

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
CAMPUS TIMÓTEO**

Marco Túlio Valeriano Santana de Oliveira

**ANÁLISE DE OBJETOS DE APRENDIZAGEM ATRAVÉS DO  
MÉTODO LORI PARA O ENSINO E APRENDIZAGEM DE  
PROGRAMAÇÃO NA EDUCAÇÃO SUPERIOR**

**Timóteo**

**2023**

**Marco Túlio Valeriano Santana de Oliveira**

**ANÁLISE DE OBJETOS DE APRENDIZAGEM ATRAVÉS DO  
MÉTODO LORI PARA O ENSINO E APRENDIZAGEM DE  
PROGRAMAÇÃO NA EDUCAÇÃO SUPERIOR**

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Viviane Cota Silva

Timóteo

2023

**ESTUDO DE OBJETOS DE APRENDIZAGEM APLICÁVEIS NO  
ENSINO DE PROGRAMAÇÃO DE COMPUTADORES NA  
GRADUAÇÃO**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, como requisito parcial para obtenção do título de Engenheiro de Computação.

Trabalho aprovado. Timóteo, 13 de Julho de 2023:



Prof. Dr. Viviane  
Cota Silva  
Orientador



Prof. Dr. Lucas  
Pantuza  
Professor Convidado



Prof. Luciano  
Moreira  
Professor Convidado

.

.

.

Dedico a minha mãe por tudo que ela fez e faz por mim

# Agradecimentos

Agradeço à minha orientadora Viviane, por não ter desistido de mim mesmo nos momentos mais difíceis e ter tido toda a paciência para me acompanhar e me guiar nessa jornada.

À Ingrid, minha namorada, por ter sido tão persistente em me fazer seguir em frente. Por toda compreensão e por sempre desejar o melhor para mim.

Aos meus amigos que levo do CEFET para a vida: Bruno Roldão, Bruno Durso, Andre, Fabi, Morandi, Rafael. Vocês fizeram essa árdua jornada mais divertida. Agradeço pela presença em todos os momentos, sejam bons ou ruins.

Ao Lucas Pantuza pela leitura deste trabalho e por fornecer importantes contribuições e sugestões.

*“Os grandes navegadores  
devem sua reputação aos temporais e tempestades”.*  
*Epicuro*

# Resumo

Esse trabalho buscou fazer uma revisão bibliográfica de avaliações de objetos de aprendizagem utilizados em disciplinas introdutórias de programação, sendo alinhada com a grade curricular com as matérias introdutórias presentes no curso de engenharia de computação da faculdade CEFET-MG Campus Timóteo. Com essa revisão feita foram escolhidos três objetos de aprendizado para serem avaliados utilizando o método de avaliação LORI para que fosse feita uma comparação entre os resultados da avaliação da literatura e os resultados da avaliação utilizando o LORI. Os objetos de aprendizagem escolhidos tiveram suas funcionalidades explicadas e foram levantados os pontos fortes e pontos fracos que os diversos autores apresentaram na literatura, seja através de estudos de caso, observação empírica ou avaliações teóricas. A avaliação utilizando o modelo LORI seguiu a cartilha do usuário proposta pelo método, construindo tabelas que avaliam oito aspectos de acordo com um determinado número de estrelas. Após a comparação das avaliações foram levantados os limites, para que fosse entendido até onde os pontos fracos interferiam em sua utilização, e possibilidades que esses objetos de aprendizagem traziam de acordo com a proposta de cada um, para que ficasse de forma resumida a melhor forma de utilização para o autor.

**Palavras-chave:** bibliographic review, learning objects, LORI, evaluation, limits and possibilities.

# Abstract

This work sought to carry out a bibliographical review of evaluations of learning objects used in introductory programming disciplines, being aligned with the curriculum with the introductory subjects present in the computer engineering course at CEFET-MG Campus Timóteo. With this review done, three learning objects were chosen to be evaluated using the LORI evaluation method so that a comparison could be made between the results of the literature evaluation and the results of the evaluation using the LORI. The chosen learning objects had their functionalities explained and the strengths and weaknesses that the various authors presented in the literature were raised, either through case studies, empirical observation or theoretical evaluations. The evaluation using the LORI model followed the user manual proposed by the method, building tables that evaluate eight aspects according to a certain number of stars. After comparing the evaluations, the limits were raised, so that it was understood to what extent the weaknesses interfered in their use, and possibilities that these learning objects brought according to the proposal of each one, so that the best way was summarized. of use for the author.

**Keywords:** domain analysis, facet analysis, information retrieval, enterprise information.



# Lista de ilustrações

Figura 1 – Tela de Criação de Classe . . . . .	38
Figura 2 – Menu de Controle CodeCombat . . . . .	39
Figura 3 – Código do Aluno e Solução Fornecida . . . . .	40
Figura 4 – Menu de Resolução do Desafio . . . . .	41
Figura 5 – Código Fonte de uma Classe no BlueJ . . . . .	43
Figura 6 – Interface Gráfica BlueJ . . . . .	43
Figura 7 – Interface de Resolução . . . . .	45

# Lista de tabelas

Tabela 1 – Atores e Funções na Metodologia INTERA . . . . .	20
Tabela 2 – Tabela de Exemplo: Avaliação LORI . . . . .	26
Tabela 3 – Pontos Fortes e Fracos: BlueJ . . . . .	47
Tabela 4 – Pontos Fortes e Fracos: CodeCombat . . . . .	48
Tabela 5 – Pontos Fortes e Fracos: CodeWars . . . . .	49
Tabela 6 – Avaliação do OA CodeCombat . . . . .	50
Tabela 7 – Avaliação do OA BlueJ . . . . .	52
Tabela 8 – Avaliação do OA CodeWars . . . . .	54
Tabela 9 – Tabela Comparativa: Pontos Fortes . . . . .	56
Tabela 10 – Tabela Comparativa: Pontos Fracos . . . . .	57

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Problema</b>	<b>12</b>
<b>1.2</b>	<b>Objetivos</b>	<b>12</b>
1.2.1	Objetivo Geral	12
1.2.2	Objetivos Específicos	12
<b>1.3</b>	<b>Justificativa</b>	<b>13</b>
<b>2</b>	<b>REVISÃO DE LITERATURA</b>	<b>14</b>
<b>2.1</b>	<b>Objetos de Aprendizagem</b>	<b>14</b>
2.1.1	Definição	14
2.1.2	Agrupamento	15
2.1.2.1	Tipos	15
2.1.2.2	Divisão em Grupos	15
2.1.2.3	Classificação	16
2.1.3	Características	16
2.1.4	Modelos Padrões	18
2.1.4.1	Padrão SCORM	18
2.1.4.2	ISO/IEC 9126	18
2.1.4.3	Metodologia INTERA	19
2.1.4.4	Outras Metodologias	20
2.1.5	Repositórios	20
<b>2.2</b>	<b>OAS no Ensino de Programação</b>	<b>21</b>
<b>2.3</b>	<b>Método de Avaliação de um OA: LORI</b>	<b>24</b>
2.3.1	Descrição	25
2.3.2	Avaliação	25
2.3.3	Aplicação da Avaliação	25
2.3.3.1	Qualidade do Conteúdo	27
2.3.3.2	Meta de Aprendizagem	27
2.3.3.3	Feedback e Adaptação	28
2.3.3.4	Motivação	28
2.3.3.5	Design	29
2.3.3.6	Usabilidade	29
2.3.3.7	Acessibilidade	30
2.3.3.8	Conformidade com os Padrões	31
<b>3</b>	<b>METODOLOGIA</b>	<b>32</b>
<b>3.1</b>	<b>Disciplinas do CEFET-MG</b>	<b>32</b>
3.1.1	Programação de Computadores I	33
3.1.2	Programação de Computadores II	33

3.1.3	Algoritmo e Estruturas de Dados I . . . . .	34
3.1.4	Algoritmo e Estruturas de Dados II . . . . .	35
<b>3.2</b>	<b>Etapas da Pesquisa . . . . .</b>	<b>36</b>
<b>3.3</b>	<b>Objetos de Aprendizagem Escolhidos . . . . .</b>	<b>36</b>
<b>4</b>	<b>DESENVOLVIMENTO . . . . .</b>	<b>37</b>
<b>4.1</b>	<b>OA's Escolhidos . . . . .</b>	<b>37</b>
4.1.1	Motivo da Escolha . . . . .	37
4.1.2	Descrição dos OA's . . . . .	37
4.1.2.1	CodeCombat . . . . .	37
4.1.2.1.1	Plataforma para Professores . . . . .	39
4.1.2.1.2	Plataforma para Alunos . . . . .	40
4.1.2.2	BlueJ . . . . .	41
4.1.2.2.1	Interface Gráfica . . . . .	42
4.1.2.3	CodeWars . . . . .	44
4.1.2.3.1	Ambiente . . . . .	44
<b>4.2</b>	<b>Conceitos Trabalhados . . . . .</b>	<b>45</b>
<b>4.3</b>	<b>Avaliações dos OAs . . . . .</b>	<b>46</b>
4.3.1	Avaliação Literatura . . . . .	46
4.3.1.1	BlueJ . . . . .	46
4.3.1.2	CodeCombat . . . . .	47
4.3.1.3	CodeWars . . . . .	49
4.3.2	Avaliação dos OAs: Método LORI . . . . .	49
4.3.2.1	CodeCombat . . . . .	50
4.3.2.2	BlueJ . . . . .	51
4.3.2.3	CodeWars . . . . .	53
4.3.3	Limites e Possibilidades . . . . .	54
4.3.3.1	BlueJ . . . . .	54
4.3.3.2	CodeCombat . . . . .	55
4.3.3.3	CodeWars . . . . .	55
4.3.4	Tabela Comparativa . . . . .	56
<b>5</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>59</b>
<b>6</b>	<b>TRABALHOS FUTUROS . . . . .</b>	<b>61</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>62</b>

# 1 Introdução

O início da aprendizagem de programação de computadores é comumente marcado por relatos de dificuldade por parte dos estudantes. Segundo Gomes, Henriques e Mendes (2008), a dificuldade reside em compreender e aplicar certos conceitos abstratos de programação e na aplicação de noções básicas na resolução de problemas. Normalmente, o estudante ingressa em um curso de graduação na área da computação sem possuir qualquer experiência em programação de computadores ou lógica computacional, e ao se deparar com um ritmo de estudo acelerado com pouco espaço para que os professores atendam às dificuldades individuais e com uma metodologia de ensino tradicional baseada na apresentação de elementos em um quadro e a necessidade de abstração desses elementos para formular a relação lógica entre eles, surge a dificuldade no aprendizado da programação.

Gomes (2010) diz que, independentemente da linguagem de programação usada, existem várias descrições de casos de insucesso em disciplinas introdutórias de programação. A alta taxa de reprovação em disciplinas introdutórias acaba sendo um fato observado em todo o mundo: Lister (2000) apresenta as altas taxas de reprovação em universidades australianas, sendo uma das piores. Dehnadi, Bornat et al. (2006) apresentam as altas taxas de reprovação em universidades britânicas, relatando que entre 30% e 60% dos alunos de ciências da computação em cada universidade reprovam na primeira disciplina de programação. McCracken et al. (2001) promoveram um estudo realizado em nove instituições de seis países diferentes com a finalidade de averiguar as competências de programação de estudantes universitários de ciências da computação, no final do seu 1º ano. Os resultados surpreenderam os pesquisadores, apontando que após realizar as disciplinas de programação, muitos alunos não sabem programar.

O motivo do grande insucesso dos estudantes observado nas disciplinas introdutórias de programação é tema debatido em vários trabalhos. Segundo Souza, Batista e Barbosa (2016), os principais problemas no ensino e aprendizagem de programação estão na dificuldade dos alunos em aprender os conceitos de programação, na dificuldade dos alunos na aplicação desses conceitos durante a construção de programas e na falta de motivação entre os alunos na realização da atividade de programação. Jenkins (2002) refere-se a várias causas do insucesso dos alunos em disciplinas de programação: baixo nível de abstração, falta de competência na resolução de problemas e inadequação dos métodos pedagógicos aos estilos de aprendizagem dos alunos. Dunican (2002) aponta que as características da programação são de difícil domínio, pois não têm correspondência concreta em nosso cotidiano.

Devido à evidente dificuldade de aprendizado de programação, alguns autores recorrem aos Objetos de Aprendizagem (OA) como uma ferramenta de apoio. Jesus, Gomes e Cruz (2013) destacam o poder de potencializar o processo de ensino-aprendizagem que o uso de OAs possui. Souza, Batista e Barbosa (2016) apresentam como forma de amenizar os problemas e dificuldades, a utilização de OAs chamadas de *serious games*.

## 1.1 Problema

Segundo Nascimento et al. (2017) a perda de estudantes que ingressam em uma faculdade mas não terminam o curso é um dos problemas que afligem as instituições que oferecem cursos superiores. Em Giraffa e Mora (2013) a preocupação com o elevado número de desistentes no curso de computação é reforçada pela falta de profissionais nessa área e pelas muitas oportunidades em aberto no mercado de trabalho. A respeito das disciplinas de programação, Hinterholz (2009) afirma que "essas disciplinas costumam ter altos índices de evasão e reprovação, sendo um dos gargalos existentes nos cursos de graduação, particularmente de computação, dificultando ou até mesmo impedindo a continuidade dos alunos no curso."

O alto índice de reprovação nas disciplinas introdutórias de programação tem uma grande influência na desistência dos estudantes em dar continuidade ao curso, fato que cria a busca de soluções para mitigar as dificuldades encontradas no aprendizado pelos estudantes.

As soluções para o insucesso dos estudantes nas primeiras disciplinas de programação são amplamente debatidas dentro do meio acadêmico, com várias propostas e metodologias diferentes, mas que acabam por possuir alguns pontos em comum. Com tantas propostas diferentes, é comum que o docente fique perdido em relação a qual adotar e ache muito trabalhoso procurar a fundo, resultando na maioria das vezes na desistência de aplicar algo diferente. Para o estudante, muitas vezes iniciando sua vida acadêmica, falta o conhecimento para procurar algo fora da metodologia de ensino tradicional.

Especificamente no ensino e aprendizagem de programação, os objetos de aprendizagem que poderiam ajudar os alunos com as dificuldades apresentadas são frequentemente vistos apenas como softwares de apoio, o que acarreta problemas em seu desenvolvimento, principalmente no que se refere ao desenvolvimento pedagógico. Diante desse cenário, quais os limites e possibilidades do uso de objetos de aprendizagem como instrumentos de ensino e aprendizagem de programação?

## 1.2 Objetivos

Os objetivos deste trabalho foram divididos em objetivo geral e objetivos específicos.

### 1.2.1 Objetivo Geral

Investigar e compreender como os objetos de aprendizagem podem contribuir para o entendimento dos conceitos estudados nas disciplinas iniciais de Programação, apontando suas possibilidades e limitações.

### 1.2.2 Objetivos Específicos

- Investigar quais são os conceitos estudados em disciplinas iniciais de Programação, tendo como base as ementas de disciplinas do curso superior de Engenharia de Computação do CEFET-MG campus Timóteo.

- Investigar como (e quais) os objetos de aprendizagem têm sido utilizados para trabalhar tais conceitos em cursos da área da computação.
- Selecionar três objetos de aprendizagem disponíveis na literatura e analisá-los, avaliando-os segundo o método de avaliação de LORI, e segundo avaliações disponíveis na Literatura.
- Explicitar possibilidades e limitações de cada OA escolhido, bem como fazer recomendações de uso de cada um, no contexto das disciplinas em questão.

### 1.3 Justificativa

Para Almeida, Chaves e Jr (2015), é necessário promover uma discussão mais ampla sobre a qualidade dos Objetos de Aprendizagem (OAs). A fim de aproveitar melhor sua utilização, os professores devem realizar uma avaliação minuciosa desses objetos, visando obter um ótimo aproveitamento em suas práticas pedagógicas.

De acordo com Souza, Marcelino e Fortunato (2018), existe uma divisão nos artigos que avaliam objetos de aprendizagem: alguns realizam avaliação, mas não conduzem experimentos empíricos com o método, enquanto outros realizam a avaliação empírica, mas carecem de uma abordagem educacional e avaliativa do OA estudado.

Com esse cenário em vista, mostra-se necessário ampliar a discussão sobre os softwares utilizados no processo de ensino e aprendizagem, a fim de garantir qualidade nos aspectos pedagógicos. Além disso, os estudos que avaliam objetos de aprendizagem complementam-se entre si, pois fornecem tanto uma avaliação metodológica baseada em algum processo quanto uma avaliação empírica.

## 2 Revisão de Literatura

### 2.1 Objetos de Aprendizagem

#### 2.1.1 Definição

O termo *objetos de aprendizagem* surgiu em 1992 e tem sua origem atribuída a Wayne Hodgins (POLSANI, 2003; PÖLDOJA et al., 2005; MCGREAL, 2004), um pesquisador de estratégias de ensino e treinamento auxiliadas por tecnologia. Segundo Hodgins (2006), ao observar seus filhos brincando com LEGO, ele teve a ideia de que os materiais de ensino podem ser construídos de maneira a tornar prático o seu agrupamento com outros materiais e, assim, formar estruturas maiores e mais completas de conteúdo.

Apesar de muitos estudiosos das áreas de tecnologia e de educação investigarem o desenvolvimento e a aplicação de um OA no processo de ensino e de aprendizagem ao longo dos anos, não existe um conceito de OA aceito universalmente (RAMOS et al., 2005). As definições, porém, não apresentam um alto grau de divergência, variando de acordo com grupos de pesquisa e enfoque pedagógico adotado no desenvolvimento dos objetos (SCHWARZELMÜLLER; ORNELLAS, 2006).

O principal ponto de divergência das definições se encontra no fato de que enquanto alguns autores propõem objetos de aprendizagem como sendo exclusivamente digitais, outros autores expandem a definição para qualquer objeto que possa ser utilizado para aprendizado. Para Mortimer (2002), "um objeto de aprendizagem é uma peça de conteúdo que é menor que um curso ou lição"; para Wisc-Online (2006): "é o mais básico bloco de uma lição ou atividade". Já Wiley et al. (2000) define OA como sendo "qualquer recurso digital que pode ser reutilizado para apoiar a aprendizagem". Braga, Pimentel e Dotta (2013) consideram que quando utilizados com propósito de aprendizagem, recursos como imagens, vídeos, softwares e animações podem ser considerados OAs. Beck (2002) entende que OA é qualquer recurso digital que possa ser reutilizado para o suporte ao ensino. Machado e Filho (2003) dizem que OAs são recursos digitais que podem ser usados, reutilizados e combinados com outros objetos para formar um ambiente de aprendizagem rico e flexível. Para South e Monson (2000) OA é definido como sendo uma "mídia digital que é concebida e/ou usada para finalidades instrucionais".

Dentro das definições que enfocam os OAs como sendo objetos digitais, se destaca o fato de que é ressaltada a reusabilidade como sendo uma das principais características de um OA. Uma vez que a criação de materiais de ensino requer alto investimento humano e tecnológico (SOUTH; MONSON, 2000), o reuso se torna uma característica muito importante, já que reduziria tais investimentos.

Neste trabalho, os OAs são considerados recursos pedagógicos digitais, dando enfoque principalmente nos objetos de aprendizagem utilizados no processo de ensino e aprendizagem de programação.



### 2.1.2 Agrupamento

Objetos de aprendizagem são comumente agrupados na literatura em: tipos, grupos e classificação.

#### 2.1.2.1 Tipos

Com base no processo de ensino e aprendizagem de programação e na classificação estipulada por Churchill (2007), os objetos de aprendizagem de interesse podem ser separados nos seguintes tipos:

- Práticas: buscam principalmente a interatividade. Tem como principal exemplo os jogos educacionais.
- Simulações: têm como objetivo simular digitalmente algum aspecto da realidade. Busca desenvolver a familiaridade de um indivíduo com determinado sistema, sem a necessidade física deste sistema.
- Conteúdos: projetados para a exposição de determinado assunto. Temos como exemplos animações com propósito educacional.
- Informações: conceitos e ideias explorados por: imagem, animação, áudio, diagrama, texto.
- Representações contextuais: apresentam conceitos e ideias aplicadas em um cenário autêntico.
- Modelos conceituais: definem conceitos e ideias de modo a torná-las interativas e facilitar sua compreensão.

#### 2.1.2.2 Divisão em Grupos

Os autores Silva, Andrade e Brinatti (2020) dividem as Tecnologias Educacionais em dois tipos: Tecnologias Pedagógicas, que são ferramentas que atuam diretamente nos processos de ensino e aprendizagem, e Tecnologias Operacionais, que são utilizadas na parte operacional e administrativa do trabalho dos professores.

Um objeto de aprendizagem é uma entidade que melhor se encaixa como Tecnologia Pedagógica e, segundo os autores Silva, Andrade e Brinatti (2020), pode ser dividido em dois grupos:

- Tecnologias Baseadas em Ferramentas
- Tecnologias Baseadas em Programas

As Tecnologias Baseadas em Ferramentas são aquelas em que sua funcionalidade permite que tal ferramenta se expanda para o campo educacional, mesmo que não tenha sido

desenvolvida para isso ou que possua uma pluralidade de finalidades. Ex: Microsoft Office, Compiladores, Internet.

Já as Tecnologias Baseadas em Programas são aquelas projetadas com fins pedagógicos. São menos customizáveis e geralmente possuem um tema de aprendizagem restrito. Ex: Jogos Educacionais, Objetos de Aprendizagem.

### 2.1.2.3 Classificação

Classificar um objeto de aprendizagem é unir seu aspecto técnico com sua efetividade no processo de ensino e aprendizagem. Wiley et al. (2000) classificou os Objetos de Aprendizagem da seguinte forma:

- Fundamental: é um OA que possui apenas um tipo de mídia, sendo projetado pensando no maior número de contextos possíveis.
- Combinado fechado: é a combinação de um pequeno número de OAs que possuem a mesma finalidade e por isso se complementam de alguma forma.
- Combinado aberto: é a combinação de um número maior de OA. Por possuir uma complexidade maior, normalmente existe uma ordem lógica das peças combinadas visando o entendimento completo do assunto.
- Gerador de apresentação: é a combinação do aspecto Fundamental com o Combinado fechado para que possam ser utilizados como ferramentas auxiliares na prática educacional.
- Gerador de instrução: tem como objetivo gerar interação com o aluno para poder avaliar o seu aprendizado sobre o assunto.

### 2.1.3 Características

Utilizar as tecnologias para complementar ou substituir os métodos de ensino tradicionais não é algo que possa ser feito com trivialidade. É necessário ter conhecimento sobre qual tecnologia usar e por que ela é a que melhor se encaixa na situação requerida.

Os objetos de aprendizagem são considerados um novo parâmetro tecnológico, pois são objetos que visam auxiliar o processo de aprendizagem (OLIVEIRA, 2013b). Sendo assim, diferentemente de softwares complexos que acabam afastando os professores da ideia de sair da metodologia de ensino tradicional devido à sua dificuldade operacional, as OAs têm como objetivo auxiliar tanto na aprendizagem quanto no ensino, e para isso, cuidados devem ser tomados durante sua elaboração.

Mesmo considerando as várias definições dadas aos objetos de aprendizagem, quando se trata especificamente de recursos digitais, já está consolidado na literatura determinadas características que as OAs devem seguir. Considerando as características abordadas na literatura, é possível fazer a divisão entre as características de aspecto técnico e as características

de aspecto pedagógico. Segundo Wiley et al. (2000), Friesen (2004), Assis (2005) e Oliveira (2013b), as características de aspecto técnico de um OA são:

- **Acessibilidade:** o usuário deve, independentemente de sua localização, conseguir acessar via internet as OAs presentes no repositório.
- **Reusabilidade:** o OA deve ter a capacidade de ser utilizado em diferentes contextos, permitindo a elaboração de materiais educacionais mais complexos.
- **Durabilidade:** assegurar que o OA não precise ser refeito devido aos avanços tecnológicos.
- **Interoperabilidade:** deve ser garantido o funcionamento do OA independentemente da plataforma, ou seja, independente de qual desenvolvedor, software ou hardware esteja sendo utilizado.
- **Granularidade:** o OA deve ser capaz de se integrar com outros.
- **Adaptabilidade:** deve ser assegurada a capacidade de se adaptar às necessidades de quem estiver utilizando.

Dentre as características citadas, a reusabilidade se destaca na maioria dos casos na literatura. A possibilidade de reutilização é um dos pilares na construção de um OA, focando em reduzir o tempo e o custo de seu desenvolvimento.

Um OA sem considerar os aspectos pedagógicos, porém, acaba falhando no espectro de aprendizagem, o que pode interferir, inclusive, nos aspectos técnicos, como, por exemplo, desestimular o reuso por simplesmente não ter garantia de sua efetividade. Galafassi, Gluz e Galafassi (2013) relata que podemos considerar como aspectos pedagógicos importantes:

- **Interatividade:** é o indicativo de que o aluno precisa interagir de alguma forma com o OA, seja por meio da visão, audição ou pelo estímulo de pensar e responder algo.
- **Autonomia:** indica se o objeto de aprendizagem estimula a tomada de decisão.
- **Cooperação:** indica a presença de coletividade para resolução de problemas ou debate de ideias.
- **Cognição:** refere-se à ativação cognitiva que o OA causa no aluno.
- **Afetividade:** diz respeito a como o aluno se sente ao utilizar o OA.

É importante ressaltar que não é necessário que o OA possua todas essas características à risca, sendo algumas delas consideradas mais como boas práticas. No entanto, a falta ou o descuido com essas características pode ocasionar dificuldade de reuso ou fazer com que o OA não atinja o seu objetivo.

### 2.1.4 Modelos Padrões

Dada a natureza dos objetos de aprendizagem, que possuem alto grau de reusabilidade, é de grande valor utilizar modelos padrões na criação dos OAs para garantir diretrizes e boas práticas, a fim de assegurar as características necessárias para um OA.

Considerando a grande quantidade de características atribuídas a um OA por alguns autores e como isso resulta em um aumento exponencial da complexidade de criação e custo, devido à interação entre as características, a adoção de uma metodologia adequada pode levar à produção de OAs com um maior número de características desejáveis, permitindo maior reuso e eficiência (BRAGA et al., 2012).

#### 2.1.4.1 Padrão SCORM

O Shareable Content Object Reference Model (SCORM) foi desenvolvido pela ADL (Advanced Distributed Learning) para garantir a qualidade de materiais educacionais. Um dos principais objetivos do modelo SCORM é buscar a padronização de recursos digitais, favorecendo principalmente a reusabilidade e a possibilidade de migração dos objetos de aprendizagem.

Segundo Silva (2015), a organização dos metadados de OAs no SCORM é feita através de arquivos Extensible Markup Language (XML), onde os objetos são referenciados por meio de tags.

O padrão SCORM pode ser dividido em três elementos: manifesto em formato XML, arquivo de esquema e definição, e o conteúdo do OA (SILVA, 2011). O manifesto em formato XML é o corpo principal e pode ser subdividido em:

- Organizations: parte responsável pela ordem cronológica e organização.
- Resources: o conteúdo diretamente apresentado ao usuário final.
- Metadados: responsável por fazer o tagging dos dados por meio de descrição, permitindo que, quando esse dado estiver em um repositório, as ferramentas de busca tenham maior facilidade e compreensão para classificá-lo.

De acordo com Silva (2015), o SCORM especifica como os conteúdos do OA são empacotados, existindo assim uma grande variedade de ferramentas e softwares capazes de realizar essa compactação e organização de dados.

#### 2.1.4.2 ISO/IEC 9126

Modelos de qualidade de software são artefatos usados para descrever os fatores de qualidade de um produto de software de qualquer natureza (DROMEY, 1996).

De acordo com Botella et al. (2004), a ISO/IEC 9126 é um dos padrões de qualidade mais difundidos, abrangendo tanto modelos de qualidade quanto métricas. Possui uma natureza genérica, sendo assim alguns dos conceitos referidos pela norma precisam ser refinados de acordo com a situação de uso.

Quando se trata especificamente de OAs, o modelo de qualidade ISO/IEC 9126 se mostra eficiente na padronização de criação destes. Segundo Braga et al. (2012), as principais diretrizes do modelo que se encaixam na criação de um OA são:

- Precisão: O OA deve produzir os resultados esperados.
- Confiabilidade: O OA não deve possuir falhas técnicas.
- Facilidade de instalação: O OA deve ser instalado facilmente.
- Portabilidade: O OA deve funcionar nos sistemas operacionais mais utilizados, assim como ser funcional nas mais diversas plataformas (computador, mobile, tablet).
- Interoperabilidade: O OA deve conversar com outros OAs.
- Usabilidade: O OA deve ser de fácil utilização e navegação.

#### 2.1.4.3 Metodologia INTERA

A metodologia INTERA tem como objetivo estabelecer diretrizes para a construção de um OA mantendo as principais características esperadas com alto grau de reusabilidade e recursividade permitindo que revisões sejam realizadas (SILVA, 2015).

Esta metodologia é dividida em duas partes: as etapas e os atores (BRAGA; PIMENTEL; DOTTA, 2013). Os atores são aqueles que desempenham determinadas funções no desenvolvimento do OA e as etapas são os processos de construção do OA propriamente dito.

Cada etapa da metodologia é constituída por três elementos: entradas, práticas e saídas (SILVA, 2015). As entradas são as contextualizações necessárias para iniciar a etapa. As práticas são as técnicas e ferramentas utilizadas na construção do OA. As saídas são os artefatos produzidos após a finalização de uma etapa.

As etapas de desenvolvimento de um OA utilizando a metodologia INTERA, segundo (SILVA, 2015), são:

- Contextualização: etapa inicial da construção de um OA. É onde é feito o levantamento de conteúdo, os limites da abordagem, o público-alvo e o objetivo de aprendizagem que o OA deve cumprir.
- Requisitos: etapa onde é feito o levantamento de quais recursos vão ser oferecidos pelo OA, por exemplo, recursos visuais, sonoros, etc.
- Arquitetura: etapa onde é feito o esboço do que será o OA.
- Desenvolvimento: etapa onde de fato o OA começa a ser construído. É aconselhável que essa etapa seja iniciada em um ambiente de testes.
- Testes: Etapa de revisão tanto técnica quanto pedagógica e de conteúdo do OA. Essa etapa acontece durante todo o desenvolvimento do projeto e deve acontecer várias vezes. É ela que garante maior precisão, reusabilidade e confiabilidade ao OA.

- Validação: Essa etapa é realizada por um grupo selecionado do público a quem se destina o OA, para que seja coletado feedback do usuário final.

Os atores e suas funções presentes na metodologia INTERA, de acordo com (BRAGA; PIMENTEL; DOTTA, 2013), são apresentados na Tabela 1.

Tabela 1 – Atores e Funções na Metodologia INTERA

Ator	Função Desempenhada
Analista	Fazer levantamento de requisitos e elaborar o planejamento de construção do OA
Conteudista	Elaborar pesquisa de conteúdo e sua revisão
Demandista	Quem solicitou que o OA fosse construído
Gerente de projetos	Elaborar cronograma e fazer acompanhamento do projeto
Designer Instrucional	Planejamento e validação da parte pedagógica
Designer de Interface	Fica a cargo da usabilidade do projeto e da sua identidade visual
Designer Técnico	Determina quais tecnologias, ferramentas e linguagens de programação serão utilizadas para a construção do OA
Equipe de Desenvolvimento	Designers, programadores e afins que efetivamente desenvolvem o projeto
Equipe de Teste	Equipe responsável por garantir que a construção do OA mantenha as características necessárias de um bom OA

Fonte: Adaptado de Braga, Pimentel e Dotta (2013)

#### 2.1.4.4 Outras Metodologias

Alguns autores abordam outras metodologias e padrões para a criação de um OA, sejam eles especificamente desenvolvidos com o objetivo de objetos de aprendizagem, sejam metodologias de desenvolvimento de software que acabam sendo aplicadas nesse contexto.

Metodologias como SCRUM e RUP, voltadas para engenharia de software, possuem diretrizes muito robustas que podem ser utilizadas na construção de um OA, mas não abordam a parte pedagógica necessária para um OA (BRAGA et al., 2012).

As metodologias SOPHIA e RIVED, apesar de terem sido desenvolvidas especificamente para a criação de OAs, demonstram, de acordo com Braga et al. (2012), falhas em vários aspectos, principalmente no que diz respeito à revisão do OA para garantir reusabilidade, confiabilidade e precisão.

#### 2.1.5 Repositórios

Os repositórios de OAs são meios de catalogar, manter e disponibilizar materiais educacionais na web (ASTUDILLO; WILLGING; GARCIA, 2011). Essas plataformas online servem

como acervo para OAs de diversos tipos de conteúdo. A importância desses repositórios reside no fato de que materiais espalhados pela internet são de difícil localização, muitas vezes não sendo referenciados como OAs. Segundo Leffa (2006), a maioria das OAs está inserida em contextos amplos, sem qualquer identificação que as relacione com o termo "objeto de aprendizagem". Henrique e Rebouças (2015) relata que, em várias situações, não é utilizado o termo "objeto de aprendizagem", mesmo que o recurso apresentado tenha sido criado para auxiliar o processo de ensino e possa ser considerado como tal.

Os repositórios podem ser divididos em dois tipos: centralizados e distribuídos (RICHARDS et al., 2002; LUCCHESI et al., 2006). Os repositórios do tipo centralizado caracterizam-se por armazenar em um único servidor tanto OAs como metadados, proporcionando rápida indexação e recuperação (ASTUDILLO; WILLGING; GARCIA, 2011). Como exemplos, temos:

- Wisc-Online: desenvolvido pela Faculdade Wisconsin Technical College System (WTCS), tem como objetivo criar OAs com um bom padrão de qualidade para auxiliar no ensino. A elaboração de OAs é feita por uma equipe multidisciplinar da própria plataforma, e o acesso é mediante registro gratuito.
- Casa da Ciência: é um projeto acadêmico de apoio ao ensino das ciências no ensino básico e secundário, criado em 2008 pela Fundação Calouste Gulbenkian. Qualquer pessoa pode submeter um OA para o repositório, mas este só ficará disponível para visualização após validação pela equipe da plataforma.

Os repositórios distribuídos armazenam apenas metadados (ASTUDILLO; WILLGING; GARCIA, 2011), e as OAs catalogadas estão descentralizadas. Ao tentar acessar algum objeto do catálogo, o usuário é direcionado ao servidor em que será possível encontrar o recurso. Como exemplo, temos:

- MERLOT (Multimedia Educational Resource for Learning and Online Teaching): é mantido pela Universidade de Alberta, Canadá, e por universidades estadunidenses. Tem como objetivo aprimorar a qualidade do ensino utilizando OAs como material de apoio. A inclusão de OAs no repositório é permitida a colaboradores registrados.

## 2.2 OAS no Ensino de Programação

Jesus et al. (2007) apresentam em seu trabalho uma proposta teórica de como utilizar objetos de aprendizagem para auxiliar o ensino de lógica de programação, reconhecendo a necessidade de métodos que contribuam para a superação das dificuldades existentes em cursos de graduação em computação e informática. O estudo conclui que a utilização de objetos de aprendizagem é muito justificada devido a boa aplicabilidade de um objeto de aprendizagem sobre os conceitos abordados num curso de lógica de programação.

Henrique e Rebouças (2015) realizam um estudo sobre o uso de objetos de aprendizagem no auxílio ao ensino de programação orientada a objetos, conteúdo que segundo os autores possui um grande número de reprovações dos alunos, devido à dificuldade que eles

enfrentam em realizar abstrações. O trabalho destaca que existem vários OAs que apoiam o ensino de programação orientada a objetos, como por exemplo, BlueJ, Object Editor e RAPTOR, mas nem sempre são referenciados na literatura como objetos de aprendizagem, embora possam ser considerados como tais.

Gomes, Henriques e Mendes (2008) propõem a criação de um ambiente computacional de apoio à aprendizagem de programação, incorporando atividades lúdicas e jogos lógicos, composto assim, por um conjunto de OAs. Os autores destacam que muitos professores continuam a achar que o ensino superior não necessita de pedagogia e que os alunos têm obrigação de se adaptar ao estilo de cada professor. Os autores preveem que essa abordagem no ensino de programação possa ter um grande impacto na motivação dos alunos, evitando que desistam de aprender mesmo diante de dificuldades.

Silva (2015) realizam um estudo sobre o comportamento dos alunos ao utilizar um objeto de aprendizagem criado especificamente para o ensino de disciplinas introdutórias de programação. Os participantes do estudo demonstraram grande interesse na automatização de algumas correções e na interatividade proporcionada pelo OA. O autor relata que é difícil construir um OA que atenda a todas as características e diretrizes propostas na literatura, mas destaca que a usabilidade e a aplicação lúdica de uma atividade despertam muito o interesse dos alunos.

Cardoso, Santos e Gatti (2015) realizam uma revisão sistemática de objetos de aprendizagem utilizados no ensino de computação, com o objetivo de identificar os tipos de OAs mais empregados na área e os assuntos mais abordados. Ao analisar mais de duzentos artigos, os autores observam que jogos e ferramentas de desenvolvimento são os OAs mais utilizados, enquanto a construção de algoritmos e engenharia de software são os assuntos mais abordados. Além disso, os autores destacam que a maioria dos OAs voltados para computação são direcionados a alunos do ensino médio e fundamental.

Santos (2015) apresenta um objeto de aprendizagem de autoria própria focado na utilização da gamificação como ferramenta de apoio para disciplinas de introdução à programação. A autora identifica as dificuldades que os alunos encontram no primeiro contato com disciplinas de programação e a possibilidade de criar um objeto de aprendizagem que se concentre nos conteúdos considerados mais difíceis dentro dessas disciplinas. Para a criação desse objeto de aprendizagem vários outros OAs foram utilizados como inspiração, principalmente aqueles que trabalham com o sistema de gamificação, como por exemplo, Progame, Entrando pelo Cano, Klouro e Monitor. Já como suporte para a teoria de criação de um OA e garantia de parâmetros foi utilizado o modelo LORI.

Tabuti, Puga e Brito (2014) constroem um objeto de aprendizagem para auxiliar no ensino da disciplina Estrutura de Dados. Os autores percebem que os alunos enfrentam dificuldades no conteúdo abordado, devido à sua natureza abstrata, e acreditam que a ludicidade oferecida por um objeto de aprendizagem pode ajudar no entendimento. Para a construção desse OA, os autores combinam noções de boas práticas em engenharia de software com as questões pedagógicas abordadas na literatura sobre OAs.



Vahldick (2007) realiza um estudo de caso em uma disciplina de programação orientada a objeto, onde é utilizado o objeto de aprendizagem BlueJ em conjunto com o framework Greenfoot como ferramentas de apoio ao ensino. O autor monitorou as notas das avaliações dos alunos durante um período escolar para entender o impacto das ferramentas. Apesar de acreditar em um impacto grande do objeto de aprendizagem, o autor traz como resultado um ganho muito pequeno em relação a média de aprovação da disciplina principalmente em relação a alunos desistentes após a primeira avaliação. Acredita-se que o framework não causou a motivação esperada pois carece de uma simulação com mais recursos, com imagens melhores, com recursos de multimídia e com um mundo mais complexo.

Haaster e Hagan (2004) descrevem em seu trabalho um estudo de caso de alunos utilizando o BlueJ: foi avaliada a performance das turmas durante três anos utilizando o BlueJ para a aprendizagem de programação orientada a objeto. Ao final das observações, os autores aplicaram um questionário para os alunos avaliarem a ferramenta com os seguintes critérios: usabilidade, suporte ao paradigma orientado a objeto e suporte ao processo de ensino e aprendizagem. A conclusão do trabalho foi de que o BlueJ foi eficiente no ensino de programação orientada a objeto, mas alguns pontos negativos foram levantados: foram identificados alguns problemas de usabilidade, como uma documentação incompleta e mensagens de erros pouco específicas.

Alkazemi e Grami (2012) apresentam um estudo sobre como o objeto de aprendizagem BlueJ pode auxiliar em disciplinas de graduação que abordam programação orientada a objeto, especificamente o conteúdo polimorfismo. Os autores separaram estudantes em dois grupos. Para o primeiro grupo foi ensinado os conceitos utilizando o objeto de aprendizagem BlueJ e para o segundo grupo os conceitos foram ensinados utilizando a ferramenta Eclipse e diagramas. O estudo mostrou um entendimento igual, nos dois grupos, de conceitos iniciais mas quando o assunto abordado ficou mais abstrato o entendimento do grupo que utilizou o BlueJ foi muito maior.

Xinogalos et al. (2006) avaliam as dificuldades dos estudantes quando introduzidos a programação orientada a objeto com a metodologia de ensino proposta por Kölling Barnes. Os autores citam que diante da dificuldade que os alunos encontram na programação orientada a objeto, vários objetos de aprendizagem foram desenvolvidos para lidar com elas, como por exemplo, BlueJ, DrJava, jGRASP. A metodologia abordada pelos autores consiste em utilizar o objeto de aprendizagem BlueJ em conjunto com as lições propostas. Os autores consideram a metodologia boa mas não livre de problemas. São levantados tópicos de melhora no que se diz respeito a interação e condução do uso do objeto de aprendizagem.

Hagan e Markham (2000) coletam de alunos opiniões a respeito do objeto de aprendizagem BlueJ após algumas aulas de programação orientada a objeto utilizando a ferramenta. Os autores buscam entender quais os pontos fortes do OA e quais as dificuldades encontradas pelos alunos para entender quais aspectos levam a motivação ou desmotivação de utiliza-lo.

Ribeiro, Merlin e Fülber (2019) estudam os impactos que a utilização de um ambiente gamificado pode ter em uma matéria introdutória do curso de Ciência da Computação. O autor procurou na literatura trabalhos que experimentaram objetos de aprendizagem com a caracte-

rística de jogo no ensino de programação e encontrou CodeCombat, Lightbot e NoBug's Snack Bar. Depois de escolher o CodeCombat como OA para o trabalho, foi aplicado o OA para os alunos e avaliou retenção de conhecimento e engajamento. O autor concluiu que o uso de um ambiente gamificado não necessariamente tem impacto na abstração dos conceitos abordados mas que o engajamento e a motivação gerado justificam a utilização.

Barbosa (2017) propõe avaliar vários objetos de aprendizagem em formato de jogo nos quesitos usabilidade, aspectos técnicos e aspectos pedagogicos, para entender como esses OAs podem contribuir para o pensamento computacional. Os objetos de aprendizagem avaliados são: Lightbot, Operação Big Hero – Code Baymax, CodeCombat e FightCode. Os autores concluem que é difícil de avaliar os aspectos pedagogicos por falta de técnicas especializadas e que um OA pode ser considerado melhor ou pior avaliada de acordo com a proposta e contexto de uso.

## 2.3 Método de Avaliação de um OA: LORI

Boff e Reategui (2005), Souza et al. (2007) e Romero, Andrade e Pietrocola (2009) são autores que discutem a respeito da importância de se avaliar um objeto de aprendizagem afim de garantir o melhor aproveitamento das praticas pedagogicas. É entendido que muitos objetos de aprendizagem não se diferem de meras ferramentas para entreter o aluno e acaba se perdendo no seu objetivo de ensino pois não tem seu impacto pedagógico avaliado.

Segundo Almeida, Chaves e Jr (2015) a ausência de padronização faz com que métodos de avaliação diferentes dêem vereditos que podem divergir muito em relação a qualidade e aplicabilidade de um OA, o que leva o professor a não ter certeza a respeito de se deve ou não utilizar a ferramenta. Para Souza, Pequeno e Filho (2005) a importância da avaliação se dá tanto para o professor que vai utilizar o objeto de aprendizagem quanto para as instituições de ensino que muitas vezes vai fazer um investimento financeiro.

Existem diferentes tipos de ferramentas que são utilizadas para se avaliar um objeto de aprendizagem na literatura. Dentre essas ferramentas se destacam: modelos de construção de um OA, modelos de avaliação de um OA, modelos de construção de software e observação empírica.

Foram encontrados na literatura trabalhos em que os autores utilizaram algum método de avaliação de um OA. Esses métodos foram: LORI (MARZAL; PEDRAZZI, 2015; FERNÁNDEZ-PAMPILLÓN, 2013; CESTEROS, 2014), MERLOT (GÜRER; YILDIRIM, 2014; ZAPATA et al., 2015), LOAM (Learning Object Attribute Metric Tool) (CESTEROS, 2014) e HEODAR (Herramienta para la Evaluación de Objetos Didácticos de Aprendizaje) (FERNÁNDEZ-PAMPILLÓN, 2013; MARZAL; PEDRAZZI, 2015).

Para esse trabalho será utilizado o modelo de avaliação de OA LORI. Um modelo de avaliação se destaca das demais formas de avaliação pois existe uma preocupação real com a parte pedagógica do OA. Dentre os modelos estudados e vistos na literatura, o LORI foi escolhido por sua clareza e respaldo de outros autores.

### 2.3.1 Descrição

O método LORI (Learnin Object Review Instrument) é utilizado pelo MERLOT (Multi-media Educational Resource for Learning and Online Teaching), um dos maiores repositórios internacionais de OA disponíveis atualmente (ALMEIDA; CHAVES; JR, 2015). Esse método consiste em um sistema de avaliação de objetos de aprendizagem que leva em consideração oito itens, entre critérios técnicos e pedagógicos, para avaliação. Este trabalho utilizará a versão 2.0 do LORI que foi disponibilizada em janeiro de 2009.

O LORI foi desenvolvido pela E-Learning Research e a Assessment Network (eLera) com o apoio da TeleLearning NCE, CANARIE Inc. e eduSourceCanada (NESBIT; BELFER; LEACOCK, 2009).

O manual oficial de utilização do LORI pode ser encontrado no endereço eletrônico: <[https://www.academia.edu/7927907/Learning\\_Object\\_Review\\_Instrument\\_LORI\\_](https://www.academia.edu/7927907/Learning_Object_Review_Instrument_LORI_)>

### 2.3.2 Avaliação

Nesbit, Belfer e Leacock (2009) apresentam, no manual de usuário do LORI, os oito tópicos avaliados. A avaliação dos tópicos é feita através de um sistema de número de estrelas onde uma estrela é o pior nível de avaliação e cinco estrelas é o melhor nível. Os oito tópicos avaliados são:

- Qualidade do Conteúdo: precisão do conteúdo, boa apresentação de ideias, nível de propriedade e detalhes e reusabilidade em vários contextos.
- Meta de Aprendizagem: características e escalada da aprendizagem com foco em uma meta.
- Feedback e Adaptação: feedback e adaptação modelados por diferentes modelos de aprendizagem.
- Motivação: capacidade de manter interesse no aprendizado.
- Design: parte visual eficiente para gerar motivação e facilitar o entendimento.
- Usabilidade: facilidade de utilização e navegação.
- Acessibilidade: controles que permitam a utilização em diferentes plataformas (computador, mobile, tablet) e a utilização de pessoas com necessidades especiais.
- Conformidade com os Padrões: adesão a normas internacionais e operacionalidade em plataformas técnicas comumente usadas.

### 2.3.3 Aplicação da Avaliação

Para um melhor entendimento de como é feita a avaliação de cada item do LORI, serão apresentados os parâmetros que determinam a classificação de um objeto de aprendizagem

em uma estrela, três estrelas ou cinco estrelas. Uma e cinco estrelas são considerados extremos da avaliação, enquanto três estrelas será o meio termo exato. A avaliação ainda vai contar com duas e quatro estrelas para representar a excelência ou problema sem que seja representado por um extremo. A descrição de cada nível de classificação será feita através de tópicos explicando as características que devem ser preenchidas ou por meio de exemplos de objetos de aprendizagem que se enquadrem em cada nível.

Essa aplicação da avaliação é um uso direto do manual do usuário do LORI proposto por Nesbit, Belfer e Leacock (2009). Os exemplos foram adaptados para se encaixarem com a proposta do trabalho em avaliar objetos de aprendizagem voltado para o ensino de programação em disciplinas introdutórias.

A tabela 2 é um exemplo de como será feita a avaliação, com o parâmetro avaliado na coluna da esquerda e a quantidade de estrelas, representadas pelo símbolo \*, na coluna da direita.

Tabela 2 – Tabela de Exemplo: Avaliação LORI

<b>Tópicos Avaliados</b>	<b>Avaliação LORI</b>
<b>Qualidade de Conteúdo</b>	* * * * *
<b>Meta de Aprendizagem</b>	* * * * *
<b>Feedback e Adaptação</b>	* * * * *
<b>Motivação</b>	* * * * *
<b>Design</b>	* * * * *
<b>Usabilidade</b>	* * * * *
<b>Acessibilidade</b>	* * * * *
<b>Conformidade com Padrões</b>	* * * * *

### 2.3.3.1 Qualidade do Conteúdo

Uma Estrela - Uma ou mais das seguintes características tornam o objeto de aprendizagem inutilizável:

- O conteúdo é impreciso.
- O conteúdo é apresentado com viés ou omissões significativas.
- O nível de detalhe não é adequado.
- O conteúdo não é adequado para reutilização.
- As diferenças culturais ou étnicas não são representadas de forma equilibrada.

Três Estrelas - A qualidade do conteúdo não é completa por si só e acaba restringindo o uso do objeto de aprendizagem em situações específicas. Podemos tomar como exemplo um objeto de aprendizagem que ensina a usar consultas em um banco de dados para visualização dos dados, mas que se limita aos recursos mais básicos de visualização. Esse objeto de aprendizagem pode cumprir sua função em determinado contexto, mas além de ser limitado pelo contexto e pelo tipo e nível do usuário, pode induzir o usuário a acreditar que determinadas funções que existem não são possíveis.

Cinco Estrelas - O conteúdo não possui erros e é apresentado sem omissões que possam induzir o usuário ao erro. As alegações são apoiadas por evidências e raciocínio. As apresentações enfatizam pontos-chave e ideias significativas com um nível adequado de detalhe. O conteúdo é aplicável em vários contextos de aprendizagem e mostra-se adequado para diferentes níveis e tipos de alunos.

### 2.3.3.2 Meta de Aprendizagem

Uma Estrela - Uma ou mais das seguintes características tornam o objeto de aprendizagem inutilizável:

- Não possui uma meta de aprendizagem clara.
- O conteúdo abordado não é compatível com o contexto para o qual foi desenvolvido.
- A meta de aprendizagem não é compatível com o tipo de usuário para o qual foi desenvolvido.

Três Estrelas - Algum aspecto do objeto de aprendizagem aborda algum assunto que está relacionado ao tema principal, mas não é um objetivo de aprendizagem do OA. Por exemplo, podemos imaginar um objeto de aprendizagem que ensina a linguagem de programação Python para o desenvolvimento de aplicativos móveis e que possui um teste para avaliar o conhecimento do aluno após a aprendizagem. No entanto, esse teste aborda, em algumas questões, o tema de segurança de rede. Embora seja um tema relevante dentro do contexto

para o qual o OA foi desenvolvido, ele não é o objetivo de aprendizagem e o OA não oferece o ensino específico que permitiria ao aluno responder a pergunta.

Cinco Estrelas - Os objetivos de aprendizagem são apropriados para o público-alvo e são explícitos, seja dentro do conteúdo do próprio OA ou por meio de documentação de fácil acesso ao usuário. As atividades de aprendizagem, conteúdo e avaliações fornecidas pelo objeto estão alinhadas com os objetivos declarados. O objeto de aprendizagem é suficiente para permitir que os alunos alcancem as metas de aprendizagem.

#### 2.3.3.3 Feedback e Adaptação

Uma Estrela - Uma ou mais das seguintes características estão presentes no objeto de aprendizagem, mesmo que ele apresente um bom grau de interatividade:

- A saída do OA não é adaptável a diferentes entradas fornecidas pelo usuário.
- Não há um sistema de aprendizado com progressão de acordo com a avaliação do aprendizado do aluno.
- Não existe feedback sobre a qualidade da resposta do usuário.

Três Estrelas - Um objeto de aprendizagem que propõe um teste para que o aluno avalie seu conhecimento, fornece um feedback diferente se a resposta estiver correta ou incorreta, mas não adapta o conteúdo com base nas áreas em que o aluno demonstra mais facilidade ou dificuldade, ainda pode ser útil, mas se torna limitado.

Cinco Estrelas - O objeto de aprendizagem tem a capacidade de adaptar mensagens ou atividades. O perfil do aluno influencia o comportamento do objeto de aprendizagem.

#### 2.3.3.4 Motivação

Uma Estrela - Uma ou mais das seguintes características tornam o objeto de aprendizagem inutilizável:

- O conteúdo não é relevante para o público que o utiliza.
- As atividades são muito triviais ou muito complexas para o público-alvo.
- Existem recursos no objeto de aprendizagem que causam distração e atrapalham o aprendizado.
- O aluno não interage e não é submetido a escolhas.
- Não há feedback para o aluno sobre seu aprendizado.

Três Estrelas - Um objeto de aprendizagem que tem como foco o aprendizado de programação pode ter atividades repetitivas que levam o aluno a sentir-se entediado e desafiado.

No entanto, esse objeto de aprendizagem ainda pode ter valor se for adotada uma abordagem diferente pelo professor, a fim de evitar que se torne uma experiência monótona, não se bastando por si só.

Cinco Estrelas - O objeto de aprendizagem gera grande motivação para o usuário, sendo relevante para seus objetivos pessoais e interesses. O objeto oferece escolha, aprendizado realista, atividades interativas, humor, drama ou desafios semelhantes a jogos. Ele cria uma expectativa realista de progresso para o aluno e possui critérios de sucesso. O feedback compara o desempenho do aluno com os critérios, mostra as consequências naturais do desempenho e explica como o desempenho pode ser melhorado. É provável que os alunos demonstrem maior interesse no tópico após trabalhar com o objeto de aprendizagem.

#### 2.3.3.5 Design

Uma Estrela - O design, a interface do usuário, a estética ou a produção são ruins a ponto de se destacarem. O objeto de aprendizagem pode ser inutilizável pelas seguintes razões:

- O tamanho da fonte atrapalha a capacidade e velocidade de leitura do aluno.
- A qualidade do vídeo ou áudio é insuficiente.
- A escolha de cores e imagens atrapalha ou distrai o aluno.
- As escolhas de design causam um esforço cognitivo desnecessário.
- O cabeçalho dos textos não possui um significado relevante.

Três Estrelas - Em um objeto de aprendizagem que ensina estruturas de dados, existem códigos das estruturas e textos de descrição para cada uma delas. No entanto, o objeto de aprendizagem apresenta todos os códigos primeiro e, em seguida, todas as descrições, sem especificar a qual código cada descrição se refere. O objeto de aprendizagem ainda pode ser utilizado, pois o aluno pode deduzir a qual código pertence cada descrição com base no entendimento do código, mas essa falha leva o aluno a um esforço cognitivo desnecessário.

Cinco Estrelas - O design presente no objeto de aprendizagem permite que o usuário aprenda de forma eficiente. As apresentações minimizam a busca visual. O texto, gráficos e tabelas são legíveis, rotulados e livres de desordem. Eventos animados ou gravados em vídeo são descritos por narração em áudio. Títulos estão presentes sinalizando o conteúdo das passagens de texto. A escrita é clara, concisa e livre de erros. Cores, música e elementos decorativos são esteticamente agradáveis e não interferem na aprendizagem.

#### 2.3.3.6 Usabilidade

Uma Estrela - Uma ou mais das características abaixo impacta significativamente na experiência de usabilidade do usuário:

- Funções de navegação como botões e hyperlinks não estão presentes ou não funcionam.
- Existe um atraso significativo na navegação.
- A usabilidade não é intuitiva e falta instruções.
- O funcionamento da interface é instável e inconsistente.

Três Estrelas - Em um objeto de aprendizagem que mostra a mesma consulta em diferentes sistemas de gerenciamento de banco de dados (SGBD), a informação sobre qual SGBD a consulta pertence só aparece quando o mouse está em cima da consulta. Embora essa situação possa gerar confusão para o usuário e exigir uma usabilidade que poderia ser mais intuitiva, o objeto de aprendizagem ainda é utilizável.

Cinco Estrelas - O design da interface do usuário informa implicitamente ou explicitamente aos alunos como interagir com o objeto de aprendizagem. A navegação é fácil, intuitiva e livre de atrasos excessivos. O comportamento da interface do usuário é consistente, previsível e livre de erros.

#### 2.3.3.7 Acessibilidade

Uma Estrela - O objeto de aprendizagem se torna inutilizável para pessoas com deficiência quando:

- Não há legenda para os vídeos.
- Não há transcrição para arquivos de áudio.
- Não existe texto alternativo para descrição de imagens.
- Gráficos dependem exclusivamente da percepção de cores para serem entendidos.

Três Estrela - Podemos utilizar para classificar o objeto de aprendizagem o Web Content Accessibility Guidelines (WCAG) 2.0. O WCAG é uma série de diretrizes que dizem respeito a acessibilidade de conteúdos WEB, considerando o nível de acessibilidade: AAA é o nível mais completo e correto e A seria o nível que torna a acessibilidade funcional mas não é ideal. Um objeto de aprendizagem que explica um assunto em vídeo seria considerado AAA se tivesse legendas sincronizadas em tempo real com o conteúdo apresentado e seria considerado A se possuir alguma outra mídia que fornecesse explicação a respeito do assunto abordado em vídeo. Na categoria quatro, três e duas estrelas estariam os critérios considerados A e AA.

Cinco Estrelas - O objeto de aprendizagem fornece um alto grau de acomodação para os alunos com deficiências sensoriais e motoras, possui ferramentas de assistência e apoio e pode ser acessado dispositivos portáteis. Os critérios da WCAG rotulados como AAA se enquadrariam nessa categoria.



### 2.3.3.8 Conformidade com os Padrões

Uma Estrela - O objeto de aprendizagem não está em conformidade com padrões internacionais de desenvolvimento de objetos de aprendizagem como por exemplo: SCORM e W3C.

Três Estrela - Os padrões internacionais de desenvolvimento de objetos de aprendizagem apresentam diversos parâmetros. Se encaixaria como três estrelas aqueles objetos de aprendizagem que seguem parcialmente os parâmetros estabelecidos sem tornar o OA inutilizável.

Cinco Estrelas - O objeto de aprendizagem adere a todos os padrões e especificações internacionais relevantes. Isso inclui os padrões IEEE Learning Object Metadata e diretrizes técnicas desenvolvido por IMS, IEEE, SCORM e W3C (diretrizes de acessibilidade não incluídas). Metadados são fornecidos no código documentado e comentado dentro do objeto e apresentados em uma página disponível para os usuários.

## 3 Metodologia

Este capítulo descreve os procedimentos metodológicos utilizados na realização deste trabalho. Com base no estudo de outros trabalhos, identificou-se que a dificuldade na etapa inicial da aprendizagem de programação de computadores é algo muito recorrente. Sabendo disso, este trabalho propõe uma revisão bibliográfica de avaliações dos objetos de aprendizagem escolhidos, utilizando como escopo de ensino a grade das disciplinas introdutórias de programação do curso de Engenharia de Computação do CEFET-MG Campus Timóteo, para concluir nesse cenário específico qual dos objetos de aprendizagem escolhidos será melhor avaliado.

Para esse trabalho, foi escolhido o modelo de pesquisa bibliográfica e documental, pois existe uma busca pelo entendimento teórico e a compreensão crítica do significado em trabalhos e fontes que antecedem esse trabalho.

Segundo Gil et al. (2002), pesquisa é definida como "o procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos". Quando falamos especificamente de pesquisa bibliográfica, é importante destacar que ela é sempre realizada para fundamentar teoricamente o objeto de estudo, contribuindo com elementos que subsidiam a análise futura dos dados obtidos (LIMA; MIOTO, 2007). Para Sousa, Oliveira e Alves (2021), a pesquisa bibliográfica está inserida principalmente no meio acadêmico e tem a finalidade de aprimoramento e atualização do conhecimento, através de uma investigação científica de obras já publicadas.

Segundo Sá-Silva et al. (2009), a pesquisa documental é muito próxima da pesquisa bibliográfica. O elemento diferenciador está na natureza das fontes: a pesquisa bibliográfica trabalha com fontes secundárias, enquanto a pesquisa documental com fontes primárias. Fontes primárias são dados originais, a partir dos quais se tem uma relação direta com os fatos a serem analisados, ou seja, é o pesquisador que analisa (SÁ-SILVA et al., 2009). Por fontes secundárias compreende-se a pesquisa de dados de segunda mão (OLIVEIRA, 2013a), ou seja, informações que foram trabalhadas por outros estudiosos e, por isso, já são de domínio científico, o chamado estado da arte do conhecimento.

### 3.1 Disciplinas do CEFET-MG

Para delimitar o escopo de assuntos abordados pelos Objetos de Aprendizagem escolhidos, foram utilizadas as disciplinas introdutórias do curso de Engenharia de Computação do CEFET-MG Campus Timóteo. É apresentada uma visão sobre os assuntos abordados nessas disciplinas, suas características e objetivos, a fim de relacionar o que for levantado com o que é apresentado pelos OA's.

As informações apresentadas acerca das disciplinas, características, ementa e afins foram todas retiradas do plano didático presente no site da instituição, disponibilizado no se-

guinte link: <<https://www.eng-computacao.timoteo.cefetmg.br/o-curso/planos-de-ensino-do-curso/>>

### 3.1.1 Programação de Computadores I

Programação de Computadores I é uma disciplina obrigatória do primeiro semestre com carga horária total de 30 horas, dividida em duas aulas semanais, totalizando dois créditos. É uma disciplina teórica que é complementada pela disciplina Laboratório de Programação de Computadores I, considerada um co-requisito, em sua parte prática, com mais 30 horas totais. Essa disciplina pertence ao eixo Fundamentos de Engenharia de Computação e ao Departamento de Computação (DECOM). É classificada pelas Diretrizes Curriculares Nacionais (DCN) como uma disciplina básica.

Ela é lecionada juntamente com as disciplinas: Cálculo I, Geometria Analítica e Álgebra Vetorial, Inglês Instrumental I, Introdução à Engenharia de Computação, Laboratório de Programação de Computadores I, Metodologia Científica e Português Instrumental.

Essa disciplina é pré-requisito para as seguintes disciplinas: Programação de Computadores II, Sistemas Digitais para Computação e Computação Gráfica.

Segundo o plano didático disponibilizado, a ementa da disciplina é composta por: Sistemas numéricos: representação e aritmética nas bases decimal, binária, octal e hexadecimal; introdução à lógica; álgebra e funções Booleanas; algoritmos estruturados: tipos de dados e variáveis, operadores aritméticos e expressões aritméticas; operadores lógicos e expressões lógicas; estruturas de controle; entrada e saída de dados; estruturas de dados; organização e manipulação de arquivos.

Esta disciplina tem como objetivos fazer o aluno:

- Conhecer os conceitos lógicos e computacionais que são essenciais para ciência da computação, visando capacitá-lo a formular corretamente um problema computacional e a construir um algoritmo para sua resolução;
- Conhecer os sistemas numéricos e sua aritmética, noções de lógica e álgebra Booleana;
- Contribuir para o desenvolvimento do raciocínio lógico-matemático abstrato.

### 3.1.2 Programação de Computadores II

Programação de Computadores II é uma disciplina obrigatória do segundo semestre do curso, com carga horária total de 30 horas, dividida em duas aulas semanais, totalizando dois créditos. É uma disciplina teórica que é complementada pela disciplina Laboratório de Programação de Computadores II, considerada um co-requisito, em sua parte prática, com mais 30 horas totais. Essa disciplina pertence ao eixo Fundamentos de Engenharia de Computação e ao Departamento de Computação (DECOM). É classificada pelas Diretrizes Curriculares Nacionais (DCN) como uma disciplina básica.

Ela é lecionada juntamente com as disciplinas: Cálculo II, Filosofia da Tecnologia, Física I, Sistemas Digitais para Computação e seu Laboratório, Matemática Discreta e Laboratório de Programação de Computadores II.

Essa disciplina é pré-requisito para as seguintes disciplinas: Métodos Numéricos Computacionais e Algoritmos e Estruturas de Dados I.

Segundo o plano didático disponibilizado, a ementa da disciplina é composta por: Conceitos de orientação a objetos: tipos abstratos de dados, objetos, classes, métodos, visibilidade, escopo, encapsulamento, associações de classes, estruturas todo-parte e generalização-especialização, interfaces; herança de interface e de classe, polimorfismo, sobrecarga, invocação de métodos; aplicações em uma linguagem de programação orientada a objetos; noções de modelagem de sistemas usando UML: diagrama de classes e de interação.

Esta disciplina tem como objetivos fazer o aluno:

- Conhecer e saber utilizar os conceitos de programação orientada a objetos;
- Projetar e implementar programas utilizando o paradigma de orientação a objetos.

### 3.1.3 Algoritmo e Estruturas de Dados I

Algoritmo e Estruturas de Dados I é uma disciplina de natureza obrigatória do terceiro semestre do curso, com carga horária total de 60 horas, dividida em duas aulas semanais, totalizando quatro créditos. É uma disciplina teórica que é complementada pela disciplina Laboratório de Algoritmo e Estruturas de Dados I, considerada um co-requisito, em sua parte prática, acrescentando mais 30 horas ao total. Essa disciplina pertence ao eixo Fundamentos de Engenharia de Computação e ao Departamento de Computação (DECOM). É classificada pelas Diretrizes Curriculares Nacionais (DCN) como uma disciplina básica.

Ela é lecionada juntamente com as disciplinas: Cálculo III, Física II, Física Experimental I, Arquitetura e Organização de Computadores I, Laboratório de Arquitetura e Organização de Computadores I, Métodos Numéricos Computacionais e Laboratório de Algoritmos e Estruturas de Dados I.

Essa disciplina é pré-requisito/co-requisito para as seguintes disciplinas: Métodos Numéricos Computacionais, Algoritmos e Estruturas de Dados I. Além disso, o plano de ensino apresenta como tendo relação de transdisciplinariedade (inter-relações desejáveis) com as disciplinas: Banco de Dados, Engenharia de Software, Sistemas Operacionais e diversas optativas.

Segundo o plano didático disponibilizado, a ementa da disciplina é composta por: Computabilidade, introdução à avaliação da complexidade de algoritmos; técnicas de projeto e análise de algoritmos; estruturas de dados lineares: listas, pilhas, filas; algoritmos para manipulação e ordenação de estruturas de dados lineares: busca, inserção, eliminação, percurso e ordenação; ponteiros; alocação dinâmica de memória; encadeamento em listas e em tabelas e tabelas de dispersão.

Esta disciplina tem como objetivos fazer o aluno:

- Compreender conceitos em computabilidade e complexidade de algoritmos;
- Saber avaliar a complexidade de algoritmos simples;
- Formar visão crítica dos limites teóricos da computação;
- Conhecer métodos e técnicas para a construção de algoritmos eficientes;
- Conhecer e implementar algoritmos para estruturas de dados lineares;
- Avaliar e comparar implementações de estruturas lineares estáticas e dinâmicas.

### 3.1.4 Algoritmo e Estruturas de Dados II

Algoritmo e Estruturas de Dados II é uma disciplina de natureza obrigatória do quarto semestre do curso com carga horária total de 60 horas, dividida em duas aulas semanais, totalizando quatro créditos. É uma disciplina teórica que é complementada pela disciplina Laboratório de Algoritmo e Estruturas de Dados II, considerada um co-requisito, que adiciona mais 30 horas de prática ao total. Essa disciplina pertence ao eixo Fundamentos de Engenharia de Computação e ao Departamento de Computação (DECOM). É classificada pelas Diretrizes Curriculares Nacionais (DCN) como uma disciplina básica. Essa disciplina encerra o eixo de ensino básico relacionado à programação.

Ela é lecionada juntamente com as disciplinas: Cálculo IV, Física III, Física Experimental II, Arquitetura e Organização de Computadores II, Laboratório de Arquitetura e Organização de Computadores II, Linguagens de Programação, Laboratório de Linguagens de Programação e Laboratório de Algoritmos e Estrutura de Dados II.

Essa disciplina é pré-requisito/co-requisito para as seguintes disciplinas: Banco de Dados I, Laboratório de Banco de Dados I, Compiladores, Inteligência Artificial, Laboratório de Inteligência Artificial Otimização I. Além disso, o plano de ensino apresenta como tendo relação de transdisciplinaridade (inter-relações desejáveis) com as disciplinas: Banco de Dados, Engenharia de Software, Sistemas Operacionais e diversas optativas.

Segundo o plano didático disponibilizado, a ementa da disciplina é composta por: Estruturas de dados hierárquicas: árvores e suas variantes; árvores binárias de busca; árvores binárias balanceadas; listas de prioridades; árvores B; árvores digitais; estruturas auto-ajustáveis; noções de grafos; busca em largura e profundidade; algoritmos para manipulação de estruturas de dados hierárquicas: busca, inserção, eliminação, percurso e ordenação; árvore geradora mínima; caminho mais curto.

Esta disciplina tem como objetivos fazer o aluno:

- Conhecer estruturas de dados não lineares;
- Estudar e programar árvores de pesquisa;
- Conhecer os principais algoritmos de busca em grafos;
- Comparar e escolher a estrutura de dados mais adequada para um problema;

- Conhecer e implementar algoritmos para estruturas de dados não lineares;
- Conhecer estruturas de dados adequadas para memória secundária.

## 3.2 Etapas da Pesquisa

Para esse trabalho o seguinte fluxo será seguido:

- Estudo dos objetos de aprendizagem usados para ensino de programação (Literatura/internet);
- Escolha de três objetos de aprendizagem;
- Levantamento das avaliações dos 3 OAs presentes na Literatura;
- Avaliação dos 3 OAs segundo o método LORI;
- Síntese e análise das avaliações;

## 3.3 Objetos de Aprendizagem Escolhidos

Após levantamento dos objetos de aprendizagem através de pesquisa de recomendações em sites e estudos na literatura, os seguintes objetos de aprendizagem foram escolhidos para serem avaliados:

- CodeCombat;
- BlueJ;
- CodeWars.

## 4 Desenvolvimento

### 4.1 OA's Escolhidos

#### 4.1.1 Motivo da Escolha

Ao realizar o levantamento dos objetos de aprendizagem que seriam objeto de pesquisa neste trabalho, algumas características foram estabelecidas como pré-requisitos:

- Trabalhar com as linguagens: Java, Python ou C++;
- Abordar os mesmos assuntos tratados nas matérias introdutórias deste trabalho;
- Facilidade de acesso ou instalação;
- Capacidade de fornecer conhecimento através de recursos gratuitos;
- Ter presença na literatura.

Os objetos de aprendizagem foram selecionados com o objetivo de evitar redundâncias em relação à forma como abordam o processo de ensino e aprendizagem. O principal critério utilizado foi a presença desses objetos na literatura, a fim de possibilitar a coleta de opiniões sobre seu uso e avaliação por outros autores.

- CodeCombat: objeto de aprendizagem que envolve o aluno em um jogo no estilo RPG. Seu principal objetivo é tornar o processo de ensino e aprendizagem mais dinâmico e divertido.
- BlueJ: objeto de aprendizagem amplamente difundido no ensino e estudo de programação orientada a objeto. Possui uma estrutura mais formal e é considerado um ambiente de desenvolvimento integrado (IDE).
- CodeWars: plataforma que contém desafios para os alunos resolverem. É um objeto de aprendizagem que possui um sistema de classificação e utiliza a competição e a comunidade para estimular o desejo de aprender.

#### 4.1.2 Descrição dos OA's

##### 4.1.2.1 CodeCombat

O CodeCombat é uma plataforma educacional de programação que utiliza a abordagem de jogos para ensinar programação aos usuários. Ao utilizar um jogo de estratégia, o CodeCombat apresenta desafios aos usuários que devem ser resolvidos através da criação de código, tornando o processo de aprendizado mais interativo e divertido. O objetivo do CodeCombat é ensinar conceitos fundamentais de programação, como loops, condicionais, funções

e variáveis, através da resolução de tarefas e desafios que se tornam progressivamente mais difíceis. Para completar cada nível, o usuário deve criar o código necessário para que seu personagem no jogo execute as ações corretas.

Ao apresentar uma abordagem prática e lúdica, esse objeto de aprendizagem se mostra especialmente efetivo para alunos que têm dificuldades de aprendizado com metodologias mais tradicionais. Além disso, a plataforma é flexível e pode ser utilizada por estudantes de diferentes níveis de conhecimento, desde iniciantes até aqueles que já possuem uma base sólida em programação.

Os recursos disponíveis no CodeCombat incluem tutoriais e vídeos explicativos, que ajudam os usuários a entenderem os conceitos fundamentais de programação, além de uma grande variedade de níveis e desafios que aumentam em dificuldade. A plataforma também permite que os usuários trabalhem em projetos colaborativos, o que incentiva a troca de conhecimentos e a ajuda mútua entre os usuários.

Um aspecto interessante presente nesse objeto de aprendizagem é a gestão de turmas. A plataforma do CodeCombat permite que o usuário selecione, durante a criação da conta, se é um aluno, um professor ou um responsável. Ao criar uma conta como professor, é possível criar e gerenciar turmas, convidar os alunos através de um link de acesso e acompanhar o progresso dos alunos por meio de painéis de controle. A Figura 1 mostra o primeiro passo de criação de uma turma.

Figura 1 – Tela de Criação de Classe

**Criar nova classe**

**Nome da Turma**

**Linguagem de Programação**

Selecione uma linguagem

Python

JavaScript

C++

**Classroom Code Autocomplete:** ☒

Whether to enable coding autocomplete in this classroom. Disabled: turns autocomplete off for all levels. Enabled: allows students to choose. We recommend leaving this enabled.

**Classroom Announcement** *Opcional*

**Class Type** *Opcional*

Selecione a melhor opção

**Variedade de Idades dos Alunos** *Opcional*

- à -

**Datas de aula estimadas** *Opcional*

dd / mm / aaaa à dd / mm / aaaa

**Frequência Estimada por Turma** *Opcional*

Turmas por Semana Minutos por Turma

**Criar Turma**

Fonte: Site CodeCombat

O CodeCombat pode ser acessado pelo seguinte endereço eletrônico: <<https://codecombat.com>>. Esse objeto de aprendizagem contempla várias linguagens de programação, mas o foco



deste trabalho será Python.

#### 4.1.2.1.1 Plataforma para Professores

O CodeCombat possui dois tipos de plataforma: uma voltada para o aluno e outra voltada para o professor. Como mencionado anteriormente, a plataforma voltada para o professor permite a criação de turmas selecionando a linguagem de programação a ser trabalhada. Após a criação da turma, o professor tem acesso a um menu de controle onde é possível adicionar alunos à turma, gerenciar os conteúdos e verificar estatísticas e o progresso dos alunos exibido na Figura 2.

Figura 2 – Menu de Controle CodeCombat

Minhas Turmas > TESTE1

**TESTE1** [view ai league team](#) [editar configurações da turma](#)

**2** Análise da Turma

Linguagem: Python  
 Estudantes: 1  
 Tempo médio de jogo por nível: 2 minutos  
 Tempo total de jogo: 41 minutos  
 Níveis médios concluídos: 19.0  
 Total de níveis concluídos: 19  
 Criado: 13/12/2022

Exportar progresso dos alunos (CSV)

**1** Adicionar Alunos:

ShakeCupBus [Copiar código da turma](#)

Os alunos podem ingressar na sua turma usando este Código de turma. Nenhum endereço de e-mail é necessário ao criar uma conta de estudante com este código de de turma.

<https://codecombat.com/stude...> [Copie o URL da turma](#)

Você também pode postar esse URL de classe exclusivo em uma página da Web compartilhada.

Convidar alunos por E-mail

**3**

Estudantes Progresso do Curso Avaliações Status da Licença AI League

Selecionar tudo

☐ Classificar por: [Último Nome](#) [Primeiro Nome](#) [Progresso](#) Selecionar turma: Desenvolvimento de Jogos 1 [Atribuir Curso](#) [Remover curso](#)

<input type="checkbox"/>	<b>Marco Tulio Valeriano</b> mtuliovaleriano@gmail.com	Completado Recentemente: Introdução à Ciência da Computação: Nível 19 <span>CS1</span>	<a href="#">editar</a> <a href="#">remover</a>
--------------------------	-----------------------------------------------------------	-------------------------------------------------------------------------------------------	------------------------------------------------

Fonte: Elaboração Própria a Partir do Site CodeCombat.

O menu de controle do CodeCombat oferece diversas ferramentas que devem ser destacadas. Na Figura 2, temos os números circulados fazendo os seguintes apontamentos:

- 1 - Ferramentas para adicionar alunos à turma: código da turma ou URL.
- 2 - Dashboard com informações gerais e médias da turma.
- 3 - Menu para acompanhamento detalhado de cada estudante.

Atualmente, no CodeCombat, é possível selecionar os seguintes cursos em Python: Introdução à Ciência da Computação, Desenvolvimento de Jogos 1, 2 e 3, Ciência da Com-

putação 1, 2, 3, 4, 5 e 6, e Desenvolvimento Web 1 e 2. Ao atribuir um curso ao aluno, vários desafios são disponibilizados, e quando concluídos, as informações como data de conclusão, tempo gasto e código utilizado para resolver o desafio são enviadas ao professor. Para cada conjunto de temas abordados, a plataforma oferece um desafio no formato de avaliação, com o objetivo de testar os conhecimentos do aluno. Esse desafio segue as mesmas características de uma atividade normal, mas os menus de ajuda são suprimidos para o aluno. Além disso, a plataforma disponibiliza para o professor as soluções mais otimizadas dos desafios, como ilustrado na Figura 3.

Figura 3 – Código do Aluno e Solução Fornecida

**Nível 10: Haunted Kithmaze** [Voltar ao topo](#)

Código Marco Tulio Valeriano (Level Completed 2 horas atrás)  
Last edited 2 horas atrás

```

1 # A melhor maneira de repetir é usando loops.
2
3 while True:
4     # Adicione comandos aqui para repetir.
5     hero.moveRight()
6     hero.moveRight()
7     hero.moveUp()
8     hero.moveUp()
9

```

**Haunted Kithmaze Solução**

```

1 # A melhor maneira de repetir é usando loops.
2
3 while True:
4     # Adicione comandos aqui para repetir.
5     hero.moveRight()
6     hero.moveRight()
7     hero.moveUp()
8     hero.moveUp()
9

```

**Nível 11: Descending Further** [Voltar ao topo](#)

Código Marco Tulio Valeriano (Level Completed 2 horas atrás)  
Last edited 2 horas atrás

```

1 # Atravesse o labirinto em menos de 5 declarações.
2 while True:
3     hero.moveRight(2)
4     hero.moveDown()
5
6
7

```

**Descending Further Solução**

```

1 # Atravesse o labirinto em menos de 5 declarações.
2
3 while True:
4     hero.moveRight(2)
5     hero.moveDown()
6

```

**Nível 11a: Riddling Kithmaze (Praticado)** [Voltar ao topo](#)

Código Marco Tulio Valeriano (Level Completed 2 horas atrás)  
Last edited 2 horas atrás

```

1 # Use um loop para repetir seus movimentos.
2
3 while True:
4     # Escreva seu código aqui para repeti-los.
5     hero.moveRight()
6     hero.moveDown()
7     hero.moveRight(2)
8     hero.moveUp()
9
10

```

**Riddling Kithmaze Solução**

```

1 # Use um loop para repetir seus movimentos.
2
3 while True:
4     # Escreva seu código aqui para repeti-los.
5     hero.moveRight()
6     hero.moveDown()
7     hero.moveRight(2)
8     hero.moveUp()
9

```

Fonte: Elaboração Própria a Partir do Site CodeCombat.

#### 4.1.2.1.2 Plataforma para Alunos

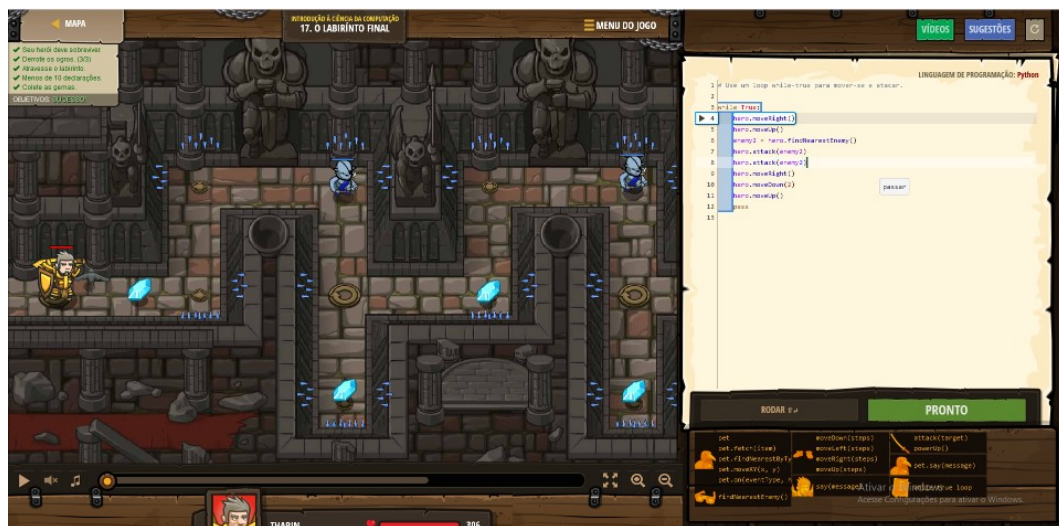
Os alunos podem estudar de forma independente ou através de uma turma, ao serem adicionados por um professor. A plataforma CodeCombat busca ser o mais amigável possível em termos de usabilidade, considerando a ampla faixa etária a que se destina o objeto de aprendizagem. O CodeCombat é um jogo no estilo RPG com características fantasiosas, onde a temática principal é explorar e sair da masmorra. O aluno avança de fase à medida que completa os níveis, que seguem um modelo didático de aprendizagem com aumento progressivo de dificuldade. Além disso, há desafios opcionais que permitem desbloquear os próximos níveis, incentivando o aluno a colocar em prática o que foi aprendido.

Os desafios geralmente são concluídos utilizando as funcionalidades simples do personagem selecionado, tais como andar, atacar, defender, construir, quebrar, resgatar e esconder. A complexidade dos desafios reside na necessidade de aplicar as técnicas de programação ensinadas para executar essas ações e encontrar a solução mais otimizada. À medida que os

níveis avançam, o objeto de aprendizagem combina as ações ensinadas e impõe restrições quanto à quantidade de linhas de código que podem ser utilizadas para resolver o problema, com o intuito de incentivar os alunos a buscar a melhor solução. Esse objeto de aprendizagem enfatiza a abordagem lúdica para estimular os alunos a resolverem os problemas, e ao buscar soluções mais otimizadas, os alunos podem desbloquear novos personagens e equipamentos.

Quando o aluno seleciona um desafio, ele é direcionado para a tela mostrada na Figura 4. Nessa imagem, podemos observar as ferramentas disponíveis para auxiliar o aluno no processo de aprendizagem e compreensão do desafio. À esquerda, temos o personagem que executará as funções que o aluno escrever, e à direita, temos a área para o aluno digitar o código. Na parte inferior direita da tela, encontram-se as funções disponíveis, e na parte superior, há dois botões que oferecem sugestões e dicas em vídeo para a solução do problema. No canto superior esquerdo da imagem, estão os objetivos que devem ser concluídos para avançar para o próximo nível.

Figura 4 – Menu de Resolução do Desafio



Fonte: Site CodeCombat.

#### 4.1.2.2 BlueJ

BlueJ é um ambiente de desenvolvimento integrado (IDE) específico para a linguagem de programação Java. Foi criado com o objetivo de ser fácil de usar e visualmente intuitivo, tornando-se uma escolha popular entre estudantes e professores que estão aprendendo Java. Esse objeto de aprendizagem pode ser encontrado no endereço <<https://www.bluej.org/>>

O BlueJ é extremamente útil para auxiliar os alunos a compreender os conceitos fundamentais da programação orientada a objetos (POO). Possui uma interface gráfica simples e uma ferramenta chamada "Objetos de Projeto", que permite aos alunos visualizar a interação entre as classes e a criação de objetos. Isso reduz a curva de aprendizado e permite que os alunos se concentrem na lógica do programa, sem se preocupar tanto com a sintaxe. Além disso, o BlueJ é uma ferramenta flexível de aprendizagem, que pode ser adaptada para dife-

rentes níveis de habilidade e utilizada em diversos contextos, desde a introdução de conceitos básicos de programação até o desenvolvimento de aplicativos complexos em Java.

Algumas das principais características do BlueJ incluem:

- Uma interface gráfica intuitiva e fácil de usar;
- Ferramentas para visualizar classes e objetos em tempo real;
- Uma ferramenta de depuração para identificar erros em códigos;
- Capacidade de criar projetos de qualquer tamanho e complexidade;
- Suporte para várias plataformas, incluindo Windows, Mac e Linux

Este objeto de aprendizagem é focado em estudantes e professores que desejam aprender ou ensinar programação orientada a objetos em Java. Sua abordagem visual e intuitiva torna a aprendizagem mais fácil e agradável, enquanto suas características avançadas permitem que os alunos progridam para projetos mais complexos à medida que adquirem habilidades e experiência.

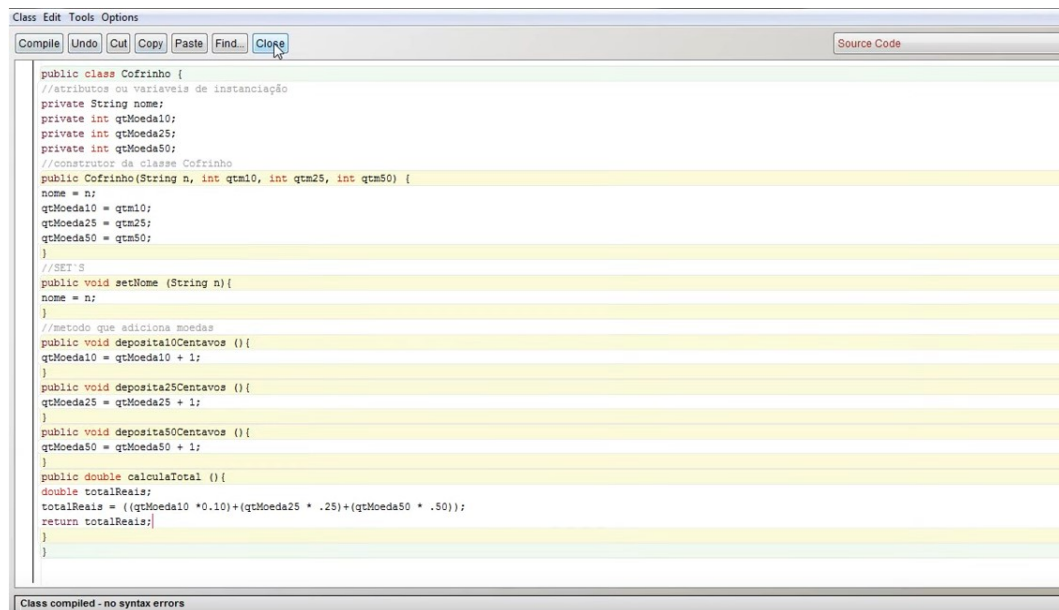
#### 4.1.2.2.1 Interface Gráfica

Ricarte (2001, p.3) enfatiza que a programação orientada a objetos tem como base quatro pilares: classes, objetos, herança e polimorfismo. Esses conceitos possuem um alto grau de abstração, e a interface gráfica do BlueJ tem como objetivo auxiliar os alunos na compreensão desses conceitos.

Ao abrir o objeto de aprendizagem e iniciar um projeto, nos deparamos com a primeira funcionalidade: criar uma nova classe. Ao criar essa nova classe, um código-fonte é aberto para escrever variáveis, construtor e métodos conforme Figura 5. Após a criação da classe, é possível criar objetos correspondentes a essa classe.

Conforme demonstrado na Figura 6, quando o objeto é criado, os métodos da classe ficam disponíveis para serem utilizados. As classes e seus relacionamentos são exibidos no quadrante direito do objeto de aprendizagem, enquanto os objetos criados são listados na parte inferior. O menu à esquerda serve para a criação de uma nova classe ou estabelecimento de hierarquia por meio dos botões com uma seta centralizada.

Figura 5 – Código Fonte de uma Classe no BlueJ

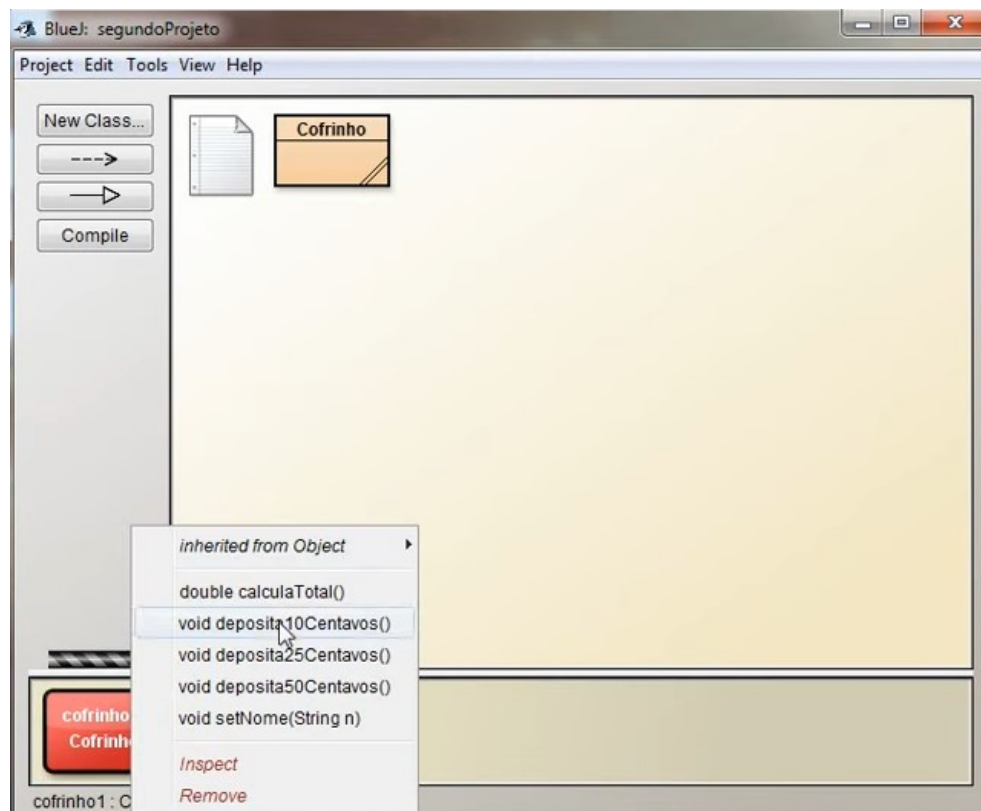


```
public class Cofrinho {  
    //atributos ou variáveis de instânciação  
    private String nome;  
    private int qtMoeda10;  
    private int qtMoeda25;  
    private int qtMoeda50;  
    //construtor da classe Cofrinho  
    public Cofrinho(String n, int qtm10, int qtm25, int qtm50) {  
        nome = n;  
        qtMoeda10 = qtm10;  
        qtMoeda25 = qtm25;  
        qtMoeda50 = qtm50;  
    }  
    //SET'S  
    public void setNome (String n){  
        nome = n;  
    }  
    //metodo que adiciona moedas  
    public void deposita10Centavos () {  
        qtMoeda10 = qtMoeda10 + 1;  
    }  
    public void deposita25Centavos () {  
        qtMoeda25 = qtMoeda25 + 1;  
    }  
    public void deposita50Centavos () {  
        qtMoeda50 = qtMoeda50 + 1;  
    }  
    public double calculaTotal () {  
        double totalReais;  
        totalReais = ((qtMoeda10 * 0.10) + (qtMoeda25 * .25) + (qtMoeda50 * .50));  
        return totalReais;  
    }  
}
```

Class compiled - no syntax errors

Fonte: Autor

Figura 6 – Interface Gráfica BlueJ



Fonte: Autor

#### 4.1.2.3 CodeWars

Esse objeto de aprendizagem pode ser encontrado no domínio: <<https://www.codewars.com>>.

O CodeWars é um site de treinamento de programação que oferece uma plataforma para que desenvolvedores aprimorem suas habilidades em diversas linguagens de programação, como JavaScript, Python, Ruby, C++, entre outras. O site possui uma variedade de desafios de codificação em vários níveis de dificuldade, sendo o nível 8 o mais básico e o nível 1 o mais avançado. Os desafios e suas dificuldades são progressivos e evoluem de acordo com os pontos acumulados pelo usuário. Cada desafio é apresentado como um "kata": uma tarefa específica de programação que o usuário deve resolver usando código.

Esse objeto de aprendizagem se destaca pelo fator comunidade: existe uma comunidade de programadores disposta a ajudar, corrigir e criar novos desafios. É possível participar de um clã focado em determinadas atividades ou linguagens de programação, e os pontos recebidos são chamados de pontos de honra.

O CodeWars oferece aos usuários a oportunidade de avaliar e discutir as soluções de outros usuários, bem como compartilhar suas próprias soluções. O site também apresenta uma série de ferramentas para ajudar os usuários a aprimorar suas habilidades de codificação, incluindo um compilador de código online, uma biblioteca de codificação e um conjunto de desafios de codificação para iniciantes. Entre os recursos de treinamento presentes nesse objeto de aprendizagem, temos exemplos, tutoriais em vídeo, guias de aprendizado e artigos de codificação escritos por desenvolvedores experientes.

##### 4.1.2.3.1 Ambiente

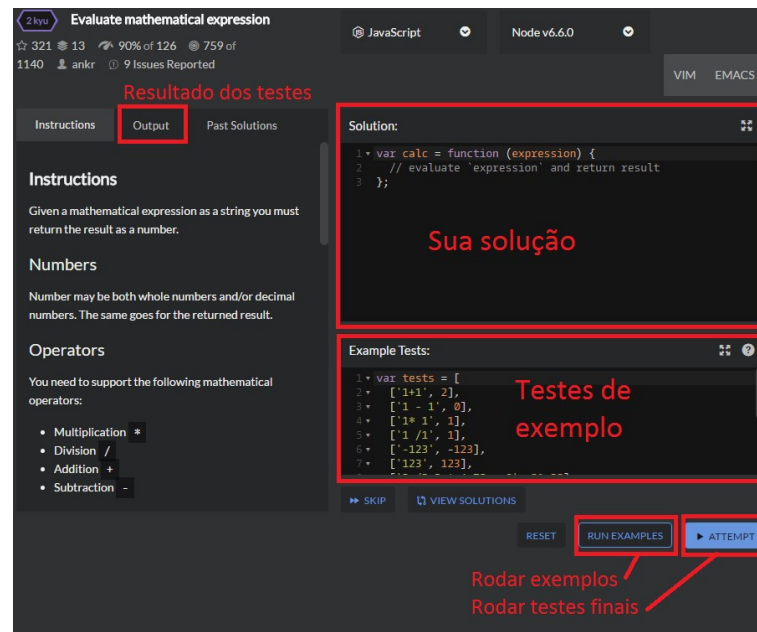
Todo o ambiente desse objeto de aprendizagem gira em torno da interação dos usuários entre si, formando uma comunidade. Existem ferramentas para os usuários discutirem as resoluções de problemas, e a competitividade é sempre estimulada como forma de motivar os usuários a produzirem códigos otimizados de maneira veloz. Existe um sistema de classificação e clãs para exibir o progresso do usuário, além de permitir competições um contra um com outros usuários na opção "duelo".

Esse objeto de aprendizagem é constantemente alimentado com novos desafios por seus usuários mais experientes. Os problemas são categorizados de acordo com o assunto que abordam. Ao selecionar a linguagem de programação, é aberto um menu para que o usuário possa escolher um problema de acordo com seu nível. Após a seleção, é aberta uma página para que o usuário possa resolver o desafio (Figura 7). Essa página de resolução possui um campo de entrada para escrever a solução e um campo de saída onde o resultado do código é exibido.

Podem ser observados vários aspectos de interação da comunidade nas páginas de resolução de problemas. Todos os problemas possuem um fórum dedicado aos usuários para debaterem suas respostas e dificuldades. Existem métricas de satisfação dos usuários em relação aos problemas propostos, estatísticas do site (como as linguagens mais utilizadas



Figura 7 – Interface de Resolução



Fonte: Elaboração Própria a Partir do Site CodeWars

para resolver um determinado problema e quantas pessoas já o resolveram) e um sistema para reportar problemas e bugs.

## 4.2 Conceitos Trabalhados

Esta seção tem como objetivo relacionar os conceitos abordados pelos objetos de aprendizagem com os conceitos presentes nas matérias introdutórias estabelecidas anteriormente neste trabalho.

O BlueJ trabalha com o ensino de programação orientada a objetos. São abordados os chamados quatro pilares da programação orientada a objetos: abstração, encapsulamento, herança e polimorfismo. A abstração diz respeito à criação do objeto propriamente dito, suas propriedades e métodos. O encapsulamento é utilizado para evitar o acesso direto ao objeto como forma de segurança, por meio dos conhecidos métodos "get" e "set". A herança refere-se a um objeto herdar características de objetos que estão acima dele. O polimorfismo é a propriedade de duas ou mais classes derivadas de uma mesma superclasse, onde o conceito de herança é aplicado, e respondem à mesma mensagem, cada uma de uma forma diferente. Esses conceitos estão diretamente relacionados ao que é proposto em Programação de Computadores II.

O CodeCombat começa abordando os conceitos mais básicos de programação: declaração de variáveis, operações aritméticas, operadores condicionais e estruturas de repetição. Este objeto de aprendizagem aprofunda-se em alguns outros temas, como classes e métodos, mas há limitações na área paga da ferramenta. Os conceitos abordados por este objeto de aprendizagem estão alinhados com o que é ensinado em Programação de Computadores I.

Não é surpreendente que na literatura ele seja comumente utilizado em trabalhos com alunos que estão iniciando a faculdade ou sendo introduzidos à programação.

O CodeWars é o objeto de aprendizagem que aborda o maior número de temas, pois possui uma grande escalabilidade no nível de seus desafios, atendendo desde iniciantes até programadores muito experientes. Os assuntos abordados incluem: lógica, declaração de variáveis, operações aritméticas, operadores condicionais, estruturas de repetição e estruturas de dados. Os assuntos abordados perpassam todas as disciplinas, dependendo do que for selecionado: Programação de Computadores I, Programação de Computadores II, Algoritmo e Estruturas de Dados I e Algoritmo e Estruturas de Dados II.

## 4.3 Avaliações dos OAs

### 4.3.1 Avaliação Literatura

#### 4.3.1.1 BlueJ

Vahldick (2007) diz que o BlueJ tende a ajudar no processo de aprendizagem por possuir recursos visuais interativos para situações que tendem a ser abstratas, como classes e relacionamentos, além da possibilidade de verificar o comportamento dos objetos em memória quando em execução. O autor cita como pontos negativos da ferramenta a sua baixa usabilidade para novos usuários, como a falta de ferramenta de auto completar ou assistente para ajudar o programador. Além disso o OA carece de uma simulação com mais recursos, com imagens melhores, com recursos de multimídia e com um mundo mais complexo.

Haaster e Hagan (2004) conclui após apresentar o BlueJ a três turmas de aluno que esse OA foi eficiente no ensino de programação orientada a objeto, mas alguns pontos negativos foram levantados: foram identificados alguns problemas de usabilidade, como uma documentação incompleta e mensagens de erros pouco específicas.

Alkazemi e Grami (2012) ao comparar o BlueJ com outras IDEs tradicionais percebe que especificamente para assuntos mais elaborados de programação orientada a objeto o BlueJ se destaca devido a capacidade de trabalhar assuntos abstratos. Um limitador que os autores levantam é a necessidade de um tempo maior para trabalhar a ferramenta nas aulas, seja em quantidade de aulas ou no tempo que o professor precisa pra aprender a ferramenta.

Xinogalos et al. (2006) atribui um consumo de tempo muito grande à utilização da ferramenta: ela se mostra eficiente mas pode não caber em um cenário com quantidade de aulas limitado. O autor atribui esse tempo gasto ao fato de que para utilizar a ferramenta explicações adicionais devem ser dadas e o fluxo cronológico de explicação de conceitos subvertidos.

Hagan e Markham (2000) observam como o BlueJ poderia ajudar os alunos de uma turma e quais seriam as dificuldades de utilização apresentadas. Dentre as dificuldades apresentadas, foi observado que alguns alunos encontraram dificuldade de instalação em casa, pois a versão disponibilizada no site oficial e a da escola eram diferentes. Alguns alunos relataram problemas de performance quando o programa foi utilizado para escrever códigos mais pesados e também houve reclamações a respeito da falta de mensagens claras de erro. A



conclusão dos autores foi de que o BlueJ se mostra eficiente no que se propõe a fazer, mas deve ser utilizado em conjunto com outras técnicas de ensino.

O objeto de aprendizagem BlueJ é considerado uma referência no ensino de programação orientada a objetos. É uma ferramenta muito elogiada por diversos autores por conseguir explicar bem conceitos abstratos. No entanto, alguns pontos negativos são frequentemente mencionados, como reclamações sobre a usabilidade, a falta de documentação e o esforço necessário, especialmente por parte do professor ou de alunos que estão estudando de forma independente, para compreender como utilizar a ferramenta.

Tabela 3 – Pontos Fortes e Fracos: BlueJ

Pontos Forte	Pontos Fraco
Aborda com Maestria um Tema Específico	Usabilidade
Bem Avaliado na Literatura	Não Aguenta Rodar Códigos Complexos
Gratuita	Documentação
Facil Instalação	Demanda Tempo para Aprender a Utilizar
	Falta de Mensagens de Erros

#### 4.3.1.2 CodeCombat

Ribeiro, Merlin e Fülber (2019) observaram que o principal benefício foi relacionado a motivação dos alunos: estes se mostraram empolgados com a ideia de aprender através de um jogo, porém os autores deixam claro que o ambiente gamificado não necessariamente causa aumento significativo da abstração dos conceitos abordados.

Barbosa (2017) avaliam vários objetos de aprendizagem nos aspectos técnicos, pedagógicos e usabilidade. Na avaliação técnica proposta o CodeCombat teve nota máxima nos atributos: navegação, mapeamento, compatibilidade espacial do conhecimento, design de tela e apresentação da informação, e teve atribuído a nota 3 nos critérios: facilidade de uso e carga cognitiva. Já no aspecto pedagógico objetividade, motivação e aprendizado cooperativo tiveram nota 3 e Valorização de erro e Controle de aluno nota 2. Os autores concluíram que a avaliação feita do ponto de vista pedagógico é difícil de ser realizada pois essas avaliações não levam em conta a natureza do objeto e não captam os aspectos da ferramenta em sua totalidade.

Zanchett, Vahldick e Raabe (2015) apresentam o CodeCombat para avaliação de um grupo de alunos. Os alunos se mostraram motivados ao utilizar o CodeCombat e relataram que acharam de fácil utilização dada a presença de vários indicadores de ajuda e tutoriais. Um lado que foi levantado como negativo é a alta repetição existente, por mais que seja uma técnica de fixação os alunos se sentiam entediados quando se deparando com conceitos simples sendo repetidos.

Severgnini (2020) avalia que o CodeCombat é intuitivo de utilizar e uma ferramenta útil para despertar a motivação dos alunos, porém ele constata que a parte do jogo acaba sendo

apenas uma desculpa para ensinar programar e que seria positivo se tivesse uma história para os alunos se interessarem já que se levado so pela jogabilidade a tendência é rapidamente a ferramenta cair na repetição. Foram destacadas como características positivas a usabilidade e acessibilidade, possuindo suporte para várias línguas.

O CodeCombat aposta na motivação e diversão como principais ferramentas para estimular o processo de ensino de programação. Em vários trabalhos, é relatada a usabilidade e a facilidade de acesso como pontos positivos. Em relação à usabilidade, a ferramenta oferece diversas formas de apoiar tanto o aluno como o professor no processo de aprendizagem. Ela deixa claro os objetivos, permite a correção dos códigos feitos e disponibiliza vários tutoriais e guias para ajudar na compreensão e resolução dos problemas. O aspecto de jogo proposto é sempre mencionado como um motivo de motivação e até mesmo surpresa, pois consegue combinar eficientemente o ensino e a diversão.

No entanto, alguns aspectos negativos são observados, como a falta de um trabalho mais refinado nos aspectos pedagógicos da ferramenta, algumas reclamações em relação aos desafios que são muito repetitivos e demoram para mudar o conceito estudado.

Tabela 4 – Pontos Fortes e Fracos: CodeCombat

Pontos Forte	Pontos Fraco
Motivação	Fraco Em Aspectos Pedagógicos
Usabilidade	Enjoa Fácil
Fácil Acesso	Falta de Uma Forma de Avaliar o Aprendizado
Disponível em Várias Línguas	Estudos Levantam que Pode Não ter Ganho na Qualidade Do Ensino, Apenas na Motivação
Boa Avaliação Técnica	
Mensagens de Erro Claras	
Disponibilidade em Português	
Plataforma para o Professor	

#### 4.3.1.3 CodeWars

Segundo Zhang et al. (2023) o CodeWars se diferencia de outras plataformas de treino de escrita de código pois além de corrigir a resposta enviada, também possui um sistema de pontuação para que o usuário se sinta motivado a sempre evoluir e buscar novos desafios.

O CodeWars não é uma ferramenta que aparece muito na literatura, porém ela é muito recomendada quando se pesquisa no Google por ferramentas que podem ser utilizadas para aprendizado de programação. Foi selecionado cinco sites para entender o que é falado sobre essa OA e o porque da recomendação:

- <<https://www.freecodecamp.org>>
- <<https://www.hostinger.com.br/tutoriais/11-sites-aprender-como-programar-de-graca>>
- <<https://www.websiteplanet.com/pt-br/blog/>>
- <<https://imasters.com.br/desenvolvimento/>>
- <<https://ppbruna.medium.com>>

O CodeWars é recomendado por sua capacidade de auto alimentação, movido por sua comunidade e por sua alta escalabilidade. É uma plataforma que atende desde o mais iniciante ao avançado em programação e possui todas as ferramentas para poder dar o suporte necessário para o usuário. O aspecto gameficado fica por conta de gerar motivação enquanto a plataforma é alimentada por seus próprios usuários. O ponto forte desse OA pode também ser seu ponto fraco, já que um usuário pode ficar sem apoio caso a comunidade não se atente para tirar suas dúvidas.

Tabela 5 – Pontos Fortes e Fracos: CodeWars

Pontos Forte	Pontos Fraco
Muitas Linguagens de Programação Disponíveis	Fraco Em Aspectos Pedagógicos
Desafios Para Todos os Níveis	Pode Faltar Suporte
Correção e Mensagens de Erro	
Motivação Através da Gameificação	

#### 4.3.2 Avaliação dos OAs: Método LORI

Nesta seção, os objetos de aprendizagem escolhidos serão avaliados de acordo com o modelo LORI e suas notas explicadas.

## 4.3.2.1 CodeCombat

Tabela 6 – Avaliação do OA CodeCombat

<b>Tópicos Avaliados</b>	<b>Avaliação LORI</b>
<b>Qualidade de Conteúdo</b>	* * *
<b>Meta de Aprendizagem</b>	* * * * *
<b>Feedback e Adaptação</b>	* * * * *
<b>Motivação</b>	* * * *
<b>Design</b>	* * * *
<b>Usabilidade</b>	* * *
<b>Acessibilidade</b>	*
<b>Conformidade com Padrões</b>	* * *

Qualidade de Conteúdo - Nota 3\*: Este objeto de aprendizagem cumpre sua função no contexto de ensino proposto, mas limita-se a ensinar apenas comandos com explicações simples sobre o que está sendo realizado. Falta um maior detalhamento do que é ensinado e da teoria por trás do código, para que o aluno possa extrapolar o aprendizado para outras situações.

Meta de Aprendizagem - Nota 5\*: O objetivo de aprendizado é claro e direcionado ao público-alvo, e as atividades e avaliações propostas estão sempre alinhadas e progridem de acordo com a meta de aprendizagem. Essa nota é facilitada pela limitação da qualidade de conteúdo: o objeto de aprendizagem aborda intencionalmente os assuntos de forma mais limitada para o contexto proposto e, portanto, não possui metas de aprendizagem ousadas.

Feedback e Adaptação - Nota 5\*: Existe um bom sistema de feedback e adaptação para o aluno. O aluno recebe feedback sobre a qualidade da resposta e como ela poderia ser otimizada. Há também um feedback sobre o tempo gasto em cada atividade, além de dicas em caso de resposta incorreta.

Motivação - Nota 4\*: Este objeto de aprendizagem utiliza o formato de jogo RPG como um pilar para motivar os alunos, com um cenário medieval, vários personagens, equipamentos, armaduras, inimigos e tesouros. Avançar no conteúdo e resolver os desafios é a chave para desbloquear novos recursos visuais para o personagem. No entanto, mesmo com vários estímulos visuais e recompensas por avançar, alguns conteúdos acabam sendo repetitivos, pois o mesmo assunto é abordado em vários níveis diferentes, o que pode resultar em falta de interesse por parte do aluno.

Design - Nota 4\*: Por se tratar de um objeto de aprendizagem em forma de jogo, em alguns momentos há muitos elementos visuais na tela que não são claros. Para pessoas familiarizadas com jogos, pode ser bastante intuitivo entender onde tudo está localizado, mas o mesmo não ocorre necessariamente para pessoas menos acostumadas com esse tipo de ambiente.

Usabilidade - Nota 3\*: O objeto de aprendizagem é suficientemente consistente para que o usuário, com um pouco de prática, não tenha dificuldades em utilizá-lo. No entanto, por se tratar de um jogo, algumas funcionalidades ficam implícitas, o que pode gerar certa confusão. Vale destacar também que, por se tratar de uma plataforma online, há suscetibilidade a problemas de conexão provenientes do servidor da plataforma.

Acessibilidade - Nota 1\*: Não há suporte para pessoas com deficiência. As funcionalidades e desafios são limitados ao que é visualizado na tela e dependem fortemente da percepção de cores e leitura das instruções, que não possuem transcrição em áudio.

Conformidade com Padrões - Nota 3\*: Este objeto de aprendizagem se preocupa mais em mesclar o ensino com um jogo do que em seguir rigidamente os modelos padrões. Ainda assim, podemos observar alguma conformidade com os padrões, principalmente no que se diz respeito à preocupação com a qualidade de ensino e à utilização em múltiplas plataformas.

#### 4.3.2.2 BlueJ

Qualidade de Conteúdo - Nota 2\*: Este objeto de aprendizagem possui as ferramentas necessárias para explicar programação orientada a objetos, mas não é auto-suficiente. Os conceitos envolvidos dependem diretamente de um professor ou de outros materiais de apoio para serem entendidos. O objeto se limita ao contexto de aprendizagem de programação orientada a objetos e não pode ser extrapolado para outros contextos.

Meta de Aprendizagem - Nota 5\*: O escopo do objeto de aprendizagem é bastante limitado e sua meta é clara: facilitar a compreensão dos conceitos de programação orientada a objetos por meio da abstração.

Feedback e Adaptação - Nota 2\*: O feedback se limita a mostrar qual erro ocorre e em qual linha do código, mas pode ser pouco intuitivo para entender o que está sendo feito de errado, pressupondo um nível intermediário de compreensão de programação por parte do aluno. A capacidade de adaptação é limitada ao que se pode esperar de uma IDE.

Motivação - Nota 3\*: A motivação depende de outra pessoa que conheça os conceitos abordados para usar o objeto de aprendizagem junto com o aluno. Quando essa dependência

Tabela 7 – Avaliação do OA BlueJ

<b>Tópicos Avaliados</b>	<b>Avaliação LORI</b>
<b>Qualidade de Conteúdo</b>	* *
<b>Meta de Aprendizagem</b>	* * * * *
<b>Feedback e Adaptação</b>	* *
<b>Motivação</b>	* * *
<b>Design</b>	*
<b>Usabilidade</b>	*
<b>Acessibilidade</b>	*
<b>Conformidade com Padrões</b>	*

é cumprida, a ferramenta aumenta o nível de motivação ao representar visualmente um tema com alto grau de abstração.

**Design - Nota 1\*:** O design é pouco amigável para novos usuários: existem botões sem tags que tornam difícil identificar sua função sem testá-los exaustivamente. Os objetos na tela são dispostos de maneira pouco intuitiva e sem sinalização clara.

**Usabilidade - Nota 1\*:** O objeto de aprendizagem é complicado de usar sozinho: é necessário testar os botões e prestar atenção para entender o que está acontecendo. A navegação não é clara e o usuário pode ficar perdido, até mesmo para iniciar alguma atividade no OA. Falta documentação ou material de apoio para a utilização do objeto de aprendizagem, e parece perder o sentido quando exige tanto conhecimento prévio para ensinar um conceito novo. O objeto de aprendizagem funciona como uma IDE na prática, mas sem uma documentação clara, torna-se difícil de usar, mesmo com várias funcionalidades agregadas.

**Acessibilidade - Nota 1\*:** Não há suporte para pessoas com deficiência.

**Conformidade com Padrões - Nota 1\*:** O objeto de aprendizagem é extremamente limitado em escopo e assunto proposto, e não está em conformidade com padrões técnicos ou

pedagógicos. Também não demonstra possibilidade de reutilização em outros contextos além de ser utilizado por um professor como complemento ao que já foi ensinado.

#### 4.3.2.3 CodeWars

**Qualidade de Conteúdo - Nota 5\*:** Essa plataforma apresenta uma ampla variedade de conteúdos e desafios que abrangem desde o nível básico até o avançado, englobando várias linguagens de programação. A qualidade dos desafios é garantida pela própria comunidade, que não só contribui com novos desafios, mas também corrige problemas e remove conteúdos inadequados.

**Meta de Aprendizagem - Nota 4\*:** A meta de aprendizagem é determinada pelo próprio usuário, mas a plataforma oferece um sistema de níveis que o usuário pode almejar.

**Feedback e Adaptação - Nota 4\*:** O sistema de feedback é realizado pela comunidade que utiliza o objeto de aprendizagem: as soluções, dúvidas e dificuldades são discutidas por programadores mais experientes, sempre com foco na aprendizagem e otimização das soluções. No entanto, a dependência de ter alguém para responder uma dúvida pode gerar atrasos na resposta.

**Motivação - Nota 5\*:** O fato de o objeto de aprendizagem estar centrado em uma comunidade ativa, que gera novos desafios diariamente, tende a ser altamente motivador. Além disso, possui um sistema de ranking e a possibilidade de desafiar outros usuários para duelos de programação.

**Design - Nota 4\*:** A plataforma é agradável de usar e não apresenta grandes dificuldades para encontrar as funcionalidades desejadas. Algumas seleções são identificadas por cores e imagens, sendo necessário realizar testes e cliques para entender o que está acontecendo.

**Usabilidade - Nota 4\*:** Não há problemas de usabilidade, mas, devido à complexidade do objeto de aprendizagem, é necessário um tempo de adaptação e compreensão por parte do usuário para realizar todas as tarefas. Os botões são fáceis de encontrar e estão identificados.

**Acessibilidade - Nota 1\*:** Não há suporte para pessoas com deficiência.

**Conformidade com Padrões - Nota 3\*:** Esse objeto de aprendizagem está em conformidade com os padrões de design e usabilidade de sites, valorizando a experiência do usuário. No entanto, ainda é limitado em relação à conformidade com os padrões específicos voltados para objetos de aprendizagem.

Tabela 8 – Avaliação do OA CodeWars

<b>Tópicos Avaliados</b>	<b>Avaliação LORI</b>
<b>Qualidade de Conteúdo</b>	* * * * *
<b>Meta de Aprendizagem</b>	* * * *
<b>Feedback e Adaptação</b>	* * * *
<b>Motivação</b>	* * * * *
<b>Design</b>	* * * *
<b>Usabilidade</b>	* * * *
<b>Acessibilidade</b>	*
<b>Conformidade com Padrões</b>	* * *

### 4.3.3 Limites e Possibilidades

Os objetos de aprendizagem avaliados neste trabalho foram selecionados para abordar diferentes limites e possibilidades, a fim de compreender melhor os cenários em que podem ser utilizados.

#### 4.3.3.1 BlueJ

O objeto de aprendizagem BlueJ possui um escopo mais limitado em relação aos assuntos abordados em comparação aos outros OAs analisados. No entanto, isso não é necessariamente um ponto negativo, pois o BlueJ possui metas de aprendizagem claras e cumpre o que é proposto, sendo amplamente utilizado em trabalhos, estudos e recomendações relacionados à programação orientada a objetos.



O maior problema do BlueJ é a falta de autonomia, havendo uma grande dependência de auxílio de um professor ou de outras ferramentas para explicar os conceitos trabalhados, uma vez que o BlueJ apenas demonstra os conceitos na prática, sem se preocupar em explicá-los adequadamente. Há também uma deficiência em termos de documentação e dificuldade de adaptação, requerendo um esforço do professor para integrar o uso do OA no contexto da aula. A adaptação também é prejudicada pelo design, pois a ferramenta parece ter sido desenvolvida sem levar em consideração aspectos de acessibilidade e experiência do usuário.

A maior possibilidade de uso do OA BlueJ é como uma ferramenta auxiliar no ensino de programação orientada a objetos, de modo que, após a explanação completa do conteúdo, o OA seja introduzido para reforçar conceitos abstratos.

#### 4.3.3.2 CodeCombat

O CodeCombat oferece um ambiente interativo em forma de jogo RPG, visando gerar motivação por meio de uma proposta de ensino diferenciada da abordagem tradicional. O acesso e a usabilidade da ferramenta são considerados pontos fortes, sendo complementados pela existência de uma plataforma para os professores acompanharem o progresso dos alunos e receberem feedbacks fornecidos pela própria ferramenta.

No entanto, esse objeto de aprendizagem apresenta limitações em relação aos assuntos abordados. Às vezes, temas simples são repetidos diversas vezes, o que pode levar a reclamações, e o escopo dos temas é limitado na versão gratuita. Além disso, os assuntos são abordados levando em consideração a introdução à programação para um público mais jovem, o que pode não estar alinhado com a proposta de ensino de um curso de graduação.

Existe a possibilidade de uso do OA CodeCombat em disciplinas introdutórias sobre programação, como uma forma de diversificar a metodologia de ensino. No entanto, não parece viável utilizá-lo em várias aulas, uma vez que seu principal ponto forte, aumentar a motivação para aprender, pode se perder ao longo do tempo.

#### 4.3.3.3 CodeWars

O CodeWars utiliza sua comunidade para promover interação, correção e discussão dos desafios propostos, além de permitir a criação de novos desafios. É uma ferramenta que abrange uma ampla variedade de linguagens de programação e aborda muitos temas que exigem lógica e otimização para serem resolvidos.

No entanto, assim como muitas ferramentas criadas para auxiliar no processo de ensino e aprendizagem de programação, falta uma fundamentação pedagógica mais sólida.

A existência de diversos desafios criados pela comunidade facilita o trabalho do professor ao propor diferentes situações para ensinar um determinado assunto. Além disso, a correção automática do código com mensagens de erro específicas dentro do próprio OA é outro ponto facilitador.

Esse OA se mostra muito eficiente ao ser utilizado para aprender conceitos novos por meio dos desafios propostos, podendo ser aplicado em qualquer disciplina introdutória de

programação devido à vasta gama de assuntos abordados. O cenário ideal seria o professor utilizar a plataforma como apoio para desafiar os alunos por meio do sistema de rankings, incentivando a competição. Os desafios propostos se assemelham muito aos encontrados em maratonas e competições de programação.

#### 4.3.4 Tabela Comparativa

Abaixo, seguem duas tabelas comparativas dos pontos fortes (Tabela 9) e fracos (Tabela 10) dos objetos de aprendizagem estudados, na visão do autor deste trabalho e na visão de outros autores presentes na literatura.

Tabela 9 – Tabela Comparativa: Pontos Fortes

<b>Objeto de Aprendizagem</b>	<b>Pontos Forte (Autor)</b>	<b>Pontos Forte (Literatura)</b>
<b>BlueJ</b>	<b>Meta de Aprendizagem, Motivação</b>	<b>Meta de Aprendizagem, Consolidado na Literatura; Gratuito</b>
<b>CodeCombat</b>	<b>Motivação, Usabilidade, Meta de Aprendizagem, Feedback e Adaptação; Design</b>	<b>Motivação, Usabilidade, Fácil Acesso, Conteúdo, Sem Erros de Conteúdo</b>
<b>CodeWars</b>	<b>Qualidade de Conteúdo, Meta de Aprendizagem, Feedback e Adaptação, Motivação, Usabilidade e Design</b>	<b>Escopo de Assuntos e Linguagens de Programação Abrangente; Comunidade Ativa; Motivador; Feedback e Adaptação</b>

Tabela 10 – Tabela Comparativa: Pontos Fracos

<b>Objeto de Aprendizagem</b>	<b>Pontos Fraco (Autor)</b>	<b>Pontos Fraco (Literatura)</b>
<b>BlueJ</b>	<b>Qualidade de Conteúdo, Usabilidade, Feedback e Adaptação, Conformidade com Padrões, Acessibilidade, Design</b>	<b>Usabilidade, Robustez, Documentação, Demante Tempo de Aprendizado, Falta Suporte ao Usuário</b>
<b>CodeCombat</b>	<b>Conformidade com Padrões, Acessibilidade, Qualidade de Conteúdo</b>	<b>Aspectos Pedagógicos, Repetitivo, Escopo de Assuntos Abordados</b>
<b>CodeWars</b>	<b>Aspectos Pedagógicos, Conformidade com Padrões, Acessibilidade</b>	<b>Aspectos Pedagógicos, Disponível Apenas Em Ingles</b>

No que se diz respeito a comparação entre a avaliação utilizando o método LORI e as avaliações presentes na literatura, alguns pontos podem ser comentados.

O BlueJ passa a mesma percepção de ser uma ferramenta objetiva e que dentro de um universo limitado, programação orietanda a objeto, ele tem grande potencial. A usabilidade da ferramenta se mostra confusa e a falta de suporte, documentação e mensagens de erro contribuem muito para esse aspecto. Alguns autores reclamam que a ferramenta não suporta rodar algoritmos complexos, mas é um caso que parece fugir do que o objeto de aprendizagem propõe. A literatura não aborda a questao de designe e interface do usuário provavelmente por o software ou os trabalhos serem antigos, porém é algo que claramente ficou datado no OA.

O CodeCombat parece ser o mais amigável dos objetos de aprendizagem testados, as mensagens são claras, possui tutoriais, correções, mensagens de erro e suporte. Não atoa a usabilidade é um ponto forte claramente abordado na literatura. O conteúdo abordado não possui erros mas as vezes se torna muito repetitivo, as tarefas a serem realizadas são iguais ou parecidas durento todas as fazes. As reclamações que aparecem na literatura são evidentes ao utilizar o OA e o foco é realmente no quesito motivação através de um jogo.

O CodeWars é a ferramenta mais atual a ter sido desenvolvida e aparentemente houve um cuidado maior com a experiência do usuário. Na literatura há citações a esse OA mas não avaliações ou trabalhos a utilizando. Logo ao utilizar a ferramenta é possível perceber diversos pontos positivos na construção da OA, mas como a maioria dos lugares que o recomendam são sites acaba uqe os pontos fracos são omitidos.

Os autores na literatura se mostram muito preocupados com os aspectos pedagógicos dos OAs que por muitas vezes são negligenciados, não atoa sendo constantemente tratado como ponto fraco dos OAs. Apesar da preocupação a revisão bibliográfica não demonstrou melhora nesa característica ao longo dos anos. Já a questão de acessibilidade é completamente ignorada tanto pelos OAs quanto pelos autores na literatura, pelo menos, no que se diz respeito a OAs para o ensino de programação.

## 5 Considerações Finais

À medida que avançamos na busca por métodos e ferramentas que possam auxiliar de maneira eficaz o processo de ensino e aprendizagem de programação, observamos a criação de numerosos softwares e plataformas. No entanto, apesar de desempenharem um papel fundamental nessa tarefa, é importante salientar que muitas dessas ferramentas não são inicialmente concebidas como objetos de aprendizagem, embora, no fundo, desempenhem essa função. Essa lacuna na percepção e compreensão desses recursos como objetos de aprendizagem acaba gerando problemas recorrentes em relação ao seu aspecto pedagógico. Assim, é fundamental destacar a necessidade de uma reflexão mais aprofundada sobre a própria natureza e essência dos objetos de aprendizagem, bem como sua aplicação adequada no contexto educacional.

Não podemos afirmar com certeza se essa dificuldade em identificar e reconhecer plenamente os objetos de aprendizagem como tal é decorrente da própria complexidade inerente ao assunto abordado ou se está relacionada à forma como essas ferramentas são concebidas e utilizadas. No entanto, é inegável que os objetos de aprendizagem desenvolvidos, por si só, não são capazes de suprir todas as necessidades do processo de ensino e aprendizagem. Eles alcançam seu potencial máximo quando são utilizados em conjunto com outras estratégias de ensino, como complementos valiosos ou mediante uma intervenção direta e orientada do professor, a fim de proporcionar uma experiência de aprendizagem enriquecedora e significativa.

Outro aspecto que merece atenção e crítica é a falta de preocupação e investimento em tornar os objetos de aprendizagem acessíveis a todos os usuários, independentemente de suas habilidades ou limitações. Infelizmente, essa questão é frequentemente negligenciada, o que acarreta uma exclusão e marginalização desnecessárias de pessoas com deficiências ou dificuldades específicas.

No âmbito da literatura acadêmica, é importante ressaltar que são escassos os estudos que analisam os objetos de aprendizagem em conformidade com padrões específicos estabelecidos. Em vez disso, é mais comum encontrar pesquisas baseadas em estudos de caso, que podem, em certos casos, fornecer resultados inconclusivos ou enviesados devido às expectativas e perspectivas individuais dos autores. É válido mencionar, entretanto, que alguns pesquisadores defendem que os objetos de aprendizagem têm um impacto significativamente maior na motivação dos alunos do que propriamente na qualidade do ensino em si.

Diante desse contexto desafiador, é essencial promover discussões mais aprofundadas, incentivar pesquisas contínuas e colaborativas, bem como investir na melhoria e aprimoramento dos objetos de aprendizagem, a fim de maximizar seu potencial educacional e garantir uma experiência enriquecedora para todos os envolvidos no processo de ensino e aprendizagem de programação.

Dos objetos de aprendizagem apresentados, o CodeWars se destaca no contexto do ensino atual de programação. Dentre suas características que o tornam especialmente eficaz, destaca-se sua dinamicidade e simplicidade de uso. O CodeWars adota uma abordagem colaborativa, alimentada pela própria comunidade de usuários, o que garante uma constante evolução e atualização dos desafios propostos. Além disso, sua interface amigável e intuitiva permite que os usuários realizem as tarefas de forma independente, estimulando o desenvolvimento de habilidades autônomas. A adaptabilidade do CodeWars aos avanços tecnológicos e às novas linguagens de programação amplia ainda mais seu valor como objeto de aprendizagem. Mas não deve-se tratar os outros OAs como piores por causa disso, cada um tem seu espaço, seu modo de uso e seu objetivo específico de aprendizagem.

No entanto, apesar das vantagens e méritos do CodeWars, é válido ressaltar que ainda há espaço para aprimoramentos em sua abordagem como objeto de aprendizagem. Um aspecto que merece atenção é a incorporação de elementos pedagógicos mais sólidos, visando a criação de uma estrutura de apoio mais completa. Isso poderia beneficiar especialmente os alunos menos experientes, fornecendo orientação adicional e recursos educacionais complementares para fortalecer seu aprendizado de forma mais estruturada.

Diante desse cenário fica claro a importância de ferramentas de avaliação de objetos de aprendizagem, podendo assim entender o melhor contexto para que seja utilizado, extraindo assim, maior proveito dele. A importância também reside no fato de que aprimoramentos devem ser feitos e essas ferramentas são muito importantes para entender aonde estão as lacunas.

As possibilidades de uso de OA no ensino de programação são enormes mas para conceitos tão abstratos com uma quantidade de aulas pequena, é necessário que seja lapidada a questão pedagógica para que os limites sejam cada vez mais raros e as possibilidades cada vez maiores.

## 6 Trabalhos Futuros

Com base na revisão bibliográfica e avaliação dos objetos de aprendizagem utilizados em disciplinas introdutórias de programação, bem como na comparação dos resultados da avaliação da literatura com os resultados da avaliação utilizando o método LORI, surgem diversas oportunidades para trabalhos futuros. Essas possibilidades podem contribuir para o aprimoramento e a eficácia do uso de objetos de aprendizagem no contexto do ensino de programação. A seguir, são apresentadas algumas sugestões de trabalhos futuros.

Uma possível extensão deste estudo seria a realização de avaliações adicionais utilizando o método LORI em um número maior de objetos de aprendizagem relacionados ao ensino de programação. Essa abordagem permitiria uma análise mais abrangente e representativa das características e funcionalidades desses objetos, a fim de identificar pontos fortes e fracos adicionais. Com uma amostra maior de objetos de aprendizagem, seria possível obter resultados mais sólidos e fornecer recomendações mais robustas para o uso desses recursos no contexto específico do curso. Vários autores na literatura relatam a dificuldade de se encontrar avaliações feitas com métodos especificamente feitos para objetos de aprendizagem.

Uma análise comparativa entre diferentes métodos de avaliação de objetos de aprendizagem, além do LORI, poderia contribuir para uma compreensão mais completa e abrangente das características dos objetos avaliados. Comparar o LORI com outros métodos, poderia fornecer insights valiosos sobre diferentes abordagens de avaliação. Essa análise comparativa poderia ser realizada considerando critérios como eficácia, eficiência, satisfação do usuário, facilidade de uso e adequação ao contexto educacional.

Estudar causa e consequência dos dois pontos mais frágeis dos objetos de aprendizagem voltados para o ensino de programação: aspectos pedagógicos e acessibilidade. Isso envolveria a revisão das diretrizes de acessibilidade, a avaliação dos objetos existentes com foco nesses temas, o aprimoramento dos recursos acessíveis e o desenvolvimento de novos objetos de aprendizagem com foco na inclusão de estudantes com necessidades especiais.

# Referências

- ALKAZEMI, B. Y.; GRAMI, G. M. Utilizing bluej to teach polymorphism in an advanced object-oriented programming course. *Journal of Information Technology Education. Innovations in Practice*, Informing Science Institute, v. 11, p. 271, 2012. Citado nas páginas 23 e 46.
- ALMEIDA, R. R.; CHAVES, A. C. L.; JR, C. F. de A. Avaliação de objetos de aprendizagem: aspectos a serem considerados neste processo. *Revista Educação & Tecnologia*, n. 13, 2015. Citado nas páginas 13, 24 e 25.
- ASSIS, L. S. d. Concepções de professores de matemática quanto à utilização de objetos de aprendizagem: um estudo de caso do projeto rived-brasil. Pontifícia Universidade Católica de São Paulo, 2005. Citado na página 17.
- ASTUDILLO, G. J.; WILLGING, P.; GARCIA, P. Estado de arte de los repositorios de materiales educativos en latinoamérica. In: *VI Congreso de Tecnología en Educación y Educación en Tecnología*. [S.l.: s.n.], 2011. Citado nas páginas 20 e 21.
- BARBOSA, A. V. d. S. Seleção e avaliação de ferramentas para a promoção do pensamento computacional. Universidade Federal da Paraíba, 2017. Citado nas páginas 24 e 47.
- BECK, R. Learning objects: What? *Center for Internation Education, University of Winsconsin*, 2002. Citado na página 14.
- BOFF, E.; REATEGUI, E. A importância do processo de avaliação de software educativo. *Seminário Nacional de Tecnologia na Educação*, v. 2, 2005. Citado na página 24.
- BOTELLA, P. et al. Iso/iec 9126 in practice: what do we need to know. In: *Software Measurement European Forum*. [S.l.: s.n.], 2004. v. 2004. Citado na página 18.
- BRAGA, J. C. et al. Desafios para o desenvolvimento de objetos de aprendizagem reutilizáveis e de qualidade. In: *Anais do Workshop de Desafios da Computação Aplicada à Educação*. [S.l.: s.n.], 2012. p. 90–99. Citado nas páginas 18, 19 e 20.
- BRAGA, J. C.; PIMENTEL, E.; DOTTA, S. Metodologia intera para o desenvolvimento de objetos de aprendizagem. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2013. v. 24, n. 1. Citado nas páginas 14, 19 e 20.
- CARDOSO, R.; SANTOS, O.; GATTI, D. C. Revisão sistemática de objetos de aprendizagem para o ensino de computação. *Anais temporários do LACLO 2015*, v. 10, n. 1, p. 389, 2015. Citado na página 22.
- CESTEROS, A. M. F.-P. Development of a spanish standard for quality assessment of digital educational material. *IEEE revista iberoamericana de tecnologías del aprendizaje*, IEEE, v. 9, n. 4, p. 151–158, 2014. Citado na página 24.
- CHURCHILL, D. Towards a useful classification of learning objects. *Educational Technology Research and Development*, Springer, v. 55, n. 5, p. 479–497, 2007. Citado na página 15.
- DEHNADI, S.; BORNAT, R. et al. The camel has two humps (working title). *Middlesex University, UK*, p. 1–21, 2006. Citado na página 11.



- DROMEY, R. G. Cornering the chimera [software quality]. *IEEE Software*, IEEE, v. 13, n. 1, p. 33–43, 1996. Citado na página 18.
- DUNICAN, E. Making the analogy: Alternative delivery techniques for first year programming courses. 2002. Citado na página 11.
- FERNÁNDEZ-PAMPILLÓN, A. M. A new aenor project for measuring the quality of digital educational materials. In: *Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality*. [S.l.: s.n.], 2013. p. 133–139. Citado na página 24.
- FRIESEN, N. Three objections to learning objects and e-learning standards. *Online education using learning objects*, p. 59–70, 2004. Citado na página 17.
- GALAFASSI, F. P.; GLUZ, J. C.; GALAFASSI, C. Análise crítica das pesquisas recentes sobre as tecnologias de objetos de aprendizagem e ambientes virtuais de aprendizagem. *Revista Brasileira de Informática na Educação*, v. 21, n. 03, p. 100, 2013. Citado na página 17.
- GIL, A. C. et al. *Como elaborar projetos de pesquisa*. [S.l.: Atlas São Paulo, 2002. v. 4. Citado na página 32.
- GIRAFFA, M. M.; MORA, M. da C. Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. In: *Congressos CLABES*. [S.l.: s.n.], 2013. Citado na página 12.
- GOMES; HENRIQUES; MENDES. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias-ISSN 1646-933X*, v. 1, n. 1, p. 93–103, 2008. Citado nas páginas 11 e 22.
- GOMES, A. d. J. *Dificuldades de aprendizagem de programação de computadores: contributos para a sua compreensão e resolução*. 2010. Tese (Doutorado), 2010. Citado na página 11.
- GÜRER, M. D.; YILDIRIM, Z. Effectiveness of learning objects in primary school social studies education: achievement, perceived learning, engagement and usability. *Egitim ve Bilim, Turk Egitim Dernegi*, v. 39, n. 176, 2014. Citado na página 24.
- HAASTER, K. V.; HAGAN, D. Teaching and learning with bluej: an evaluation of a pedagogical tool. *Issues in Informing Science & Information Technology*, CiteSeer, v. 1, 2004. Citado nas páginas 23 e 46.
- HAGAN, D.; MARKHAM, S. Teaching java with the bluej environment. In: *Proceedings of Australasian Society for Computers in Learning in Tertiary Education Conference ASCILITE*. [S.l.: s.n.], 2000. Citado nas páginas 23 e 46.
- HENRIQUE, M. S.; REBOUÇAS, A. D. D. S. Objetos de aprendizagem para auxiliar o ensino de conceitos do paradigma de programação orientada a objetos. *RENOTE*, v. 13, n. 2, 2015. Citado na página 21.
- HINTERHOLZ, O. Tepequém: uma nova ferramenta para o ensino de algoritmos nos cursos superiores em computação. In: *XVII-Anais do Workshop sobre Educação em Informática*. [S.l.: s.n.], 2009. v. 20, p. 21. Citado na página 12.
- HODGINS, H. W. The future of learning objects. *Educational Technology*, JSTOR, p. 49–54, 2006. Citado na página 14.
- JENKINS, T. On the difficulty of learning to program. In: CITESEER. *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*. [S.l.], 2002. v. 4, n. 2002, p. 53–58. Citado na página 11.

- JESUS, Â.; GOMES, M. J.; CRUZ, A. Case based learning digital: proposta para estruturação da formação. *XII CIGP Psicopedagogia 2013*, Universidade do Minho, Instituto de Educação, Centro de Investigação em Educação, 2013. Citado na página 11.
- JESUS, A. N. de et al. Objetos de aprendizagem no ensino de lógica de programação. *Revista de Informática Aplicada*, v. 3, n. 2, 2007. Citado na página 21.
- LEFFA, V. J. Nem tudo o que balança cai: objetos de aprendizagem no ensino de línguas. *Polifonia*, v. 12, n. 12 (2), 2006. Citado na página 21.
- LIMA, T. C. S. d.; MIOTO, R. C. T. Procedimentos metodológicos na construção do conhecimento científico: a pesquisa bibliográfica. *Revista katálysis*, SciELO Brasil, v. 10, p. 37–45, 2007. Citado na página 32.
- LISTER, R. On blooming first year programming, and its blooming assessment. In: *Proceedings of the Australasian conference on Computing education*. [S.l.: s.n.], 2000. p. 158–162. Citado na página 11.
- LUCCHESI, E. M. et al. Construindo objetos de aprendizagem e pensando em geometria. *RENOTE*, v. 4, n. 2, 2006. Citado na página 21.
- MACHADO, E. d. C.; FILHO, C. S. S. O computador como agente transformador da educação e o papel do objeto de aprendizagem. *SEMINÁRIO NACIONAL ABED DE EDUCAÇÃO A DISTÂNCIA*, 2003. Citado na página 14.
- MARZAL, M. Á.; PEDRAZZI, S. Educational potential of topic maps and learning objects for m-learning in the knowledge society. *Transinformação*, SciELO Brasil, v. 27, p. 229–244, 2015. Citado na página 24.
- MCCRACKEN, M. et al. A multi-national, multi-institutional study of assessment of programming skills of first-year cs students. In: *Working group reports from ITiCSE on Innovation and technology in computer science education*. [S.l.: s.n.], 2001. p. 125–180. Citado na página 11.
- MCGREAL, R. Learning objects: A practical definition. *International Journal of Instructional Technology and Distance Learning (IJITDL)*, v. 9, n. 1, 2004. Citado na página 14.
- MORTIMER, L. Learning objects of desire: Promise and practicality. *Learning Circuits*, 2002. Citado na página 14.
- NASCIMENTO, P. do et al. Recomendação de objetos de aprendizagem baseada em modelos de estilos de aprendizagem: Uma revisão sistemática da literatura. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2017. v. 28, n. 1, p. 213. Citado na página 12.
- NESBIT, J.; BELFER, K.; LEACOCK, T. Learning object review instrument (lori): User manul. 2009. Citado nas páginas 25 e 26.
- OLIVEIRA, M. M. d. Como fazer pesquisa qualitativa. In: *Como fazer pesquisa qualitativa*. [S.l.: s.n.], 2013. p. 232–232. Citado na página 32.
- OLIVEIRA, V. L. P. d. Vivenciando objetos de aprendizagem na perspectiva da aprendizagem significativa: análise de uma formação continuada desenvolvida com um grupo de professores de matemática de ipatinga (mg). Programa de Pós-Graduação em Educação Matemática. Departamento de Matemática . . . , 2013. Citado nas páginas 16 e 17.

- PÖLDOJA, H. et al. Designing learning objects for social constructivist learning. In: *eLearning in Higher education. EUDORA intensive program ELHE during summer school in Viljandi, Estonia, 2004*. [S.l.: s.n.], 2005. p. 196–213. Citado na página 14.
- POLSANI, P. Use and abuse of reusable learning objects journal of digital information. Retrieved via the Wayback Machine on, v. 14, n. 10, p. 2012, 2003. Citado na página 14.
- RAMOS, A. F. et al. Uma experiência com objetos de aprendizagem no ensino da matemática. In: *Anais do 12º Congresso Internacional de Educação à Distância. Florianópolis, Setembro*. [S.l.: s.n.], 2005. Citado na página 14.
- RIBEIRO, F.; MERLIN, B.; FÜLBER, H. Avaliação do impacto de ambientes gamificados no processo de ensino-aprendizagem de programação de computadores: uma comparação entre elementos monousuário e multiusuário. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2019. v. 30, n. 1, p. 803. Citado nas páginas 23 e 47.
- RICHARDS, G. et al. The evolution of learning object repository technologies: Portals for on-line objects for learning. *International Journal of E-Learning & Distance Education/Revue internationale du e-learning et la formation à distance*, v. 17, n. 3, p. 67–79, 2002. Citado na página 21.
- ROMERO, T. R. L.; ANDRADE, R. d.; PIETROCOLA, M. Parâmetros para análise de roteiros de objetos de aprendizagem. *Simpósio Nacional de Ensino de Física*, v. 18, 2009. Citado na página 24.
- SÁ-SILVA, J. R. et al. Pesquisa documental: pistas teóricas e metodológicas. *Revista brasileira de história & ciências sociais*, São Leopoldo, v. 1, n. 1, p. 1–15, 2009. Citado na página 32.
- SANTOS, M. W. A. d. Trivia, objeto de aprendizagem com gamificação para apoio à disciplina de introdução à programação. 2015. Citado na página 22.
- SCHWARZELMÜLLER, A. F.; ORNELLAS, B. Os objetos digitais e suas utilizações no processo de ensino-aprendizagem. In: *Primeira Conferência Latinoamericana de Objetos de Aprendizaje*. [S.l.: s.n.], 2006. Citado na página 14.
- SEVERGNINI, L. F. Serious games e o desenvolvimento do pensamento computacional: uma abordagem vigotskiana. 2020. Citado na página 47.
- SILVA, B. V. Objeto de aprendizagem como instrumento de ensino-aprendizagem de conteúdos da disciplina programação de computadores em um curso de engenharia de computação: Uma análise à luz da teoria da atividade. In: . [S.l.]: Dissertação CEFET-MG campus Timoteo Curso de Engenharia de Computação, 2015. Citado nas páginas 18, 19 e 22.
- SILVA, R. S. da. Objetos de aprendizagem para educação a distância. 2011. Citado na página 18.
- SILVA, S. L. R. d.; ANDRADE, A. V. C. d.; BRINATTI, A. M. Ensino remoto emergencial. *dos Autores*, 2020. Citado na página 15.
- SOUSA, A. S. de; OLIVEIRA, G. S. de; ALVES, L. H. A pesquisa bibliográfica: princípios e fundamentos. *Cadernos da FUCAMP*, v. 20, n. 43, 2021. Citado na página 32.
- SOUTH, J. B.; MONSON, D. W. A university-wide system for creating, capturing, and delivering learning objects. *The instructional use of learning objects*, AECT, 2000. Citado na página 14.

- SOUZA, D. M.; BATISTA, M. d. S.; BARBOSA, E. F. Problemas e dificuldades no ensino e na aprendizagem de programação: Um mapeamento sistemático. *Revista Brasileira de Informática na Educação*, v. 24, n. 1, p. 39–52, 2016. Citado na página 11.
- SOUZA, M. d. F. C. de et al. Locpn: redes de petri coloridas na produção de objetos de aprendizagem. *Revista Brasileira de Informática na Educação*, v. 15, n. 3, 2007. Citado na página 24.
- SOUZA, M. de F.; PEQUENO, M. C.; FILHO, J. A. C. Professor x software educativo: a difícil tarefa de escolher. 2005. Citado na página 24.
- SOUZA, M. T. de; MARCELINO, R.; FORTUNATO, I. O lori como método de avaliação de objetos de aprendizagem: estudo de revisão. *Revista de estudos aplicados em Educação*, v. 3, n. 5, 2018. Citado na página 13.
- TABUTI, L. M.; PUGA, S.; BRITO, M. M. Construção do objeto de aprendizagem paciência como recurso didático para o ensino da disciplina estruturas de dados. In: *Nuevas Ideas en Informática Educativa Memorias del IXX Congreso Internacional. TISE*. [S.l.: s.n.], 2014. p. 55–62. Citado na página 22.
- VAHLICK, A. Uma experiência lúdica no ensino de programação orientada a objetos. In: *Workshop de Ambientes de Apoio à Aprendizagem de Algoritmos e Programação—Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2007. Citado nas páginas 23 e 46.
- WILEY, D. A. et al. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. *The instructional use of learning objects*, v. 2830, n. 435, p. 1–35, 2000. Citado nas páginas 14, 16 e 17.
- XINOGALOS, S. et al. Teaching oop with bluej: A case study. In: IEEE. *Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06)*. [S.l.], 2006. p. 944–946. Citado nas páginas 23 e 46.
- ZANCHETT, G.; VAHLICK, A.; RAABE, A. Jogos de programar como uma abordagem para os primeiros contatos dos estudantes com a programação. In: *Workshop do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2015. Citado na página 47.
- ZAPATA, A. et al. Evaluation and selection of group recommendation strategies for collaborative searching of learning objects. *International Journal of Human-Computer Studies*, Elsevier, v. 76, p. 22–39, 2015. Citado na página 24.
- ZHANG, Z. et al. Evaluating gpt's programming capability through codewars' katas. *arXiv preprint arXiv:2306.01784*, 2023. Citado na página 49.

**ESTUDO DE OBJETOS DE APRENDIZAGEM APLICÁVEIS NO  
ENSINO DE PROGRAMAÇÃO DE COMPUTADORES NA  
GRADUAÇÃO**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, como requisito parcial para obtenção do título de Engenheiro de Computação.

Trabalho aprovado. Timóteo, 13 de Julho de 2023:



Prof. Dr. Viviane  
Cota Silva  
Orientador



Prof. Dr. Lucas  
Pantuza  
Professor Convidado



Prof. Luciano  
Moreira  
Professor Convidado

Timóteo  
2024