29,71	3434,38	102035,52	65991,86
39	29,99	3303,50	99071,89	66613,78
13	30,00	3581,86	107455,92	66636,00
18	30,12	3432,48	103396,44	66902,54
14	30,60	3778,99	115637,08	67968,73
5	30,67	3788,08	116180,30	68124,21
1	30,81	3639,59	112135,88	68435,17
35	31,04	2918,92	90603,41	68946,05
8	31,36	3435,92	107750,40	69656,83
26	31,95	2877,89	91948,43	70967,34
10	32,11	3073,73	98697,55	71322,73
9	32,17	3643,41	117208,59	71456,00
38	32,20	3797,16	122268,45	71522,64
36	32,26	2805,51	90505,90	71655,91
4	32,34	3593,19	116203,88	71833,61
20	32,38	2893,72	93698,72	71922,46
31	32,99	3765,02	124208,01	73277,38
19	33,02	3530,93	116591,40	73344,02
11	33,11	3168,66	104914,40	73543,93
25	33,11	2871,34	95070,10	73543,93
15	33,13	2911,76	96466,68	73588,36
6	33,19	3500,71	116188,68	73721,63
32	33,42	3798,52	126946,40	74232,51
37	34,13	3057,61	104356,15	75809,56
28	34,19	3006,10	102778,44	75942,83
2	34,35	2954,66	101492,45	76298,22
29	34,53	3730,00	128796,81	76698,04
12	34,62	3544,88	122723,89	76897,94
16	34,66	3034,50	105175,61	76986,79
34	34,93	3109,96	108630,89	77586,52
21	35,35	3388,38	119779,11	78519,42
27	35,46	3330,69	118106,26	78763,75
33	35,77	2745,12	98193,10	79452,32
30	35,77	3004,87	107484,24	79452,33

Area	Area	m² CUG L.	L. CUG budget	L CUB budget
26,56	26,56	2628,15	69803,75	48514,50
28,7	28,70	2340,62	67175,83	52423,42
28,77	28,77	2562,67	73728,01	52551,27
28,99	28,99	2341,18	67870,72	52953,13
29,22	29,22	2236,46	65349,24	53373,26
29,63	29,63	2343,24	69430,06	54122,16
29,71	29,71	2178,85	64733,63	54268,29
29,99	29,99	2095,81	62853,44	54779,73
30	30,00	2272,41	68172,45	54798,00
30,12	30,12	2177,65	65590,68	55017,19
30,6	30,60	2397,48	73362,76	55893,97
30,67	30,67	2403,24	73707,39	56021,82
30,81	30,81	2309,04	71141,52	56277,55
31,04	31,04	1851,83	57480,84	56697,66
31,36	31,36	2179,82	68359,28	57282,17
31,95	31,95	1825,80	58334,15	58359,87
32,11	32,11	1950,05	62615,95	58652,13
32,17	32,17	2311,46	74359,77	58761,72
32,2	32,20	2409,00	77569,85	58816,52
32,26	32,26	1779,88	57418,98	58926,12
32,34	32,34	2279,60	73722,36	59072,24
32,38	32,38	1835,84	59444,57	59145,31
32,99	32,99	2388,61	78800,35	60259,53
33,02	33,02	2240,10	73968,21	60314,33
33,11	33,11	2010,27	66560,05	60478,73
33,11	33,11	1821,64	60314,61	60478,73
33,13	33,13	1847,29	61200,63	60515,26
33,19	33,19	2220,93	73712,71	60624,85
33,42	33,42	2409,86	80537,65	61044,98
34,13	34,13	1939,81	66205,89	62341,86
34,19	34,19	1907,14	65204,95	62451,46
34,35	34,35	1874,50	64389,09	62743,71
34,53	34,53	2366,39	81711,59	63072,50
34,62	34,62	2248,95	77858,80	63236,89
34,66	34,66	1925,15	66725,77	63309,96
34,93	34,93	1973,03	68917,88	63803,14
35,35	35,35	2149,66	75990,56	64570,81
35,46	35,46	2113,06	74929,27	64771,23
35,77	35,77	1741,57	62295,91	65337,48
35,77	35,77	1906,36	68190,42	65337,48

Figure 5. Budget results. Source: The authors, 2021

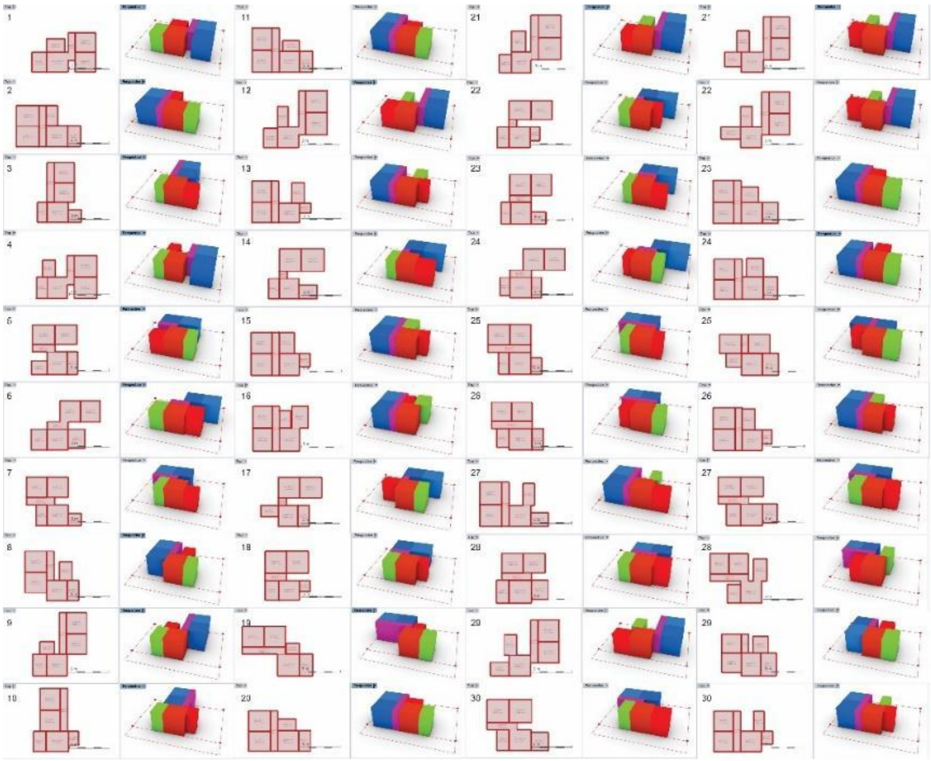


Figure 6. The formal results. Source: Source: The authors, 2021

5 Discussions

From the programming developed in this research, it was noted that the automation of the layout generation process helps increase the number of possibilities for distributing rooms. On the other hand, it also shows how this process is complex and involves a large number of restrictions in order to generate viable solutions. In this sense, the importance of using dynamic programming as a method for improving the algorithm becomes clear.

The use of programming also allows designers to evaluate compositions according to their costs, making it possible to find more interesting architectural designs for a given construction cost. This result was possible by applying information and results obtained in the initial stages of the project. Interpreting budget data can also help students understand how the form of project contributes to the final cost of the work.

The results found also demonstrate the importance of applying and teaching programming and logic practices within architecture and urbanism. The growing use of technologies that imply the use of programming (visual and written), such as simulations, is something that is already part of the design practice routine, and its correct use can result in the improvement of various design characteristics.

It is intended in future works to further analyze the results of the budgets generated by the programming, evaluating the various compositions and their consequent cost. It is also intended to improve the programming, as its results are generated randomly without the possibility of being saved, and there is no way to return to a given composition. If this is implemented, programming can receive a generative process and be able to select the most suitable proposal for a given value.

Acknowledgements. The authors would like to thank the Universidade Federal de Santa Catarina – UFSC, the PósARQ program and the Grupo de Modelagem Avançada - GMA group**.**

References

ALFARIS, A.; MERELLO, R. (2008). The Generative Multi-Performance Design System. Acadia 8: Silicor + Skin, n. 8, p.448-457.

AMORE, C. S.; SHIMBO, L. Z.; RUFINO, M. B. C. (2015). Minha Casa e a cidade? Avaliação do Programa Minha Casa Minha Vida em Seis Estados Brasileiros.

CALIXTO, V. (2016). Geração automatizada de layouts com uso de algoritmos evolutivos: aplicação em arquitetura e urbanismo. 181 f. Dissertação (Mestrado) – Curso de Engenharia Civil, Arquitetura e Urbanismo, Tecnologia e Cidade, Universidade Estadual de Campinas, Campinas.

CELANI, G. et al. (2006). A gramática da forma como metodologia de análise e síntese em arquitetura. Conexão – Comunicação e Cultura, Ucs, Caxias do Sul, v. 5, n. 10, p. 182-196, 10.

CREWS, C. (2019). What machine learning can learn from foresight: a human-centered approach. Reserch-Tecnology Management. V. 62, n. 1, p.30-33.

DVORKIN, M. (2016). Jobs involving routine tasks aren't growing. On the Economy.

FGV IBRE (2020). Instituto Brasileiro de Economia da Fundação Getúlio Vargas (org.). INCC - Índice Nacional de Custo da Construção. 2020. Disponível em: https://portal.fgv.br/fgv-ibre.

GONÇALVES, C. M.; CEOTTO, L. H. (2014). Custo sem susto: um método para gestão do custo de edificações. São Paulo: O Nome da Rosa. 160 p.

KNILL, O. (2009). Probability Theory and Stochastic Processes with Applications: probability and stochastic processes with applications. Overseas Press.

KOVACS, A. T.; SZOBOSZLAI, M.; CSUSZ, I. (2019). Key for Entering Industry 4.0 in the AEC Sector: BIM Organisation Development. Ecaade 37 / Sigradi 23: BUILDING INFORMATION MODELLING, v. 1, n. 1, p.275-282.

LIMA, F. et al. (2013). Custo Unitário Geométrico: uma proposta de método de estimativa de custos na fase preliminar do projeto de edificações. SBQP 2013: Campinas, p. 1-12.

LIMA, F.S.A. (2013). Custo Unitário Geométrico: Uma Proposta de Método de Estimativa de Custos na Fase Preliminar do Projeto de Edificações. 116 f. Dissertação (Mestrado) - COPPE, Universidade Federal do Rio de Janeiro.

LIMA, F. de et al. (2016). Orçamento Na Fase Da Viabilidade: comparativo dos parâmetros custo unitário básico e custo unitário geométrico. EntaC. São Paulo, p. 3924-3937.

MASCARÓ, L. (1985). O Custo das Decisões Arquitetônicas. São Paulo: Nobel.

MCNEEL, R. (2014). Associates. Rhinoceros 3D modelling Software. v. 5. 2014. [Computer software]. https://www.rhino3d.com/

MORAES, A. F. S.; PICCHI, F.; GRANJA, A. (2016). Variáveis E Índices Geométricos De Projeto Arquitetônico Relacionados Ao Custo De Empreendimentos Residenciais. In: ENTAC 15, 2016, São Paulo. p. 3722-3732.

ROSSI, Andrea. (2017). Plugin para o Grasshopper, DDu Digital Design Unit. [Computer software]. https://github.com/ar0551/Wasp

RUTTEN, David; MCNEEL, Robert & Associates, (2014). Grasshopper 3D, EULA. [Computer software]. www.grasshopper3d.com.

SECRETARIA NACIONAL DE HABITAÇÃO. (2009). PROJETO BRA/00/019 - PNUD HBB: Como delimitar e regulamentar Zonas Especiais de Interesse Social ZEIS de Vazios Urbanos. Brasília. 50 p.

SENADO FEDERAL. (2001) LEI NO 10.257, DE 10 DE JULHO DE 2001: Estatuto da Cidade. Brasília: Subsecretaria de Edições Técnicas. 101 p.

STANGL, B. (2013). How Digital Tools Shape the Way Architects Think: the digital design process in architecture. 2013. 127 f. Dissertação (Mestrado) - Technische Universität Wien Für Architektur Und Raumplanung, Viena.

TOMASETTI, Thornton. TT toolbox for Grasshopper, (2017) CORE studio. [Computer software]. http://core.thorntontomasetti.com/colibri-release/

YANASSE, H. H; SOMA, N. Y.; MACULAN, N. (200.) An algorithm for determining the K-best solutions of the one-dimensional Knapsack problem. Pesquisa Operacional, v. 20, p. 117-134, UNIFESP (SciELO). http://dx.doi.org/10.1590/s0101-74382000000100011.