

# Java-Judge-OO: Uma Ferramenta Educacional para Avaliação Automatizada de Programação Orientada a Objetos em Java

Ramon Mayor Martins<sup>1</sup>

<sup>1</sup>Área de Telecomunicações – Instituto Federal de Santa Catarina (IFSC),  
São José – SC – Brazil

ramon.mayor@ifsc.edu.br

**Resumo.** *A avaliação de códigos em disciplinas de Programação Orientada a Objetos (POO) representa um desafio significativo devido à complexidade e subjetividade envolvidas na análise de conceitos como encapsulamento, herança e polimorfismo. O Java-Judge-OO é uma ferramenta educacional que automatiza a avaliação de códigos Java utilizando uma rubrica estruturada, oferecendo feedback consistente e objetivo. Desenvolvida com foco na justiça e padronização do processo avaliativo, a ferramenta implementa métricas específicas para análise de elementos essenciais da POO. O sistema apresenta uma interface web acessível, hospedada no Hugging Face Spaces, e utiliza critérios bem definidos a partir de uma rubrica de pontuação, incluindo aspectos como definição de classes, métodos, atributos, encapsulamento, herança e polimorfismo.*

## 1. Introdução

A avaliação de códigos em disciplinas de Programação Orientada a Objetos (POO) apresenta desafios únicos relacionados à complexidade dos conceitos e à subjetividade na análise das implementações (Krismadinata, 2023). Uma avaliação justa e consistente requer a consideração de múltiplos aspectos do paradigma orientado a objetos, desde a estruturação básica de classes até conceitos avançados como herança e polimorfismo. Nesse contexto, o Java-Judge-OO foi desenvolvido para oferecer um método sistemático e justo de avaliação, baseado em uma rubrica clara e objetiva.

Diferentemente das abordagens baseadas em *Large Language Models (LLMs)* para análise de código-fonte, que podem apresentar resultados variáveis devido à sua natureza probabilística (Lyu et al., 2024), o Java-Judge-OO implementa um sistema determinístico com rubrica estruturada que busca garantir consistência avaliativa e transparência do processo. Seu diferencial também reside na capacidade de personalização pelo educador, que pode ajustar diretamente no código-fonte as ponderações específicas para cada conceito de POO, adaptando a ferramenta aos objetivos pedagógicos específicos - uma flexibilidade raramente encontrada em avaliadores automáticos convencionais (Keuning et al., 2018).

Este resumo apresenta a ideia da ferramenta, sua metodologia pedagógica e um relato de primeira impressão a partir dos resultados obtidos em aplicações com estudantes.

## **2. Metodologia**

O desenvolvimento do Java-Judge-OO seguiu uma abordagem centrada em critérios específicos de avaliação, implementando uma rubrica que considera os principais aspectos da POO. A rubrica está pautada em critérios essenciais com o valor de 60 pontos distribuídos entre:

- Classes e Objetos (20 pontos): avalia definição e uso de classes, quantidade de objetos instanciados e organização geral
- Métodos (20 pontos): analisa definição, organização e propósito dos métodos
- Atributos (20 pontos): examina presença, tipos adequados e organização dos campos.

E pautado em critérios bonificados com o valor de 40 pontos distribuídos entre:

- Encapsulamento (10 pontos): verifica uso de modificadores e getters/setters
- Herança (10 pontos): analisa hierarquia e reutilização de código
- Polimorfismo (10 pontos): avalia sobrescrita de métodos
- Abstração (10 pontos): examina uso de classes abstratas e interfaces.

## **3. Implementação**

A ferramenta foi desenvolvida como uma aplicação web hospedada no Hugging Face Spaces, oferecendo uma interface intuitiva para submissão e avaliação de códigos Java. O sistema implementa: análise sintática e estrutural do código fonte, verificação automatizada dos critérios da rubrica, geração de feedback detalhado com pontuações específicas e interface web responsiva e acessível globalmente.

A ferramenta foi construída em Python e a interface com Gradio, permitindo execução eficiente e em navegadores sem necessidade de instalação. Além disso, o código-fonte está disponível no GitHub, possibilitando personalizações e extensões por educadores e desenvolvedores.

## **4. Níveis de Proficiência**

A ferramenta estabelece quatro níveis de proficiência baseados na pontuação total: Excelente: 90-100 pontos; Bom: 75-89 pontos; Satisfatório: 60-74 pontos e Necessita Melhorias: 0-59 pontos. Estes níveis orientam o usuário de forma primária quanto a qualidade do seu código.

## **5. Discussão**

Aplicações foram conduzidas com um total de 40 estudantes das disciplinas de Programação Orientada a Objetos dos cursos de Análise e Desenvolvimento de Sistemas e Engenharia de Telecomunicações no Instituto Federal de Santa Catarina. A ferramenta teve importância para os estudantes, pois conseguiram com o avaliador ter uma percepção da qualidade de seus códigos com um sistema fixo baseado em expressões regulares e condicionais, livre da parcialidade humana e da aleatoriedade das inteligências artificiais (ex. ChatGPT, Llama, DeepSeek, etc.) Para o professor foi uma ferramenta importante para balancear a avaliação manual.

## 5. Conclusão

O Java-Judge-OO representa uma solução para o desafio da avaliação em disciplinas de POO, oferecendo um método sistemático e justo através de uma rubrica bem definida. A ferramenta não apenas padroniza o processo avaliativo, mas também fornece feedback construtivo para o desenvolvimento dos estudantes.

Como trabalhos futuros, planeja-se a avaliação da rubrica de pontuação, e o teste com um painel de especialista para obter feedbacks para melhorias.

## References

- Keuning, Hieke; Jeuring, Johan; Heeren, Bastiaan (2019). A Systematic Literature Review of Automated Feedback Generation for Programming Exercises. ACM Transactions on Computing Education, v. 19, n. 1, art. 3, p. 1-43.
- Krismadinata E., Jama, J. e Mulya, R (2023) "A Systematic Literature Review of Teaching and Learning on Object-Oriented Programming Course," International Journal of Information and Education Technology vol. 13, no. 2, pp. 302-312.
- Lyu, Michael R.; Ray, Baishakhi; Roychoudhury, Abhik; TAN, Shin Hwei; Thongtanunam, Patanamon. (2024) "Automatic Programming: Large Language Models and Beyond". ACM Transactions on Software Engineering and Methodology. <https://doi.org/10.1145/3708519>.

## YouTube link

Video Java-Judge-OO (2024). Disponível em: <https://youtu.be/JLZGmqKjBIg>

## Material

Java-Judge-OO (2025). <https://huggingface.co/spaces/rmayormartins/java-judge-oo>

Rubrica do Java-Judge-OO (2025).

[https://huggingface.co/spaces/rmayormartins/java-judge-oo/blob/main/assets/rubric\\_table.PNG](https://huggingface.co/spaces/rmayormartins/java-judge-oo/blob/main/assets/rubric_table.PNG)

Github do projeto Java-Judge-OO (2025).

<https://github.com/rmayormartins/SPACE-PY-java-inspector-judge/tree/main/java-judge-oo>