Artifacts Available CBSoft 2026
Artifacts Functional CBSoft 2026

Programming Dojo as a Pedagogical Strategy in Object-Oriented Programming Education: A Multi-Semester Study at a Public and a Private Institution

Talita Vieira Ribeiro
Universidade do Estado do Rio de Janeiro
Rio de Janeiro, Brazil
Centro Universitário IBMEC-RJ
Rio de Janeiro, Brazil
talita.ribeiro@ime.uerj.br

Rebeca Campos Motta
Universidade Federal Fluminense
Niterói, Brazil
rcmotta@ic.uff.br

Abstract

This paper reports on the adoption of Programming Dojo in undergraduate Java Object-Oriented Programming (OOP) courses across four cohorts at two Brazilian institutions in 2025.1 and 2025.2, with 76 students surveyed anonymously. Two formats were compared: a whole-class format with single-pass active participation (2025.1) and a small-group format with multi-pass participation (2025.2). The structural variable most associated with affective outcomes was participation dose, not audience size: under single-pass participation, engagement collapsed after each student's turn, a relief effect that the small-group format eliminated. Students also rated the Dojo as more useful for appreciating OOP design than for OOP learning in general (Wilcoxon r = .56 and .37 in the 2025.1 cohorts), and reducing the audience shifted the valence of perceived pressure from predominantly negative to predominantly positive in the cohort most affected by the whole-class format. Recommendations for educators adopting collaborative coding practices in OOP courses are discussed.

Keywords

Programming Dojo, Object-Oriented Programming Education, Active Learning, Empirical Study

1 Introduction

The growth of Computer Science and Information and Communication Technologies courses in Brazil underscores the increasing demand for training in these areas. Data from the Brazilian Higher Education Census (published by INEP/MEC [5]) indicate that, in 2024 alone, these courses registered approximately 500,000 new students; yet the accumulated dropout rate remains close to 50%, pointing to structural challenges in Computer Science education, especially in its initial stages [6].

Teaching computer programming is widely recognized as a critical factor behind this scenario. Introductory courses exhibit significant failure and dropout rates, often associated with students' difficulty in developing adequate mental models for problem-solving, gaps in logical-mathematical preparation, and motivational factors [13, 14]. Object-Oriented Programming (OOP) adds a further layer of complexity since it requires higher levels of abstraction (encapsulation, inheritance, polymorphism) and the structuring of solutions around objects and their interactions [8]. The resulting transition from algorithmic to object-oriented thinking is hard for many students and directly affects performance and software outcomes [15].

Active learning methodologies have been widely investigated as alternatives to mitigate these difficulties [3, 22]. Among them, the Coding Dojo stands out as a practice originating in agile software development, especially Extreme Programming (XP), in which participants collaboratively solve programming problems in structured sessions, alternating roles and building solutions incrementally [18]. This dynamic favors technical learning along with communication skills and exposure to different problem-solving strategies, while its rapid feedback loop brings the learning process closer to real software development practice [4].

This paper presents an experience report on the application of Coding Dojo sessions in Object-Oriented Programming with Java courses, offered across four undergraduate cohorts at two Brazilian institutions in 2025.1 and 2025.2. Two structurally different formats were adopted: a whole-class format with single-pass active participation per student and a small-group format with multi-pass participation, allowing a multi-cohort comparison under harmonized materials and a single instructor. We frame the work as an *experience report*: the data are students' self-reported perceptions collected through an anonymous questionnaire, and the quantitative analysis is primarily descriptive, complemented by exploratory non-parametric tests. We do not claim experimental control over the differences between cohorts, and the comparisons we draw are interpreted as associations rather than causal effects (Section 7).

The paper describes the planning, execution, and assessment design of the Dojo across the four cohorts and derives lessons learned and practical recommendations for educators adopting collaborative coding practices in OOP courses. Its main contribution, though, lies in the findings themselves related to a relief effect associated with single-pass participation, channeling the Dojo toward design appreciation rather than first-exposure content acquisition, and a shift in the valence of perceived pressure from negative to positive when the audience shrinks.

2 Theoretical Background and Related Work

2.1 Teaching Programming

Learning to program is regarded as one of the most demanding challenges in undergraduate computing education, as students must master language syntax, develop problem decomposition skills, and construct adequate mental models of program execution. Despite decades of research and pedagogical innovation [14, 17], dropout rates in introductory programming courses remain high across educational systems and institutional contexts [6, 12]. Recurring obstacles include limited prior mathematical preparation, difficulties with control flow and data structures, and motivational factors [13]; on the instructor's side, large classes and heterogeneous student profiles complicate the design of effective learning experiences [16]. In Brazil, structural differences between public and private institutions create distinct learning environments worth considering [9]. A consistent finding across this literature is that approaches engaging students actively (requiring them to produce, discuss, and reflect on code) tend to yield better learning outcomes and higher retention than purely expository instruction [3, 22].

2.2 Teaching Object-Oriented Programming

Beyond the procedural and algorithmic reasoning discussed above, OOP requires a mental model in which computation is organized around objects, their states, and relationships [8] — a transition that is not merely syntactic but demands a level of abstraction many students are not yet equipped to handle [13].

Research has identified a consistent set of conceptual barriers. *Encapsulation*, while syntactically straightforward, is frequently misunderstood semantically, as students often treat access modifiers as bureaucratic formalities rather than a design principle that enforces information hiding and reduces coupling [21]. *Inheritance* is often overused as a code-reuse mechanism, with its semantic implications for type hierarchies overlooked [11]. *Polymorphism* is widely reported as the concept most resistant to consolidation, as it requires reasoning about behavior resolved at runtime [11]. Collectively, these difficulties point to a gap between syntactic competence (writing code that runs) and conceptual competence (designing solutions that are coherent, extensible, and maintainable).

Java, one of the most widely adopted languages for teaching OOP, enforces the paradigm from the start and combines a strict type system with verbose syntax, imposing additional load on novices [2]. A well-documented consequence is the persistence of procedural thinking patterns. Students grounded in structured programming produce solutions that are nominally object-oriented but structurally procedural [8]. This disconnect is difficult to detect solely through automated assessment, which motivates the adoption of collaborative, observable coding practices.

2.3 Active Methods for Teaching and Learning Programming

Active learning is broadly defined as any instructional approach that engages students in meaningful activities requiring them to think about what they are doing, rather than passively receiving information [1]. In computing education, this distinction carries particular weight, since programming competence is acquired through repeated cycles of construction, failure, and reflection; active approaches have consistently been found to produce better outcomes than lecture-based instruction [3, 22]. Investigated strategies include problem-based learning, gamification, and collaborative coding practices such as pair programming and live coding [3, 4, 22]. Collaborative practices bring a dimension particularly relevant to OOP: they make the problem-solving process visible, externalizing reasoning that remains implicit in individual work. Since students can produce syntactically correct code that conceals deep conceptual misunderstandings, formats that require design decisions to be articulated and defended in real time create conditions in which such misunderstandings surface. Among these practices, the Coding Dojo operationalizes collaborative coding within an explicit pedagogical framework.

2.4 Coding Dojo Practices

Coding Dojo is a collaborative programming practice originated in the context of agile methodologies, particularly XP, in which a group collectively solves a programming problem in a structured, non-competitive environment [18]. *Randori* is the most widely adopted format: participants alternate between the roles of **Pilot**, who controls the keyboard and writes the code, and **Copilot**, who guides the Pilot's decisions and monitors the overall direction of the solution. The remaining participants form the audience, who observe, may contribute suggestions when invited, and prepare to take their turn. The defining mechanic is the timed rotation: at fixed intervals (commonly five minutes), the Pilot steps out and rejoins the audience, the Copilot becomes the new Pilot, and a new Copilot is drawn from the audience, so that every participant cycles through both active roles. The solution is built incrementally by a succession of pairs, each inheriting and extending the code left by the previous one, forcing the reasoning behind each design decision to be made explicit so that the next pair can carry it forward. Sessions typically conclude with a retrospective in which participants reflect collectively on what was learned and what could be improved [18]. The two concrete instantiations of this cycle used in our study (a whole-class rotation and a within-group rotation) are detailed in Section 3 and illustrated in Figure 1.

Empirical studies on Coding Dojo in educational contexts have produced generally positive findings. Estácio *et al.* [7] conducted a controlled experiment comparing Randori Dojo and Pair Programming in the acquisition of programming skills and concluded that both practices outperformed individual programming, with fewer code anomalies introduced in Randori sessions and motivation often rated higher in Pair Programming. In the Brazilian educational context, [4] reported that Dojo sessions fostered a collaborative and engaging environment, with increased motivation and a sense of collective ownership of the solution, while also identifying exposure anxiety, the discomfort of writing code in front of peers, as a persistent barrier that instructors must actively manage. The most directly related study is Schoeffel *et al.* [20], who applied Coding Dojo specifically in an OOP course and found that the format can support OOP concept acquisition, particularly through reading and continuing a colleague's code; their study, however, covered a single institution and cohort, and did not examine student perception of specific Dojo dynamics such as role rotation, assessment criteria, or awareness of design quality as distinct constructs. More recently, [19] reported positive effects on motivation, collaboration, and perceived learning in introductory programming courses in Brazil.

Taken together, the literature establishes Coding Dojo as a promising pedagogical strategy. Our findings confirm the generally positive affective and collaborative perceptions reported by [4, 7, 19, 20] and the salience of exposure anxiety noted by [4]. What they add, and what to our knowledge has not been reported together, is the relief effect under single-pass participation, the contrast between a public and a private institution under the same instructor and materials, the channeling of the Dojo toward design appreciation in an OOP setting, and the discussion of assessment criteria, including soft skills such as courage, proactivity, and collaboration.

3 Experience Context and Design

3.1 Institutional and Course Context

The experience was conducted in OOP/Programming courses at two Brazilian higher education institutions in the two semesters of 2025. **UERJ** is a public university; the course "LP2" (Programming Language 2) is part of the Computer Science (CS) undergraduate program and may be taken from the third semester onward. **IBMEC** is a private institution; the course "POO" (Object-Oriented Programming) is offered across multiple programs: from the second semester in the Systems Analysis and Development (SAD) technologist program, and from the third semester onward in Computer Engineering (CE), Software Engineering (SE), and Data Science & Artificial Intelligence (DS&AI) programs. An overview of the cohorts is presented in Table 1.

The same instructor taught all four cohorts. The lecture content, worked examples, homework exercises, set of Dojo problems, and the four assessment criteria were identical across cohorts; what differed was the Dojo *format*, the host institution and its student profile (Table 1; Section 6.6), and, for PRIVATE 2025.1, the distribution of weight among the assessment criteria (Table 2). "LP2" at PUBLIC and "POO" at PRIVATE belong to different programs and curricula, but both are first courses dedicated to the object-oriented paradigm in Java, cover the same syllabus (encapsulation, relationships, inheritance, and polymorphism), and were delivered over a comparable course load; we treat this equivalence of *materials and instructor* (not of institutional curriculum) as the basis for the cross-cohort comparison. Problems followed a progressive-complexity model: 30 minutes for abstraction problems (classes and objects), 40 minutes for relationship problems (dependencies and associations), and 50 minutes for inheritance and polymorphism problems. Dojo sessions were held only after each topic had been introduced and practiced in class and through homework; the Dojo was thus positioned as a consolidation activity, not as the first encounter with each concept.

Table 1: Cohort profiles.

Cohort	Program(s)	Enrolled	Reg. Att.	Resp.	Resp. Rate
PUBLIC 2025.1	CS — 3rd sem+	50	38	20	53%
PRIVATE 2025.1	SAD 2nd sem+; CE/SE/DS&AI 3rd sem+	53	45	30	67%
PUBLIC 2025.2	CS — 3rd sem+	15	11	6	55%
PRIVATE 2025.2	SAD 2nd sem+; CE/SE/DS&AI 3rd sem+	36	30	20	67%

3.2 Dojo Format — 2025.1: Whole-Class Format

In the first semester (2025.1), both cohorts used the classical whole-class Dojo format (Figure 1a). A pilot and a copilot coded at the front of the room, visible to all students. Every five minutes, the copilot assumed the pilot role, the former pilot returned to the audience, and a new copilot was selected by random number generation mapped to the class attendance list. Students who had already participated as both pilot and copilot were excluded from subsequent draws. The session continued until the problem was solved or the time limit for that problem category was reached. In both formats, and unlike the classical Randori described in Section 2.4, in which the audience may be invited to contribute, only the active pilot and copilot were allowed to speak and intervene in the solution: the remaining students — the whole class in 2025.1, the waiting group members in 2025.2 — observed in silence, since the assessment focused on the performance of the two active roles.

Given the class sizes, covering all students required multiple sessions: PUBLIC 2025.1 held 5 Dojo sessions (one per class day), while PRIVATE 2025.1 required 8 sessions, with some days hosting more than one Dojo to accommodate 45 regular attendees. Under this format, each student participated actively as pilot exactly once and as copilot exactly once across the entire semester, for a total of 2 active rounds per student.

3.3 Dojo Format — 2025.2: Small-Group Format

In the second semester (2025.2), the format was redesigned. The class was divided into groups of three to five students, and each group conducted its own Dojo simultaneously. Groups were composed by the instructor, who distributed higher-performing students (based on prior assessments) homogeneously across groups to ensure each group had at least one student capable of supporting peers during problem-solving. Role rotation occurred within the group, not across the class (Figure 1b).

Only two Dojo sessions were held per cohort: one for abstraction problems (30 minutes) and one for inheritance/polymorphism problems (50 minutes). Within each session, students rotated roles multiple times, resulting in each student acting as pilot at least twice and as copilot at least twice per session, for a minimum of 8 active rounds across the semester, a fourfold increase over the 2025.1 format. Figure 1 highlights the differences between the two formats.

3.4 Assessment Design

Dojo participation was assessed subjectively by the instructor according to four criteria, applied consistently across all cohorts: (1) Courage: willingness to participate and engage despite uncertainty; (2) Proactivity/Leadership as pilot: initiative in problem decomposition and coding decisions; (3) Technical understanding: correctness and quality of the produced solution; (4) Collaboration as copilot: quality of guidance and support provided to the pilot. Students knew the four criteria in advance, but did not know how points were distributed among them. The total score and its weight in the final grade varied across institutions, ranging from 20% to 25% of the final grade, as shown in Table 2.

A notable difference across cohorts concerns the weight assigned to the "courage" criterion: 50% of the Dojo score in PRIVATE 2025.1

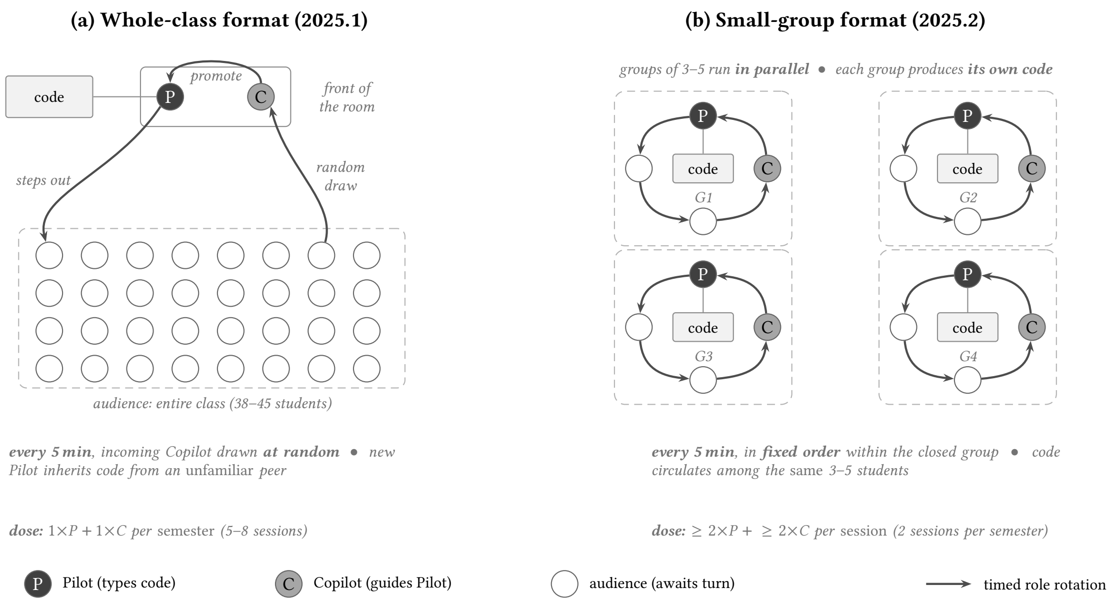

Figure 1: Session dynamics of the two Randori formats: (a) whole-class (2025.1), a single pair rotating in front of the entire class with random selection; (b) small-group (2025.2), parallel groups rotating in a fixed order, each producing its own code.

Table 2: Assessment criteria weights by cohort.

Criterion	2025.1 PUB	2025.1 PRV	2025.2 PUB	2025.2 PRV
(i) Courage / Participation	10%	50%	10%	10%
(ii) Proactivity / Leadership	30%	20%	40%	40%
(iii) Technical Understanding	40%	20%	30%	30%
(iv) Collaboration as Copilot	20%	10%	20%	20%
Total Dojo points	5	10	5	10
Weight in final grade	25%	20%	25%	20%

versus 10% in all other cohorts. This adjustment reflected early resistance among PRIVATE students who feared that Dojo participation might negatively impact their grades; the criterion was rebalanced in subsequent cohorts as students became comfortable with the dynamic and peers from prior cohorts attested that grades had not been adversely affected. PUBLIC students did not raise this concern explicitly in advance.

4 Data Collection and Analysis

During each Dojo session, the instructor observed and took notes on each student's performance in both roles, recording decisions made, the quality of support offered to the pair, and the solution being developed. After the session, the instructor reviewed the final code produced and discussed it with the students as part of the assessment process. These observations informed the subjective participation grade but were not transcribed into a structured performance instrument (e.g., code-quality rubrics or concept-mastery measures); consequently, the present study reports students' *perceptions* of the experience, not objective learning or code-quality outcomes. This scoping is revisited in Sections 7 and 8.

On the last class day of each semester, students were invited to complete an anonymous, voluntary online questionnaire about their perceptions of the Dojo experience. Before accessing the questionnaire, students were presented with a consent statement stating that participation was entirely voluntary, which the instructor reinforced verbally. The questionnaire (Google Forms) consisted of Likert-scale items (1–5), dichotomous yes/no questions, multiple-choice items (select all that apply), and open-ended questions, covering: (1) prior experience with the Dojo format; (2) perceived utility for learning and for understanding OOP specifically; (3) motivational impacts; (4) perceived role difficulty and learning value; (5) fairness of the assessment; and (6) suggestions for improvement. Responses were not linked to individual Dojo scores, as the instrument was anonymous; this protected response sincerity but precluded cross-referencing perceptions with actual grades. Response rates ranged from 53% to 67% of regular attendees (Table 1), yielding 76 valid responses in total.

Quantitative data were analyzed mainly descriptively, with Likert items treated as ordinal (medians, modes, frequencies, and cross-tabulations by cohort and institution). For the two paired Likert items on perceived utility (OOP learning versus OOP design), a within-respondent Wilcoxon signed-rank test was additionally applied as an exploratory confirmation of the direction of the observed gap (Section 5.5); these tests are exploratory and not pre-registered, and no family-wise correction was applied (Section 7).

Open-ended responses were analyzed thematically following an inductive coding procedure. The first author conducted all of the coding by reading the open-ended responses and assigning initial codes that captured the respondents' main messages. Recurring codes were then grouped into higher-level categories, which were refined and merged as new responses were incorporated. The coding unit was the individual response to an open-ended item. The second author did not code independently; instead, she reviewed the complete set of responses together with the codes and themes produced by the first author, and divergences in interpretation were resolved through discussion. Because a single coder produced the codes, we do not report inter-rater agreement statistics; the coding scheme and the coded responses are included in the replication package (Section "Artifact Availability") to support transparency and independent inspection.

5 Results

5.1 Prior Experience

Of the 76 respondents, 75 (98.7%) reported having no prior experience with the Programming Dojo before taking the course, with a single respondent in PRIVATE 2025.2 reporting prior exposure. This near-unanimous lack of prior familiarity strengthens the comparability of the cohorts: the format variations in Section 3 represent first-time Dojo experiences for virtually all participants.

5.2 Reliance on AI Tools

Because generative AI assistants are a salient confound in programming education, a 5-point item asked how much each respondent felt they needed to rely on AI tools to cope with the OOP course, relative to other programming courses. Reliance was modest and differed by institution rather than format: PRIVATE medians of 3.0 in both semesters versus PUBLIC medians of 2.0 and 2.5. We did not instrument whether AI tools were used *during* sessions; the live, timed rotation structurally limits private use, but we treat it as uncontrolled (Section 7).

5.3 General Motivational Impact

The broader motivational and affective consequences of Dojo participation were investigated with two instruments. The first was a categorical item asking respondents whether, after taking part in a Dojo session, they had lost motivation to keep engaging with subsequent Dojos (yes/no). Across the four cohorts, 28 of 76 respondents (36.8%) reported losing motivation after their own participation, with rates of 65.0% in PUBLIC 2025.1 (13/20), 30.0% in PRIVATE 2025.1 (9/30), 16.7% in PUBLIC 2025.2 (1/6), and 25.0% in PRIVATE 2025.2 (5/20). The second was a multiple-selection checklist of ten possible motivational effects (positive engagement responses, anxiety-related responses, and the valence of perceived pressure), with respondents instructed to mark all items that applied. Counts and percentages are reported in Table 3.

Table 3 rewards a closer look. Positive and anxiety-related responses coexist at similar magnitudes at the top of the table: paying more attention in lectures and worrying about meeting instructor expectations were each selected by 52.6% of respondents, and greater concern with code quality (51.3%) sits alongside embarrassment over public mistakes (42.1%). The valence of perceived pressure, in contrast, varied across cohorts. In PUBLIC 2025.1, negative pressure (50%) far exceeded positive pressure (15%); in the three remaining cohorts, positive pressure equaled or exceeded negative pressure (PRIVATE 2025.1: 43% vs. 23%; PUBLIC 2025.2: 67% vs. 17%; PRIVATE 2025.2: 40% vs. 30%).

The number that stands out most, though, is the 65% rate of post-participation motivation loss in PUBLIC 2025.1: more than twice the next-highest rate (PRIVATE 2025.1, 30%) and nearly four times the rate observed under the small-group format at PUBLIC 2025.2 (17%). One qualitative response from that cohort put it bluntly: "After my Dojo, I gradually lost focus on the subject. I sensed this in my classmates too — 'I've already done the Dojo, since this is a Dojo class, I don't need to go.' " We return to these patterns in Section 6.

5.4 Role Perception: Pilot vs. Copilot

Respondents were asked which of the two roles, pilot (typing) or copilot (guiding), they had found more challenging, and which had provided more learning. Pooled across cohorts, 60.5% (46/76) identified the pilot as more challenging and 61.8% (47/76) as more learning-rich. The pattern held in three of four cohorts; PRIVATE 2025.2 inverted on challenge (55% chose copilot), but not on learning, a pattern that returned in Section 6.5. Role exchange was endorsed as beneficial by a clear majority overall (78.9%, 60/76) and in every cohort.

5.5 Perceived Utility

Students rated the perceived utility of the Dojo on two 5-point Likert items: one concerning their learning of OOP in general (the **Learning item**) and one concerning their understanding of the importance of good design and code in OOP (the **Design item**), with 1 = not useful at all and 5 = extremely useful. Responses were treated as ordinal: central tendency is summarized using the median (with interquartile range and mode), and the distribution is grouped into three ordered categories: Low (1–2), Neutral (3), and High (4–5). Table 4 reports these summaries by cohort and item.

The Design item was consistently rated higher than the Learning item, with Design medians equal to or higher than Learning medians in every cohort. Also, all four cohorts converged on a median of 4.0 for Design, with High-rating proportions ranging from 60.0% to 66.7%, whereas for Learning, the range was 30.0% to 83.3%. On top of these sits a semester movement: both items were rated higher in 2025.2 than in 2025.1, most visibly on the Learning item, whose High-rating proportion rose from 30.0% to 83.3% at PUBLIC and from 46.7% to 50.0% at PRIVATE.

Because the same respondents answered both items, we applied the within-subjects Wilcoxon signed-rank test to the paired differences (Design − Learning), reporting also the direction of the difference and the effect size $r = |z|/\sqrt{N}$ (Table 5). In both 2025.1 cohorts the direction favored the Design item and the test was significant, with a large effect at PUBLIC ($p = .012$, $r = .56$) and a medium effect at PRIVATE ($p = .042$, $r = .37$); in both 2025.2 cohorts the direction was more evenly distributed and neither test reached significance. The attenuation of the asymmetry between semesters is examined in Section 6.

5.6 OOP Concepts with Perceived Improvement

Whereas Section 5.5 examined perceived utility in the aggregate, this subsection examines which specific OOP concepts students

Table 3: Self-reported motivational effects of the Dojo, overall and by cohort.

Motivational effect	Type	Total	Overall %	% by cohort			
				PUB. 2025.1	*PRV. 2025.1*	*PUB. 2025.2*	*PRV. 2025.2*
Paid more attention in lectures	*Positive*	40	52.6%	60%	50%	50%	50%
Worried about meeting instructor expectations	*Anxiety*	40	52.6%	55%	53%	67%	45%
Greater concern with code quality	*Positive*	39	51.3%	60%	53%	67%	35%
Solved more exercises at home	*Positive*	37	48.7%	45%	50%	83%	40%
Embarrassed by making mistakes publicly	*Anxiety*	32	42.1%	50%	40%	50%	35%
Practiced more exercises in class	*Positive*	31	40.8%	30%	43%	50%	45%
Afraid of receiving a low grade	*Anxiety*	30	39.5%	35%	40%	33%	45%
Felt pressure as positive	*Positive*	28	36.8%	15%	43%	67%	40%
Felt pressure as negative	*Negative*	24	31.6%	50%	23%	17%	30%
Opportunity to demonstrate knowledge	*Positive*	20	26.3%	20%	20%	33%	40%

Note. $N = 76$; multiple selection permitted, so percentages do not sum to 100.

Table 4: Perceived utility of the Dojo by cohort — medians and grouped frequency distribution.

Cohort	n	Median	IQR	Mode	Frequency distribution (%)		
					Low (1–2)	*Neutral (3)*	*High (4–5)*
Panel A. Perceived utility for OOP learning							
PUBLIC 2025.1	20	3.0	[2.00, 4.00]	3	35.0%	35.0%	30.0%
PRIVATE 2025.1	30	3.0	[3.00, 5.00]	3	16.7%	36.7%	46.7%
PUBLIC 2025.2	6	4.0	[4.00, 4.00]	4	0.0%	16.7%	83.3%
PRIVATE 2025.2	20	3.5	[2.75, 5.00]	5	25.0%	25.0%	50.0%
Panel B. Perceived utility for understanding the importance of good OOP design							
PUBLIC 2025.1	20	4.0	[3.00, 5.00]	4/5	15.0%	25.0%	60.0%
PRIVATE 2025.1	30	4.0	[3.00, 5.00]	5	10.0%	26.7%	63.3%
PUBLIC 2025.2	6	4.0	[3.25, 4.75]	4/5	16.7%	16.7%	66.7%
PRIVATE 2025.2	20	4.0	[3.00, 4.25]	4	10.0%	25.0%	65.0%

Note. IQR = interquartile range $[Q_1, Q_3]$. Row percentages may not sum to 100 due to rounding.

Table 5: Within-cohort comparison of perceived utility for OOP learning versus OOP design — Wilcoxon signed-rank tests.

Cohort	Median		Direction of paired difference			Wilcoxon signed-rank test			Effect size	
	Learn.	*Design*	$D > L$	*Tied*	$L > D$	W	z	p	r	*Interp.*
PUBLIC 2025.1	3.0	4.0	55%	35%	10%	11.0	−2.51	**.012**	.56	*large*
PRIVATE 2025.1	3.0	4.0	33%	57%	10%	18.0	−2.03	**.042**	.37	*medium*
PUBLIC 2025.2	4.0	4.0	17%	50%	33%	2.5	−0.00	1.000	.00	*negligible*
PRIVATE 2025.2	3.5	4.0	40%	35%	25%	30.0	−1.13	.260	.25	*small*

Note. Direction columns report the percentage of respondents rating the Dojo as more useful for design ($D > L$), tied, or more useful for learning ($L > D$). The Wilcoxon test operates on the non-tied pairs (normal approximation with continuity and tie correction); the effect size $r = |z|/\sqrt{N}$ uses the full cohort size N. Interpretation of r: negligible $< .10$, small $\geq .10$, medium $\geq .30$, large $\geq .50$. Bold p-values denote significance at $\alpha = .05$ (two-tailed).

associated with perceived improvement. Respondents were presented with a checklist of eight core OOP concepts, ranging from foundational structural notions (abstraction, encapsulation, inheritance, inter-class relationships) to more advanced topics (direction of relationships, polymorphism, abstract classes, interfaces), and marked all concepts for which they had perceived an improvement in understanding as a result of Dojo participation. Figure 2 reports the proportion endorsing each concept.

A clear gradient is visible in Figure 2. The four most frequently selected concepts are precisely those associated with the early modeling steps of an object-oriented solution: identifying which classes the problem requires, what attributes and responsibilities each class should hold, how classes relate to one another, and how data and behavior should be encapsulated within them. The four least frequently selected are those typically introduced later in the curriculum and required only when problems demand richer structural flexibility.

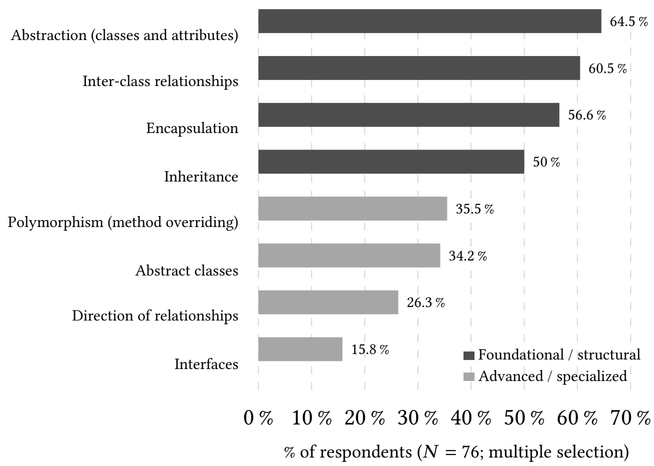

Figure 2: Proportion of respondents reporting perceived improvement in each OOP concept (N = 76; multiple selection permitted). Concepts are ordered by frequency; shading distinguishes foundational/structural concepts (darker) from advanced/specialized ones (lighter).

5.7 Assessment Fairness and Open-Ended Feedback

Overall, 66 of 76 respondents (86.8%) judged the Dojo assessment fair, with affirmative rates ranging from 80% to 100% across cohorts. The open-ended items nuance this aggregate figure in three ways. First, redistribution preferences were *symmetric* around the courage/participation criterion: in the PRIVATE cohorts, where courage weighed 50% of the Dojo score in 2025.1, the recurring request was to move weight from courage toward copilot collaboration or leadership — the direction in which the assessment was in fact modified between semesters (courage: 50% ⟶ 10%; Section 6.4) — whereas in PUBLIC 2025.1, where courage weighed only 10%, three respondents proposed creating or raising an explicit participation point. The two extremes thus converge, from opposite directions, on a moderate participation weight.

Second, the single most frequent suggestion across all cohorts, by a wide margin, was *more time in the active roles*: 31 of 76 respondents (41%) asked for longer turns as pilot or copilot, or for dedicated time to analyze the problem before coding, including 14 of the 20 PUBLIC 2025.1 respondents. The requests targeted the per-turn rotation slot rather than total session length, and they persist under the small-group format reframed accordingly: 2025.2 respondents asked for longer rotation intervals and smaller groups rather than for longer sessions.

Third, four respondents across the two 2025.1 cohorts identified a fairness concern that the closed items do not capture. Since active participation was single-pass while the Dojo problems grew progressively harder, students drawn early in the semester faced easier material than students drawn late. Two smaller threads also recur: four respondents asked for individualized feedback or for the reference solutions to be distributed (the latter adopted in 2025.2), and three asked for the assessment criteria to be restated before each session, likely reflecting absences on the day the criteria were introduced. These threads are taken up in Sections 6.1, 6.4, and 6.7.

6 Discussion

6.1 The Relief Effect: Participation Dose as a Structural Variable

The most consequential structural difference between the whole-class format (2025.1) and the small-group format (2025.2) was not audience size but participation dose. As detailed in Section 3, the same student went from two active rounds across the whole semester to at least eight. This difference in practice frequency, we argue, is the structural variable most plausibly associated with the observed motivational and utility-perception differences between semesters. Because format, number of sessions, and session length all changed together (Section 7), we present this as the most coherent reading of a convergent pattern rather than as an isolated causal effect.

We propose the term **relief effect** to denote a specific failure mode of the whole-class format: once a student has discharged their single active-participation obligation, the perceived stake in subsequent sessions collapses, and engagement declines correspondingly. The post-participation motivation-loss rates reported in Section 5.3 — highest at PUBLIC 2025.1 and nearly four times the rate of its small-group counterpart — are consistent with this pattern. The qualitative testimony quoted in Section 5.3 highlights that the respondent not only reports their own disengagement but explicitly attributes it to a collective behavior among peers, suggesting that the relief effect is a socially observed and collectively rationalized phenomenon, not merely an individual one. The small-group format eliminates the structural precondition for this effect by making every session an active participation event for every student.

The open-ended responses reveal a second failure mode arising from the same structural root. Because participation was single-pass while the Dojo problems followed a progressive-complexity sequence, the moment at which a student was drawn determined the difficulty they faced — and were assessed on. Four respondents across the two 2025.1 cohorts flagged this inequity explicitly (Section 5.7); one, drawn early, reported that the task "was not complicated enough to offer learning." Like the relief effect, this is a consequence of single-pass participation rather than of the Dojo dynamic itself, and the small-group format removes it by the same mechanism: every student cycles through every difficulty level. A related cost raised in both 2025.1 cohorts is class time: five to eight whole-class sessions absorb several lecture days while each student is active in only one of them ("there were too many dojos. . . more lectures and worked examples were missing," PRIVATE 2025.1), an opportunity cost the parallel small-group format does not incur.

A secondary mechanism specific to the whole-class format is the cognitive cost of code inheritance, once each new pilot inherits code written under a different mental model and must continue it without full shared context — a difficulty respondents described explicitly ("picking up a colleague's reasoning in just five minutes. . . is quite a complicated task," PUBLIC 2025.1), even though others valued precisely the practice of continuing someone else's code. In the small-group format, code circulates among the same three to five students, so shared context emerges organically, and the cost of entry is reduced. The coding also delimits the motivation-loss item. The two PRIVATE 2025.2 respondents who reported disengagement attributed it to falling behind on course content, not to the Dojo format.

6.2 Public Exposure and the Valence of Pressure

The reduction in audience size from the entire class to a small peer group also appears to have shifted the affective experience of the Dojo. Expressed as positive-to-negative pressure ratios, the four cohorts correspond to 0.30:1 in PUBLIC 2025.1 (strongly unfavorable), 1.87:1 in PRIVATE 2025.1, 3.94:1 in PUBLIC 2025.2 (strongly favorable), and 1.33:1 in PRIVATE 2025.2. The change at PUBLIC was particularly large: the ratio inverted from 0.30:1 to 3.94:1 between semesters. The PRIVATE cohort, by contrast, showed a small decline between semesters (1.87:1 → 1.33:1), an observation we return to in Section 6.4.

These results are consistent with the challenge–threat appraisal framework derived from Lazarus and Folkman's transactional theory of stress [10]: the same observable performance task is appraised as a challenge when perceived coping resources meet demands, and as a threat when demands exceed them, with opposing motivational consequences. PUBLIC 2025.1 combined maximally exposing conditions (audience of 38, single-pass participation, no participation-based grade floor) with the cohort whose socioeconomic profile (Section 6.6) plausibly lowers the threshold at which appraisal flips to threat. The ratio inversion fits the reading that the small-group format lowered perceived demand enough for the task to be reappraised as a challenge: it does not eliminate the motivational stimulus of observable performance (endorsement of "embarrassed by making mistakes publicly" remains substantial in 2025.2), but it keeps more students on the productive side of the appraisal threshold. Notably, reduced exposure was a demand students themselves voiced in the 2025.1 open-ended feedback — respondents at both institutions asked for group-based or parallel formats, citing anxiety and shyness — and that feedback was an input to the redesign.

6.3 The Dojo as a Vehicle for Design Appreciation

Sections 5.5 and 5.6 converge on the cognitive impact of the Dojo. Dojo was perceived as more useful for understanding the importance of good OOP design than for broader OOP learning, a pattern statistically significant in both 2025.1 cohorts (r = .56 at PUBLIC; r = .37 at PRIVATE). The convergence across two institutions with distinct student populations suggests a feature of the Dojo dynamic itself. A plausible mechanism is that its dialogic conditions (role rotation, public reasoning about code choices, and explicit verbalization of decisions) bring into focus precisely those aspects of programming that are typically invisible in solitary practice: naming, responsibility assignment, cohesion, coupling, and design trade-offs. Learning new OOP content is largely an individual cognitive achievement that lectures or solo practice can advance equally well; learning to appreciate design quality benefits disproportionately from the collaborative articulation of choices that the Dojo enforces, in line with prior reports on pair- and ensemble-based practices [4, 7, 20]. The open-ended responses trace this mechanism directly: one PUBLIC 2025.1 respondent reported being pushed to write cleaner code, "always thinking that whoever comes next needs to understand my reasoning," and a PRIVATE 2025.1 respondent observed that the pair is effectively "obliged" to leave the code understandable so future participants can continue it — design-facing behaviors rather than new content acquisition. The concept-endorsement pattern of Section 5.6 reinforces this reading: the four concepts most associated with perceived improvement (abstraction, inter-class relationships, encapsulation, inheritance) are the ones most directly implicated in *design* decisions.

The asymmetry then attenuates in 2025.2, since neither cohort showed a significant Design–Learning gap. Read alongside the participation-dose argument of Section 6.1, a single Dojo experience appears sufficient to expose students to the value of good design but insufficient for them to attribute their conceptual learning to the practice; repeated participation seems to close that gap. Given the small PUBLIC 2025.2 sample (n = 6) and the concurrent changes in session length and count, we claim only that the more intensive 2025.2 format is associated with a more uniformly positive perception of the Dojo across both utility dimensions.

6.4 Assessment Weight and the PRIVATE Trajectory: A Plausible Co-Factor

PRIVATE 2025.1 operated under the whole-class format, the one associated with strongly negative outcomes at PUBLIC 2025.1, yet exhibited a favorable pressure ratio (1.87:1), lower motivation loss (30% vs. 65%), and a smaller design-over-learning asymmetry. One contributing difference may be the 50% weight of the courage/participation criterion, an implicit grade floor under which sheer engagement guaranteed half the assessment, plausibly buffering students against the appraisal threat the whole-class format induces. The institutions differ along several other dimensions (Section 6.6) that could account for part or all of the gap, and the present data cannot adjudicate among them.

In PRIVATE 2025.2, the courage criterion was reduced to 10% concurrently with the format change, and this was the only cohort showing an inter-semester decline in the pressure ratio (1.87:1 → 1.33:1) and an increase in fear of a low grade (40% → 45%). One reading is that the format change reduced anxiety while the increased performance demands partially offset that benefit, format and assessment weight acting as co-dependent affective levers; a simpler alternative is that PRIVATE 2025.1 already operated near a favorable ceiling. The data do not discriminate between these readings; we treat the co-dependence hypothesis as a direction for future work in which format and weight are varied independently.

The redistribution preferences outlined in Section 5.7 provide independent support for a moderate participation weight. Students under the 50% courage weight asked to reduce it, while students under the 10% weight asked to raise it — the two cohorts converging, from opposite directions, on the intermediate design we recommend in Section 6.7.

6.5 Trade-Offs and Residual Challenges of the Small-Group Format

While the small-group format outperformed the whole-class format on most indicators, the qualitative data highlighted a reduced audience incentive. In the whole-class format, any student may be called as the next pilot or copilot at any time, sustaining attention; in the small-group format, with rotation predictable within a closed group, members awaiting their turn showed signs of disengagement. Two respondents in PRIVATE 2025.2 noted that the third and fourth members of their groups were often idle, and one observed that the stronger student tended to dominate the pilot role, reducing the copilot role to relatively passive instruction-following. The students themselves converged on the mitigation we propose in Section 6.7: four 2025.2 respondents suggested letting the waiting members contribute ideas or splitting tasks so that everyone works simultaneously, and one PUBLIC 2025.2 respondent singled out the mandatory silence outside the active roles as the hardest part of an otherwise highly valued dynamic.

These observations also illuminate the role-perception inversion of Section 5.4, in which PRIVATE 2025.2 was the only cohort to identify the copilot role as more challenging. As a conjecture for future investigation: the copilot must think *through* another person — sustaining a model of the solution, translating it into calibrated instructions, and monitoring execution — a load that can exceed piloting itself, where a single coherent mental model is externalized directly into code.

6.6 Institutional Differences and Generalizability

The contrasting trajectories of the PUBLIC and PRIVATE cohorts warrant discussion. PUBLIC improved substantially between semesters across most indicators (notably motivation loss, 65% → 17%, and the Learning High-rating proportion, 30% → 83%), while PRIVATE's improvement was smaller and mixed. One plausible reading is that PUBLIC 2025.1 was operating closer to a negative threshold and had more room to gain from the format change, while PRIVATE 2025.1 was already in a more favorable affective region. Whether this differential margin reflects the protective effect of the high courage weight, the informal between-cohort transmission of reassurance among PRIVATE students, baseline institutional differences, or some combination, cannot be settled with the present data.

Institutional context also seems to influence the feedback. PUBLIC students are predominantly from lower-income backgrounds, often dependent on scholarships or part-time work, while PRIVATE students come mostly from more financially comfortable families in programs that cultivate an entrepreneurial professional identity — differences that plausibly shape baseline performance anxiety and familiarity with presentation-oriented practices. The present design did not collect respondent-level demographics to test these conjectures; future replications should treat institutional type, socioeconomic background, and program orientation as explicit contextual variables.

6.7 Lessons Learned

What worked. Moving from the whole-class to the small-group format raised the participation dose from two active rounds per student to at least eight, and coincided with the sharpest improvements recorded: the collapse of post-participation motivation loss and the inversion of the pressure ratio at PUBLIC (Sections 6.1–6.2). Holding the instructor, materials, and Dojo problems constant across cohorts also proved a sound design choice: it is what allows the cross-cohort contrasts to be attributed to format and institution rather than to content.

What went wrong, and how we adjusted. At PRIVATE 2025.1, students initially resisted the Dojo, fearing it would depress their grades; the courage/participation criterion was set at 50% of the Dojo score, functioning as an implicit grade floor that appears to have buffered early anxiety (Section 6.4). As students grew comfortable and later cohorts could be reassured by peers, courage was reduced to 10% and the weight redistributed to leadership and technical correctness. The move to the small-group format itself was informed by the 2025.1 open-ended feedback, in which respondents at both institutions asked for reduced public exposure and for group-based or parallel formats. The small-group format then surfaced a problem absent from the whole-class format: with rotation predictable inside a closed group, audience members disengaged, and stronger students sometimes dominated the pilot role (Section 6.5).

What we would do differently. We would keep the small-group format, since groups run in parallel and each student still accumulates several active rounds per session; however, we would assign explicit active roles to non-piloting members (e.g., note-taker, standards reviewer, test-case challenger) to counter idle-audience disengagement, and retain a modest participation weight in the assessment. Guided by the open-ended feedback, we would calibrate the per-role rotation slot and the initial problem-analysis time to problem complexity — more time in the active roles was by far the most frequent request (31 of 76 respondents) — complement the collective post-session correction with brief individualized feedback, keep distributing the reference solutions, and restate the assessment criteria at the start of each session. We would also instrument objective outcomes (code-quality rubrics, concept checks) alongside perceptions, so that the perceived design benefit reported here can be tested against measured ability.

7 Threats to Validity

We highlight the threats most consequential for interpretation, following the framework of Wohlin *et al.* [23]; given the non-experimental, single-instructor, anonymous-survey design of this experience report, several are inherent to the design and not mitigable.

Construct validity. Every dependent variable is a *self-reported perception*, operationalized by a single questionnaire item, which precludes internal-consistency estimation and conflates each construct with item-specific wording. The design includes no objective measure of learning or code quality, and the instructor's in-session observations were neither transcribed into a structured instrument nor linkable to the anonymous responses. Claims about the Dojo's value for design appreciation should therefore be read as claims about *perceived* value, partially corroborated by the convergent concept-endorsement pattern of Section 5.6; whether the perception corresponds to measurable gains is an open question for future work.

Internal validity. The 2025.1 vs. 2025.2 comparison confounds the format change with three concurrent changes: number of sessions (5–8 → 2), participation dose (2 → ≥ 8 active rounds), and, at PRIVATE, the courage weight (50% → 10%); instructor maturation between semesters is a further uncontrolled factor. The participation-dose interpretation of Section 6 is therefore the most plausible reading of a convergent pattern, not a causal claim. Anonymity also precludes linking perceptions to actual Dojo grades, so we cannot determine whether motivation loss is concentrated among lower-performing students.

Conclusion validity. Cohorts are small (n = 20, 30, 6, 20; PUBLIC 2025.2 especially), the analyses were exploratory and not preregistered, and no family-wise correction was applied; the design-over-learning conclusion rests on the convergent direction of the paired differences and on the concept-endorsement alignment rather than on the p-values alone. Response rates of 53% to 67% imply possible non-response bias of indeterminable direction, so absolute rates describe respondents rather than full cohorts; the central comparisons, however, are within-respondent and relative across cohorts under a constant instrument, and thus less sensitive to a uniform self-selection effect.

External validity. The study covers two Brazilian institutions and a single instructor, leaving open whether the relief effect is a structural property of the whole-class format or partly an artefact of facilitation style. The problems were Java OOP problems in which design judgment was itself a learning objective, so the design-over-learning finding is not directly supported for settings where syntactic mastery is the primary goal; the affective response to public performance is also culturally inflected, as the within-Brazil PUBLIC/PRIVATE contrast itself suggests.

8 Conclusion

This experience report compared two Dojo configurations, a whole-class format with single-pass participation (2025.1) and a small-group format with multi-pass participation (2025.2), across four cohorts at two Brazilian institutions under a single instructor and harmonized materials. The clearest result concerns participation dose. It was dose, not audience size, that best tracked affective outcomes: single-pass participation produced a relief effect (once the participation obligation was discharged, engagement collapsed), which the small-group format eliminated by making every session active for every student. The shift coincided with sharp reductions in motivation loss and a reversal of the pressure ratio at PUBLIC.

The Dojo also turned out to function, in both 2025.1 cohorts, primarily as a vehicle for design appreciation, not first-exposure content acquisition. We read this as a property of its dialogic conditions (role rotation, public reasoning about code choices, and verbalization of design trade-offs), which make visible the dimensions of programming that solitary practice leaves implicit.

The PRIVATE trajectory, finally, suggests that format and assessment weight may act as co-dependent affective levers: a high weight on participation in the grade can substitute for some of the exposure-reducing benefits of a small-group format. The present data cannot establish this co-dependence. Practitioners redesigning format and assessment together should nonetheless monitor affective outcomes, since the two adjustments may not combine additively.

We therefore recommend the small-group format with at least two rotations per role per session whenever cohort size precludes multiple whole-class rounds per student; because groups run in parallel, this format scales to large classes while preserving a high participation dose. To mitigate the residual challenges of Section 6.5, the assessment should retain some weight on participation, and non-piloting group members should be assigned explicit active roles. Adopters should position each Dojo as a consolidation activity *after* the corresponding topic has been introduced and practiced, size groups at three to five students, calibrate the per-role rotation time to problem complexity, complement the closing collective correction with individualized feedback and shared reference solutions. The central limitation to address in future work is the reliance on perceptions: subsequent studies should pair the questionnaire with objective indicators (code-quality rubrics, concept checks, or graded artifacts) and vary participation dose independently of audience size, session count, and assessment weight, replicating across institutional contexts.

Artifact Availability

The replication package is publicly available at https://doi.org/10.6084/m9.figshare.33021725.v4 and includes the questionnaire, the 76 anonymized responses, a description of each question, and the qualitative coding scheme with the coded open-ended responses (Section 4).

Acknowledgements

Generative AI was used as an assistive tool for text and image refinement and data cross-checking. The intellectual content and all conclusions are entirely the authors' own.

References

[1] Charles C Bonwell and James A Eison. 1991. *Active learning: Creating excitement in the classroom. 1991 ASHE-ERIC higher education reports.* ERIC.
[2] Neil CC Brown, Pierre Weill-Tessier, Maksymilian Sekula, Alexandra-Lucia Costache, and Michael Kölling. 2023. Novice use of the Java programming language. *ACM Transactions on Computing Education* 23, 1 (2023), 1–24. doi:10.1145/3551393
[3] Diana-Margarita Córdova-Esparza, Julio-Alejandro Romero-González, Karen-Edith Córdova-Esparza, Juan Terven, and Rocio-Edith López-Martínez. 2024. Active learning strategies in computer science education: A systematic review. *Multimodal Technologies and Interaction* 8, 6 (2024), 50. doi:10.3390/mti8060050
[4] Peterson Luiz da R. Rodrigues, Luiz Paulo Franz, Jean Felipe P Cheiran, João Pablo S da Silva, and Andrea S Bordin. 2017. Coding Dojo as a transforming practice in collaborative learning of programming: an experience report. In *Proceedings of the XXXI Brazilian Symposium on Software Engineering*. 348–357. doi:10.1145/3131151.3131180
[5] Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira | Inep. 2025. Censo da Educação Superior - 2024. Retrieved April, 2026 from https://www.gov.br/inep/pt-br/areas-de-atuacao/pesquisas-estatisticas-e-indicadores/censo-da-educacao-superior/resultados
[6] Rafael Peixoto de Moraes, Valéria Franklin da Costa, and Ricardo EP Scholz. 2022. Mapeamento sistemático do ensino introdutório de programação nos ensinos técnico e superior no Brasil. *Revista Brasileira de Informática na Educação* 30 (2022), 628–647.
[7] Bernardo Estácio, Roberto Oliveira, Sabrina Marczak, Marcos Kalinowski, Alessandro Garcia, Rafael Prikladnicki, and Carlos Lucena. 2015. Evaluating collaborative practices in acquiring programming skills: Findings of a controlled experiment. In *2015 29th Brazilian Symposium on Software Engineering*. IEEE, 150–159. doi:10.1109/SBES.2015.24
[8] Luz E Gutiérrez, Carlos A Guerrero, and Héctor A López-Ospina. 2022. Ranking of problems and solutions in the teaching and learning of object-oriented programming. *Education and Information Technologies* 27, 5 (2022), 7205–7239. doi:10.1007/s10639-022-10929-5
[9] Lucas Krzyzanowski, Carlos Beleti Jr, Robertino Santiago Jr, and Raimundo Alberto Tostes. 2019. Ensino de programação: um estudo preliminar nos cursos de licenciatura em Computação no Brasil. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, Vol. 8. 21.

[10] Richard S Lazarus and Susan Folkman. 1984. *Stress, appraisal, and coping.* Springer, New York.

[11] Neomi Liberman, Catriel Beeri, and Yifat Ben-David Kolikant. 2011. Difficulties in learning inheritance and polymorphism. *ACM Transactions on Computing Education (TOCE)* 11, 1 (2011), 1–23. doi:10.1145/1921607.1921611

[12] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A Becker, Michail Giannakos, Amruth N Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, et al. 2018. Introductory programming: a systematic literature review. In *Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education.* 55–106. doi:10.1145/3293881.3295779

[13] Rodrigo Pessoa Medeiros, Taciana Pontual Falcão, and Geber Lisboa Ramalho. 2020. Ensino e aprendizagem de introdução à programação no ensino superior brasileiro: Revisão sistemática da literatura. In *Workshop Sobre Educação em Computação (WEI).* SBC, 186–190.

[14] Rodrigo Pessoa Medeiros, Geber Lisboa Ramalho, and Taciana Pontual Falcão. 2019. A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE Transactions on Education* 62, 2 (2019), 77–90. doi:10.1109/TE.2018.2864133

[15] André Menolli and Bruno Strik. 2025. Educational Insights from Code: Mapping Learning Challenges in Object-Oriented Programming through Code-Based Evidence. In *Anais do XXXIX Simpósio Brasileiro de Engenharia de Software* (Recife/PE). SBC, Porto Alegre, RS, Brasil, 544–554. doi:10.5753/sbes.2025.11095

[16] Ceres Germanna Braga Morais. 2022. *Ensino e aprendizagem de programação: estudo de caso no Ensino Superior.* Ph. D. Dissertation. Universidade do Minho (Portugal).

[17] Anthony Robins, Janet Rountree, and Nathan Rountree. 2003. Learning and teaching programming: A review and discussion. *Computer science education* 13, 2 (2003), 137–172. doi:10.1076/csed.13.2.137.14200

[18] Danilo Toshiaki Sato, Hugo Corbucci, and Mariana Vivian Bravo. 2008. Coding dojo: An environment for learning and sharing agile practices. In *Agile 2008 Conference.* IEEE, 459–464. doi:10.1109/AGILE.2008.11

[19] Adriana Paula Zamim Scherer and Filipo Novo Mór. 2020. Uso da técnica Coding DOJO em aulas de programação de computadores. In *Workshop sobre Educação em Computação (WEI).* SBC, 6–10.

[20] Pablo Schoeffel, Daniel Felipe Rosa, and Raul Sidnei Waslawick. 2016. Um experimento do uso de coding dojo na aprendizagem de programação orientada a objetos. *iSys-Journal of Information Systems* 9, 2 (2016).

[21] Ronit Shmallo and David Ginat. 2025. Teaching Object-Oriented Threshold Concepts Using the Learning-from-Errors Approach. *ACM Transactions on Computing Education* 26, 1 (2025), 1–25. doi:10.1145/3770064

[22] Fábio Aiub Sperotto and Claudio Cleverson de Lima. 2025. Metodologias Ativas na Aprendizagem da Programação Orientada a Objetos. *Revista Thema* 24, 2 (2025).

[23] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, Anders Wesslén, et al. 2012. *Experimentation in software engineering.* Vol. 236. Springer. doi:10.1007/978-3-642-29044-2