

# Construindo Novos Paradigmas: Pensamento Computacional no Ensino de Arquitetura

**Juliano V. C. Pita<sup>1</sup>, Karin R. Kagi<sup>2</sup>, Luciano B. de Paula<sup>2</sup>**

<sup>1</sup>Instituto Federal de São Paulo (IFSP) *Campus São Paulo.*

<sup>2</sup>Instituto Federal de São Paulo (IFSP) *Campus Bragança Paulista.*

juliano.pita@ifsp.edu.br, karinkagi@ifsp.edu.br, lbernardes@ifsp.edu.br

**Abstract.** *This paper presents the preliminary results of the design, implementation, and assessment of a curriculum that incorporates Computational Thinking in the field of design. As part of the research outcomes, a lesson plan was developed for a course in a Bachelor's degree program in Architecture and Urbanism, where the proposal is currently being applied. The initial results indicate the potential for fostering Computational Thinking in the educational process of designers, primarily through the pillar of Abstraction and the combined use of visual programming and textual programming tools.*

**Resumo.** *Este artigo apresenta os resultados parciais da estruturação, aplicação e verificação de uma sequência didática que considere o Pensamento Computacional na área de design. Como resultados da pesquisa, foi produzido um plano de ensino para uma disciplina de um curso de Bacharelado em Arquitetura e Urbanismo, no qual está sendo realizada a aplicação efetiva da proposta. Os resultados iniciais apontam para uma possibilidade de desenvolvimento do Pensamento Computacional no processo formativo de designers através principalmente do pilar da Abstração e do uso combinado de recursos de programação visual e programação textual.*

## 1. Introdução

O uso de recursos computacionais no processo de *design* (planejamento e desenvolvimento de projetos) teve sua adoção primariamente por meio do uso destes recursos de forma instrumental, objetivando a substituição de processos manuais de desenho e modelagem tridimensional, sem alterações profundas nas metodologias tradicionais de projeto [Oxman 2006]. Em escritórios de arquitetura, por exemplo, os computadores foram (e ainda são) utilizados como uma “prancheta eletrônica”, na qual as capacidades computacionais são utilizadas somente em substituição ao traço em papel, aprimorando sua precisão e rapidez de execução sem alterar o processo tradicional de *design*. Entretanto, novos processos estão se consolidando, especialmente com a disseminação do BIM (Building Information Modeling), mas ainda persiste o conflito entre as metodologias tradicionais e a mudança de paradigma necessária para exploração máxima do potencial das novas tecnologias [Succar 2009], [Eastman et al. 2008].

Tais mudanças paradigmáticas ensejam também transformações no Ensino, uma vez que estes recursos podem fazer parte do processo projetual desde sua concepção, superando sua utilização como meras ferramentas [Terzidis 2006]. Neste contexto,

surge a indagação: a introdução dos quatro pilares do Pensamento Computacional — Decomposição (identificação de um problema e sua fragmentação para facilitar a análise e a solução), Reconhecimento de Padrões (observação de semelhanças desses subproblemas com problemas anteriores), Abstração (observação somente do que é relevante ao contexto, desconsiderando especificidades), e Algoritmos (sequência de passos para resolução do problema) [Vicari et al. 2018] - na etapa de formação profissional poderia possibilitar uma melhor exploração de potencialidades do uso de recursos computacionais no processo de *design*?

Um desafio que se coloca é que a natureza da resolução dos problemas de *design*, por meio de Síntese, se diferencia da resolução por Análise, própria da área da Ciência da Computação. Enquanto na primeira se buscam padrões de soluções às condicionantes iniciais, com regras definidas posteriormente, na segunda se buscam regras que regem os elementos envolvidos em um problema e das quais derivam as soluções [Cross 2006]. Ademais, outra característica dos problemas de *design* é sua natureza como *wicked problems*, sem limites e formas definidas, na qual as soluções adotadas em sua resolução integram as novas iterações futuras do problema [Sabet 2010], que impossibilita a compreensão da totalidade dos dados. Essa característica enseja a produção de respostas preliminares para verificação de como elas se adequam ao problema, algo impensável nas Ciências Exatas [Jones 1970 apud Cross 2006]. Como se poderia harmonizar a abordagem sintética, característica do *design*, e a analítica, característica da Computação, visando a autonomia do profissional em formação no uso de recursos computacionais?

Neste sentido, o objetivo da pesquisa é discutir diretrizes para a introdução de conceitos do Pensamento Computacional no Ensino de *design*, visando alicerçar fundamentos para a construção de soluções projetuais com uso de recursos computacionais. Para materializar uma aplicação prática destas diretrizes, todo o planejamento e desenvolvimento da proposta foram realizados prevendo sua aplicação na concepção e desenvolvimento de uma disciplina sobre o tema, em um curso de Arquitetura e Urbanismo. As inovações deste estudo são a construção e introdução da disciplina como obrigatória na grade do curso, considerando que a computação no *design* não se limita ao ensino de ferramentas computacionais, mas demanda uma fundamentação teórica sólida sobre o assunto, e o fato de que o Pensamento Computacional é explorado de forma teórico-prática, estruturando o desenvolvimento de metodologias inovadoras em processos de projeto que serão exploradas posteriormente ao longo do curso.

## 2. Metodologia

Para o desenvolvimento da disciplina proposta, foi realizada uma revisão da literatura existente, seguida por uma Revisão Integrativa buscando compreender as principais experiências desenvolvidas na área e identificar trabalhos correlatos [Pita e Kagi 2025]. O plano de ensino foi analisado com o suporte da Taxonomia de Bloom revisada [Krathwohl 2002] e as ações propostas foram organizadas segundo o nível de complexidade e o objetivo de aprendizagem desejado e planejado, de forma a estruturar a abordagem para alcance de cinco objetivos principais: (1) Recordação de conceitos fundamentais sobre Computação; (2) Entendimento e aplicação da lógica de processamento de dados; (3) Análise das potencialidades e limites dos uso dos recursos computacionais em arquitetura e urbanismo; (4) Aplicação do pensamento computacional no enfrentamento de questões da Arquitetura e Urbanismo e (5) Avaliação do uso dos recursos computacionais.

nais, permitindo uma análise fundamentada do desenvolvimento das atividades didáticas. Também foram desenvolvidos os materiais de apoio ao ensino, composto por diagramas, dinâmicas, leituras, artefatos computacionais. Neste momento, a avaliação dos resultados foi realizada através das respostas aos exercícios propostos e da observação dos responsáveis pela disciplina.

Para o planejamento e desenvolvimento da disciplina foram levadas em consideração as características próprias do ensino de Arquitetura e Urbanismo no Brasil. Em sua quase totalidade, os cursos fazem uso da metodologia do Ateliê, caracterizado pela existência de uma situação-problema definida de maneira inicial e pela procura de soluções para essa situação [Artigas e Lira 2004]. Essa dinâmica se dá pelo estabelecimento de uma relação horizontal aluno-aluno, aluno-professor e professor-professor. Tendo em vista essa dinâmica, no lugar de aulas expositivas tradicionais, estudantes de arquitetura têm grande receptividade a dinâmicas que envolvam metáforas espaciais, a extração de padrões a partir de dados incompletos, a adaptabilidade a situações inusitadas e flexibilidade de reavaliação a partir de novos dados, e, sobretudo, a capacidade de abstração por meio da síntese [Celani e Vaz 2012], [Cross 2006], [Leitão 2013].

Assim, na preparação das aulas tornou-se necessário o uso de abordagens pedagógicas que explorassem essas características para uma abordagem mais eficiente. Dessa forma, além de poucos momentos limitados de aulas expositivas, foram utilizadas metáforas espaciais, discussões em rodas de conversa, adoção de sala de aula invertida [Holzer 2019], desenvolvimento em grupos e avaliação por pares, seminários no formato Pecha Kucha [Klentzin et al. 2010], desenvolvimento integrado de programação visual e textual e desenvolvimento de projeto a partir de uma técnica computacional, adotadas no desenvolvimento do conteúdo programático. Segundo a progressão cumulativa dos objetivos educacionais delineados, foram definidas três fases para desenvolvimento da disciplina: Estruturação, Enfrentamento e Sedimentação.

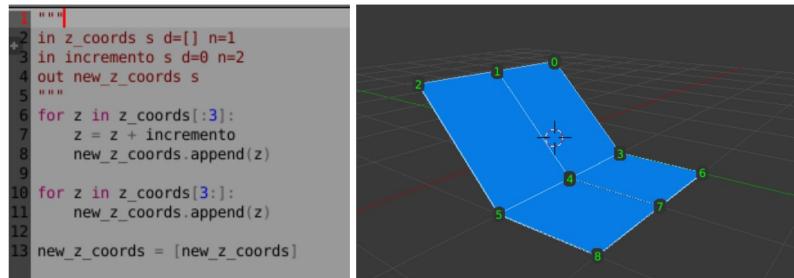
## 2.1. Desenvolvimento e aplicação

A Revisão Integrativa indicou que experiências semelhantes já haviam sido desenvolvidas em contextos de disciplinas optativas ou em nível de pós-graduação, o que impede a verificação de um impacto mais abrangente do ensino do Pensamento Computacional na formação geral. Dessa forma, a proposta foi aplicada em uma disciplina obrigatória do quarto semestre do curso de Arquitetura e Urbanismo do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, *campus São Paulo*, com carga horária de 28,5 horas, distribuída em 19 aulas semanais de 1h30.

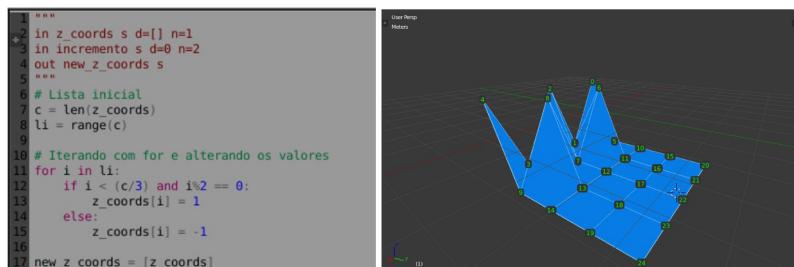
Ao longo das primeiras seis aulas do semestre, foi desenvolvida a fase de Estruturação, que teve como objetivo conceituar pontos-chave, construindo o repertório necessário para o desenvolvimento posterior das atividades da disciplina, valendo-se principalmente da leitura prévia de textos para discussão em sala de aula sobre a relação da arquitetura e computação e de dinâmicas de Computação Desplugada, introduzindo os conceitos fundamentais da Computação, tais como processamento, variáveis, estruturas de dados, classes etc.

Após esse primeiro momento, passou-se à segunda fase, Enfrentamento, na qual os pilares do Reconhecimento de Padrões e Algoritmo foram desenvolvidos. Esta fase demandou dos autores/educadores maior desenvolvimento de artefatos computacionais,

como códigos em linguagem Python em arquivos do aplicativo Blender, com estruturas de decisão e laços de repetição (Figuras 1 e 2), entre outros que pudessem facilitar aos estudantes a apropriação dos elementos básicos das linguagens de programação, identificação de padrões e desenvolvimento de algoritmos. Constituiu-se no elemento prático da disciplina, no qual efetivamente a discussão sobre o uso de recursos computacionais em arquitetura, e especialmente, o conceito de Algitectura [Terzidis 2006] (exemplo na Figura 3) tornou-se mais evidente. Essa fase ocupou mais cinco aulas da disciplina.



**Figura 1. Estruturas de repetição aplicadas a uma lista de pontos e a representação gráfica do resultado da execução do algoritmo. Fonte: Os autores.**



**Figura 2. Estruturas de repetição aplicadas a uma lista de pontos, com subsequentes estruturas de decisão e a representação gráfica do resultado da execução do algoritmo. Fonte: Os autores.**



**Figura 3. Referências de Algitectura apresentados aos alunos. Fonte: Alejandro Rodriguez (esq.), Enrico da Prato (centro), Karim Soliman (dir.)**

Por fim, na terceira fase da disciplina, chamada de Sedimentação, houve o desenvolvimento de trabalho em duplas, com o objetivo da aplicação, análise e avaliação do Pensamento Computacional por meio da criação de algoritmos simples desenvolvidos com vistas ao enfrentamento de situações-problema típicas da área de Arquitetura e Urbanismo, consolidando assim a necessidade de um maior domínio dos recursos computacionais no exercício contemporâneo da profissão. Durante as sete aulas desta fase,

foram propostas atividades a partir de quatro técnicas computacionais, cada uma dessas implementadas por quatro duplas de alunos. A divisão em duplas permite que se implemente a estratégia de programação em pares [Williams 2001], importante instrumento para desenvolvimento do raciocínio lógico e analítico por meio de um processo dialógico de discussão do problema proposto. A atribuição de um mesmo problema para quatro duplas permite que se estabeleça esse mesmo processo em uma escala mais ampla, através da análise comparativa de soluções, encaminhamentos e percursos. As técnicas computacionais propostas foram:

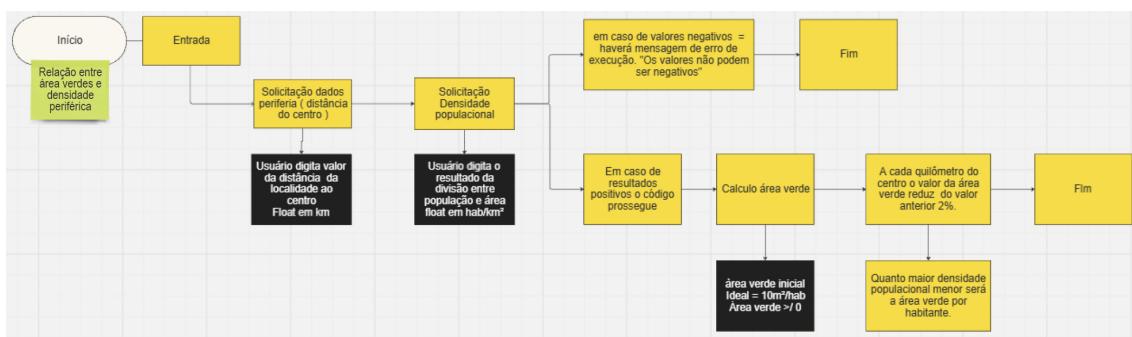
**Estudo, desenvolvimento e manipulação de formas complexas** - Utilização de recursos computacionais para a concepção e desenvolvimento de formas complexas de elementos arquitetônicos, como coberturas, cascas, divisórias, etc. Por forma complexa, define-se aquelas que fazem uso de geometrias não euclidianas, que possuem múltiplas curvaturas e são resultado de representações imagéticas de funções matemáticas [Tramontano 2016]. O desenvolvimento do algoritmo se baseia na definição dessas funções como regras implementadas e na manipulação de listas e condições;

**Gramática da Forma** - Construção de padrões geométricos complexos a partir de formas iniciais e regras simples [Knight 1999] e a associação destas formas a aspectos não geométricos, como é o caso do uso de algoritmos implementados para estudo de distribuição de espaços em um plano de edificação, oferecendo soluções diversas. Neste caso, o desenvolvimento do algoritmo ocorrerá com base na definição de regras iniciais, com condições para relação entre partes da forma, e o uso de recursividade;

**Criação de objetos parametrizados** - Desenvolvimento de algoritmos que implementam um conjunto de regras para a criação de um determinado objeto arquitetônico a partir da abstração dos elementos definidores da tipologia do objeto [Leitão et al. 2012]. O algoritmo deve implementar regras simples que produzam resultados diversos do mesmo objeto a partir de entradas do usuário;

**Criação de padrões a partir de atratores e repulsores** - Estruturação de algoritmos que determinam a distribuição de objetos ou a localização de elementos no espaço a partir de regras simples ou, ainda, de bases aleatórias, bem como o estabelecimento de relações de proximidade em relação a estes. O algoritmo deverá estabelecer relações espaciais entre diferentes objetos e suas características fundamentais, como altura ou área de sua projeção.

Os alunos foram incentivados a, dentro dos temas propostos, propor uma situação-problema a ser resolvida, como por exemplo, o cálculo de uma cobertura em relação aos ventos dominantes de uma região, de forma a maximizar a captação da ventilação natural e minimizar a carga estrutural acidental nas telhas, ou ainda, na distribuição de elementos de revestimento, como ladrilhos, a partir de regras flexíveis. Foi solicitado aos alunos que entregassem uma descrição textual do caso concreto, uma descrição textual do algoritmo proposto e do seu funcionamento, com o fluxo de informações. Também foi solicitado um diagrama de abstração, na ideia de um diagrama de classes (Figura 4), mas sem a formalidade demandada para a Programação Orientada a Objetos, e um diagrama do fluxo de informações (Figura 5), desde sua entrada, processamento, desvios lógicos e retorno. Além disso, foi solicitado a entrega dos códigos efetivamente desenvolvidos em Python, como uma resposta ao desenvolvimento anterior (Figura 6).



**Figura 4. Diagrama de fluxo de informações desenvolvidos por alunos. Fonte: Os autores.**



**Figura 5. Diagrama de abstração desenvolvidos pelos alunos. A discussão em conjunto, com o uso dos quadros da sala de aula foi essencial para o avanço da discussão. Fonte: Os autores.**

### 3. Resultados

Consideramos que o objetivo principal da pesquisa, o de estruturar, aplicar e verificar a construção de um sequência didática voltado à superação das barreiras à introdução do Pensamento Computacional no processo formativo para desenvolvimento de projetos (*design*) foi preliminarmente alcançado. A receptividade do plano proposto somente pode ser parcialmente verificada, mas os resultados até o momento são encorajadores. Por fim, como suporte a esses desenvolvimentos, foram produzidos artefatos computacionais, na forma de dinâmicas sem o uso de computadores, e da estruturação do ambiente misto de programação visual e textual. Estes artefatos estão disponíveis em <https://github.com/julianopita/Pensamento-Computacional> e <https://github.com/julianopita/Pensamento-Computacional>, em repositório público. Os artefatos estão disponibilizados sob a licença Creative Commons Attribution-NonCommercial (CC BY-NC).

A análise dos trabalhos finais da disciplina demonstrou a construção, por parte dos alunos, de uma capacidade de aproximação lógica a um problema proposto em consonância dos quatro pilares do pensamento computacional. A maior parte dos alunos foi capaz de subdividir um problema em partes menores, observar padrões na inter-relação entre os diferentes componentes do problema, abstrair os elementos principais e essenciais para o desenvolvimento de um algoritmo capaz de resolver o problema inicialmente

```

1 import numpy as np
2 import mathutils
3 import pandas as pd
4 import os
5 import math
6 import random
7 import mathutils
8 import math
9 import mathutils
10 import mathutils
11 import mathutils
12 import mathutils
13 import mathutils
14 import mathutils
15 import mathutils
16 import mathutils
17 import mathutils
18 import mathutils
19 import mathutils
20 import mathutils
21 import mathutils
22 import mathutils
23 import mathutils
24 import mathutils
25 import mathutils
26 import mathutils
27 import mathutils
28 import mathutils
29 import mathutils
30 import mathutils
31 import mathutils
32 import mathutils
33 import mathutils
34 import mathutils
35 import mathutils
36 import mathutils
37 import mathutils
38 import mathutils
39 import mathutils
40 import mathutils
41 import mathutils
42 import mathutils
43 import mathutils
44 import mathutils
45 import mathutils
46 import mathutils
47 import mathutils
48 import mathutils
49 import mathutils
50 import mathutils
51 import mathutils
52 import mathutils
53 import mathutils
54 import mathutils
55 import mathutils
56 import mathutils
57 import mathutils
58 import mathutils
59 import mathutils
60 import mathutils
61 import mathutils
62 import mathutils
63 import mathutils
64 import mathutils
65 import mathutils
66 import mathutils
67 import mathutils
68 import mathutils
69 import mathutils
70 import mathutils
71 import mathutils
72 import mathutils
73 import mathutils
74 import mathutils
75 import mathutils
76 import mathutils
77 import mathutils
78 import mathutils
79 import mathutils
80 import mathutils
81 import mathutils
82 import mathutils
83 import mathutils
84 import mathutils
85 import mathutils
86 import mathutils
87 import mathutils
88 import mathutils
89 import mathutils
90 import mathutils
91 import mathutils
92 import mathutils
93 import mathutils
94 import mathutils
95 import mathutils
96 import mathutils
97 import mathutils
98 import mathutils
99 import mathutils
100 import mathutils
101 import mathutils
102 import mathutils
103 import mathutils
104 import mathutils
105 import mathutils
106 import mathutils
107 import mathutils
108 import mathutils
109 import mathutils
110 import mathutils
111 import mathutils
112 import mathutils
113 import mathutils
114 import mathutils
115 import mathutils
116 import mathutils
117 import mathutils
118 import mathutils
119 import mathutils
120 import mathutils
121 import mathutils
122 import mathutils
123 import mathutils
124 import mathutils
125 import mathutils
126 import mathutils
127 import mathutils
128 import mathutils
129 import mathutils
130 import mathutils
131 import mathutils
132 import mathutils
133 import mathutils
134 import mathutils
135 import mathutils
136 import mathutils
137 import mathutils
138 import mathutils
139 import mathutils
140 import mathutils
141 import mathutils
142 import mathutils
143 import mathutils
144 import mathutils
145 import mathutils
146 import mathutils
147 import mathutils
148 import mathutils
149 import mathutils
150 import mathutils
151 import mathutils
152 import mathutils
153 import mathutils
154 import mathutils
155 import mathutils
156 import mathutils
157 import mathutils
158 import mathutils
159 import mathutils
160 import mathutils
161 import mathutils
162 import mathutils
163 import mathutils
164 import mathutils
165 import mathutils
166 import mathutils
167 import mathutils
168 import mathutils
169 import mathutils
170 import mathutils
171 import mathutils
172 import mathutils
173 import mathutils
174 import mathutils
175 import mathutils
176 import mathutils
177 import mathutils
178 import mathutils
179 import mathutils
180 import mathutils
181 import mathutils
182 import mathutils
183 import mathutils
184 import mathutils
185 import mathutils
186 import mathutils
187 import mathutils
188 import mathutils
189 import mathutils
190 import mathutils
191 import mathutils
192 import mathutils
193 import mathutils
194 import mathutils
195 import mathutils
196 import mathutils
197 import mathutils
198 import mathutils
199 import mathutils
200 import mathutils
201 import mathutils
202 import mathutils
203 import mathutils
204 import mathutils
205 import mathutils
206 import mathutils
207 import mathutils
208 import mathutils
209 import mathutils
210 import mathutils
211 import mathutils
212 import mathutils
213 import mathutils
214 import mathutils
215 import mathutils
216 import mathutils
217 import mathutils
218 import mathutils
219 import mathutils
220 import mathutils
221 import mathutils
222 import mathutils
223 import mathutils
224 import mathutils
225 import mathutils
226 import mathutils
227 import mathutils
228 import mathutils
229 import mathutils
230 import mathutils
231 import mathutils
232 import mathutils
233 import mathutils
234 import mathutils
235 import mathutils
236 import mathutils
237 import mathutils
238 import mathutils
239 import mathutils
240 import mathutils
241 import mathutils
242 import mathutils
243 import mathutils
244 import mathutils
245 import mathutils
246 import mathutils
247 import mathutils
248 import mathutils
249 import mathutils
250 import mathutils
251 import mathutils
252 import mathutils
253 import mathutils
254 import mathutils
255 import mathutils
256 import mathutils
257 import mathutils
258 import mathutils
259 import mathutils
260 import mathutils
261 import mathutils
262 import mathutils
263 import mathutils
264 import mathutils
265 import mathutils
266 import mathutils
267 import mathutils
268 import mathutils
269 import mathutils
270 import mathutils
271 import mathutils
272 import mathutils
273 import mathutils
274 import mathutils
275 import mathutils
276 import mathutils
277 import mathutils
278 import mathutils
279 import mathutils
280 import mathutils
281 import mathutils
282 import mathutils
283 import mathutils
284 import mathutils
285 import mathutils
286 import mathutils
287 import mathutils
288 import mathutils
289 import mathutils
290 import mathutils
291 import mathutils
292 import mathutils
293 import mathutils
294 import mathutils
295 import mathutils
296 import mathutils
297 import mathutils
298 import mathutils
299 import mathutils
300 import mathutils
301 import mathutils
302 import mathutils
303 import mathutils
304 import mathutils
305 import mathutils
306 import mathutils
307 import mathutils
308 import mathutils
309 import mathutils
310 import mathutils
311 import mathutils
312 import mathutils
313 import mathutils
314 import mathutils
315 import mathutils
316 import mathutils
317 import mathutils
318 import mathutils
319 import mathutils
320 import mathutils
321 import mathutils
322 import mathutils
323 import mathutils
324 import mathutils
325 import mathutils
326 import mathutils
327 import mathutils
328 import mathutils
329 import mathutils
330 import mathutils
331 import mathutils
332 import mathutils
333 import mathutils
334 import mathutils
335 import mathutils
336 import mathutils
337 import mathutils
338 import mathutils
339 import mathutils
340 import mathutils
341 import mathutils
342 import mathutils
343 import mathutils
344 import mathutils
345 import mathutils
346 import mathutils
347 import mathutils
348 import mathutils
349 import mathutils
350 import mathutils
351 import mathutils
352 import mathutils
353 import mathutils
354 import mathutils
355 import mathutils
356 import mathutils
357 import mathutils
358 import mathutils
359 import mathutils
360 import mathutils
361 import mathutils
362 import mathutils
363 import mathutils
364 import mathutils
365 import mathutils
366 import mathutils
367 import mathutils
368 import mathutils
369 import mathutils
370 import mathutils
371 import mathutils
372 import mathutils
373 import mathutils
374 import mathutils
375 import mathutils
376 import mathutils
377 import mathutils
378 import mathutils
379 import mathutils
380 import mathutils
381 import mathutils
382 import mathutils
383 import mathutils
384 import mathutils
385 import mathutils
386 import mathutils
387 import mathutils
388 import mathutils
389 import mathutils
390 import mathutils
391 import mathutils
392 import mathutils
393 import mathutils
394 import mathutils
395 import mathutils
396 import mathutils
397 import mathutils
398 import mathutils
399 import mathutils
400 import mathutils
401 import mathutils
402 import mathutils
403 import mathutils
404 import mathutils
405 import mathutils
406 import mathutils
407 import mathutils
408 import mathutils
409 import mathutils
410 import mathutils
411 import mathutils
412 import mathutils
413 import mathutils
414 import mathutils
415 import mathutils
416 import mathutils
417 import mathutils
418 import mathutils
419 import mathutils
420 import mathutils
421 import mathutils
422 import mathutils
423 import mathutils
424 import mathutils
425 import mathutils
426 import mathutils
427 import mathutils
428 import mathutils
429 import mathutils
430 import mathutils
431 import mathutils
432 import mathutils
433 import mathutils
434 import mathutils
435 import mathutils
436 import mathutils
437 import mathutils
438 import mathutils
439 import mathutils
440 import mathutils
441 import mathutils
442 import mathutils
443 import mathutils
444 import mathutils
445 import mathutils
446 import mathutils
447 import mathutils
448 import mathutils
449 import mathutils
450 import mathutils
451 import mathutils
452 import mathutils
453 import mathutils
454 import mathutils
455 import mathutils
456 import mathutils
457 import mathutils
458 import mathutils
459 import mathutils
460 import mathutils
461 import mathutils
462 import mathutils
463 import mathutils
464 import mathutils
465 import mathutils
466 import mathutils
467 import mathutils
468 import mathutils
469 import mathutils
470 import mathutils
471 import mathutils
472 import mathutils
473 import mathutils
474 import mathutils
475 import mathutils
476 import mathutils
477 import mathutils
478 import mathutils
479 import mathutils
480 import mathutils
481 import mathutils
482 import mathutils
483 import mathutils
484 import mathutils
485 import mathutils
486 import mathutils
487 import mathutils
488 import mathutils
489 import mathutils
490 import mathutils
491 import mathutils
492 import mathutils
493 import mathutils
494 import mathutils
495 import mathutils
496 import mathutils
497 import mathutils
498 import mathutils
499 import mathutils
500 import mathutils
501 import mathutils
502 import mathutils
503 import mathutils
504 import mathutils
505 import mathutils
506 import mathutils
507 import mathutils
508 import mathutils
509 import mathutils
510 import mathutils
511 import mathutils
512 import mathutils
513 import mathutils
514 import mathutils
515 import mathutils
516 import mathutils
517 import mathutils
518 import mathutils
519 import mathutils
520 import mathutils
521 import mathutils
522 import mathutils
523 import mathutils
524 import mathutils
525 import mathutils
526 import mathutils
527 import mathutils
528 import mathutils
529 import mathutils
530 import mathutils
531 import mathutils
532 import mathutils
533 import mathutils
534 import mathutils
535 import mathutils
536 import mathutils
537 import mathutils
538 import mathutils
539 import mathutils
540 import mathutils
541 import mathutils
542 import mathutils
543 import mathutils
544 import mathutils
545 import mathutils
546 import mathutils
547 import mathutils
548 import mathutils
549 import mathutils
550 import mathutils
551 import mathutils
552 import mathutils
553 import mathutils
554 import mathutils
555 import mathutils
556 import mathutils
557 import mathutils
558 import mathutils
559 import mathutils
560 import mathutils
561 import mathutils
562 import mathutils
563 import mathutils
564 import mathutils
565 import mathutils
566 import mathutils
567 import mathutils
568 import mathutils
569 import mathutils
570 import mathutils
571 import mathutils
572 import mathutils
573 import mathutils
574 import mathutils
575 import mathutils
576 import mathutils
577 import mathutils
578 import mathutils
579 import mathutils
580 import mathutils
581 import mathutils
582 import mathutils
583 import mathutils
584 import mathutils
585 import mathutils
586 import mathutils
587 import mathutils
588 import mathutils
589 import mathutils
590 import mathutils
591 import mathutils
592 import mathutils
593 import mathutils
594 import mathutils
595 import mathutils
596 import mathutils
597 import mathutils
598 import mathutils
599 import mathutils
600 import mathutils
601 import mathutils
602 import mathutils
603 import mathutils
604 import mathutils
605 import mathutils
606 import mathutils
607 import mathutils
608 import mathutils
609 import mathutils
610 import mathutils
611 import mathutils
612 import mathutils
613 import mathutils
614 import mathutils
615 import mathutils
616 import mathutils
617 import mathutils
618 import mathutils
619 import mathutils
620 import mathutils
621 import mathutils
622 import mathutils
623 import mathutils
624 import mathutils
625 import mathutils
626 import mathutils
627 import mathutils
628 import mathutils
629 import mathutils
630 import mathutils
631 import mathutils
632 import mathutils
633 import mathutils
634 import mathutils
635 import mathutils
636 import mathutils
637 import mathutils
638 import mathutils
639 import mathutils
640 import mathutils
641 import mathutils
642 import mathutils
643 import mathutils
644 import mathutils
645 import mathutils
646 import mathutils
647 import mathutils
648 import mathutils
649 import mathutils
650 import mathutils
651 import mathutils
652 import mathutils
653 import mathutils
654 import mathutils
655 import mathutils
656 import mathutils
657 import mathutils
658 import mathutils
659 import mathutils
660 import mathutils
661 import mathutils
662 import mathutils
663 import mathutils
664 import mathutils
665 import mathutils
666 import mathutils
667 import mathutils
668 import mathutils
669 import mathutils
670 import mathutils
671 import mathutils
672 import mathutils
673 import mathutils
674 import mathutils
675 import mathutils
676 import mathutils
677 import mathutils
678 import mathutils
679 import mathutils
680 import mathutils
681 import mathutils
682 import mathutils
683 import mathutils
684 import mathutils
685 import mathutils
686 import mathutils
687 import mathutils
688 import mathutils
689 import mathutils
690 import mathutils
691 import mathutils
692 import mathutils
693 import mathutils
694 import mathutils
695 import mathutils
696 import mathutils
697 import mathutils
698 import mathutils
699 import mathutils
700 import mathutils
711

```

**Figura 6. Conjunto de algoritmos desenvolvidos pelos alunos para resolução dos problemas propostos. Fonte: Os autores.**

proposto. O desenvolvimento do algoritmo, entretanto, foi somente ensaiada, com poucos alunos efetivamente construindo códigos funcionais, ainda que, em uma análise à guisa de leitura de pseudocódigo, a organização esteja coerente com o desenvolvimento anterior. Esse resultado é esperado, tanto pela pequena carga horária da disciplina, quanto pelas limitações dos equipamentos e *softwares* disponibilizados, mas consideramos que o exercício de implementação efetiva, ainda que incompleto, seja importante para concretizar o processo, algo bastante importante para alunos do curso de Arquitetura e Urbanismo, conforme já discutido. Espera-se que, em uma disciplina posterior da sequência do curso dedicada exclusivamente à modelagem paramétrica arquitetônica, seja possível concentrar-se no desenvolvimento de algoritmos funcionais. A análise mais detalhada dos resultados será realizada a seguir, na discussão.

#### 4. Conclusões e desenvolvimentos futuros

Os resultados parciais da pesquisa foram avaliados, neste momento, por meio das atividades entregues e do envolvimento dos alunos nas dinâmicas propostas, bem como as suas respostas no decorrer da disciplina. Dessa forma, a discussão terá a forma de um relato de experiência baseado na percepção e observação do autor responsável pela disciplina, em discussão constante com os outros autores do estudo. Apesar das limitações deste instrumento avaliativo, entende-se que é adequado para apresentar um retrato instantâneo de um estudo em desenvolvimento e apontar suas potencialidades e limites iniciais, além de fornecer parâmetros mais precisos para futuros instrumentos de avaliação. Assim, por tratar-se de uma pesquisa de longo, prazo serão utilizados em momento posterior questionários e entrevistas estruturadas, que demandam aprovação dos comitês de Ética competentes e uma maior amostragem para a produção de resultados significativos.

Durante a fase de Estruturação em diversos momentos foram observadas instâncias de conversação efetiva entre os alunos [Pask 1976], ou seja, a informação emitida foi recebida pelos colegas, que transformaram e elaboraram novas informações sobre o conteúdo inicial. Em geral, observou-se a participação espontânea de mais da metade da turma nas discussões, de forma estruturada e aprofundada. Um aspecto relevante é que, nesta fase, o docente colocou-se no papel de mediador da discussão, preparado para abordar pontos não discutidos ao longo do debate dos textos, provocando e redirecionando a troca de informações, mas foi raramente necessária essa intervenção.

Durante a fase de Enfrentamento, foi observada uma mudança na dinâmica da turma em relação ao objeto de trabalho. A relação professor-aluno passou a se estruturar de maneira mais tradicional, e observaram-se momentos de frustração com os equipamentos e o ambiente de trabalho. Apesar de esperado, já que o enfrentamento pressupõe um deslocamento da zona de conforto dos alunos, essas questões foram certamente agravadas devido à infraestrutura disponível e o ambiente de trabalho complexo do aplicativo Blender com *add-on* Sverchok, utilizado como suporte à programação textual e visual.

Entretanto, após as primeiras aulas essas questões foram sendo equalizadas; os alunos passaram a concentrar as suas incertezas na questão da programação em si, e no objetivo do conteúdo. Neste momento, exemplos de aplicação da programação na arquitetura foram introduzidos e foi realizada a comparação com o aprendizado de uma nova linguagem, em que os conceitos mais básicos e fundamentais são trabalhados, o vocabulário é expandido para só então poder realizar-se a leitura de obras mais complexas.

Durante a fase de Sedimentação, a aplicabilidade mais direta do conteúdo em situações-problema da Arquitetura estabeleceu mais uma ligação entre o conteúdo da disciplina e a prática arquitetônica, e o avanço durante as aulas e construção mediada do produto final a ser entregue diminui as incertezas sobre o final do processo.

Na continuidade desta pesquisa, pretende-se verificar a articulação dos conceitos trabalhados de Pensamento Computacional com as duas outras disciplinas voltadas à computação na grade do curso, dedicadas à modelagem computacional. Essas disciplinas são especialmente importantes pois, ao contrário da disciplina aqui discutida, são integradas ao exercício projetual desenvolvido no ateliê de Arquitetura, Urbanismo e Paisagismo. Espera-se, dessa forma, que os alunos possam criar a partir dos conceitos aplicados às técnicas a serem desenvolvidas nas disciplinas e aplicadas no ateliê, completando assim o ciclo completo preconizado na Taxonomia de Bloom [Krathwohl 2002].

Outro desdobramento futuro da pesquisa é a sua aplicabilidade em relação aos demais docentes do curso. A apropriação dos recursos computacionais só poderá ser sedimentada se sua aplicabilidade permear as demais disciplinas do curso, de forma a construir uma continuidade de seu uso. Assim, pode ser importante a construção de cursos de atualização e formação continuada para os docentes, oferecendo recursos para serem introduzidos nas disciplinas sob sua responsabilidade.

Outra vertente esperada é a do desenvolvimento de estratégias de *design* voltado à formação de professores da Educação Básica, especificamente direcionado a uso de recursos de laboratórios *makers*. A proposta compartilha das mesmas premissas mencionadas no parágrafo anterior: Pensamento Computacional e adoção efetiva de tecnologias em Educação deve permear o currículo escolar, sem se restringir a disciplinas específicas, como Computação ou Robótica.

Em um panorama mais amplo, observa-se a possibilidade de continuidade da pesquisa ao longo dos próximos anos, com o refinamento do plano de ensino desenvolvido e com os desdobramentos posteriores nas disciplinas subsequentes. Também pretende-se desenvolver a vertente voltada aos docentes de níveis diversos, entendendo que, na contemporaneidade, o Pensamento Computacional deve ser desenvolvido como forma de interação plena com o mundo altamente digitalizado.

## Referências

- Artigas, R. e Lira, J. T. C. d. (2004). *Caminhos da arquitetura. Vilanova Artigas*. Cosac Naify, São Paulo, 4a ed. edition.
- Celani, G. e Vaz, C. E. V. (2012). Cad scripting and visual programming languages for implementing computational design concepts: A comparison from a pedagogical point of view. *International Journal of Architectural Computing*, 10(1):121–137.
- Cross, N. (2006). *Designerly Ways of Knowing*, pages 1–13. Springer London, London.
- Eastman, C., Teicholz, P., Sacks, R., e Liston, K. (2008). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Wiley Publishing.
- Holzer, D. (2019). Teaching computational design and bim in the age of (semi)flipped classrooms. In *Haeusler, M. and Schnabel, M. A. and Fukuda, T. (eds.), Intelligent Informed - Proceedings of the 24th CAADRIA Conference - Volume 2, Victoria University of Wellington, Wellington, New Zealand, 15-18 April 2019*, pp. 715-724.
- Klentzin, J. C., Paladino, E. B., Johnston, B., e Devine, C. (2010). Pecha kucha: using “lightning talk” in university instruction. *Reference Services Review*, Vol. 38 No. 1, pp. 158-167.
- Knight, T. (1999). Applications in architectural design, and education and practice. In *NSF/MIT Workshop on Shape Computation*, volume 67.
- Krathwohl, D. R. (2002). A revision of bloom’s taxonomy: An overview. *Theory Into Practice*, 41(4):212–218.
- Leitão, A., Santos, L., e Lopes, J. (2012). Programming languages for generative design: A comparative study. *International Journal of Architectural Computing*, 10:139–162.
- Leitão, A. M. (2013). Teaching computer science for architecture - a proposal. In *FUTURE TRADITIONS [1st eCAADe Regional International Workshop Proceedings / ISBN 978-989-8527-03-5], University of Porto, Faculty of Architecture (Portugal), 4-5 April 2013*, pp. 95-104.
- Oxman, R. (2006). Theory and design in the first digital age. *Design Studies*, 27.
- Pask, G. (1976). *Conversation Theory. Applications in education and epistemology*. Elsevier.
- Pita, J. V. d. C. e Kagi, K. R. (2025). Desenvolvimento do pensamento computacional em ensino de design: um estudo e aplicação na área de arquitetura. Trabalho de Graduação em Análise e Desenvolvimento de Sistemas. Orientador: Prof. Dr. Luciano Bernardes de Paula.
- Sabet, A. (2010). Wickedness, governance and collective sanctions: : Can corruption be tamed? In Salminen, A., editor, *Ethical Governance: a citizen perspective.*, pages 91–112. Vaasa University Press.
- Succar, B. (2009). Building information modelling framework: A research and delivery foundation for industry stakeholders. *Automation in Construction*, 18(3):357–375.
- Terzidis, K. (2006). *Algorithmic Architecture*. Routledge, 1th edition.

- Tramontano, M. (2016). Quando pesquisa e ensino se conectam. design paramétrico, fabricação digital e projeto de arquitetura. *Arquitectos, São Paulo, ano 16, n. 190.01, Vitruvius, mar. 2016.*
- Vicari, R. M., Moreira, A. F., e Menezes, P. F. B. (2018). *Pensamento computacional : revisão bibliográfica.* UFRGS/MEC, Porto Alegre.
- Williams, L. (2001). Integrating pair programming into a software development process. In *Proceedings 14th Conference on Software Engineering Education and Training. 'In search of a software engineering profession' (Cat. No.PR01059)*, pages 27–36.