

**CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
MESTRADO PROFISSIONAL EM EDUCAÇÃO E NOVAS
TECNOLOGIAS**

GILBERTO ANTONIO MÜLLER SOBRINHO

**AULA GAMIFICADA DE PROGRAMAÇÃO ORIENTADA A
OBJETOS**

CURITIBA

2023

**CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
MESTRADO PROFISSIONAL EM EDUCAÇÃO E NOVAS TECNOLOGIAS**

GILBERTO ANTONIO MÜLLER SOBRINHO

AULA GAMIFICADA DE PROGRAMAÇÃO ORIENTADA A OBJETOS

CURITIBA

2023

GILBERTO ANTONIO MÜLLER SOBRINHO

AULA GAMIFICADA DE PROGRAMAÇÃO ORIENTADA A OBJETOS

Dissertação apresentada ao Programa de Pós-Graduação – Mestrado Profissional em Educação e Novas Tecnologias, como parte dos requisitos necessários para obtenção do grau de Mestre em Educação e Novas Tecnologias.

Área de Concentração: Educação

Orientador: Prof. Dr. Rodrigo Otávio dos Santos

CURITIBA

2023

M958a Müller Sobrinho, Gilberto Antonio
Aula gamificada de programação orientada a objetos /
Gilberto Antonio Müller Sobrinho. – Curitiba, 2023.
145 f. : il. (algumas color.)
Orientador: Prof. Dr. Rodrigo Otávio dos Santos
Dissertação (Mestrado Profissional em Educação e Novas
Tecnologias) – Centro Universitário Internacional Uninter.
1. Gamificação. 2. Programação orientada a objetos
(Computação). 3. Aprendizagem. 4. Java (Linguagem de
programação de computador). 5. Tecnologia educacional. I.
Título.

CDD 371.334

Catálogo na fonte: Vanda Fattori Dias - CRB-9/547



CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
PRO-REITORIA DE POS-GRADUAÇÃO, PESQUISA E EXTENSÃO-PGPE
PROGRAMA DE MESTRADO E DOUTORADO PROFISSIONAL EM EDUCAÇÃO E NOVAS TECNOLOGIAS
Secretaria do Mestrado e Doutorado Profissional em Educação e Novas Tecnologias

Defesa Nº 008/2023

**ATA DE DEFESA DE DISSERTAÇÃO PARA CONCESSÃO DO GRAU DE MESTRE EM
EDUCAÇÃO E NOVAS TECNOLOGIAS**

No dia 12 de junho de 2023, às 16h, reuniu-se via web conferência a Banca Examinadora designada pelo Colegiado do Programa de Mestrado e Doutorado Profissional em Educação e Novas Tecnologias, composta pelos professores doutores: Rodrigo Otávio dos Santos (Presidente-Orientador-PPGENT/UNINTER); João Augusto Mattar Neto (Integrante Externo/ PUC-SP);

Luciano Frontino de Medeiros (Integrante Interno Titular - PPGENT/UNINTER); André Luis Cavazzani (Integrante Interno Suplente - PPGENT/UNINTER), para julgamento da dissertação: "AULA GAMIFICADA DE PROGRAMAÇÃO ORIENTADA A OBJETOS", do mestrando Gilberto Antonio Muller Sobrinho. O presidente abriu a sessão apresentando os professores membros da banca, passando a palavra em seguida ao mestrando, lembrando-lhe de que teria até vinte minutos para expor oralmente o seu trabalho. Concluída a exposição, o candidato foi arguido oralmente pelos membros da banca.

Concluída a arguição, a Banca Examinadora reuniu-se e comunicou o Parecer Final de que o mestrando foi:

(X) APROVADO, devendo o candidato entregar a versão final no prazo máximo de 60 dias.

() APROVADO somente após satisfazer as exigências e, ou, recomendações propostas pela banca, no prazo fixado de 60 dias.

() REPROVADO.



O Presidente da Banca Examinadora declarou que o candidato foi aprovado e cumpriu todos os requisitos para obtenção do título de Mestre em Educação e Novas Tecnologias, devendo encaminhar à Coordenação, em até 60 dias, a contar desta data, a versão final da dissertação devidamente aprovada pelo professor orientador, no formato impresso e PDF, conforme procedimentos que serão encaminhados pela secretaria do Programa. Encerrada a sessão, lavrou-se a presente ata que vai assinada pela Banca Examinadora.

Recomendações: Fazer uma grande revisão de texto, diferenciar título da dissertação e do produto, rever estrutura e metodologia do trabalho conforme apontado pela banca.



Dr. Rodrigo Otávio dos Santos
Presidente da Banca

Documento assinado digitalmente
gov.br JOAO AUGUSTO MATTAR NETO
Data: 16/06/2023 10:05:07-0300
Verifique em <https://validar.itl.gov.br>

Dr. João Augusto Mattar Neto
Integrante Externo



Dr. Luciano Frontino de Medeiros
Integrante Interno Titular

Dra. André Luis Moscaleski Cavazzani
Integrante Interno Suplente



Gilberto Antonio Muller Sobrinho
Mestrando

Para Rachel, Sophia, Antônio e Joana

AGRADECIMENTOS

Gostaria de expressar minha sincera gratidão a todos que me apoiaram durante o processo de elaboração desta dissertação e durante toda a jornada do programa do mestrado.

Primeiramente, agradeço à minha esposa, que esteve ao meu lado em todos os momentos importantes da minha vida, oferecendo seu apoio incondicional. Aos meus filhos, que são minha inspiração e motivo de orgulho, e me ajudaram a manter a motivação em todos os momentos.

Também gostaria de agradecer à minha sogra, uma referência excepcional tanto na vida profissional quanto pessoal, e ao meu professor orientador, que demonstrou uma paciência ímpar, ofereceu orientações valiosas e compartilhou conhecimento e inspiração durante nossos encontros na sala de aula, no grupo de pesquisa e nas videoconferências.

Agradeço aos professores do programa e colegas de classe pela rica troca de experiências e pelo aprendizado proporcionado durante as atividades acadêmicas. Vocês fazem um trabalho incrível em todo o Brasil.

Sou grato também à secretária do programa, sempre muito prestativa. Aos amigos que participaram da roda de conversa e aos professores da banca examinadora, que dedicaram seu valioso tempo e conhecimento para contribuir com este trabalho.

Sem o apoio e incentivo dessas pessoas extraordinárias, a realização desta dissertação não teria sido possível. A todos vocês, minha mais sincera gratidão.

“A educação é a arma mais poderosa que você pode usar para mudar o mundo.”

- Nelson Mandela

AULA GAMIFICADA DE PROGRAMAÇÃO ORIENTADA A OBJETOS

RESUMO

Esta pesquisa tem como objetivo desenvolver uma aula gamificada sobre programação orientada a objetos (POO), com base nos princípios e nas boas práticas identificadas na literatura e de profissionais da área, adaptando os elementos e mecânicas de jogos às características e necessidades dos estudantes e aos objetivos pedagógicos e conteúdos a serem ensinados. Toma como questão de pesquisa: como a gamificação pode ser aplicada no ensino da programação orientada a objetos e quais são seus impactos na aprendizagem e na motivação dos estudantes ao aprender POO? As considerações teóricas incluem fundamentação sobre a linguagem de programação Java, a programação orientada a objetos e a gamificação, com conceitos e aplicações práticas, e a compreensão de aula no contexto da educação. Para coletar dados empíricos, foi realizada uma roda de conversa com base em Weller (2010) com profissionais da área de engenharia de software em vários países, a fim de compreender a relevância da linguagem Java no mercado de trabalho internacional e sua importância. A pesquisa tem como resultado uma aula que utiliza elementos de gamificação, incluindo pontuação, feedback instantâneo, desafios e recompensas, para ensinar programação orientada a objetos com a linguagem Java. Conclui-se que a gamificação pode ser uma estratégia de motivação para ensinar programação orientada a objetos, tornando o processo de aprendizagem mais envolvente para o engajamento dos estudantes. O envolvimento gera mais e melhor aprendizagem.

Palavras-chave: Gamificação, Aprendizagem, Programação orientada a objetos, Java.

GAMIFIED OBJECT-ORIENTED PROGRAMMING CLASS

ABSTRACT

This research aims to develop a gamified class on object-oriented programming (OOP), based on the principles and best practices identified in the literature and from professionals in the area, adapting the elements and mechanics of games to the characteristics and needs of students and the pedagogical objectives and content to be taught. The research question is: How can gamification be applied to the teaching of object-oriented programming and what are its impacts on students' learning and motivation when learning OOP? Theoretical considerations include grounding on the Java programming language, object-oriented programming, and gamification, with concepts and practical applications, and understanding class in the context of education. To collect empirical data, a conversation wheel was held based on Weller (2010) with software engineering professionals in various countries in order to understand the relevance of the Java language in the international job market and its importance. The research results in a class that uses gamification elements, including scoring, instant feedback, challenges, and rewards, to teach object-oriented programming with the Java language. It is concluded that gamification can be a motivational strategy to teach object-oriented programming, making the learning process more engaging for student engagement. Involvement generates more and better learning.

Keywords: Gamification, Learning, Object-oriented programming, Java.

LISTA DE QUADROS E TABELAS

Quadros	p.
Quadro 1 – Eixos estruturais da Política Nacional de Educação Digital	19
Quadro 2 – Palavras-chave da definição de gamificação	81
Tabelas	
Tabela 1 – Oferta de cursos de nível superior	28
Tabela 2 – Guia rápido de recompensas	132

LISTA DE ABREVIATURAS E SIGLAS

AI	<i>Artificial Intelligence</i> ou Inteligência Artificial em inglês
ALGOL	<i>Algorithmic Language</i> ou Linguagem Algorítmica em inglês
API	<i>Application Programming Interface</i> ou Interface de Programação da Aplicação em inglês
AR	<i>Augmented Reality</i> ou Realidade Aumentada em inglês
BASIC	<i>Beginners All-purpose Symbolic Instruction Code</i> ou Código de Instrução Simbólica para Iniciantes para Todos os Fins em inglês
BTD	Banco de Tese e Dissertações
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
COBOL	<i>Common Business Oriented Language</i> OU Linguagem Comum Orientada aos Negócios em inglês
DDD	<i>Domain Driven Development</i> ou Desenvolvimento Orientado à Domínio em inglês
DevOps	<i>Development and Operations</i> ou Desenvolvimento e Operações em inglês, uma metodologia de desenvolvimento de sistemas
ECA	Ensaio Controlado Aleatório
FORTRAN	<i>Formula Translation</i> ou Tradução da Fórmula em inglês
GPS	<i>Global Positioning System</i> ou Sistema de Posicionamento Global em inglês
GSA	<i>Google Search Appliance</i> ou Aparelho de Busca Google em inglês
GUI	<i>Graphical User Interface</i> ou Interface Gráfica de Usuário em inglês
GUJ	Grupo de Usuários Java
HTML	<i>Hypertext Markup Language</i> ou Linguagem de Marcação de Hipertexto em inglês
IA	Inteligência Artificial
IBM	<i>International Business Machines</i> ou Máquinas de Negócios Internacionais em inglês
IDE	<i>Integrated Desktop Environment</i> ou Ambiente de trabalho integrado em inglês
IoT	<i>Internet of Things</i> ou Internet das Coisas em inglês

GRN	<i>Gamification Research Network</i> ou Rede de Pesquisa em Gamificação em inglês
HTTP	<i>Hypertext Transfer Protocol</i> ou Protocolo de Transferência de Hipertexto em inglês
JDK	<i>Java Development Kit</i> ou Kit de Desenvolvimento Java em inglês
JEP	<i>JDK Enhancement Proposal</i> ou Proposta de Melhoria do JDK em inglês.
JPMS	<i>Java Platform Module System</i> ou Sistema de módulos da plataforma Java em inglês
JSF	<i>JavaServer Faces</i> ou Faces de Servidor Java em inglês
JSP	<i>JavaServer Pages</i> ou Páginas de Servidor Java em inglês
JVM	<i>Java Virtual Machine</i> ou Máquina Virtual Java em inglês
LDBEN	Lei de Diretrizes e Bases da Educação Nacional
LTS	<i>Long Term Support</i> ou Suporte a Longo Prazo em inglês
MDA	<i>Mechanics, Dynamics, Aesthetics</i> ou Mecânica, Dinâmica, Estética em inglês
ML	<i>Machine Learning</i> ou Aprendizado de Máquina em inglês
MOODLE	<i>Modular Object-Oriented Dynamic Learning Environment</i> ou Ambiente de aprendizado dinâmico modular orientado a objetos
OOP	<i>Object Oriented Programming</i> ou Programação Orientada a Objetos em inglês
OSGi	<i>Open Services Gateway initiative</i> ou Iniciativa Portal de Serviços Abertos em inglês
PARC	<i>Palo Alto Research Center</i> ou Centro de Pesquisa Palo Alto em inglês
PBL	<i>Problem-Based Learning</i> ou Aprendizado baseado em problemas em inglês
PHP	<i>Hypertext Processor</i> ou Pré-processador de Hipertexto em inglês
POO	Programação Orientada a Objetos
PRNG	<i>Pseudo-Random Number Generators</i> ou Geradores de números pseudoaleatórios em inglês
RDBMS	<i>Relational Databases Management System</i> ou Sistema de Gerenciamento de Banco de Dados Relacional em inglês

RTR	Reflexivo Tempo Real
SQL	<i>Structured Query Language</i> ou Linguagem de Dados Estruturada em inglês
SUV	<i>Sport Utility Vehicle</i> ou Veículo Utilitário Esportivo em inglês
TDD	<i>Test Driven Development</i> ou Desenvolvimento orientado a testes em inglês
TIC	Tecnologia da Informação e Comunicação
UI	<i>User Interface</i> ou Interface de Usuário em inglês
UX	<i>User Experience</i> ou Experiencia de Usuário em inglês
VR	<i>Virtual Reality</i> ou Realidade Virtual em inglês
ZDP	Zonas de Desenvolvimento Proximal

SUMÁRIO

1 INTRODUÇÃO	17
2 ESTRATÉGIAS DE ENSINO SUSTENTADAS EM CONCEPÇÕES DE APRENDIZAGEM	26
2.2 AS RELAÇÕES DA APRENDIZAGEM NO ENSINO	38
3 FUNDAMENTOS DE LINGUAGENS DE PROGRAMAÇÃO, JAVA E ORIENTAÇÃO A OBJETOS	49
3.1 LINGUAGENS DE PROGRAMAÇÃO	49
3.2 LINGUAGEM JAVA	58
3.3 PROGRAMAÇÃO ORIENTADA A OBJETOS	66
4 GAMIFICAÇÃO.....	71
5 DESENVOLVENDO O APRENDIZADO DE PROGRAMAÇÃO ORIENTADA A OBJETOS COM GAMIFICAÇÃO – PRODUTO DA DISSERTAÇÃO	105
5.1 RODA DE CONVERSA	105
5.1.1 Dados da Roda de Conversa	109
5.2 AULA GAMIFICADA DE PROGRAMAÇÃO ORIENTADA A OBJETOS.....	112
5.2.1 Introdução	114
5.2.2 Configuração do Ambiente de Desenvolvimento Java	115
5.2.3 Classes e Objetos Java.....	119
5.2.4 Herança e Polimorfismo	123
5.2.5 Encapsulamento.....	127
5.2.6 Conclusão e Recompensas	130
6 CONSIDERAÇÕES FINAIS.....	133
7 REFERÊNCIAS.....	136
ANEXO I - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO.....	146

1 INTRODUÇÃO

Ao longo dos últimos anos, a sociedade tem vivenciado uma crescente “era da informação”, na qual o desenvolvimento tecnológico tem se tornado cada vez mais sofisticado e onipresente no cotidiano da população. Nesse contexto, as Tecnologias da Informação e Comunicação (TIC) desempenham um papel crucial na promoção do aprimoramento de habilidades essenciais aos indivíduos que buscam adaptar-se a essa nova realidade.

Uma habilidade particularmente relevante é a programação de computadores, que envolve o uso de linguagens de programação específicas para transmitir instruções às máquinas. As linguagens de programação funcionam como uma interface entre os seres humanos e os computadores, permitindo a criação de soluções tecnológicas inovadoras e a resolução de problemas complexos em diversas áreas do conhecimento.

Para se tornarem proficientes em programação, os indivíduos precisam adquirir domínio sobre as diversas dimensões das linguagens de programação, abrangendo aspectos básicos, teóricos e práticos. Embora muitos busquem cursos especializados para aprender tais linguagens, há dificuldade para assimilar as informações teóricas e aplicá-las em contextos práticos. Esta lacuna no aprendizado pode resultar em profissionais com conhecimentos fragmentados e habilidades limitadas no campo da programação.

Um dos principais desafios enfrentados pelos aprendizes de programação no Brasil é a escassez de recursos e materiais didáticos em língua portuguesa, que sejam capazes de proporcionar um aprendizado eficiente e a aplicação prática dessas linguagens de programação. A falta de materiais adequados em português pode criar barreiras para o acesso à informação, dificultando a aquisição de competências e conhecimentos necessários para o sucesso na área. Além disso, Richter (2018, p.5) afirma que “o ensino e a aprendizagem de programação são considerados por muitos professores e alunos um processo difícil”.

Desse modo, é importante promover a adoção de metodologias pedagógicas inovadoras e centradas no aluno, que estimulem a aprendizagem ativa, a criatividade e a resolução de problemas, e que possibilitem a aplicação dos conhecimentos teóricos em contextos práticos e reais. A integração de tecnologias emergentes, como a inteligência artificial e a realidade virtual, também pode contribuir para o enriquecimento do processo de ensino-

aprendizagem e facilitar a aquisição de habilidades em programação. A gamificação é uma das atividades ligadas à informática dentre as metodologias ativas, que podem ser “possibilidades dinâmicas e criativas de construir conhecimentos do aprendizado e pelas relações professor-alunos e alunos-alunos que assumem papéis de protagonistas e transformadores dos processos de aprendizagem” como argumentam Conceição et al. (2021, p. 568).

Ao abordar esses desafios e investir na formação de profissionais qualificados em programação, a sociedade brasileira estará mais bem preparada para enfrentar os desafios da "era da informação" e contribuir para o avanço tecnológico e o desenvolvimento sustentável do país.

De fato, nem todos os usuários ou estudantes possuem tempo suficiente ou recursos adequados para investigar e aprender de maneira abrangente, e assim aprofundada os conhecimentos à cerca das linguagens de programação. Bereiter e Scardamalia (2014) defendem que a aprendizagem deve ser vista como um processo contínuo e dinâmico, de construção do conhecimento, englobando três dimensões principais: a aquisição de informações e habilidades, a elaboração de conceitos e teorias, e a aplicação prática desses conhecimentos em contextos reais e complexos.

Cobo e Moravec (2011) enfatizam que o conhecimento é um elemento-chave para impulsionar a inovação e a evolução da sociedade. Eles argumentam que o conhecimento, incluindo novas teorias, ideias, técnicas e descobertas científicas, é fundamental para fomentar a mudança e a transformação da sociedade. Além disso, a capacidade inerente da mente humana de criar e expandir constantemente o conhecimento é um fator determinante para a adaptação e o progresso das sociedades.

Os benefícios de uma Ciência evoluída e construída se mostram nas atitudes cotidianas das sociedades. A assimilação das descobertas científicas, transformadas pela tecnologia, tem impacto direto sobre o modo de vida do ser humano (MEDEIROS, 2012).

Assim, a questão central desta pesquisa é: como a gamificação pode ser aplicada no ensino da programação orientada a objetos e quais são seus impactos na aprendizagem e na motivação dos estudantes ao aprender POO? Este estudo busca respostas às seguintes questões de pesquisa: (1) Como a gamificação pode ser aplicada no ensino da programação orientada a objetos? (2) Qual é o impacto da gamificação na aprendizagem e na motivação dos

estudantes ao aprender programação orientada a objetos? (3) Quais são os fatores que influenciam o sucesso da gamificação no ensino da programação orientada a objetos?

Ressalta-se que na atualidade a educação digital tem sido indicada como necessária para a formação dos sujeitos. Nesta perspectiva foi aprovado em 04 de agosto de 2022 a Política Nacional de Educação Digital. O texto aprovado é o substitutivo do deputado Professor Israel Batista (PSB-DF) ao Projeto de Lei 4513/20, transformado na Lei Ordinária 14533/2023, conforme a Agência Câmara de Notícias. No teor da lei, que altera a Lei 9394/96 estão destacados os eixos estruturais da Política Nacional de Educação Digital. O quadro a seguir, transcrito da Agência Câmara, apresenta esses eixos.

Quadro 1 - Eixos estruturais da Política Nacional de Educação Digital

Inclusão Digital da população brasileira	Educação Digital para alunos e professores	Capacitação e Especialização Digital dos trabalhadores	Pesquisa Digital para inovação e novos conhecimentos
Ampliação e facilidade de acesso da população	Formação de professores	Identificação das necessidades de mercado	Programa Nacional de Incentivo à Pesquisa e Desenvolvimento
Promoção de ferramentas online	Práticas de educação em ambientes digitais	Promoção de qualificação em tecnologias	Parcerias entre o Brasil e centros internacionais
Treinamento de competências	Promoção de tecnologias digitais	Rede nacional de Curso profissionais e superior	Qualificação avançada de pesquisadores
Universalização da conectividade das escolas	Incentivo à inovação pedagógica do ensino	Requalificação profissional de trabalhadores	Incentivo à ciência aberta

Fonte: Agência Câmara de Notícias

Desse modo, esta pesquisa se articula com essa recém aprovada lei na Câmara dos Deputados¹ que reconhece a necessidade da formação digital da população tanto quanto ao acesso, educação escolar, capacitação dos trabalhadores como para a promoção da pesquisa e inovação.

Do ponto de vista do pesquisador, e nesse breve parágrafo explico-me na primeira pessoa do singular, a justificativa se expressa que ao longo da minha carreira no campo da programação, deparei-me repetidamente com a dificuldade

¹ Após a aprovação na Câmara dos Deputados foi remetida para aprovação no Senado.

de encontrar materiais de estudo em língua portuguesa, assim como observei colegas de trabalho enfrentando desafios semelhantes. Esta situação é comum não apenas entre os profissionais, mas também nos cursos de formação em tecnologia da informação. Frequentemente, para expandir e aprimorar meus conhecimentos em programação, foi necessário recorrer a materiais em língua inglesa, como livros e artigos de revistas, devido à maior disponibilidade e qualidade dessas publicações no idioma. A escassez de recursos em português constitui um obstáculo significativo no processo de aprendizagem e desenvolvimento de habilidades em programação para falantes dessa língua, evidenciando a necessidade de investir em materiais e metodologias educacionais mais acessíveis e eficazes.

Assim, essa é uma das dificuldades para a população brasileira, em geral, pois segundo a pesquisa do *British Council* (2014, p. 6) apenas 5,1% da população brasileira de 16 anos ou mais afirma possuir algum conhecimento do idioma inglês, ainda que entre os mais jovens de 18 a 24 anos, o percentual dos que afirmam falar inglês dobra, chegando a 10,3% das pessoas nessa faixa etária, o que é ainda muito pouco. Essa questão é reafirmada como uma das dificuldades enfrentadas pela população brasileira, em geral, pois de acordo com o relatório *EF English Proficiency Index* (EF EPI) de 2021, o Brasil ocupa a 58ª posição entre 111 países no ranking de proficiência em inglês. Ainda que o conhecimento do idioma seja mais comum entre os jovens de 18 a 24 anos, a proficiência geral no país ainda é considerada insuficiente, o que representa um desafio significativo para a população.

Atualmente, existe uma vasta gama de linguagens de programação no mercado, cada uma com suas especificidades e aplicações, variando desde as mais recentes até aquelas com várias décadas de história. Entre as linguagens de programação mais conhecidas e utilizadas, podemos citar: C, C++, C#, PHP, Java, Go, Kotlin, Ruby, R, Scala, TypeScript, Swift, Python, Rust, Lua, Groovy, F# e Haskell. Essa diversidade de linguagens de programação atende a diferentes necessidades e contextos, desde aplicações web, sistemas embarcados e desenvolvimento de aplicativos móveis, até análise de dados e inteligência artificial. Ao escolher uma linguagem de programação para estudo ou aplicação em um projeto específico, é importante considerar fatores como facilidade de aprendizado, portabilidade, desempenho e integração com outras tecnologias.

Para Horstmann (2016) Java é uma das linguagens de programação mais populares e está presente em dispositivos como smartphones e microchips de cartões de crédito. Bosch (2019) também destaca que Java é uma das linguagens de programação mais relevantes no mundo atual. Um dos fatores que contribuem para sua importância é sua praticidade, versatilidade e poder em comparação com outras linguagens e alternativas disponíveis. Deitel e Deitel (2016) e Bloch e Gafter (2018) comentam que Java é particularmente atraente como primeira linguagem de programação e ressalta a importância que a linguagem pode ter na experiência do indivíduo.

Entretanto Hadjerrouit (2008) aponta que alguns estudantes sentiram que programar em Java era uma tarefa desafiadora. Luxton-Reilly et al. (2018) também observam que alguns estudantes consideram a programação em Java um desafio, o que justifica proporcionar maior oferta de atividades de aprendizagem.

A Programação Orientada a Objetos (POO) é um paradigma de programação que utiliza objetos e suas interações para modelar e solucionar problemas. Essa abordagem permite uma melhor compreensão e organização do código, proporcionando maior reutilização e manutenibilidade do software. Entretanto, muitos estudantes enfrentam dificuldades em aprender os conceitos e aplicá-los na prática, devido à sua complexidade e abstração.

Nesse contexto, a gamificação surge como uma estratégia promissora para melhorar o ensino e a aprendizagem da POO. Ela é uma técnica que utiliza elementos e mecânicas de jogos em contextos não lúdicos, com o objetivo de aumentar a motivação, o engajamento, a mudança de comportamento e produtividade conforme Marczweski (2013). A aplicação da lógica e da mecânica dos games em diferentes aspectos do cotidiano, caracterizando, portanto, ambientes que contenham elementos de jogos como gamificados.

Essa abordagem tem sido amplamente adotada em diversas áreas do conhecimento, incluindo a educação, com resultados positivos em termos de desempenho e satisfação dos estudantes conforme reforçam Deterding et al. (2011). Para Marczweski (2013) a gamificação tem sido aplicada em contextos como educação, em que estudantes recebem recompensas, como estrelas ou distintivos, por completar atividades propostas, ou em competições com pontuações entre diferentes grupos.

A gamificação do ensino da POO envolve a criação de um ambiente de aprendizagem interativo e lúdico, no qual os estudantes são desafiados a resolver problemas e aplicar os conceitos de orientação a objetos de maneira prática e significativa. Nesse ambiente, os estudantes são incentivados a assumir o papel de jogadores, enfrentando desafios, superando obstáculos e progredindo através de níveis de dificuldade crescente, à medida que desenvolvem habilidades e compreensão dos conceitos de POO.

Kapp (2012, p. 33) também oferece uma definição complementar, caracterizando gamificação como “o uso de mecânicas baseadas em jogos, estética e pensamento de jogos para encorajar pessoas, motivar ações, promover o aprendizado e resolver problemas”. Ele ressalta a importância do “pensamento de jogos” ao transformar atividades cotidianas em processos sociais, competitivos e cooperativos.

Diversos estudos como os de Bayliss (2013), Ibanez et al. (2014), Leutenegger e Edgington (2007) e Papastergiou (2009) têm demonstrado que a gamificação pode melhorar o ensino e a aprendizagem de programação e seus derivados como programação orientada a objetos, proporcionando uma experiência mais rica e envolvente para os estudantes. No entanto, ainda há lacunas na literatura sobre a efetividade dessa abordagem em diferentes contextos educacionais, bem como sobre os fatores que influenciam a sua implementação e sucesso conforme apontam os autores acima. Além disso, a criação de aulas gamificadas requer um planejamento cuidadoso e um entendimento profundo dos objetivos pedagógicos, dos conteúdos a serem ensinados e das dinâmicas de jogo que melhor se adaptam às características e necessidades dos estudantes.

Espera-se que esta pesquisa contribua para o avanço do conhecimento sobre a aplicação da gamificação no ensino da programação orientada a objetos, fornecendo ideias valiosas sobre a eficácia dessa abordagem, as boas práticas pedagógicas, os fatores que influenciam a implementação e o sucesso de aulas gamificadas. Além disso, espera-se que os resultados desta pesquisa possam ser utilizados por educadores e pesquisadores interessados em desenvolver e implementar estratégias de ensino inovadoras e eficazes para a aprendizagem de POO, promovendo uma maior compreensão e engajamento dos estudantes e, conseqüentemente, melhorando a qualidade da formação em programação e ciência da computação.

Adicionalmente, este estudo busca fomentar o debate e a reflexão sobre a importância da inovação e da adoção de novas abordagens pedagógicas na educação em programação e ciência da computação, bem como sobre o papel da gamificação como uma ferramenta poderosa para aumentar a motivação e a aprendizagem dos estudantes. Com a crescente demanda por profissionais qualificados na área de tecnologia da informação, é fundamental que os educadores estejam preparados para enfrentar os desafios e as oportunidades que surgem no processo de ensino-aprendizagem, utilizando estratégias e recursos didáticos que possam facilitar a aquisição de habilidades e competências necessárias para o sucesso no mercado de trabalho.

O Objetivo Geral é:

Desenvolver uma aula gamificada sobre programação orientada a objetos (POO), com base nos princípios e nas boas práticas identificadas na literatura e com os profissionais da área, adaptando os elementos e mecânicas de jogos às características e necessidades dos estudantes e aos objetivos pedagógicos e conteúdos a serem ensinados.

Os Objetivos Específicos são:

- Abordar as referências de boas práticas do ensino e aprendizagem de acordo com a literatura da área.
- Fundamentar as linguagens de programação, a linguagem Java e orientação a objetos.
- Fundamentar a gamificação aplicada ao ensino.
- Situar a composição da aula gamificada com as práticas realizadas por profissionais da área de engenharia de software via roda de conversa e com os elementos da programação orientada a objetos (POO).

A metodologia de investigação adota a abordagem qualitativa, tendo como procedimento revisão bibliográfica da literatura atual pertinente ao tema. Nessa revisão, foram consultados autores de referência, considerando a literatura composta por obras relevantes e, preferencialmente, atualizadas.

Quanto ao tipo de pesquisa, é exploratória quanto aos objetivos de investigação, considerando as proposições de Severino (2016, p. 129): “A pesquisa exploratória busca apenas levantar informações sobre um determinado objeto, delimitando assim um campo de trabalho.”

Incluiu um levantamento de dados por meio de uma roda de conversa envolvendo profissionais que atuam na área de programação com mais detalhes adiante.

A dissertação está organizada da seguinte maneira: o capítulo 2 aborda as referências de boas práticas do ensino e aprendizagem na educação formal e não-formal para sustentar o estudo do produto na área da educação. Em seguida contextualiza as atuais metodologias inovadoras de ensino. Depois, o capítulo aborda uma discussão fundamentada sobre aprendizagem.

Em seguida, o capítulo 3 trata de circunstanciar as linguagens de programação, contexto histórico e diferenças entre diferentes linguagens disponíveis no mercado. Trata da linguagem Java, tomando como ponto de partida o histórico de sua criação, passando por elucidação da sua maturidade, e como detalhes de como tem sido utilizada na atualidade em diferentes frentes ao redor do globo. Na sequência aborda a orientação à objetos, tema principal da aula.

Já o capítulo 4 aborda a metodologia do estudo desenvolvida que é a gamificação. A gamificação do ensino de programação orientada a objetos é apenas um exemplo de como a educação pode se beneficiar da interdisciplinaridade e da aplicação de conhecimentos de áreas como a psicologia, a pedagogia e o design de jogos para criar experiências de aprendizagem significativas e motivadoras. Neste sentido, este estudo também procura inspirar outros pesquisadores e educadores a explorar as possibilidades e potencialidades da gamificação e de outras abordagens inovadoras no ensino de diferentes áreas do conhecimento, contribuindo para a construção de uma educação mais dinâmica, flexível, adaptada às necessidades e características dos estudantes do século XXI.

Na continuidade, o capítulo 5 descreve e analisa os dados coletados que advém das falas dos participantes da pesquisa em uma roda de conversa com profissionais da área que já passaram (ou estão passando) por esta fase de aprendizado em uma linguagem de programação, reforçando a escolha da linguagem Java para este estudo e também é dedicado a expor a aula gamificada de programação orientada à objetos, produto desse estudo e sendo uma contribuição do estudo realizado ao ensino deste conteúdo

No capítulo 6, apresentaremos as conclusões desta dissertação, destacando as principais contribuições e avanços no campo do ensino de

programação orientada a objetos com a aplicação da gamificação. Além disso, a presente pesquisa pretende chamar a atenção para a importância da avaliação e da investigação empírica das práticas pedagógicas baseadas na gamificação, a fim de proporcionar uma base sólida para a tomada de decisões e implementação de melhorias no processo educacional. Com a aula gamificada sobre POO, espera-se fornecer evidências concretas dos benefícios e desafios dessa abordagem, bem como identificar os elementos e fatores que podem potencializar o ensino nessa abordagem.

2 ESTRATÉGIAS DE ENSINO SUSTENTADAS EM CONCEPÇÕES DE APRENDIZAGEM

Este capítulo tem por finalidade proporcionar uma discussão sobre boas práticas de ensino identificadas na literatura, considerando um dos objetivos desse estudo: desenvolver uma aula gamificada sobre POO, com base nos princípios e nas boas práticas identificadas na literatura, adaptando os elementos e mecânicas de jogos às características e necessidades dos estudantes, e aos objetivos pedagógicos e conteúdos a serem ensinados.

Diferentes modalidades de educação possibilitam o desenvolvimento de boas práticas de ensino, contribuindo na formação profissional dos estudantes e da população em geral. Este capítulo toma, como ponto de partida, o entendimento do que é educação, contextualizando a educação formal e não-formal, modalidades nas quais efetiva-se o ensino, incluindo as teorias de aprendizagem. A compreensão desses componentes se apresenta como suporte ao desenvolvimento de boas práticas de ensino nas quais se insere a aula gamificada, objeto de investigação desse estudo.

2.1 RELAÇÃO DA EDUCAÇÃO FORMAL E NÃO-FORMAL COM O ENSINO

O ensino compreendido na atualidade se confunde com a origem da escola pública, universal, laica, integrado à educação e socialmente referendado. Assim, ao iniciar a argumentação, retoma-se Brandão (2003, p. 7) que é enfático ao afirmar “Ninguém escapa da educação”, pois está inserida no processo de humanização e socialização da vida humana. Ela se insere no cotidiano das relações sociais entre as pessoas, desde o ambiente familiar, o ambiente do trabalho e das organizações sociais, ou seja, ela “existe misturada com a vida em momentos de trabalho, de lazer, de camaradagem ou de amor” (Brandão, 2003, p. 19). Assim, esse autor entende a educação como uma prática social presente na vida dos grupos sociais que a criam e recriam, fomentando as formas de ser e de produzir a cultura e os meios de subsistência da vida. São modos de criação e preservação de saberes presentes nos diversos grupos sociais, desde as tribos, os códigos sociais de conduta, as regras do trabalho, as formas de inventar, as artes, as religiões, as tecnologias que os grupos humanos obtêm para se reinventarem, todos os dias, salienta Brandão (2003).

Desse modo, a educação se insere na história, como ressalta Severino (2001). Destaca o autor que a educação se torna um “processo pedagógico sistematizado de intervenção na dinâmica da vida social” podendo ser analisada como objeto “de estudos científicos com vistas à definição de políticas estratégicas para o desenvolvimento integral das sociedades”. (SEVERINO, 2001, p. 65). A educação proposta por Freire (1996) é o processo constante de criação do conhecimento e de busca da transformação-reinvenção da realidade, envolvendo a ação-reflexão humana de um processo intencional.

A espécie e a extensão da educação distribuídas por uma sociedade a seus membros são reflexos de seu estado de desenvolvimento material e cultural. Estes são as que determinam as possibilidades da educação tanto em qualidade (conteúdo e métodos) como em quantidade (a quem e quantos será distribuída) (VIEIRA PINTO, 1993).

O ensino, como processo intencional institucionalizado e desenvolvido, integrado à educação como política de estado para possibilitar a todos a inserção na vida social, é um processo formal e reconhecido socialmente. Nesse aspecto, na Constituição de 1988, a educação é definida no Art. 205: “A educação, direito de todos e dever do Estado e da família, será promovida e incentivada com a colaboração da sociedade, visando ao pleno desenvolvimento da pessoa, seu preparo para o exercício da cidadania e sua qualificação para o trabalho”.

Esse modo de entendimento da educação se manifesta de modo expressivo na Lei de Diretrizes e Bases da Educação Nacional – LDBEN, Lei 9394/96, que em seu Artigo 1º define “A educação abrange os processos formativos que se desenvolvem na vida familiar, na convivência humana, no trabalho, nas instituições de ensino e pesquisa, nos movimentos sociais e organizações da sociedade civil e nas manifestações culturais” reafirmando o princípio constitucional. Entende a Lei 9394/96 que a composição da educação nacional compreende: I - educação básica, formada pela educação infantil, ensino fundamental e ensino médio; II - educação superior. Com efeito, a educação formal pode ser compreendida como escolarizada e institucionalizada.

Retomando Brandão (1981, p. 26):

O ensino formal é o momento em que a educação se sujeita à Pedagogia (a teoria da educação), cria situações próprias para o seu exercício, produz os seus métodos, estabelece suas regras e tempos e constitui executores especializados. É quando aparecem a escola, o aluno e o professor.

A seguir são apresentadas as definições da UNESCO (2012) condensadas por Bendrath (2014, p. 57 e 58) sobre educação formal, não-formal e informal:

- Educação formal: ocorre em estabelecimentos de ensino e formação, é reconhecida pelas autoridades nacionais competentes e confere diplomas e qualificações. A aprendizagem formal é estruturada de acordo com dispositivos como currículos, qualificações e requisitos de ensino-aprendizagem.
- Educação não-formal: se refere a atividade educativa complementar que não corresponda exatamente à definição de educação formal. Geralmente, ocorre em contextos comunitários locais, no local de trabalho e por meio de atividades de organizações da sociedade civil e pode atender pessoas de qualquer idade. Em alguns casos, também é estruturada de acordo com dispositivos de educação e treinamento, mas de modo flexível. De acordo com o contexto de cada país, poderá abarcar programas educacionais para alfabetização de adultos, educação básica de crianças para escolas e habilidades para a vida ativa, para o trabalho e cultura geral. Os programas de educação não-formal não são necessariamente regidos por um sistema de classificação e podem ter durações diferentes e não conferem certificações de aprendizagem.
- Educação informal: é a aprendizagem que ocorre na vida cotidiana, na família, no local de trabalho, nas comunidades e é mediada pelos interesses ou atividades das pessoas. Pode ser reconhecida, validada sofrer acreditação por meio de exames, provas em que se expressam as competências obtidas semelhantes às adquiridas na educação formal.

A educação formal, institucionalizada, de caráter público e alvo das políticas educacionais, no caso específico da formação na área de ciências da computação, define o seu ensino na Base Comum Nacional Curricular – BNCC, como competências desde a educação infantil ao final do ensino médio, na educação básica. Na educação superior se constituem cursos de bacharelado, licenciaturas e tecnólogos.

De acordo com os dados do Instituto Nacional de Estudos e Pesquisas Pedagógicas Anísio Teixeira – INEP, divulgados em novembro de 2022, apontam para oferta de muitos cursos de graduação no nível superior. No que tange à grande área da tecnologia da informação, temos um panorama na tabela abaixo:

Cursos de graduação	Número
Computação e Tecnologias da Informação e Comunicação (TIC) em biociências e saúde	7
Computação formação de professor	90
Engenharia de informação	1
Inteligência artificial	8

Internet das coisas	18
Programas interdisciplinares abrangendo computação e Tecnologias da Informação e Comunicação (TIC)	25
Redes de computadores	240
Segurança da informação	49
Sistemas de informação	1.316
Sistemas para internet	138

Fonte: INEP, Sinopse Estatística, 2022.

Os dados apontam que a maior oferta de curso é de Sistemas de Informação, correspondendo ao número de 1.316, em seu total. Os demais cursos são em menor número, com Rede de Computadores com 240 cursos, Sistemas para internet com 138. Em seguida o número de cursos ofertados é reduzido para menos de uma centena com 90 cursos de Computação e formação de professor (foco na licenciatura), 49 em Segurança da informação. Para Cunha et al. (2013) a formação realizada na educação superior envolve examinar as estruturas, os processos e os resultados, em que os conhecimentos das diversas disciplinas, objeto do ensino, ao serem dinamizados, possibilitem a formação técnica e profissional, formação cultural e crítica para fortalecer o compromisso de transformação da realidade social.

Ao considerar a tabela do Conselho Nacional de Desenvolvimento Científico e Tecnológico verifica-se que a Ciência da Computação é composta por conjunto de disciplinas que envolvem Teoria da Computação, Computabilidade e Modelos de Computação, Linguagem Formais e Automatos e de Programação, Análise de Algoritmos e Complexidade de Computação, Lógicas e Semântica de Programas, Matemática da Computação, Matemática Simbólica, Modelos Analíticos e de Simulação, Metodologia e Técnicas da Computação, Engenharia de Software, Banco de Dados, Sistemas de Informação, Processamento Gráfico, Hardware, Arquitetura de Sistemas de Computação, Software Básico Teleinformática que compõe, na maioria as disciplinas dos cursos dessa área. Assim, o objeto dessa dissertação se situa nesse campo, e a aula de gamificação se insere como ensino em uma abordagem metodológica inovadora, também nominada como metodologia ativa.

Durante o desenvolvimento desses cursos, torna-se evidente que a aprendizagem da programação envolve inúmeras situações que requerem um ensino engajador, como enfatizado por Souza, Batista e Barbosa (2016). Entre as situações desafiadoras constam: a) a compreensão dos conceitos de programação; b) a transferência desses conceitos para a aplicação elaboração

de programas; c) o pouco envolvimento dos estudantes na realização de atividades de programação.

Para melhorar o ensino, no enfrentamento dessas questões, a literatura da área da educação tem indicado, de modo geral, como alternativas: a) a existência de bons laboratórios de informática nas instituições de ensino; b) utilização de visualização de programas e algoritmos; b) utilização da gamificação; c) desenvolvimento de ambientes pedagógicos para o ensino e aprendizagem de programação, entre elas práticas que envolvem metodologias inovadoras de ensino.

Em relação às metodologias inovadoras a inserção de tecnologias digitais se expressa com intensidade. Richter (2018) ao desenvolver estudos sobre ensino de programação orientada a objetos realizou um levantamento em pesquisas sobre a inserção de ferramentas de software efetivamente exploradas entre elas: Scratch (25,8%); Alice (24,2%); LEGO® Mindstorms (9,1%); MIT AppInventor (3,0%); Arduino IDE (3,0%); Blockly (3,0%); Scratch for Arduino (3,0%); CFácil (3,0%); Moodle (3,0%); Pygame (1,5%) GameMaker Studio (1,5%) Vprog (1,5%) FARMA-ALG (1,5%); E62 RoboEduc (1,5%) RaspiBlocos (1,5%); Lord of Cod (1,5%); CodeBench (Juiz online) (1,5%); RHODES 2.0(1,5%) Robocode (1,5%) Mblock (1,5%) e Hour of Code (1,5%). Esses dados expressam uma diversidade de possibilidades de desenvolvimento do ensino direcionado a objetos.

Os mais utilizados no ensino são apresentados a seguir:

- O Scratch apresenta uma interface gráfica visual e de mídias destinadas à programação, permitindo a construção das sintaxes algorítmicas no conceito de orientação a objeto. A versão 3.0, encontra-se disponível para smartphones, tablets e computadores e usa a base da tecnologia voltada para web com o HTML5 (sigla de *Hypertext Markup Language*, expressão inglesa que significa “Linguagem de Marcação de Hipertexto”).
- Alice é uma plataforma para a iniciação dos estudantes com o conceito da lógica de programação: utiliza linguagem e recursos simples que ajudam os estudantes na aprendizagem; é empregada na criação de desenhos gráficos e em jogos que utilizam de recursos de 2D e 3D (CATUNDA e LOPES, 2020).

- O LEGO Mindstorms é utilizado pelos estudantes para realizar projetos de robótica.
- O App Inventor é um software web criado pela universidade americana *Massachusetts Institute of Technology* (MIT) que permite desenvolver aplicativos Android usando um navegador da Web e um telefone ou emulador conectados.
- Arduino é uma plataforma de prototipagem eletrônica open-source que se baseia em hardware e software flexíveis e fáceis de usar. É destinado a criar objetos ou ambientes interativos.
- Blockly é uma biblioteca que adiciona um editor visual para a criação de aplicações para a web e para o Android.
- MOODLE é o acrônimo de “*Modular Object-Oriented Dynamic Learning Environment*”, um software livre, de apoio à aprendizagem, executado num ambiente virtual.

Em relação às metodologias inovadoras Richter (2018) identificou, nos estudos examinados, uma forte tendência de introdução dessas metodologias no ensino direcionado a objetos, tais como: Aprendizagem baseada em problemas (PBL ou *Problem-Based Learning* em inglês); Aprendizagem por projetos; Competição/Olimpíada; Ensino Híbrido; Mapas conceituais; Método de composição de resultado; Metodologia baseada em semiótica para concepção, elaboração e análise de práticas. Seguidas de emprego de Objetos de aprendizagem; Oficinas; Práticas colaborativas; Sala de aula invertida e Sistema Personalizado de Ensino. Nessa perspectiva Veiga e Fernandes (2020) e seu grupo de pesquisa em didática têm realizado estudos intensos na última década² desde aprendizagem baseada em problemas, mapas conceituais, sala de aula invertida, realidade aumentada entre outros.

Com efeito, a aprendizagem por problemas (PBL) segundo Leitinho e Carneiro (2013) é forma de organização do ensino implementada inicialmente nos cursos de Medicina, nas Universidades de MacMaster (Canadá) e Maastricht (Holanda). Disseminou-se pelo mundo ampliando seu campo de atuação na área da saúde, mas que vem sendo direcionada também a outras áreas. Essa metodologia toma como ponto de partida problemas e ideias geradoras, sobre as quais são processadas as investigações seguidas de reflexão, crítica e

² Consultar D'Ávila e Veiga (2012); Veiga (2013); Veiga (2017); Veiga (2020).

transformação. O currículo sistematizado no projeto pedagógico de um curso PBL torna-se um elemento estruturante a partir de temáticas centrais abordadas de modo interdisciplinar e transdisciplinar. O propósito, tanto na organização do currículo como nas aulas, visa promover aprendizagens significativas, organizadas a partir de situações-problema. Para o ensino de programação vários cursos têm optado por utilizar a metodologia de PBL associada a ferramentas visuais: Alice/NetBeans (Java) e Scratch/DevC++(C) (AMBRÓSIO e COSTA, 2010).

Os mapas conceituais, também conhecidos como mapas de conceitos, são uma forma organizada de facilitar a aprendizagem por meio de diagramas que indicam as relações entre conceitos ou palavras usadas para representá-los. Segundo Moreira (2012), esses diagramas estabelecem relações significativas e estabelecem hierarquias conceituais. Diferentemente dos mapas mentais, que são livres e associativos, os mapas conceituais não se ocupam apenas de relacionar conceitos, mas também de hierarquizá-los. Essa abordagem se integra ao processo de aprendizagem baseado na teoria proposta por Ausubel, em que os novos conceitos e ideias não são aprendidos superficialmente, mas sim integrados e relacionados com as ideias prévias dos estudantes. Essas associações entre conceitos favorecem a aprendizagem, o que é especialmente útil no contexto da programação, conforme ressaltado por Richter (2018).

A Composição de Resultado foi concebida a partir de ideias como o *Test Driven Development* (TDD ou Ambiente Orientado à Testes em inglês), e seu objetivo é oferecer um método de resolução de problemas de programação. O TDD é uma técnica de desenvolvimento de software baseada em um ciclo curto de repetições, em que são estabelecidas teses permitindo que funcionalidades possam ser criadas, assim como a realização de mudanças para aperfeiçoar o código, ambas avaliadas pelos testes, conforme argumentam Beck et al. (2015).

Já o ensino híbrido provém do termo *blended learning*, ou seja, “aprendizagem misturada” que segundo Moran (2015, p. 27) “significa misturado, mesclado”. [...] “é a articulação de processos de ensino e aprendizagem mais formais com aqueles informais, de educação aberta e em rede”. Nascimento et al. (2020, p. 953)³ ao revisarem o conceito destacam que

³ No artigo Engajamento de docentes na educação superior: implementando ensino híbrido, Nascimento et al. (2020) relacionam vários estudos sobre o ensino híbrido.

em 2013 Christensen, Horn e Staker, associam essa metodologia a uma perspectiva de “inovação sustentada e disruptiva, tendo como base o uso das tecnologias digitais de informação e comunicação para integrar momentos presenciais e virtuais em prol da aprendizagem do sujeito”. Com efeito, a proposição de um ensino disruptivo implica em romper, interromper, desfazer, rescindir, ou seja, buscar novas direções, novos modos de realizar o ensino. A origem desse ensino é marcada pela inserção das TICS como possíveis geradores de aprendizagem por disponibilizar, no ciberespaço, o acesso ao conhecimento. Para Horn e Staker (2015) o ensino online foi empregado para os estudantes que buscavam recuperar suas notas, seu desempenho nas avaliações. Em várias escolas a aprendizagem de línguas passou a empregar cursos online.

Hodges et al. (2020) desenvolveram uma classificação para os métodos de organização do ensino, dividindo-os em: totalmente online; híbrido (com aproximadamente 50% ou entre 25% e 50% das atividades online); e presencial com suporte online. Essas categorias consideram diferentes aspectos, como o ritmo de aprendizagem, que podem ser individualizados ou sincronizados com a participação do grupo. Além disso, a classificação leva em conta o número proporcional de estudantes em cada grupo, o processo de avaliação, os meios de comunicação entre professores e alunos, o papel dos estudantes na experiência educacional e as oportunidades para receber feedback.

Não basta incluir um laboratório de informática para sustentar o ensino híbrido, as atividades espontâneas de busca de informação não constituem um ensino híbrido, pois pressupõe uma programação com acompanhamento dos estudantes, ou seja, integra a educação formal segundo Horn e Staker (2015). Esse ensino envolve parte online e parte presencial realizado em uma instituição de ensino.

Essas contribuições das metodologias inovadoras valorizam a participação ativa, colaborativa, coletiva na busca da melhoria do ensino que resulte em aprendizagens para edificar uma formação científica, acadêmica e profissional da educação formal. Como enfatiza Araújo (2017) as metodologias ativas envolvem a participação, a colaboração, a cooperação, o compartilhamento. O propósito dessas abordagens vai além da aprendizagem, pois pressupõem a compreensão do contexto em que realiza o ensino e a problematização do conhecimento e não apenas sua reprodução. Destarte, nos

cursos da área de ciências da computação, devido a complexidade dos conhecimentos abordados que exigem a compreensão, aplicação de conhecimentos interdisciplinares, há um desafio permanente para a realização de um ensino que desenvolva aprendizagens significativas. Além disso, a educação formal pode ser complementada pela educação não-formal.

Assim, na focalização desse estudo que trata de desenvolver uma aula gamificada para o aprendizado de programação na linguagem Java, essa aula estará disponível também como ferramenta na educação não-formal. É importante destacar que Bendrath (2014) salienta o aprendizado na educação não-formal para qualificar a mão de obra para determinadas funções sistemáticas (destreza, conhecimento, habilidades) associada ao mundo do trabalho, mas ao mesmo tempo organizada e proposta por valores definidos pelo Estado. O autor destaca que na formação dos trabalhadores a escola pode se constituir como um processo de desajustes com o setor produtivo se a estrutura do currículo mantém e persiste em realizar a educação por um currículo homogêneo, que não considera as demandas do sistema produtivo. O sistema produtivo se apresenta cada vez mais heterogêneo o que exige da escola e dos currículos uma atualização constante. O desajuste pode ocorrer quanto não há sincronia entre o número de graduados e a demanda de cargos de trabalho, ou seja, os sistemas de ensino precisam observar o número de profissionais necessários em cada área quando estabelecem novos cursos. Ainda, o desajuste pode ser percebido no problema de superqualificação da mão-de-obra empregada em algumas áreas da indústria e setor de serviços, que nem sempre trazem satisfação do trabalho realizado devido a produção inadequada de conhecimentos científicos e técnicos. (BENDRATH, 2014, p. 52).

Desde os anos de 1960, a educação formal desenvolvida pelas instituições escolares, tem sido alvo de críticas, entre as mais relevantes a realizada por Coombs (1968) como ressaltam Guimarães e Faria-Fortecoëf (2021). Coombs ao diagnosticar o sistema educacional, observa que ele abrange os elementos formais (cursos oficiais), não-formais e informais (atualização profissional, técnica e rural, alfabetização de adultos etc.) e diante dos resultados insuficientes da educação formal propões a sua complementariedade pelas diferentes modalidades de educação.

A UNESCO (1977) incorpora a importância atribuída às modalidades educativas que se prolongam pela vida e ocorrem em contextos e momentos

muito diversos, para além da escola. (GUIMARÃES E FARIA-FORTECOËF, 2021). Palhares (2009) reforça que, nas sociedades ocidentais, as aprendizagens não-escolares são consideradas complementares e suplementares à educação escolar, mesmo quando a escola enfrenta crises. Isso significa que a educação não-formal é insuficiente para a formação dos indivíduos, mesmo que a escola esteja passando por dificuldades. Para o autor, essa perspectiva decorre devido a escola estar na pauta no processo de formação para a cidadania democrática. Com efeito, a origem da escola moderna é marcada com essa possibilidade de educação e assim a busca da excelência escolar é examinada com interesse social.

Aliás, como seguidamente veremos, o não-formal se por um lado emerge associado à incapacidade da escola em cumprir as promessas da modernidade, tendo sido ensaiado como alternativa preferencial em países e contextos desfavorecidos, por outro lado, hoje ressurgue com uma aura renovada capaz de ajudar a resolver os problemas de educação e formação nos países ocidentais. (PALHARES, 2009, p. 55)

Nessa direção, os diversos modos da educação permanente e global se efetivam na educação formal, não-formal e informal. Os documentos publicados pela UNESCO assumem a perspectiva da educação permanente, nominada como educação ao longo da vida, por Delors como expõe Bendrath (2014).

Quanto aos programas de ações na educação não-formal Bendrath (2014) registra que no Brasil, a alfabetização de adultos se constitui na centralidade dessas ações, reconhecendo a necessidade de tornar a população linguística e matematicamente apta, com possibilidades de se contribuir na promoção do desenvolvimento humano. Segue ações para a “vida em comunidade” e “desenvolvimento comunitário” para a melhoria das condições de vida. Essas ações estão articuladas às iniciativas de associações civis e organizações não governamentais, configurando a opção pela terceira via na perspectiva macroeconômica, e não como ações direcionadas pelos governos.

A educação não-formal se direciona para programas de complementação da educação formal na educação de trabalhadores em programas de treinamento com aprimoramento para o mercado de trabalho; programas para desenvolvimento de habilidades pessoais que não necessitam de diplomas e certificados. Esses programas contribuem em formação direcionadas para a flexibilização da força de trabalho e recolocação profissional. Ainda há programas dirigidos para a formação em cultura e lazer dos trabalhadores, mais presentes no hemisfério norte. (BENDRATH, 2014, p. 62).

Ressalta Bendrath (2014, p. 63) a educação não-formal, muitas vezes contribui no atendimento de “necessidades emergentes de reestruturação social em países afetados por crises naturais, armadas ou econômicas”. A educação não-formal se insere nessas ações devido a sua não estruturação rígida, que possibilita de modo efetivo e prático, em curto espaço de tempo, organizar programas emergentes que atendem problemas urgentes como desemprego e falta de qualificação para postos de trabalho. A educação não-formal ao ser desenvolvida por organizações não governamentais favorece atuar em ambientes hostis e promoção de uma cultura de paz, ajudando no desenvolvimento humano.

As categorias de ações envolvem, portanto, “Aprendizagem Habilidades para a vida”; “Vida em Comunidade”; “Desenvolvimento Comunitário” e “Treinamento Vocacional”, que contribuem para o desenvolvimento do capital humano e capital social em ações realizadas pela terceira via⁴. (BENDRATH, 2014, p. 148). Esse direcionamento de ações se relaciona com as indicações de Gohn (2006) quando aponta a educação não-formal para a educação para cidadania, justiça social, para direitos humanos, sociais, políticos, culturais etc. Os programas podem abranger a formação em Educação para a liberdade e democracia, assim como proporcionar oportunidades para a promoção da diversidade, como a Educação contra a discriminação e a Educação para a valorização das diferenças culturais.

Na área da computação é usual a realização de atividades de ensino para a compreensão do raciocínio lógico e, conhecimentos computacionais na educação não-formal em ambientes corporativos e para habilitar o uso de computadores às pessoas comuns, em diferentes ambientes como plataformas, Cursos Online Abertos e Massivos (MOOC), museus, espaços de educação não-formal, como destacam os artigos publicados em Anais do Simpósio Brasileiro de Informática na Educação⁵, em periódicos da área e conforme destaca Mattar (2013).

Os estudos de Gohn (2006) levantam uma série de pontos que a educação não-formal pode ajudar a desenvolver, dentre elas cita-se: quando

⁴ “a tese do Estado mínimo está estreitamente ligada a uma visão peculiar da sociedade civil como um mecanismo autogerador de solidariedade social. Nesse sentido, estimula-se a criação de associações civis sem a intervenção do Estado, como mecanismo de autorregulação e promoção do bem-estar social.” BENDRATH, 2014, p. 100.

⁵ <https://sol.sbc.org.br/index.php/sbic/index>

presente em programas com crianças ou jovens adolescentes, a educação não-formal resgata o sentimento de valorização de si próprio (o que a mídia e os manuais de autoajuda denominam, simplificada e como a autoestima), ou seja, dá condições aos indivíduos para desenvolverem sentimentos de autovalorização, de rejeição dos preconceitos que lhes são dirigidos, o desejo de lutarem para ser reconhecidos como iguais – enquanto seres humanos, dentro de suas diferenças raciais, étnicas, religiosas, culturais, etc. De tal forma que os indivíduos adquirem conhecimento de sua própria prática, aprender a ler e interpretar a mundo que os cerca.

Já a educação informal se direciona para a socialização dos indivíduos: “[...] desenvolve hábitos, atitudes, comportamentos, modos de pensar e de se expressar no uso da linguagem, segundo valores e crenças de grupos que se frequenta ou que pertence por herança, desde o nascimento” reforça Gohn (2006, p. 29). A educação informal não é programada, ela é circunstancial, espontânea.

A educação não-formal, ao ser sistematizada, amplia essa inserção no mundo, pois capacita os indivíduos para a cidadania por meio de ações direcionadas ao acesso ao conhecimento, mas suas ações não são programáticas e decorrem do processo de desenvolvimento, participação e interação dos sujeitos, portanto complementa a educação formal. (GOHN, 2006, p. 29). Com efeito, a educação não-formal tem como resultados, segundo Gohn (2006, p. 30) a possibilidade de ampliar: a consciência e organização de como agir em grupos coletivos e a construção e reconstrução de concepção de mundo, sobre o mundo formando o sujeito para a vida. Além disso, os sujeitos adquirem conhecimento de sua própria prática, o que contribui para a formação da identidade dos indivíduos, bem como a sua própria valorização.

Assim, verifica-se que o caráter de complementariedade da educação não-formal em suas ações está agregado a situações em que a escolarização não foi efetivada, como no caso da alfabetização de adultos. Enquanto metodologia, os indicativos se direcionam mais para uma perspectiva dialógica com incorporação dos saberes, da experiência e da cultural. Contudo, não se trata da permanência nesses saberes, e sim na sua ampliação e articulação com saberes técnicos e científicos, mas mantendo sua interação com a prática, ainda que se constituam muito próximos de saberes da escolarização formal.

Essas indicações, sobre o caráter de complementariedade da educação não-formal, parecem estar alinhadas com o direcionamento desta pesquisa, que busca desenvolver uma aula sobre a orientação a objetos. A proposta se assenta em conhecimentos da área de tecnologia da informação, mas não se guia como conhecimentos disciplinares, não se pretende desenvolver uma disciplina acadêmica. Assim, é fundamental que a proposta tome como direcionamento uma perspectiva assentada na experiência de profissionais da área de engenharia desoftware. Esses saberes advindos da prática são o centro da proposta, o que a impulsiona para essa modalidade de educação não-formal.

Na compreensão das dimensões da aula como afirma Araújo (2011) são presentes os sujeitos que mantém uma interação intencional, organizada e sistemática, a técnica que abrange os objetivos, os conhecimentos, métodos e técnicas, tecnologias, dispositivos e avaliação, o ensinar e o aprender para responder a finalidade da sua institucionalização na sociedade, desse modo a seguir se aborda a aprendizagem completando a abordagem desse capítulo.

2.2 AS RELAÇÕES DA APRENDIZAGEM NO ENSINO

A aprendizagem é uma questão muito complexa, não existe uma definição única geralmente aceita para o conceito. Pelo contrário, observa-se o desenvolvimento constante de muitas explicações por meio de teorias mais ou menos singulares ou sobrepostas, algumas delas referindo-se a visões mais tradicionais, outras tentando explorar novas possibilidades e modos de pensar (ILLERIS, 2013).

Santos et al. (2019) comenta que a aprendizagem é uma atividade que se processa em primeira pessoa, pois exige a apropriação consciente das informações. Salaria também que a ela não é uma atividade totalmente imanente, isto é, fruto apenas das atividades do sujeito aprendiz. Completando a possibilidade de aprendizado por reflexão, de maneira subjetiva e imanente, pelas informações advindas dos sentidos (*sense data*), pela conversação e diálogo e, até mesmo, pela leitura.

Conforme Marcial (2015), as transformações desse século como a globalização da economia, a intensificação da internet que mudaram as relações de tempo e espaço, as comunicações em tempo real, a interciência em diversidade de saberes, a geopolítica com suas diferentes demarcações culturais e a transformação do trabalho tem produzido modificações intensas nas

exigências de aprendizagens, como também mudam as formas e modos de aprender. A autora aponta no texto que o crescimento da população ocorre em taxas marginais decrescentes, com envelhecimento populacional; a intensificação de movimentos migratórios; adaptação do papel do Estado frente aos novos desafios sociais e populacionais; a intensificação da urbanização; o empoderamento dos indivíduos e da sociedade civil organizada, com aumento da classe média em escala global, tudo isso têm provocado alterações significativas nos modos de entendimento da aprendizagem. Além disso, Marcial (2015) destaca os impactos das novas tecnologias quanto à aceleração do desenvolvimento tecnológico, multidisciplinar, com aplicações tecnológicas cada vez mais integradas. E mais, as convergências tecnológicas associadas à conectividade e à interatividade são potenciais que geram impactos nos ambientes possibilitando o acesso à informação interagindo no modo de aprender.

Para Illeris (2013) na composição da aprendizagem ocorre a interação e integração de dois processos divergentes: um processo externo e um processo interno. No processo externo se integram o ambiente social, cultural e de condições materiais e no processo interno a elaboração e aquisição de aprendizagem pelo indivíduo. Embora esses processos sejam distintos entre si, estão permanentemente conectados, e não podem ser examinados dissociados. Portanto, para aprender, o sujeito necessita de uma condição biológica, das condições internas e externas, e de que o conhecimento aprendido possa ser aplicado. Na aprendizagem estão envolvidos um conteúdo a ser aprendido tais como conhecimentos, habilidades, atitudes; a sensibilidade que envolve emoções, motivações e vontade e interação abrangendo ação, comunicação e cooperação (ILLERIS, 2013, p. 19).

Sobre os tipos de aprendizagem Illeris (2013) destaca a aprendizagem cumulativa ou mecânica. Essa aprendizagem é mais recorrente nos primeiros anos de vida, quando a criança é condicionada a aprender com situações de repetição que são memorizadas, mas não explicadas. A criança faz, mas não se explica a ela como e porque faz.

Sobre as abordagens das teorias da aprendizagem Giusta (2013)⁶, considera como primeira corrente a denominada de ambientalismo ou empirismo:

O conceito de aprendizagem emergiu das investigações empiristas em Psicologia, ou seja, de investigações levadas a termo com base no pressuposto de que todo conhecimento provém da experiência. Isso significa afirmar o primado absoluto do objeto e considerar o sujeito como uma tábula rasa, uma cera mole, cujas impressões do mundo, formadas pelos órgãos dos sentidos, são associadas umas às outras, dando lugar ao conhecimento. (GIUSTA, 2013)

Nessa aprendizagem, o conhecimento é constituído como uma cadeia de ideias adquiridas a partir do registro dos fatos e reduzida a uma simples reprodução. Trata-se de uma aprendizagem sem elaboração pelo sujeito, portanto determinada por fatores externos ao indivíduo. Essa aprendizagem também é identificada como associacionismo, tendo como expressão mais importante o behaviorismo, que aborda a teoria psicológica como derivada da concepção empirista. A definição de aprendizagem, associada a esse modelo, é a mudança de comportamento resultante do treino ou da experiência. O sujeito é visto como tábula rasa e, por isso, a aprendizagem é identificada com condicionamento (GIUSTA, 2013). Na corrente empirista, o processo ensino-aprendizagem é centrado no professor, que planeja e organiza os conteúdos para serem internalizados pelos alunos receptores de informações que serão armazenadas em suas memórias. O modelo de ensino é fechado, acabado, formalizado e não admite a aprendizagem relacionada ao ambiente, e menos ainda uma aprendizagem não-formal ou informal. Há uma preocupação centrada no planejamento do ensino baseando-se em que conteúdos bem transmitidos produzem aprendizagem. O conhecimento não está em interação com o sujeito, pois provém de livros e do professor e não da elaboração e da reflexão, pois a autoaprendizagem não é reconhecida.

O impacto da abordagem empirista pode ser constatado nos programas educacionais, elaborados com o objetivo de estimular e intervir no desenvolvimento das aprendizagens em que a educação formal é suficiente para transformar o indivíduo e mudar a sociedade, pois a função da escola é supervalorizada, e o aluno permanece no papel de reproduzidor do conhecimento. Para Giusta (2013), o behaviorismo ignora as condições históricas dos sujeitos

⁶ O texto foi originalmente em 1987 e republicado em 2013 em homenagem à autora.

e rejeita a consciência, a subjetividade. Assim, na concepção de ensino racionalista e tecnicista, o professor é um treinador auxiliar do aluno, que centraliza a aprendizagem no fazer, na repetição.

Outro parâmetro de aprendizagem é o referente à assimilação, na qual ocorre a adição de nova aprendizagem a um esquema já existente. É uma aprendizagem típica de disciplinas em que novos conteúdos são adicionados aos anteriores, já compreendidos. Nessa aprendizagem há memorização, mas não se reduz a reprodução, decorrem explicações e aplicações do aprendido e ela é internalizada. (ILLERIS, 2013).

Para Giusta (2013) a segunda corrente de aprendizagem que toma por base a teoria de Piaget, ainda que ele afirme tratar-se de uma teoria do desenvolvimento do pensamento, é a de que o conhecimento não procede, nem da experiência única dos objetos, nem de uma ampla programação inata, pré-formada no sujeito, mas de construções sucessivas com elaborações constantes de estruturas novas. Na teoria piagetiana as relações entre sujeito e objeto são fundamentais para provocar a desestruturação das estruturas existentes, o que gera uma nova acomodação, isto é a modificação das estruturas do pensamento. Assim, o modelo de ensino originário desta teoria é da provocação da aprendizagem por meio da problematização, em que o processo de reflexão em torno do objeto produz uma interação.

Uma terceira corrente de aprendizagem examinada por Giusta (2013) relativa às práticas pedagógicas fundamentadas na concepção interacionista de aprendizagem assentada em Vygotsky, que reconhece que todo conhecimento provém da prática social e de que o conhecimento é uma construção coletiva, em que o sujeito é participante e não gera a aprendizagem em solidão. Ainda que Vygotsky possa ser compreendido com uma concepção epistemológica interacionista/construtivista, suas proposições são diferenciadas das de Piaget, pois Vygotsky afirma que o ambiente social e cultural é determinante do desenvolvimento humano e que isso acontece, fundamentalmente, pela relação entre aprendizagem e linguagem. Para Vygotsky o sujeito é um ser histórico, resultante de um conjunto de relações sociais, na qual a consciência é engendrada no social, a partir das relações que os sujeitos estabelecem entre si, por meio da mediação da linguagem. Os signos são os instrumentos que, agindo internamente no sujeito, promovem transformações internas fazendo-o deixar de ser apenas um ser biológico para se tornar um ser sócio-histórico. Para

Giusta (2013) a aprendizagem no interacionismo está associada ao desenvolvimento, quanto mais aprendizagem mais desenvolvimento, quanto mais desenvolvimento mais aprendizagem. Com efeito, o desenvolvimento humano é entendido não decorrente de elementos isolados que amadurecem, tampouco de fatores ambientais que atuam sobre o sujeito controlando seu comportamento, mas sim resultante de trocas recíprocas estabelecidas durante toda a vida, entre indivíduo e meio, cada fator influenciando sobre o outro. Desse modo, essa corrente parece mais apropriada no entendimento da aprendizagem da educação não-formal e informal, mesmo que seja reconhecida na educação formal.

Nesta concepção se admite a existência de Zonas de Desenvolvimento Proximal (ZDP), isto é, a aprendizagem em elaboração pelo sujeito em que a ajuda do outro pode se constituir num apoio para provocar a aprendizagem. Assim, entendendo que o aprendizado de orientação a objetos com a linguagem de programação Java e com a gamificação como metodologia, pode ser um elemento de interação para ajudar o desenvolvimento de aprendizagem. Considerando a ZDP, a aula pode interagir com o sujeito e o sujeito interagir com aula gerando aprendizagem da programação orientada a objetos, pela criação de condições acessíveis à linguagem, em português, favorecendo a compreensão desse tema. A aula, quando bem elaborada, pode constituir como uma ferramenta de interação de aprendizagem.

Em relação a zona de desenvolvimento proximal, Prestes (2010) tem chamado atenção de que as obras de Vygotsky, traduzidas do inglês para o português. O autor esclarece que Vygotsky se referia não só ao que uma criança sabe, mas ao que ela pode vir a aprender. O estudo do desenvolvimento “não só determina o nível de desenvolvimento atual [...], mas sonda as funções que ainda não concluíram seu desenvolvimento e se encontra na zona de desenvolvimento imediato (Vygotsky 2004, p. 536-537 in Prestes 2010, p.171). Com efeito, para Prestes a “zona de desenvolvimento imediato define as funções ainda não amadurecidas, mas que se encontram em amadurecimento, as funções que estarão amadurecidas amanhã, e eu hoje estão em estado embrionário” (Vygotsky, 2004, p. 379 in Prestes 2010, p.173), ou seja, zonas de amadurecimento iminente.

Com o advento da informática e da educação a distância, os canadenses George Siemens e Stephen Downes, desenvolverem as proposições sobre a

aprendizagem no conectivismo, considerando que a aprendizagem é o processo facilitador da construção de redes de informação. A aprendizagem conectivista centra-se na construção e manutenção de conexões em rede, o que a torna flexível, possibilitando sua aplicação e situações problema emergentes. Coelho (2019), ao discorrer sobre o conectivismo, sinaliza que esta teoria foi desenvolvida considerando o ambiente propiciado pelas nova tecnologias que favorecem a conexão entre as pessoas e com o conhecimento. A tecnologia modificou o modo como as pessoas vivem, como se comunicam e como aprendem. Siemens (2004, p. 8) destaca que o conhecimento está disponível nas redes de informação: “Aprendizagem e conhecimento repousam na diversidade de opiniões.”

O acesso à informação foi ampliado e é abundante, assim na aprendizagem não há necessidade de memorização, portanto a capacidade de encontrar o conhecimento é fundamental. Assim, os principais pontos em que apoia essa teoria consistem em:

A aprendizagem e conhecimento apoiam-se na diversidade de opiniões.

A aprendizagem é um processo de conectar nós especializados ou fontes de informação.

A aprendizagem pode residir em dispositivos não humanos.

A capacidade de saber mais é mais crítica do que aquilo que é conhecido atualmente.

É necessário cultivar e manter conexões para facilitar a aprendizagem contínua.

A habilidade de enxergar conexões entre áreas, ideias e conceitos é uma habilidade fundamental.

Atualização (conhecimento acurado e em dia) é a intenção de todas as atividades de aprendizagem conectivistas. (Siemens 2004, in Coelho, 2019, p. 15).

Além disso, o conectivismo argumenta que o processamento mental e da resolução de problemas pode ser descarregado em máquinas, em Siemens (2004) levanta a possibilidade de que “a aprendizagem pode residir em um utensílio não humano”. No dizer de Siemens (2004, p. 5):

A aprendizagem é um processo que ocorre dentro de ambientes nebulosos onde os elementos centrais estão em mudança – não inteiramente sob o controle das pessoas. A aprendizagem (definida como conhecimento acionável) pode residir fora de nós mesmos (dentro de uma organização ou base de dados), é focada em conectar conjuntos de informações especializados, e as conexões que nos capacitam a aprender mais são mais importantes que nosso estado atual de conhecimento.

Siemens (2004) alega que o sujeito nas relações sociais internaliza o conhecimento, assim é suficiente que seja ativado para que se converta em aprendizagem, o que ele chama de “conhecimento acionável”, como destaca

Coelho (2019). Assim, em dispositivos não humanos que armazenam conhecimento, podem aprender.

O conectivismo têm sido alvo de muitas críticas, que não serão exploradas neste texto, uma vez que a dissertação focaliza a criação de aula gamificada para o aprendizado de orientação a objetos com a linguagem Java. Assim, esses pressupostos podem ajudar na argumentação da contribuição desta aula como um espaço de conexão, em que os sujeitos podem aprender a orientação a objetos e a linguagem Java.

Uma quarta forma de aprendizagem apresentado por Illeris (2013) é a referente à aprendizagem significativa, expansiva, transicional e transformadora. Essa aprendizagem modifica o próprio sujeito de modo profundo, trata-se de uma reestruturação dos padrões existentes. Esse tipo de aprendizagem é também assentado na teoria de Vygotsky, em que o objeto e o sujeito interagem por meio da mediação. A mediação pode ter a contribuição de um outro sujeito e de materiais e condições de promoção de aprendizagem. O modelo se complexifica ao entender que esse sujeito e o objeto se situam em um contexto que também se envolve na aprendizagem, tanto como comunidade e como artefatos, além das regras estabelecidas e usuais nesses envolvimentos. Os elementos da aprendizagem se ampliam ao envolver sujeito, artefatos, comunidade, regras, ações, em colaboração. Uma aprendizagem interdependente.

Especificamente, na aprendizagem da educação não-formal para Guimarães e Faria-Fortecoëf (2021) e, ao analisarem histórias de vida em biografias de imigrantes, identificaram como saberes constituídos nas aprendizagens: saber aprender; saber compreender; saber projetar-se. Assim, o saber aprender se expressa nas relações com os pares na educação formal, mas também não-formal e informal, pois aprende sem ser ensinada. E há aprendizagem em espaços diferentes da escola, como em reuniões sociais, leitura de jornais, entre outros artefatos. As inqueridas na pesquisa de Guimarães e Faria-Fortecoëf (2021) tiveram possibilidades de obterem apoios para o desenvolvimento de seu saber aprender.

Em relação ao saber compreender, a educação (formal, não-formal e informal) foram identificadas como opções das famílias ainda na infância, pois a família valorizava a escola e as aprendizagens que ela proporcionava. Por essa razão Guimarães e Faria-Fortecoëf (2021) dizem que a educação formal se apresenta como possibilidade de aprendizagem compreensiva, em que a

autoaprendizagem desempenha papel importante. Nas dificuldades escolares, o fato de buscar melhorar por si mesma é fator preponderante de promoção da aprendizagem. E o saber projetar-se trata de uma aprendizagem marcada pela autoformação, ressaltam Guimarães e Faria-Fortecoëf (2021, p. 6) “Não estando em oposição relativamente às aprendizagens anteriores, mas antes em complementaridade, saber projetar-se consiste em identificar e respeitar as regras estabelecidas por outros”. Uma vez entendidas as regras, os sujeitos são capazes de criar as próprias regras. Dessa forma, saber projetar-se implica “inscrever as ações dos sujeitos ou os seus atos no mundo em mudança”. Essa aprendizagem permite ao sujeito compreender quando precisa de mais formação, ou seja, uma autocompreensão de si, fundamentais para compor uma educação não-formal e informal, como afirmam os autores “elas não deixam de perspectivar os obstáculos como oportunidades de desenvolvimento de novas aprendizagens profissionais e pessoais, através de modalidades educativas formais, não formais e informais” (GUIMARÃES e FÁRIA-FORTECOËF, 2021, p. 7). Há uma complementariedade entre educação (formal, não-formal e informal) e a (auto, hetero e eco) formação que passam a compor as aprendizagens, ao longo da vida.

Como esse estudo trata da aprendizagem de adultos é importante considerar as proposições de Freire (1996) sobre aprendizagem. Uma das principais indicações de Freire (1996) é de que uma aprendizagem autêntica é aquela que transforma o sujeito, ou seja, o ensino considera como ponto de partida os saberes e experiências do aprendiz, mas não persiste, pois eles são reconstruídos e ampliados. “Nas condições de verdadeira aprendizagem, os educandos vão se transformando em reais sujeitos da construção e da reconstrução do saber ensinado, ao lado do educador igualmente sujeito do processo” (FREIRE, 1996, p. 26). Ao realizar a reflexão sobre seus saberes os aprendizes tornam-se autônomos, questionadores construindo sua emancipação. Em Freire (1996) a educação é libertadora, uma vez que a sociedade é compreendida como uma grande comunidade educativa, portanto a aprendizagem é resultado da troca, da cooperação e da colaboração entre as pessoas. Peroza (2014) salienta que para Freire “A “não percepção” de algo “novo” pode ser consequência de inúmeros fatores, como: passividade, falta de criatividade ou o cansaço e a resignação diante da pressão das forças condicionantes da realidade”. Portanto para provocar a aprendizagem é “que se

buscam - com insistência - ações seguidas de reflexões para problematizar as situações-limite e provocar sua transformação, a consciência alarga os horizontes da criatividade ou inventividade e supera a condição de inércia” (PEROZA, 2014, p. 145).

O essencial em Freire na compreensão da aprendizagem, para esta investigação, são as marcas das aprendizagens como arte de socialização e de interação, quer dizer, ao referir-se as proposições freirianas, busca de transformação dos sujeitos em sujeitos que não sabem ou sabem pouco, em sujeitos capazes de inserir o que sabem para melhorar suas vidas.

O novo homem e a nova mulher jamais serão o resultado de uma ação mecânica, mas de um processo histórico e social profundo e complexo. O novo homem nasce aos poucos, vem tornando-se, não nasce feito, mas se faz no cotejamento das reflexões que desenvolve no processo de socialização. (PEROZA, 2014, p. 148).

Assim, entre as proposições de Paulo Freire (1987), conforme Reis (2021) está a superação da educação bancária, a superação de uma aprendizagem centrada na memorização, produzida por um ensino que focaliza a transmissão de conteúdos. Essa aprendizagem mantém as pessoas alienadas, dominadas e oprimidas, em que os sujeitos se tornam objetos e não pessoas. Lembra a autora que Freire (1987) entende conteúdos como “retalhos da realidade desconectados da totalidade em que se engendram e em cuja visão ganhariam significação”. Essa aprendizagem não ajuda os alunos a serem “protagonistas da produção do conhecimento; pelo contrário, atua na castração da curiosidade dos educandos, além de torná-los passivos e acríticos” (REIS, 2021).

Uma segunda proposição de Paulo Freire, é uma aprendizagem libertadora e problematizadora, base do processo de humanização de professores e alunos, uma aprendizagem que focaliza uma educação como “prática social humanizadora” (FREIRE, 2019, p. 13, in REIS, 2021). Os sujeitos e seus educadores são envolvidos, provocados, não se trata de uma aprendizagem que elimina a busca, ou seja, é uma aprendizagem que não apresenta o conhecimento acabado.

A educação libertadora, proposta e defendida por Freire em diálogo com Freire e Shor (2011), é, “fundamentalmente, uma situação na qual tanto os professores como os alunos devem ser os que aprendem; devem ser os sujeitos cognitivos, apesar de serem diferentes. [...]”, como indica Reis (2021), pois os professores e os alunos são parceiros críticos para juntos conhecerem. Nessa perspectiva a educação não é uma ação ingênua, mas busca o pensamento

crítico. Como destaca Reis (2021) que “O processo libertador não é só um crescimento profissional. É uma transformação ao mesmo tempo social e de si mesmo, um momento no qual aprender e mudar a sociedade caminham juntos”. Ao citar Freire e Shor (2011, p. 90), o educador e educando se educam constantemente, considerando o inacabamento do ser humano (FREIRE, 1987).

Reis ainda destaca que Paulo Freire (2019) valoriza a educação transformadora, ou seja, uma aprendizagem nascida nas relações entre alunos, professores, escola e sociedade, para contribuir com transformação social. Essa aprendizagem é gerada no diálogo e escuta, uma metodologia que acolhe o compromisso de uma educação libertadora e humanizadora, portanto a não neutralidade da aprendizagem, mas o reconhecimento da educação como um ato político, um conhecimento colocado em prática (FREIRE e SHOR, 2011, in REIS, 2021).

E finalmente, ainda que não de modo exaustivo, são feitas algumas considerações sobre a aprendizagem no trabalho e para o trabalho. Em consulta aos periódicos da base de dados Educa da Fundação Carlos Chagas esta situação é abordada na perspectiva do ensino profissionalizante, na formação profissional e não constam artigos na vertente da aprendizagem para e no trabalho. Essa consulta foi realizada em agosto de 2022. Assim, são consideradas as proposições de Wildemeersch e Stroobants (2013) sobre a perspectiva da aprendizagem transacional que focalizada uma teoria de aprendizagem gerada no trabalho.

Wildemeersch e Stroobants (2013) em sua teoria de aprendizagem transacional examinam as experiências advindas pelo exercício do trabalho considerando narrativas de história de vida de trabalhadoras. Entendem como aprendizagem no trabalho uma possibilidade de emancipação do sujeito, ainda que o trabalho remunerado não se destine a essa finalidade, entretanto, afirmam que o exercício do trabalho pode ajudar a pessoa descobrir e transformar um processo de aprendizagem.

Segundo os autores, essa possibilidade ocorre quando o sujeito enfrenta as mudanças ocorridas no trabalho como desafios. Nesse processo, quatro etapas são presentes: adaptação, desenvolvimento (crescimento), distinção e resistência. Essas etapas são perpassadas por tensões que podem superar o desânimo e estimular a busca de atividades para a reconstrução de si. As estratégias presentes nesse processo envolvem a estimulação das demandas

pessoais e sociais em que o sujeito se reconstrói ao perceber oportunidades para si diante das mudanças. Ao redesenhar essas situações é possível apresentar dois polos que dialogam com as práticas baseadas em uma percepção crítica de resistência, de modo pragmático e na possibilidade de adaptação às mudanças, e conseqüentemente gerar aprendizagens.

Para Wildemeersch e Stroobants (2013) a aprendizagem transacional não decorre de um processo linear intencional, mas pode se constituir uma possibilidade de conexão entre os conflitos pessoais e as possíveis interações realizadas no ambiente de trabalho.

Assim, pode-se sumarizar esse item que abordou as diferentes concepções da aprendizagem, desde o associacionismo, e tem como expressão mais importante o behaviorismo, as proposições da teoria do desenvolvimento do pensamento desenvolvidas por Piaget, que originaram o construtivismo, a concepção interacionista de aprendizagem assentada em Vygotsky, que reconhece que todo conhecimento provém da prática social e a ela retorna e a de que o conhecimento é uma construção coletiva. A essas proposições se somam os postulados de uma aprendizagem libertadora e transformadora em Paulo Freire e por fim as contribuições Wildemeersch e Stroobants (2013) para a aprendizagem transacional realizadas no trabalho e para o trabalho. Ainda, foram consideradas as proposições do conectivismo devido a proposta de uma plataforma para possíveis aprendizagens da linguagem Java, portanto um espaço de conhecimento em rede.

Diante desse conjunto de concepções, essa investigação na formulação do produto busca considerar as proposições assentadas em pressupostos de uma aprendizagem coletiva, colaborativa, tal como afirmado nas abordagens de ensino que contribuam para a formação de programadores em Java. Considera importante, nesta proposição de plataforma os indicativos que advém de autores que pesquisam o assunto, mas valoriza os conhecimentos de uma aprendizagem transacional de profissionais da área de engenharia de software. Entende que os saberes produzidos na experiência são importantes para o entendimento de como realizar uma programação em Java, principalmente para iniciantes em programação.

3 FUNDAMENTOS DE LINGUAGENS DE PROGRAMAÇÃO, JAVA E ORIENTAÇÃO A OBJETOS

As linguagens de programação desempenham um papel fundamental no desenvolvimento de sistemas de computador, permitindo que os desenvolvedores criem aplicações e soluções para uma ampla gama de problemas. Entre essas linguagens, a linguagem Java se destaca como uma das mais populares e versáteis, utilizada em diversos contextos, desde o desenvolvimento de aplicações e sites, que são acessíveis e funcionais em navegadores de internet (web), e dispositivos móveis, como smartphones e tablets (mobile) até sistemas corporativos e aplicações científicas. Além disso, Java foi projetado com um forte foco na programação orientada a objetos (POO), um paradigma que oferece uma abordagem eficiente e reutilizável para organizar e estruturar o código.

Neste capítulo, exploraremos as linguagens de programação em geral, examinando seu propósito, evolução e principais características. Em seguida, focaremos na linguagem Java, discutindo sua história, principais recursos e razões para sua popularidade duradoura. Por fim, abordaremos a programação orientada a objetos, apresentando seus conceitos fundamentais e como eles são aplicados na prática.

3.1 LINGUAGENS DE PROGRAMAÇÃO

Os microcomputadores são capazes de processar grandes quantidades de dados e oferecem inúmeras possibilidades em todas as áreas da vida humana. Hoje, os computadores fazem parte do nosso dia a dia, desde smartphones e tablets até computadores pessoais, laptops e *smart* TVs. Os carros elétricos também contam com sistemas integrados, como GPS (*Global Positioning System* ou Sistema de Posicionamento Global em inglês) com atualização pela internet e telas com mapas mundiais atualizados, capazes de calcular em tempo real a autonomia e distância entre o ponto de partida e chegada.

Essas possibilidades são possíveis graças ao desenvolvimento de metodologias que permitem a tradução de ideias e necessidades em instruções para a máquina. Existem diversas aplicações dessas instruções, como destacado por Sebesta (2018), incluindo aplicações científicas, empresariais, inteligência artificial e software para a Web. Cada uma dessas aplicações

desempenha um papel importante em nossas vidas, permitindo avanços em áreas como ciência, negócios e tecnologia, ajudando a impulsionar a inovação em todas as áreas da sociedade.

A comunicação com computadores é possível graças às linguagens de programação, que são linguagens escritas e formais, que especificam um conjunto de instruções e regras utilizadas para gerar programas.

A história das linguagens de programação remonta ao século XIX, quando foram desenvolvidos os primeiros dispositivos de programação como o tear Jacquard e o Motor Analítico de Charles Babbage. Sebesta (2018) afirma que a primeira linguagem de programação, entretanto, foi desenvolvida em meados do século XX, com a criação do FORTRAN (*Formula Translation*) em 1954 pela IBM⁷. Nos anos seguintes, várias outras linguagens de programação foram desenvolvidas, incluindo COBOL (*Common Business Oriented Language*), BASIC (*Beginners All-purpose Symbolic Instruction Code*), e ALGOL (*Algorithmic Language*).

Com as linguagens de programação, é possível escrever códigos que realizam diversas tarefas, desde simples operações matemáticas até o desenvolvimento de aplicações complexas de software. As linguagens de programação também permitem que os programadores criem programas para diferentes plataformas e sistemas operacionais, incluindo desktops, dispositivos móveis e servidores.

As linguagens de programação foram desenvolvidas para facilitar o processo de escrita de instruções e comandos sequenciais, que serão interpretados e processados pelos computadores a fim de alcançar os resultados desejados. Entretanto, conforme apontam Louden e Lambert (2011), antes de meados da década de 1940, os operadores de computador "conectavam" seus programas, ajustando switches e fiação interna dos computadores para executar tarefas específicas.

A noção de fornecer comandos aos computadores para processarem e produzirem resultados esperados pode parecer inicialmente simples segundo Skiena (2020); entretanto, essa prática pode evoluir rapidamente para um cenário extremamente complexo e intrincado. Para Goodrich et al. (2021) ao buscar a proficiência em uma linguagem de programação específica, não é

⁷ A International Business Machines Corporation (IBM), apelidada de Big Blue, é uma multinacional americana de tecnologia sediada nos EUA e presente em mais de 175 países.

suficiente apenas aprender a linguagem propriamente dita; é imprescindível também dominar os fundamentos de diversas linguagens de programação. Tais fundamentos incluem, entre outros, algoritmos e estruturas de dados.

A palavra algoritmo vem do nome do matemático persa Abu Abdullah Mohammad Ibn Musa al-Khwarizmi, nascido em Khawarizm (Kheva), ao sul do mar Aral, que viveu no século XVII. A influência de Khawarizmi no crescimento da ciência em geral, particularmente na matemática, astronomia e geografia, é bastante reconhecida. Também é considerado o fundador da álgebra (MEDINA e FERTIG, 2018). Pode-se também definir como: Está associada a um conjunto de regras e operações bem definidas e ordenadas, destinadas à solução de um problema ou de uma classe de problemas, em um número finito de passos (MANZANO e OLIVEIRA, 2016).

Uma das principais funcionalidades de um computador é a manipulação de informações, por isso é necessário que haja formas de trabalhar com diferentes dados em um programa. Apesar de internamente o computador manipular unicamente números, as linguagens de programação permitem que utilizemos outros tipos de informações nos programas de forma transparente (MEDINA e FERTIG, 2018).

Estruturas de dados são maneiras de armazenar informações para que possam ser posteriormente manipuladas de forma eficiente e adequada. Elas podem incluir tipos primitivos, como números inteiros, ou estruturas mais complexas e sofisticadas, como listas, *arrays*, árvores binárias e grafo⁸. Essas estruturas são amplamente utilizadas visando a praticidade, reusabilidade do código e troca de informações entre diferentes camadas ou componentes de aplicativos, seja internamente ou externamente, como na utilização de *Application Programming Interfaces* (APIs).

Conforme Loudon e Lambert (2011), o domínio desses conceitos é essencial para obter a perspectiva necessária e compreender o impacto que as linguagens de programação e suas estruturas têm na forma como solucionamos problemas com computadores e no modo como pensamos sobre computação.

Além de algoritmos e estruturas de dados, existem outras abordagens elegantes e de alto nível que também são valiosas para aprender e podem ser aplicadas em diferentes linguagens de programação. Entre elas, destacam-se os

⁸ Grafo se tornou notoriamente famosa como principal forma de visualização de estrutura de redes sociais.

padrões de projeto e conceitos relacionados à interface e experiência do usuário, como UI (*User Interface*) e GUI (*Graphical User Interface*), bem como UX (*User Experience*), que é uma tendência atual no campo de desenvolvimento de software. Essas abordagens complementam o aprendizado de linguagens de programação e enriquecem o conhecimento do desenvolvedor.

As linguagens de programação são importantes porque permitem a automação de tarefas repetitivas e rotineiras, como o processamento de dados e a execução de algoritmos complexos. Com o uso das linguagens de programação, é possível acelerar o desenvolvimento de soluções tecnológicas e melhorar a eficiência em diversas áreas da sociedade.

A maneira que nos comunicamos tem influência em como pensamos, e vice-versa. Similarmente, como nós programamos os computadores influencia como nós pensamos sobre computação e vice-versa (LOUDEN e LAMBERT, 2011). Como a leitura de mundo de cada sujeito é diversa cada programador opta e prefere uma determinada linguagem. Segundo Travassos Sarinho (2003), a diversidade existente de linguagens provoca entre os programadores intensos e intermináveis debates sobre os méritos de cada uma delas, tornando-se muitas vezes quase um ato de fé. O debate no meio acadêmico, entre cientistas da computação, se direciona mais para a busca de desenvolvimento de “linguagens que combinem poder de expressão com simplicidade e eficiência” (TRAVASSOS SARINHO, 2003, p.16).

As linguagens de programação podem ser classificadas com base em vários critérios, incluindo sua sintaxe, nível de abstração e paradigmas. Elas podem ser classificadas com base em sua sintaxe, que se refere às regras e à estrutura da linguagem. elas podem ser divididas em dois tipos principais: linguagens de procedimento e linguagens declarativas, conforme mencionado por Sebesta (2018).

Linguagens de procedimento, como C e Pascal, são baseadas em sequências de instruções que o programa executa em uma ordem específica. Nessas linguagens, o programador deve especificar explicitamente o processo passo a passo que o programa deve seguir para alcançar um resultado desejado. Por outro lado, linguagens declarativas, como SQL (*Structured Query Language*) e Prolog, funcionam de maneira diferente. Em vez de especificar uma sequência de instruções, o programador define um conjunto de regras ou condições que descrevem o comportamento esperado do programa. A computação ocorre de

forma mais abstrata, deixando para o sistema a responsabilidade de determinar a melhor maneira de resolver o problema, com base nas regras fornecidas. A classificação das linguagens de programação também pode ser feita com base no seu nível de abstração. Isso se refere ao quanto de detalhe o programador precisa considerar ao escrever o código. Linguagens de alto nível de abstração exigem menos detalhes e estão mais próximas da linguagem humana, enquanto linguagens de baixo nível de abstração envolvem mais detalhes técnicos e estão mais próximas da linguagem de máquina.

Existem três níveis principais de abstração em linguagens de programação: baixo nível, alto nível e muito alto nível. Cada nível de abstração reflete o quão próximo ou distante a linguagem de programação está da linguagem da máquina do computador e o grau de simplicidade ou complexidade envolvido na escrita do código.

Linguagens de baixo nível, como Assembly, C, FORTRAN e COBOL, são mais próximas da linguagem da máquina e geralmente exigem que os programadores gerenciem detalhes técnicos específicos do hardware do computador. Isso pode tornar essas linguagens mais difíceis de aprender e usar, mas também permite maior controle e otimização do desempenho do programa.

As linguagens de alto nível, mencionadas por Van Roy e Haridi (2004), como C++ e Java, oferecem um nível de abstração maior, facilitando o aprendizado e a portabilidade do código entre diferentes sistemas e plataformas. Essas linguagens abstraem muitos detalhes do hardware, permitindo aos programadores se concentrarem na lógica do programa, enquanto o compilador cuida da conversão do código para a linguagem de máquina.

Por fim, as linguagens de muito alto nível, conforme Loudon e Lambert (2011), como Python e Ruby, fornecem ainda mais abstração, permitindo expressividade e facilidade de uso na escrita de código. Essas linguagens são projetadas para simplificar o desenvolvimento de software, ocultando detalhes técnicos e permitindo que os programadores se concentrem na solução de problemas e na criação de algoritmos de alto nível.

Dessa forma, a escolha da linguagem de programação adequada depende das necessidades do projeto e das habilidades do programador, levando em consideração o nível de abstração desejado e o equilíbrio entre controle do hardware, portabilidade e facilidade de uso.

As linguagens de programação também podem ser classificadas com base em seus paradigmas de programação, o que se refere à abordagem ou metodologia utilizada pela linguagem para resolver problemas. Existem vários paradigmas de programação, incluindo procedimental, orientada a objetos, funcional e de script. Alguns autores se referem às linguagens de *scripting* como uma categoria à parte (SEBESTA, 2018).

Linguagens de programação orientadas a objetos, como Java e C++, baseiam-se no conceito de objetos ou instâncias de classes, que por sua vez contêm dados e métodos. Linguagens de programação funcional, conforme Van Roy e Haridi (2004) como Haskell e Lisp, são baseadas no conceito de funções. Já as linguagens de *scripting*, como Python e Ruby, são projetadas para tarefas de prototipagem rápida e automação.

As linguagens de programação possuem uma série de características que as tornam apropriadas para diversas aplicações e cenários de uso. De acordo com Sebesta (2018) e Van Roy e Haridi (2004), as principais características das linguagens de programação incluem:

Os tipos de dados suportam diferentes tipos de dados, como inteiros (usados para representar números inteiros), números de ponto flutuante (para representar números reais com precisão simples ou dupla), strings (sequências de caracteres), listas (sequências ordenadas de valores) e dicionários (pares de chave-valor).

As estruturas de controle ajudam os programadores a gerenciar o fluxo de um programa. Essas estruturas incluem loops (*for* e *while*) para repetir blocos de código várias vezes, instruções condicionais (*if*, *else* e *elif*) para executar blocos de código apenas quando uma condição específica for verdadeira e funções que permitem a reutilização de código e a criação de blocos de código que realizam uma tarefa específica, facilitando a manutenção e o entendimento do programa.

O gerenciamento de memória pode ser alcançado por mecanismos variados, alguns exemplos incluem coleta automática de lixo em linguagens como Java e Python, que é responsável por liberar automaticamente a memória não mais utilizada. Em linguagens como C, a alocação e desalocação de memória são feitas manualmente usando funções específicas, como *malloc* e *free*. Já em C++, os ponteiros inteligentes gerenciam automaticamente a liberação de recursos quando não são mais necessários.

A entrada e saída possibilita diversas maneiras para a manipulação de dados, que envolve a leitura e escrita em arquivos; a programação de redes, que abrange a comunicação entre sistemas e dispositivos através de protocolos de rede; e a entrada/saída de consoles, que envolve a interação do usuário com o programa por meio de interfaces de texto ou gráficas.

As bibliotecas são conjuntos modulares de código pré-escritos, desenvolvidos para auxiliar na realização de tarefas comuns. Elas podem otimizar tempo e esforço, além de aprimorar a qualidade e confiabilidade do código desenvolvido. Algumas funções comuns fornecidas pelas bibliotecas incluem processamento de strings (busca, substituição, formatação e manipulação de textos), manipulação de datas (cálculo de intervalos de tempo, conversões entre formatos de data e manipulação de fusos horários), matemática avançada (trigonometria, álgebra linear e estatística), comunicação e protocolos de rede (facilitando a implementação de aplicações de rede, como servidores web, clientes e protocolos específicos) e acesso a banco de dados (provendo mecanismos para interagir com sistemas de gerenciamento de bancos de dados).

Com base nessas características, os programadores podem selecionar a linguagem de programação mais adequada às necessidades de suas aplicações, garantindo maior eficiência e eficácia na resolução dos problemas propostos. Cada linguagem de programação possui seus próprios pontos fortes e fracos, de tal maneira que é importante avaliar criteriosamente a aplicação em questão e as habilidades da equipe de desenvolvimento.

Vale salientar também que as linguagens de programação são utilizadas em uma ampla gama de aplicações, incluindo desenvolvimento de software, desenvolvimento web, ciência de dados, inteligência artificial e desenvolvimento de jogos.

No desenvolvimento de software, as linguagens de programação desempenham um papel crucial na criação de aplicações de software para desktops, dispositivos móveis e ambientes web. Algumas das linguagens de programação mais populares e amplamente utilizadas no desenvolvimento de software incluem Java, C++ e Python (SEBESTA, 2018).

Já no desenvolvimento Web, as linguagens de programação são usadas para criar websites e aplicações web. Linguagens de programação populares para desenvolvimento web incluem JavaScript, PHP e Ruby

Por sua vez, na ciência de dados, segundo Louden e Lambert (2011), as linguagens de programação são usadas para realizar análise e visualização de dados. Linguagens de programação populares para ciência de dados incluem Python, R e SQL.

Na inteligência artificial, conforme Van Roy e Haridi (2004), as linguagens de programação são usadas para criar sistemas inteligentes que podem realizar tarefas complexas, tais como processamento de linguagem natural, visão computacional e aprendizagem de máquina. Linguagens de programação populares para IA incluem Python, Java e MATLAB.

No desenvolvimento de jogos, as linguagens de programação são usadas para criar jogos de vídeo e seus motores. O motor de um jogo de videogame, também conhecido como *game engine*, é o software responsável por fornecer as funcionalidades e recursos necessários para criar e executar um jogo. Ele é o coração do jogo, cuidando de aspectos como a renderização gráfica, física, áudio, animações, inteligência artificial, gerenciamento de recursos e interação do jogador. Linguagens populares para esta finalidade incluem C++, Java e Python.

Para Costa (2018) entre as características das linguagens de programação estão a legibilidade, a confiabilidade, a eficiência, a modificabilidade, a portabilidade, a reusabilidade e facilidade de aprendizado.

A legibilidade se refere aos requisitos de facilidade com que o programador enxerga seu código-fonte. Essa facilidade para o entendimento do código, é uma das possibilidades de sua aprendizagem como destaca Costa (2018). A segunda característica é a confiabilidade – que reporta ao quanto pode-se confiar em um código, em outras palavras, uma linguagem expressa maior confiabilidade quando tende a ser menos sujeita a erros de programação. Quando uma linguagem possui recursos como o de tratamento de exceções, sua confiabilidade é aumentada, pois a possibilidade de cometer erro é menor devido o programador observar com atenção as exceções. Ressalta Costa (2018) que a aplicação de boas práticas no tratamento de exceções no código, traz, entre outras vantagens, a confiança no código, o que amplia suas características de desenvolvimento de qualidade.

A eficiência: se caracteriza por atingir seus objetivos de maneira rápida e segura. Diz Costa (2018, p. 31) “Fazer o código correto, livre de erros, não é um objetivo fácil de se cumprir, principalmente para alguns tipos de aplicações, como

as bancárias, que exigem um nível alto de segurança e rapidez de acesso à movimentação”. Ao se observar a eficiência os vários tipos de riscos são reduzidos, aumentando a viabilidade da realização de um projeto.

A modificabilidade: se reporta as facilidades oferecidas pela linguagem para realizar mudanças de uma parte do código, sem a necessidade de mudar as outras partes. Essa possibilidade de realizar mudanças parciais nos códigos sem alterações do todo é considerada como vantagem principalmente o aprendizado em cursos de computação, pois se durante o estudo de programação, possíveis erros ocorrerem, será possível realizar os ajustes das partes sem mudanças em totalidade.

A portabilidade: se refere a capacidade do código poder ser utilizado em diferentes plataformas ou sistemas operacionais, independentemente da ferramenta empregada na sua tradução. Assim, um mesmo programa pode ser utilizado em diversos ambientes e situações sem que seja necessário fazer uma reescrita de código para adaptá-lo em um novo ambiente ou situação, como indica Costa (2018). Essa é uma vantagem para as empresas de desenvolvimento, pois um software que pode ser aplicado em várias plataformas favorece o aumento das demandas para sua utilização. Ainda, para o desenvolvedor essa característica ajuda pois não será necessário a reescrita do código, exigindo apenas realizar adaptações.

A reusabilidade: “É a capacidade de se utilizar o mesmo código, ou trecho do código em outra situação” (COSTA, 2018, p. 31). Essa característica favorece o desenvolvimento de um projeto de software com rapidez e agilidade, pois o reaproveitamento de código, reduz o tempo de desenvolvimento.

Por fim, a facilidade de aprendizado é uma característica que está diretamente relacionado à legibilidade, pois quando um código não tem boa legibilidade, não tem uma fácil compreensão, o que tornará o aprendizado mais difícil. Facilidade de aprendizado traz benefícios ao programador no domínio da linguagem e pode ocorrer de diversos modos, tanto por meio de uma sintaxe mais simples, que simplifique o entendimento, ou mesmo pela legibilidade de verificação dos códigos.

Em conclusão, as linguagens de programação desempenham um papel fundamental na era da computação moderna, permitindo que os desenvolvedores se comuniquem com os computadores e criem uma ampla gama de soluções, como aplicativos de software, websites e outros programas.

Essas linguagens podem ser classificadas de acordo com a sintaxe, o grau de abstração e os paradigmas de programação, cada uma com características específicas que a tornam mais apropriada para determinadas aplicações.

Além disso, as linguagens de programação oferecem recursos importantes, como tipos de dados, estruturas de controle, gerenciamento de memória, operações de entrada/saída e bibliotecas, que são essenciais para abordar diferentes problemas e necessidades do mundo real. Elas são empregadas em diversos campos, como desenvolvimento de software, web, ciência de dados, inteligência artificial e criação de jogos.

À medida que a tecnologia avança e as demandas do setor evoluem, as linguagens de programação continuarão a se adaptar, aprimorar e diversificar-se, dando origem a novos paradigmas de programação e técnicas inovadoras. Portanto, a compreensão e a adoção dessas linguagens por profissionais e entusiastas da área continuarão sendo cruciais para impulsionar o progresso tecnológico e enfrentar os desafios emergentes no mundo da computação.

3.2 LINGUAGEM JAVA

A linguagem Java foi criada, inicialmente concebida como parte de um projeto chamado *Oak*⁹, em 1991 na Sun Microsystems por James Gosling, Mike Sheridan e Patrick Naughton no Projeto Verde com intuito de escrever uma linguagem para aplicação em pequenos dispositivos eletrônicos. Conforme Gosling et al. (2021) a equipe logo percebeu que sua criação tinha aplicações muito mais amplas. Em 1995, a linguagem foi oficialmente renomeada Java e liberada para o público. Em 2009 a Oracle adquiriu a Sun Microsystems.

O principal objetivo deste projeto era desenvolver uma linguagem de programação para sistemas embarcados, tais como dispositivos de telecomunicações, aparelhos e outros eletrônicos de consumo. Seu nome se deve à um tipo de grão de café, inspirada pela popularidade da bebida entre desenvolvedores de software. Sua ênfase inicial era a portabilidade e uma pequena superfície para o código, como capturado no slogan “*Write once, run anywhere*” (WORA)¹⁰. Para atingir este objetivo, os programas Java compilariam para código de byte independente da máquina, e cada plataforma alvo forneceria

⁹ Carvalho em inglês

¹⁰ Nota do autor: “Escreva uma vez, rode em qualquer lugar” em inglês. No jargão da programação, rodar ou executar um programa ou aplicativo significa processá-lo. É o que acontece quando se clica uma (no celular) ou duas vezes (no computador) sobre o ícone de um determinado aplicativo ou programa.

sua própria máquina virtual Java ou intérprete para este código de byte (SEBESTA, 2018).

Ao longo de sua história, o Java tem evoluído continuamente, incorporando funcionalidades que impactam significativamente o desenvolvimento de software. A introdução do conceito de WORA e a JVM (Máquina Virtual Java) em 1996 possibilitou a criação de aplicativos que podem ser executados em diversas plataformas sem recompilação, simplificando a distribuição de software.

Em 1998, o Java 1.2 trouxe o Quadro de Coleções, melhorando a manipulação de conjuntos de dados, e a API gráfica Swing, facilitando a criação de interfaces de usuário mais sofisticadas e personalizáveis. Essas inovações permitiram o desenvolvimento de aplicações empresariais mais complexas e com interfaces gráficas aprimoradas.

Em 2004, o Java 5 introduziu recursos como genéricos, anotações, autoboxing e melhorias no suporte à concorrência, beneficiando os desenvolvedores com um código mais seguro, legível e fácil de manter. Essa versão permitiu a construção de aplicativos mais robustos e eficientes, como servidores de aplicação e sistemas de gerenciamento de conteúdo.

Com o lançamento do Java 7 em 2011, a linguagem simplificou o gerenciamento de recursos como arquivos e conexões de rede através da declaração de recursos experimentais, enquanto a nova API de arquivos I/O (NIO.2) proporcionou maior controle e flexibilidade no acesso e manipulação de arquivos. Essas melhorias permitiram a criação de aplicativos com maior desempenho e escalabilidade, especialmente em ambientes de nuvem e big data.

O Java 8, lançado em 2014, introduziu expressões lambda e fluxos, proporcionando uma abordagem mais funcional à programação e permitindo o desenvolvimento de código mais conciso e expressivo. A nova API de Data e Hora facilitou o trabalho com datas e fusos horários. Com essa versão, foram desenvolvidas aplicações que lidam com grandes volumes de dados e processamento paralelo, como sistemas de análise de dados e *machine learning*.

Em 2017, o Java 9 trouxe o *Java Platform Module System* (JPMS), que possibilitou a criação de aplicações Java mais modularizadas, favorecendo a organização e a manutenção do código. A modularização contribuiu para a redução do tamanho das aplicações e melhor desempenho em ambientes de

micro serviços. Desde 2017, a comunidade Java se comprometeu a lançar novas versões a cada seis meses, permitindo à linguagem acompanhar o ritmo do desenvolvimento de software moderno e responder rapidamente às necessidades dos desenvolvedores e do mercado.

Essa cadência de lançamento também facilitou a adoção de inovações e aprimoramentos na linguagem, beneficiando os desenvolvedores de diversas maneiras. Algumas dessas melhorias incluem a inferência de tipos locais no Java 10, o suporte ao protocolo HTTP/2 e a API de cliente HTTP no Java 11, aprimoramentos no desempenho da JVM no Java 12 e 13, a introdução dos registros (*Records*) e *Pattern Matching* para *instanceof* no Java 14, e o *sealed classes* e interfaces no Java 15.

A evolução constante e as melhorias incorporadas em cada lançamento fazem do Java uma linguagem de programação relevante e versátil. A cadência de lançamento semestral permite à linguagem acompanhar as demandas do desenvolvimento de software moderno, proporcionando aos desenvolvedores uma ampla variedade de recursos para criar soluções eficientes e eficazes em diversas indústrias e cenários de aplicação.

O compromisso da comunidade Java em manter a relevância da linguagem é evidente nas numerosas Propostas de Melhorias Java (JEPs) que abordam vários aspectos, como desempenho, segurança e usabilidade. À medida que o Java continua evoluindo, novos recursos e aprimoramentos são adicionados, proporcionando ainda mais benefícios para os desenvolvedores.

Por exemplo, o Java 17 trouxe o *Pattern Matching* para *switch statements*, aumentando a expressividade e a legibilidade do código, e uma API mais flexível e segura para geração de números pseudoaleatórios. O Java 18 introduziu o Project Loom, um projeto experimental destinado a simplificar a programação concorrente e melhorar o desempenho de aplicações com muitos threads. O Java 19 adicionou o Project Valhalla, que visa melhorar o desempenho e a eficiência da memória ao lidar com tipos de valor, oferecendo aos desenvolvedores a capacidade de criar aplicações com menor consumo de memória e melhor desempenho em cenários específicos.

O Java 20, lançado em 2023, representa o 11º lançamento de recursos entregue dentro do prazo através da cadência de lançamento de seis meses. A previsibilidade dos lançamentos possibilita que os desenvolvedores gerenciem

facilmente a adoção de inovações, graças a um fluxo constante de melhorias esperadas.

Para Sharan (2021) a abrangência e adaptabilidade do Java levaram a uma ampla adoção em diferentes setores, como finanças, saúde, telecomunicações e comércio eletrônico, sendo que Eckel menciona que grandes empresas de tecnologia, como Google, Amazon e Netflix, também dependem do Java para seus sistemas de backend e infraestrutura.

À medida que o Java continua a evoluir, os desenvolvedores se beneficiam dos recursos adicionados e das melhorias constantes na linguagem. Essa evolução permite que os desenvolvedores enfrentem novos desafios e criem soluções cada vez mais inovadoras e eficientes. A comunidade Java, por sua vez, mantém-se comprometida em continuar a desenvolver e aprimorar a linguagem, garantindo sua relevância e competitividade no cenário do desenvolvimento de software moderno.

Segundo Ivanović et al. (2013) Java pode ser aplicado em vários lugares, utilizando sequências de cursos introdutórios de ciências da computação, para curso em programação orientada à objeto em diversos níveis. As tecnologias Java Servlet e JavaServer Pages (JSP), assim como os populares frameworks para internet como Spring e JavaServer Faces (JSF), contribuíram para sua adoção generalizada no desenvolvimento web.

Um framework é uma estrutura pré-concebida de classes, interfaces e componentes de software que serve como base para o desenvolvimento de aplicações em uma linguagem de programação específica. Ele fornece uma arquitetura genérica e reutilizável para simplificar e agilizar o processo de desenvolvimento, fornecendo soluções para problemas comuns e permitindo que os desenvolvedores se concentrem na lógica específica da aplicação. Frameworks geralmente seguem padrões de projeto e arquiteturais que promovem a manutenibilidade, a escalabilidade e a modularidade do código.

A legibilidade de Java, os princípios orientados a objetos e a extensa documentação fizeram dela uma escolha popular para ensinar fundamentos de programação a iniciantes. Guzdial e Ericson (2021) assertam que muitas universidades e escolas adotaram o Java como a linguagem primária para cursos de programação introdutória.

Algumas das universidades que adotaram o Java como sua linguagem principal para cursos introdutórios de programação incluem: Universidade de

São Paulo (USP), Universidade Estadual de Campinas (UNICAMP) e Instituto Tecnológico de Aeronáutica (ITA) no Brasil; Universidade de Oxford no Reino Unido; Universidade de Stanford, Universidade de Harvard, Instituto de Tecnologia de Massachusetts (MIT) e Universidade da Califórnia em Berkeley, além do *College Board*, que é uma organização sem fins lucrativos dos Estados Unidos responsável pela elaboração e aplicação do exame de Colocação Avançada em Ciência da Computação, bem como de outros exames importantes, como o SAT (*Scholastic Assessment Test*) e o AP (*Advanced Placement*) nos Estados Unidos.

Há muitas outras escolas e universidades que também adotaram o Java como sua linguagem principal para cursos introdutórios de programação. É importante destacar que essa escolha pode variar de acordo com os objetivos e recursos de cada instituição.

O aprendizado de Java, para Eckel (2021), fornece uma base sólida para a transição para outras linguagens de programação. Seus princípios orientados a objetos, sintaxe e construções de programação são similares aos de C++, C# e outras linguagens populares. Além disso, a JVM do Java permite o uso de linguagens alternativas, como Kotlin e Scala, que podem rodar na mesma plataforma, facilitando ainda mais a transição para novas linguagens.

Sendo assim qualquer plataforma de hardware ou equipamento eletrônico que possa executar uma máquina virtual conseguirá executar Java, além disso, de acordo com Furgeri (2018) a linguagem pode oferecer um amplo conjunto de operadores destinados à realização de operações aritméticas, lógicas e relacionais, com a possibilidade de formar expressões de qualquer tipo. Além dos operadores matemáticos, existem também operadores lógicos e relacionais.

El-Sheikh et al. (2008) relatam que cursos introdutórios de programação de computador, geralmente, representam um desafio devido à variação na formação e experiência dos alunos, além do extenso tempo necessário para concluir os projetos de programação. São cursos importantes para alunos com especialização em computação, tecnologia da informação ou disciplinas de engenharia de software, por servirem como a base do currículo nessa área.

É notório que a questão da seleção de uma linguagem adequada para um curso de programação introdutório tem sido objeto de considerável reflexão. Embora possa haver divergência de opiniões, o consenso predominante na atualidade e tendência no ensino de programação é o favorecimento de

linguagens orientadas a objetos. Java gradativamente se tornou a linguagem de escolha para se ensinar iniciantes (IVANOVIĆ et al., 2013).

Java tem sido uma força motriz no cenário do desenvolvimento de software por quase três décadas. Sua história, sua ampla utilização em várias indústrias e sua importância no campo do aprendizado solidificaram seu lugar como uma linguagem de programação fundacional e influente.

Como uma linguagem versátil e independente de plataforma, Java tem sido amplamente adotada em diversos setores, incluindo finanças, saúde, telecomunicações e comércio eletrônico. Sua robustez e escalabilidade também o tornaram uma escolha popular para o desenvolvimento web, com tecnologias como servlets, JavaServer Pages e frameworks como Spring e JavaServer Faces impulsionando seu uso.

O banco de dados distribuído Apache Cassandra¹¹ é um exemplo notável de um projeto escrito em Java que tem sido amplamente utilizado em grandes empresas multinacionais, como Apple e Netflix, que lidam com um grande volume de dados e requisições diariamente. Criado em 2008 para atender às necessidades do Facebook em seu sistema de busca de mensagens, o Cassandra foi doado como software livre e continua sendo uma escolha popular para sistemas distribuídos. Além disso, o sistema operacional Android, criado pelo Google e amplamente utilizado em celulares em todo o mundo, foi originalmente escrito em Java, o que resultou em uma longa batalha legal entre Google e Oracle. Embora o Kotlin tenha se tornado a linguagem preferencial para o desenvolvimento de aplicativos Android, o Java ainda é uma alternativa viável e popular para muitos desenvolvedores.

No campo da aprendizagem, a legibilidade do Java, os princípios orientados a objetos e a extensa documentação, fizeram dele uma escolha atraente para o ensino de fundamentos de programação. Seu destaque nos currículos universitários e nos exames padronizados, ressalta ainda mais sua importância na educação em programação.

Além disso, o aprendizado de Java fornece uma base sólida para a transição para outras linguagens de programação, pois seus princípios, sintaxe e construções compartilham semelhanças com linguagens como C++, C# e

¹¹ O Apache Cassandra é um banco de dados distribuído de código aberto NoSQL em que milhares de empresas confiam para escalabilidade e alta disponibilidade sem comprometer o desempenho. A escalabilidade linear e a comprovada tolerância a falhas em hardware de *commodity* ou infraestrutura de nuvem fazem dele a plataforma perfeita para dados de missão crítica - <https://cassandra.apache.org>.

outras. A JVM do Java também suporta linguagens alternativas como Kotlin e Scala, facilitando aos desenvolvedores a exploração de novas linguagens dentro da mesma plataforma.

À medida que tecnologias emergentes como a Inteligência Artificial (AI), a Aprendizagem de Máquinas (ML) e a Internet das coisas (IoT) ganham impulso, Java parece estar bem posicionada para desempenhar um papel vital em seu desenvolvimento. Para Sharan (2021), a independência da plataforma Java, sua escalabilidade e robustez fazem dela uma escolha atraente para estas aplicações de ponta. Além disso, Guzdial e Ericson (2021) comentam o próspero ecossistema de bibliotecas e estruturas Java, como TensorFlow Java API para aprendizagem de máquinas e Eclipse Kura para IoT, apoia sua adoção nestes domínios.

TensorFlow é uma biblioteca de código aberto desenvolvida pelo Google *Brain Team*, lançada em 2015, para computação numérica e construção de modelos de aprendizado de máquina. É uma plataforma flexível e eficiente que oferece suporte a uma ampla variedade de algoritmos de aprendizado de máquina, incluindo redes neurais profundas e outras técnicas de inteligência artificial. Já Kura é um projeto de código aberto sob o guarda-chuva da Eclipse Foundation, que fornece uma plataforma de integração de IoT baseada em Java e OSGi (*Open Services Gateway initiative*). Kura visa simplificar o processo de desenvolvimento, implantação e gerenciamento de aplicativos IoT, fornecendo um ambiente consistente e modular para implementação em gateways de campo, dispositivos de borda (*edge devices*) e sistemas embarcados.

Espera-se que o papel de Java na educação em programação evolua à medida que novas metodologias e ferramentas de ensino forem sendo introduzidas. Ambientes de programação interativa como os Jupyter Notebooks foram adaptados para uso com Java, proporcionando aos alunos uma experiência mais envolvente e interativa. Jupyter Notebook é uma aplicação web de código aberto que permite criar e compartilhar documentos que contêm código-fonte, equações, visualizações e texto explicativo. É um ambiente interativo de programação que oferece suporte a várias linguagens de programação, como Python, R, Julia, Scala e muitas outras, embora Python seja a linguagem mais comum usada na plataforma.

Além disso, a integração do Java em plataformas de aprendizagem online, como Coursera e edX, expande o acesso à educação Java de qualidade

para alunos do mundo inteiro. Como as instituições educacionais continuam a se adaptar e inovar, o papel de Java na programação de ensino provavelmente evoluirá, assegurando sua contínua relevância no campo da aprendizagem.

A comunidade Java desempenha um papel crucial para o sucesso contínuo da linguagem. O ecossistema vibrante de desenvolvedores, educadores e colaboradores garante que Java continue sendo uma linguagem ativa e próspera. Para Eckel (2021) projetos de código aberto, como OpenJDK (*Open Java Development Kit*) e AdoptOpenJDK, demonstram o espírito colaborativo que impulsiona a inovação dentro da comunidade. O crescimento contínuo e o engajamento da comunidade Java serão essenciais para moldar a direção e o impacto futuros da linguagem.

OpenJDK é um projeto de código aberto que visa desenvolver uma implementação de código aberto do Java SE (*Java Platform, Standard Edition*). O OpenJDK foi iniciado pela Sun Microsystems, que foi posteriormente adquirida pela Oracle. Ele fornece o conjunto de bibliotecas e ferramentas básicas para executar e compilar aplicações Java e é a base para muitas outras implementações da plataforma Java. AdoptOpenJDK, por outro lado, é uma comunidade e projeto apoiados por várias organizações e membros da comunidade Java, como IBM, *Red Hat* e Microsoft, entre outros. Seu principal objetivo é fornecer compilações (builds) binárias pré-compiladas e bem testadas do OpenJDK para uma variedade de plataformas e sistemas operacionais. O AdoptOpenJDK também oferece suporte a longo prazo (LTS) para algumas versões do OpenJDK, o que é benéfico para organizações e desenvolvedores que precisam de estabilidade e suporte contínuo.

Concluindo, durante as últimas décadas, Java teve um impacto duradouro no cenário de desenvolvimento de software e na educação em programação. Sua história, amplo uso e significado no campo da aprendizagem consolidaram sua posição como uma linguagem de programação fundacional e influente. Como a linguagem continua a evoluir, espera-se que Java mantenha sua posição de influência no cenário de desenvolvimento de software.

A adaptabilidade de Java, a inovação contínua e o envolvimento com tecnologias emergentes, tais como inteligência artificial, aprendizagem de máquinas e a Internet das Coisas, tentam garantir que a linguagem permaneça relevante e competitiva. Além disso, o papel de Java na educação em

programação poderá evoluir à medida que novas metodologias e ferramentas de ensino forem introduzidas.

O contínuo crescimento e engajamento da comunidade aparentemente desempenhará um papel crucial no futuro da linguagem Java.

3.3 PROGRAMAÇÃO ORIENTADA A OBJETOS

A programação orientada a objetos (OOP) é um paradigma de programação amplamente adotado que gira em torno do conceito de objetos e suas interações para projetar e implementar programas de software. É uma forma de organizar o código em objetos que possuem propriedades e métodos, permitindo aos desenvolvedores criar aplicações complexas e modulares que podem ser facilmente estendidas e modificadas.

As raízes da programação orientada a objetos podem ser traçadas desde os anos 60 e 70, quando cientistas da computação começaram a explorar novas formas de organizar o código. Simula, uma linguagem de programação desenvolvida na Noruega nos anos 60, é frequentemente creditada por ser a primeira linguagem orientada a objetos. Ela introduziu o conceito de objetos e classes, o que permitiu aos desenvolvedores criar estruturas de dados complexas e hierárquicas, sendo uma pedra angular do desenvolvimento de software moderno devido a sua modularidade, capacidade de manutenção e reutilização.

Entretanto, foi o desenvolvimento do Smalltalk, uma linguagem de programação desenvolvida nos anos 70 no PARC (Centro de Pesquisa Palo Alto) da Xerox, que segundo Goldberg e Robson (1983) realmente popularizou a programação orientada a objetos. Smalltalk introduziu muitos dos conceitos que agora estão associados ao OOP, incluindo encapsulamento, herança e polimorfismo. Smalltalk foi amplamente utilizado na academia e pesquisa, e influenciou o desenvolvimento de muitas outras linguagens de programação orientadas a objetos, incluindo Java e C++.

A programação orientada a objetos é uma abordagem popular para o desenvolvimento de software que se baseia nos conceitos de classes e objetos. Segundo Budd (2019), os quatro princípios-chave da OOP são encapsulamento, herança, polimorfismo e abstração.

Em termos simples, uma classe define um modelo ou *blueprint* para criar objetos, que encapsulam dados e comportamento. Ela descreve as

propriedades, também chamadas de atributos, e os métodos, ou ações, que os objetos dessa classe possuirão. Um objeto, por sua vez, é uma instância específica de uma classe e representa uma entidade concreta com seus atributos e métodos.

O encapsulamento, refere-se à prática de agrupar dados e métodos que operam sobre os dados dentro de uma única unidade, ou objeto. Isso permite que os desenvolvedores ocultem detalhes de implementação do mundo externo, tornando mais fácil mudar o funcionamento interno de um objeto sem impactar sua interface externa, como afirmado por Saini e Kaur (2021).

Isso é importante porque ajuda a proteger os dados e métodos de uma classe contra alterações indesejadas ou acesso não autorizado. Por exemplo, um desenvolvedor pode usar o encapsulamento para impedir que um usuário altere diretamente o valor de um atributo privado de uma classe, garantindo assim a integridade dos dados. O encapsulamento também pode tornar o código mais modular e mais fácil de entender, uma vez que cada objeto pode ser considerado como uma unidade autônoma e independente. Ele permite que os desenvolvedores protejam os dados e métodos de uma classe contra alterações indesejadas ou acesso não autorizado e tornem o código mais modular e fácil de entender.

Isto permite que, segundo Shelly et al. (2013), os dados sejam acessados somente através dos métodos fornecidos pelo objeto, o que proporciona uma camada de abstração e ajuda a evitar modificações não intencionais nos dados. Para Horstmann (2013) a herança é o mecanismo que permite que uma classe (um plano para um objeto) herde as propriedades e métodos de outra classe). Esta prática permite criar novas classes baseadas em classes existentes. As novas classes, chamadas de subclasses ou classes derivadas, herdam propriedades e métodos da classe original, chamada de superclasse ou classe base. Isto permite aos desenvolvedores reutilizar código e evitar duplicação de esforços.

Budd (2019) afirma que o polimorfismo permite que objetos, de diferentes classes, sejam tratados como objetos de uma superclasse comum e possibilita utilizar única interface para representar múltiplos tipos de objetos. Isto permite, segundo Horstmann (2013) aos desenvolvedores escreverem códigos que podem trabalhar com múltiplos tipos de objetos sem precisar conhecer os

detalhes de cada tipo, simplificando assim, a base de código e reduzindo a necessidade de múltiplas implementações.

Abstração é o processo de simplificação de sistemas complexos segundo Saini e Kaur (2021), dividindo-os em componentes menores e mais gerenciáveis. Este princípio permite que os desenvolvedores foquem nas características essenciais de um objeto, ocultando detalhes desnecessários e promovendo a modularidade.

Para Shelly et al. (2013) há várias vantagens no uso de programação orientada a objetos. Uma das maiores vantagens é que ela permite aos desenvolvedores criar código modular e reutilizável. Os objetos podem ser criados uma vez e depois reutilizados em várias partes de uma aplicação, o que pode economizar tempo e reduzir a probabilidade de erros.

A programação orientada a objetos também permite uma manutenção mais eficiente do código como salienta Horstmann (2013). Como o código está organizado em objetos, é mais fácil modificar e atualizar objetos individuais sem afetar o resto da base de código. Além disso, conforme Budd (2019), o OOP facilita o desenvolvimento de sistemas de software complexos e de grande escala, fornecendo uma estrutura organizacional clara.

Isto pode ajudar a reduzir o risco de introduzir bugs ou erros ao fazer alterações no código. Outra vantagem da programação orientada a objetos é que ela pode ajudar a melhorar a qualidade do código. Shelly et al. (2013) afirma que os objetos encapsulam dados e métodos, o que facilita a escrita de códigos mais seguros e menos propensos a erros. Além disso, como o OOP enfatiza o uso de herança e polimorfismo, ele pode ajudar a reduzir a duplicação de código e melhorar a legibilidade do código.

O OOP segundo Beck (2021) também influenciou a evolução das práticas e metodologias de desenvolvimento de software. O desenvolvimento ágil, por exemplo, enfatiza a importância da modularidade, adaptabilidade e refinamento iterativo, que se alinham com os princípios do OOP. Além disso, o desenvolvimento orientado a testes (TDD), uma prática amplamente adotada, é bem adequado ao OOP devido a seu foco em código modular e testável.

Apesar de suas muitas vantagens, a programação orientada a objetos (OOP) não está isenta de críticas. Uma crítica comum à OOP é que ela pode ser difícil de aprender e dominar, principalmente para desenvolvedores que estão

acostumados a paradigmas procedimentais ou funcionais de programação segundo Armstrong (2006) e Bibeau (2015).

Para Felleisen et al. (2008) a programação orientada a objetos exige que os desenvolvedores pensem em termos de objetos e classes, o que pode ser uma mudança mental desafiadora para aqueles que não têm experiência prévia com este paradigma. Martin (2018) ressalta que a abordagem baseada em objetos pode parecer mais abstrata e menos intuitiva do que abordagens procedimentais, onde a lógica do programa é mais linear e fácil de seguir.

Além disso, a programação orientada a objetos pode levar a uma maior complexidade no design e na estrutura do código segundo Bibeau (2015). O uso extensivo de herança, polimorfismo e encapsulamento pode tornar o código-fonte mais difícil de entender e manter. Além disso, a sobrecarga de funções e o uso de padrões de projeto podem aumentar a complexidade do código e a curva de aprendizado para novos desenvolvedores.

Apesar dessas críticas, a programação orientada a objetos continua sendo uma abordagem popular e eficaz para a construção de sistemas de software. Wasserman (2013) aponta que os desenvolvedores que dominam os conceitos fundamentais da OOP podem criar soluções mais modularizadas, reutilizáveis e extensíveis, o que pode levar a uma manutenção mais fácil e a uma maior flexibilidade na evolução do sistema ao longo do tempo.

A programação orientada a objetos não é o único paradigma de programação disponível para os desenvolvedores, mesmo tendo em conta sua popularidade, existem vários paradigmas alternativos que oferecem diferentes abordagens para organizar o código e resolver problemas de programação.

A programação funcional é uma alternativa ao OOP que ganhou popularidade nos últimos anos. A programação funcional enfatiza o uso de funções como os principais blocos de construção do código, ao invés de objetos ou classes. Desenvolvimentos recentes na engenharia de software têm visto uma mudança em direção a novos paradigmas, como programação funcional e design orientado por domínio. Entretanto, o OOP continua sendo uma abordagem predominante e essencial, muitas vezes complementando estas técnicas emergentes. Por exemplo, o OOP é usado em conjunto com programação funcional em linguagens como Scala e Kotlin, combinando os pontos fortes de ambos os paradigmas conforme Odersky et al. (2020).

Além disso, o desenvolvimento orientado a domínio (DDD), uma metodologia para projetar sistemas de software complexos, depende muito dos princípios do OOP, como encapsulamento e abstração, para criar código bem estruturado e de fácil manutenção.

Várias linguagens modernas de programação, segundo Saini e Kaur, (2021), tais como Python, Java e C#, utilizam princípios OOP para facilitar a criação de software robusto e de fácil manutenção.

OOP pode desempenhar um papel crucial no projeto e implementação de sistemas de inteligência artificial (IA). Para Fowler (2022), ao aproveitar os princípios de abstração, modularidade e encapsulamento, os desenvolvedores podem criar modelos e algoritmos complexos de IA que são mantidos, escaláveis e reutilizáveis. Além disso, o OOP pode facilitar o desenvolvimento de bibliotecas e estruturas de aprendizagem de máquinas, tais como TensorFlow e PyTorch, que servem como ferramentas essenciais para pesquisadores e praticantes de IA

A programação orientada a objetos é um paradigma de programação amplamente utilizado que tem sido popularizado por linguagens como Java e C++. O OOP enfatiza o uso de objetos e classes como os principais blocos de construção de código, oferecendo diversas vantagens sobre outros paradigmas de programação, incluindo a modularidade do código, manutenção eficiente e melhor qualidade do código. OOP tem sido um pilar fundamental do desenvolvimento de software por décadas, oferecendo inúmeros benefícios, tais como modularidade, capacidade de reutilização e manutenção. Apesar do surgimento de novos paradigmas, o OOP continua sendo uma abordagem essencial na moderna engenharia de software, muitas vezes complementando e aprimorando estas novas técnicas. Como o desenvolvimento de software continua a evoluir, o OOP provavelmente continuará a ser um paradigma predominante e indispensável, fornecendo uma base sólida para a construção de sistemas de software complexos e adaptáveis.

4 GAMIFICAÇÃO

Um dos primeiros trabalhos realizados na adaptação das práticas de jogo dentro do local de trabalho pode ser datado em 1984, quando Charles Coonradt explorou o valor de adicionar elementos de jogo no trabalho através de seu livro *O Jogo do Trabalho* (CHOU, 2015).

O termo “gamificação” do inglês *gamification* foi cunhado pela primeira vez, segundo Deterding et al., (2011) por Nick Pelling, um programador de computador britânico, em 2003. Nick descreveu a gamificação como um processo de mudança de interface para tornar as transações eletrônicas mais divertidas e parecidas com o jogo, mas mais tarde foi ampliada com a ênfase no uso de elementos de jogo.

Deterding et al., (2011) afirmam que, em linhas gerais, “gamificação é o uso do design de jogos em contextos de não-jogo”. O conceito tem raízes em pesquisas e práticas anteriores que visavam alavancar as qualidades motivacionais e envolventes dos jogos para fins não ligados ao jogo. O surgimento da gamificação conforme Hamari et al., (2014) pode ser atribuído a vários fatores, tais como os avanços na tecnologia, a ubiquidade dos dispositivos digitais e a crescente popularidade dos videogames.

Alves e Teixeira (2014, p. 131) apresentam outra perspectiva ao definir gamificação como “a aplicação da lógica e da mecânica dos games em diferentes aspectos do cotidiano, caracterizando ambientes que contenham elementos de jogos como gamificados”. Eles argumentam que uma sala de aula pode se tornar gamificada através da ludicidade e dinamicidade possibilitadas pelos jogos, promovendo o aprendizado autônomo e divertido. Contudo, os autores ressaltam que a percepção de diversão é subjetiva e depende da contribuição dos participantes nas atividades propostas.

A gamificação é uma abordagem que incorpora elementos de jogo em contextos não relacionados a jogos, com o objetivo de aumentar a motivação, o engajamento e a retenção dos usuários. Para entender melhor como a gamificação funciona e por que é eficaz, é importante analisar algumas das teorias e estruturas teóricas que a fundamentam, conforme proposto por Seaborn e Fels (2015), Sailer et al. (2017) e outros estudiosos.

Comportamentalismo: Essa teoria, fundamentada nos estudos de Skinner e atualizada por Lopes e Bittencourt (2019), baseia-se na ideia de que o comportamento humano é moldado por reforços e punições externos. Na

gamificação, os reforços podem incluir recompensas, como pontos, medalhas ou conquistas, que incentivam os usuários a completar tarefas ou alcançar objetivos (ROBSON et al., 2015).

Psicologia cognitiva: Essa abordagem, enraizada nos trabalhos de Piaget e Vygotsky e expandida por Kiili et al. (2018), enfoca a importância dos processos mentais, como a atenção, a memória e a tomada de decisão, na compreensão do comportamento humano. A gamificação pode ajudar a melhorar esses processos ao proporcionar feedback e desafios adequados, promovendo a aprendizagem e o desenvolvimento de habilidades (LANDERS et al., 2018).

Teoria da autodeterminação: Essa teoria, de Ryan e Deci (2017), sugere que a motivação intrínseca - o desejo de realizar uma atividade pelo prazer e satisfação que ela proporciona - é essencial para o engajamento e a persistência dos indivíduos. A gamificação pode aumentar a motivação intrínseca ao proporcionar aos usuários um senso de autonomia, competência e conexão social.

Para compreender melhor como os diferentes elementos de jogo se combinam para criar experiências envolventes, Hunicke et al. (2021) propuseram a estrutura Mecânica, Dinâmica, Estética (MDA). A Mecânica refere-se às regras e sistemas do jogo, a Dinâmica descreve o comportamento emergente dos jogadores à medida que interagem com a mecânica, e a Estética diz respeito à experiência emocional e intelectual dos jogadores. Essa estrutura ajuda a identificar os componentes-chave da gamificação e como eles se relacionam entre si para criar uma experiência eficaz e atraente.

A mecânica é o conjunto de regras e princípios fundamentais que governam o funcionamento de um jogo. Isso inclui as regras, objetivos, interações entre os elementos do jogo e os sistemas que os sustentam. A mecânica é responsável por criar a estrutura do jogo e estabelecer as condições para o engajamento do jogador. Aqui estão alguns exemplos:

- Pontuação: A atribuição de pontos aos jogadores com base em seu desempenho ou realizações no jogo.
- Níveis: Divisão do jogo em diferentes estágios, cada um com desafios e objetivos distintos.
- Conquistas: Recompensas simbólicas concedidas aos jogadores por atingir certos marcos ou completar tarefas específicas.

- **Cooperação e competição:** Oportunidades para os jogadores trabalharem juntos ou competirem uns contra os outros no jogo.

A dinâmica refere-se à forma como os jogadores interagem com a mecânica do jogo e como essa interação se traduz em experiências e comportamentos específicos. Isso pode incluir a forma como os jogadores recebem feedback (positivo ou negativo), os desafios que enfrentam e as mudanças e adaptações que ocorrem ao longo do tempo. A dinâmica é responsável por criar uma experiência de jogo envolvente e variada, que mantém os jogadores interessados e motivados. Exemplos:

- **Progressão:** Como os jogadores avançam no jogo, enfrentando desafios crescentes e desbloqueando novas habilidades ou conteúdo.
- **Estratégia:** A tomada de decisão do jogador e a escolha de abordagens específicas para resolver problemas ou superar obstáculos.
- **Adaptação:** A capacidade dos jogadores de aprender e se adaptar às mudanças no ambiente do jogo ou às ações de outros jogadores.
- **Engajamento social:** Interações entre jogadores, como comunicação, colaboração ou rivalidade, que enriquecem a experiência de jogo.

A estética é a experiência emocional e sensorial que os jogadores têm ao interagir com o jogo. Isso pode incluir as sensações, sentimentos e significados que o jogo evoca, bem como a forma como o jogo se comunica visual, auditiva e tátilmente com os jogadores. A estética é responsável por criar uma atmosfera e um ambiente de jogo que atraem e envolvem os jogadores em um nível mais profundo e emocional. Alguns dos exemplos são:

- **Narrativa:** A história ou enredo subjacente que dá contexto e significado à experiência de jogo.
- **Arte e design:** Os elementos visuais e gráficos do jogo, como personagens, cenários e animações, que criam uma atmosfera e estilo específicos.
- **Música e som:** A trilha sonora e os efeitos sonoros que acompanham a ação do jogo e ajudam a criar uma experiência imersiva.

- Atmosfera e imersão: A sensação de estar totalmente envolvido e “presente” no mundo do jogo, proporcionando uma experiência emocionalmente envolvente e cativante.

Já Chou (2015) desenvolveu outra estrutura para explicar a eficácia da gamificação, chamada Octalysis. Esta estrutura destaca oito elementos centrais da gamificação, incluindo significado, realização, empoderamento, posse, influência social, escassez, imprevisibilidade e evitação. O Octalysis ajuda a entender como cada um desses elementos pode ser usado para motivar e engajar os usuários em diferentes contextos e situações.

Chou (2015) explora o conceito de gamificação e como ele pode ser usado para motivar pessoas em vários contextos. Ele identifica oito motivadores principais que impulsionam o comportamento humano, incluindo autonomia, maestria, propósito, influência social, escassez, imprevisibilidade, evitação e propriedade. E sugere que, ao aproveitar esses motivadores, as organizações podem projetar experiências envolventes e motivadoras para seus usuários, maximizando o envolvimento e a retenção. Cada um destes conceitos, conhecido como Octalysis Framework, foca em um motivador diferente para impulsionar o comportamento humano em experiências gamificadas:

A autonomia refere-se à sensação de liberdade e controle que um indivíduo possui ao tomar suas próprias decisões e escolhas. Na gamificação, isso pode ser representado por oferecer várias opções e permitir que os usuários escolham seus próprios caminhos, desafios e recompensas.

Já a maestria envolve o desejo de se aprimorar e dominar habilidades específicas. Na gamificação, isso pode ser traduzido em níveis de dificuldade progressivos, feedback constante e a oportunidade de aprender e crescer ao longo do tempo.

Por sua vez, o propósito trata da motivação intrínseca de realizar algo maior que si mesmo, contribuindo para um objetivo comum ou causa. Em experiências gamificadas, isso pode ser incorporado ao fornecer um enredo envolvente, missões significativas ou recompensas que beneficiam a comunidade.

A Influência Social, por sua vez, relaciona-se à interação e cooperação com outros indivíduos. A gamificação pode incluir elementos como competição, colaboração, compartilhamento de conquistas e a criação de um ambiente socialmente gratificante.

A escassez é a percepção de valor baseada na raridade de um item ou experiência. Na gamificação, isso pode ser alcançado através da implementação de recompensas exclusivas, acesso a áreas restritas ou oportunidades de tempo limitado.

A imprevisibilidade pode ser resumida como o interesse humano em situações incertas ou mistérios. Na gamificação, isso pode ser incorporado por meio de eventos aleatórios, recompensas surpresa e enigmas que desafiam o jogador a descobrir o desconhecido.

A evitação, em seu turno, é o impulso para evitar consequências negativas, como perda de status ou punições. Na gamificação, isso pode ser aplicado através de sistemas de penalidades, prazos ou a possibilidade de perder conquistas.

A propriedade, como o nome indica, é o desejo de possuir, personalizar e controlar recursos. Na gamificação, isso pode ser representado por meio de colecionáveis, personagens personalizáveis ou a oportunidade de construir e desenvolver espaços virtuais.

Conforme Ryan e Deci (2000) ao incorporar elementos de jogo como competição, colaboração e feedback, a gamificação pode fomentar a motivação intrínseca, que é crucial para o engajamento e a mudança de comportamento a longo prazo.

A competição introduz a rivalidade entre os jogadores ou equipes, incentivando-os a se esforçarem para superar os outros e atingir um determinado objetivo. A competição pode aumentar a motivação intrínseca ao fornecer um senso de desafio e progresso, incentivando os usuários a se esforçarem para melhorar suas habilidades e conhecimentos. Na gamificação, a competição pode ser incorporada por meio de sistemas de ranking, tabelas de classificação e desafios temporizados, onde os participantes são recompensados com pontos, insígnias ou outras conquistas.

Já a colaboração refere-se ao trabalho em equipe e cooperação entre os participantes para atingir um objetivo comum. Através da colaboração, os usuários podem desenvolver habilidades interpessoais, compartilhar conhecimentos e recursos e sentir-se parte de uma comunidade. A gamificação pode incentivar a colaboração por meio de missões ou atividades em grupo, onde os usuários precisam cooperar e comunicar-se efetivamente para ter

sucesso. Isso pode incluir jogos cooperativos, resolução de problemas em grupo e outras atividades que exigem a contribuição de todos os membros da equipe.

A retroalimentação é uma parte fundamental do processo de aprendizagem, pois fornece aos usuários informações sobre seu desempenho e progresso. Na gamificação, o *feedback* pode ser oferecido de várias formas, como pontuação, conquistas, insígnias, comentários ou visualizações de progresso. Um bom sistema de feedback deve ser oportuno, específico e claro, para que os usuários compreendam o que estão fazendo corretamente e onde podem melhorar. O *feedback* ajuda a aumentar a motivação intrínseca, pois permite que os usuários vejam seu progresso e melhoria ao longo do tempo, incentivando-os a continuar aprendendo e se esforçando.

A gamificação para Mora et al. (2019) tem o potencial de promover a autodeterminação e a motivação intrínseca ao criar um ambiente que estimula o senso de autonomia, competência e relacionamento:

A autonomia é a capacidade de um indivíduo de ter controle sobre suas próprias ações e decisões. Na gamificação, conforme Zichermann e Cunningham (2011) isso pode ser alcançado oferecendo aos usuários várias opções e caminhos a seguir, permitindo que eles determinem suas estratégias, escolhas e metas.

A competência, por sua vez, refere-se à habilidade de um indivíduo para executar tarefas com sucesso e atingir metas. A gamificação pode ajudar a promover um senso de competência ao proporcionar desafios progressivos e a oportunidade de dominar habilidades específicas. Feedbacks constantes para Hamari et al. (2014), metas claras e sistemas de recompensa que reconhecem o progresso do usuário são formas de aumentar a percepção de competência.

O relacionamento envolve a conexão social e interação entre os indivíduos, criando um senso de pertencimento e apoio. A gamificação pode promover o relacionamento ao incluir elementos de interação social, como colaboração, competição e compartilhamento de conquistas.

Ao combinar esses três elementos - autonomia, competência e relacionamento - a gamificação tem o poder de criar experiências mais envolventes e significativas. Para Mora et al. (2019) essas experiências, por sua vez, aumentam a motivação intrínseca e a autodeterminação dos usuários, levando a resultados mais efetivos no aprendizado, engajamento e mudança de comportamento a longo prazo.

Sailer et al. (2023) destacam a eficácia da gamificação em promover resultados positivos no contexto educacional, como maior motivação, engajamento e aprendizagem. Alguns exemplos de intervenções educativas gamificadas são:

Distintivos digitais são recompensas virtuais concedidas aos estudantes por completarem tarefas, alcançarem objetivos específicos ou demonstrarem habilidades e conhecimentos específicos. Esses distintivos funcionam como incentivos para que os alunos se empenhem nas atividades, aumentando a motivação e o engajamento. Além disso, os distintivos podem ser compartilhados nas redes sociais, proporcionando reconhecimento público das conquistas e estimulando o sentimento de competência.

Quadros de liderança ou tabelas de classificação, são rankings que mostram o desempenho dos estudantes em relação aos outros, geralmente com base em pontos ou outros indicadores de progresso. Esses quadros promovem uma competição saudável entre os alunos, incentivando-os a se esforçarem mais para melhorar suas posições no ranking. Quadros de liderança também podem ajudar a aumentar o senso de competência, à medida que os estudantes veem suas posições melhorarem como resultado de seus esforços.

Quests são missões ou atividades projetadas para desafiar os alunos a aplicarem e aprimorarem seus conhecimentos e habilidades de forma prática. As quests podem ser estruturadas em vários níveis de dificuldade, com objetivos claros e feedbacks contínuos, proporcionando uma experiência de aprendizagem atraente e envolvente. Ao completar quests, os estudantes podem ganhar recompensas, como pontos, insígnias ou acesso a conteúdo exclusivo, o que aumenta a motivação e o engajamento.

Esses exemplos de intervenções educativas gamificadas ilustram como a gamificação pode ser aplicada na educação para aumentar a motivação, o engajamento e os resultados de aprendizado dos estudantes. Ao incorporar esses elementos de jogo no processo de ensino, os educadores podem criar experiências de aprendizagem mais envolventes e significativas para os alunos, pode-se melhorar o desempenho acadêmico dos estudantes, a capacidade de resolução de problemas e a retenção do conhecimento.

O campo da aprendizagem baseada em *games* (game-based learning), que envolve o design, o desenvolvimento e o uso de jogos digitais (educacionais,

mas também de entretenimento e comerciais) como apoio ao processo de ensino e aprendizagem vem crescendo intensamente (MATTAR, 2020).

Foi demonstrado que a gamificação melhora os resultados de aprendizagem e o desempenho em ambientes educacionais. Hamari et al. (2014) dizem que a gamificação tem um efeito positivo no aprendizado, motivação e resultados cognitivos em contextos educacionais.

Com a utilização de plataformas de aprendizado adaptativo, a gamificação pode ser integrada a plataformas de aprendizado adaptativo que ajustam a dificuldade e o ritmo das atividades com base no desempenho do aluno. Essas plataformas podem incluir elementos de jogo como pontos, insígnias e tabelas de classificação para motivar os alunos a progredir e aprender de acordo com suas habilidades individuais.

Simulações e jogos educacionais incorporam elementos de jogo podem ser usados para ensinar habilidades específicas ou conceitos complexos de forma envolvente. Por exemplo, um jogo de simulação de negócios pode ensinar aos alunos sobre economia, gestão e estratégia de negócios em um ambiente interativo e competitivo. Da mesma forma, para Sailer et al. (2017) nos ambientes de trabalho, a gamificação foi encontrada para aumentar a produtividade, desempenho e motivação dos funcionários.

Em programas de treinamento gamificados, a gamificação pode ser aplicada a programas de treinamento para melhorar a retenção de informações e o engajamento dos funcionários. Por exemplo, uma empresa pode desenvolver um jogo interativo de treinamento em vendas, em que os funcionários ganham pontos por concluir tarefas e responder a perguntas relacionadas a técnicas de vendas e produtos.

Com desafios e competições no local de trabalho as empresas podem criar desafios e competições gamificados que incentivem os funcionários a trabalhar em equipe e melhorar suas habilidades. Por exemplo, uma competição de inovação em que os funcionários são incentivados a propor e desenvolver ideias inovadoras, com recompensas e reconhecimento para as melhores contribuições.

Burke (2015) enfatiza que a gamificação cria modelos de engajamento, visando a novas comunidades e motivando-as a atingir metas que podem ser desconhecidas para elas. Exemplos de como a gamificação pode ser aplicada para alcançar esses objetivos:

Um aplicativo de condicionamento físico gamificado pode envolver e motivar os usuários a adotarem hábitos saudáveis, como fazer exercícios regulares e ter uma alimentação equilibrada. Por meio de recompensas, desafios e interações sociais, os usuários são incentivados a atingir metas de saúde que talvez não soubessem que eram capazes de alcançar.

A gamificação pode ser usada para promover práticas sustentáveis e conscientização ambiental em comunidades locais. Por exemplo, um aplicativo gamificado que recompensa os usuários por reciclar ou reduzir o consumo de energia pode ajudá-los a atingir metas de sustentabilidade que talvez não tenham considerado antes.

A gamificação pode ser aplicada para aumentar a participação cívica e a colaboração entre os membros de uma comunidade. Um exemplo disso seria um aplicativo ou plataforma que premia os cidadãos por envolverem-se em projetos locais, como limpeza de parques ou planejamento de eventos comunitários. Isso motiva os indivíduos a contribuir para o bem comum e a alcançar metas comunitárias.

Aplicativos de aprendizado de idiomas gamificados podem ajudar a envolver e motivar os usuários a aprenderem novos idiomas e se conectarem a comunidades globais. Esses aplicativos podem apresentar desafios, níveis progressivos e recompensas que incentivam os usuários a expandir continuamente suas habilidades linguísticas e alcançar metas que, de outra forma, poderiam ser desconhecidas para eles. Conforme Hew et al. (2016) indicam, a gamificação tem um impacto positivo no engajamento dos estudantes em ambientes de aprendizagem online, melhorando tanto a participação quanto a retenção.

Ela aumenta a participação dos estudantes ao tornar a aprendizagem online mais atraente e interativa. Ao introduzir elementos de jogo, como desafios, metas, recompensas e competição, a gamificação proporciona uma experiência de aprendizado lúdica e cativante. Isso pode encorajar os estudantes a se envolverem mais ativamente no processo de aprendizagem e a participarem de atividades e discussões online.

Além disso, a gamificação também pode melhorar a retenção dos estudantes em ambientes de aprendizagem online. A introdução de mecânicas de jogo pode tornar a aprendizagem mais memorável, ao criar experiências de aprendizado emocionalmente envolventes. O feedback contínuo e o

reconhecimento do progresso dos estudantes também podem ajudar a reforçar os conceitos aprendidos e melhorar a retenção de informações a longo prazo.

Outro aspecto importante é o senso de pertencimento e conexão social que a gamificação pode promover em ambientes de aprendizagem online. Através de elementos de jogo, como colaboração em equipe, competições e compartilhamento de conquistas, os estudantes podem se conectar e interagir uns com os outros, o que pode aumentar o engajamento e a retenção ao longo do tempo.

Sailer et al. (2023) ressaltam a importância de integrar princípios e práticas de gamificação aos currículos educacionais e ao desenvolvimento de recursos e materiais de treinamento para melhorar a compreensão e a navegação dos usuários em ambientes gamificados. A implementação bem-sucedida da gamificação na educação envolve várias etapas:

Planejamento e projeto: A integração da gamificação aos currículos educacionais requer planejamento cuidadoso e um projeto bem pensado. Os educadores devem identificar os objetivos de aprendizado específicos, bem como as áreas em que a gamificação pode melhorar o envolvimento e a motivação dos alunos. Isso pode incluir a criação de atividades e tarefas que incentivem a competição, a colaboração e a experimentação.

Desenvolvimento de recursos e materiais: Para garantir que os estudantes compreendam e aproveitem ao máximo os ambientes gamificados, é essencial desenvolver recursos e materiais de treinamento adequados. Isso pode incluir tutoriais, guias do usuário e atividades práticas que apresentem os elementos de jogo e as mecânicas de gamificação de forma clara e acessível.

Implementação e avaliação: Depois que a gamificação for incorporada ao currículo e os recursos de treinamento forem desenvolvidos, os educadores devem implementar e monitorar de perto a eficácia dessas abordagens. Isso pode envolver o acompanhamento do engajamento e do desempenho dos estudantes, bem como a realização de avaliações e pesquisas para coletar feedback e identificar áreas de melhoria.

Ajuste e melhoria contínua: Com base no feedback e nas avaliações dos alunos, os educadores devem estar dispostos a ajustar e aprimorar continuamente os elementos gamificados e os recursos de treinamento para garantir que eles continuem a promover o engajamento e a aprendizagem efetiva. Isso pode incluir a adaptação das atividades e tarefas gamificadas para

melhor atender às necessidades dos alunos ou a incorporação de novas tecnologias e abordagens para melhorar a experiência gamificada.

Para Sobreiro (2017), atividades cotidianas podem ser consideradas situações fora do contexto de jogo. Por exemplo, no ambiente de trabalho, mesmo com a presença de regras, interatividade e feedback entre colegas, clientes, superiores, computadores e diversos sistemas, além de resultados mensuráveis e respostas emocionais, ainda faltam dois elementos cruciais para a definição de jogo: “jogadores” e “desafio abstrato”. Nesse contexto, o funcionário não atua como jogador nem está envolvido em um desafio abstrato. Embora possam criar desafios pessoais em certas situações, como “completar todo o relatório em 10 minutos”, por padrão, o funcionário não está em uma situação de jogo no ambiente de trabalho.

O Quadro 2 apresenta as palavras-chave da definição de gamificação, esclarecendo os principais elementos envolvidos no processo. Os elementos de jogos incluem aspectos como regras, estética, narrativa, pontuação, recompensas e interatividade. Esses elementos são aplicados em contextos não-jogo, como atividades escolares, laborais ou domésticas. Ao adicionar esses elementos de jogos, a gamificação visa aumentar o engajamento e a motivação dos indivíduos nas atividades do cotidiano que, por padrão, não são realizadas em um contexto de jogo. Assim, ao tornar tarefas mais atraentes e interessantes, a gamificação contribui para a resolução de problemas e a obtenção de melhores resultados.

Quadro 2 - Palavras-chave da definição de gamificação

ELEMENTO	DESCRIÇÃO
ELEMENTOS DE JOGOS	São os elementos presentes em jogos, digitais ou analógicos, tais como regras, a estética, narrativa, pontuação, recompensas e interatividade, competição e cooperação, desafios, objetivos e metas, ranking e pontuações, entre outros.
CONTEXTO DE NÃO-JOGO	Situações do cotidiano, tais como ambiente escolar, atividades desempenhadas no trabalho, atividades domésticas, entre outras.
AUMENTAR ENGAJAMENTO E MOTIVAÇÃO	Despertar, através da inserção dos elementos de jogos acima, um maior interesse e comprometimento na realização destas tarefas do cotidiano, que, por padrão, não são realizadas num contexto de jogo.
RESOLUÇÃO DE PROBLEMAS	Através de maior motivação, ajudar na resolução de problemas e obtenção de melhores resultados

Fonte: Sobreiro (2017).

Podemos assumir que “gamificação consiste na utilização de elementos de jogos, digitais ou analógicos, em aplicações fora do contexto de jogos, com o propósito fundamental de aumentar o engajamento e motivação dos indivíduos e ajudar na resolução de problemas” (SOBREIRO, 2017, p. 55).

Ao aplicar a gamificação em contextos não-jogo, é possível observar diversos exemplos no mercado que incorporam essa abordagem, como programas de fidelidade que incluem elementos semelhantes a jogos, como pontos, níveis e desafios. Essa estratégia tem sido amplamente utilizada como uma ferramenta de marketing, onde o objetivo é aumentar o envolvimento do cliente com a marca ou produto. Liang et al. (2022) comentam que essas estratégias de marketing game-like têm demonstrado aumentar o envolvimento, a satisfação e as intenções de compra dos clientes.

Estratégias de marketing game-like são técnicas de marketing que incorporam elementos de jogos e mecânicas de gamificação para envolver e motivar os consumidores. Essas estratégias visam melhorar a experiência do cliente, aumentar a retenção, fidelidade à marca e, em última instância, impulsionar vendas e conversões. O uso de estratégias de marketing game-like pode tornar o processo de marketing mais divertido, envolvente e interativo para os clientes.

Algumas das estratégias de marketing game-like mais comuns incluem:

- **Recompensas e incentivos:** Oferecer pontos, prêmios, descontos ou outros incentivos aos clientes por realizar ações específicas, como fazer compras, compartilhar conteúdo nas redes sociais, escrever avaliações de produtos ou se inscrever em newsletters.
- **Desafios e competições:** Criar competições ou desafios que incentivem os clientes a participar e competir com outros usuários para ganhar prêmios ou reconhecimento. Isso pode incluir concursos de fotografia, quizzes, ou jogos de trivia relacionados à marca ou produto.
- **Níveis e progressão:** Estabelecer um sistema de níveis que permita aos clientes progredir através de diferentes estágios à medida que realizam ações específicas ou acumulam pontos. Isso pode ajudar a incentivar a retenção e a lealdade do cliente, à medida que eles

buscam alcançar níveis mais altos e desbloquear novos benefícios ou recompensas.

- Personalização e avatares: Permitir que os clientes criem avatares ou perfis personalizados que representem sua identidade dentro da plataforma de marketing. Isso pode ajudar a aumentar o engajamento do cliente e a sensação de pertencimento à comunidade da marca.
- Integração com redes sociais: Incentivar os clientes a compartilhar suas experiências e conquistas com seus amigos e seguidores nas redes sociais, aumentando a exposição e o alcance da marca.
- Narrativa e *storytelling*: Desenvolver uma narrativa envolvente ou história que envolva os clientes e os conecte emocionalmente à marca ou produto. Isso pode ser feito através de campanhas publicitárias, experiências interativas ou conteúdo de marca.

A gamificação, conforme Cugelman (2023), tem sido aplicada em contextos de saúde e bem-estar para promover comportamentos saudáveis, tais como atividade física, adesão a medicamentos e gerenciamento de estresse. Exemplos de intervenções de saúde gamificada incluem aplicativos móveis que recompensam os usuários por completar desafios de exercício ou alcançar metas pessoais de saúde:

- Headspace: Um aplicativo de meditação e atenção plena que utiliza gamificação para incentivar a prática regular de meditação, oferecendo aos usuários metas, recompensas e lembretes personalizados.
- Fitbit: Um rastreador de atividades e aplicativo que permite aos usuários monitorar sua atividade física, sono e dieta, estabelecer metas pessoais e competir com amigos em desafios semanais para incentivar a prática de exercícios.
- MySugr: Um aplicativo de gerenciamento de diabetes que utiliza gamificação para motivar os usuários a monitorar seus níveis de glicose no sangue e aderir ao tratamento prescrito, oferecendo pontos, feedback e recompensas.
- Nike+ Run Club é um aplicativo de corrida que utiliza a gamificação para motivar os usuários a se exercitarem e alcançarem seus objetivos de condicionamento físico. O aplicativo inclui desafios,

conquistas, competições e programas de treinamento personalizados para manter os usuários engajados e focados em suas metas de saúde.

- **Zombies, Run!:** Um aplicativo de corrida que transforma exercícios em uma experiência de jogo de aventura, onde os usuários devem correr e completar missões para escapar de zumbis e salvar a humanidade.

O Apple Watch é outro exemplo de um dispositivo que utiliza elementos de gamificação para promover comportamentos saudáveis e bem-estar. Com seu aplicativo de saúde e fitness integrado, o Apple Watch permite aos usuários monitorar sua atividade física, batimentos cardíacos e outros dados de saúde.

Além disso, o Apple Watch oferece recursos gamificados, como:

- **Anéis de Atividade:** Os usuários são incentivados a completar três anéis de atividade diariamente, representando metas de movimento (calorias queimadas), exercício (minutos de atividade moderada) e ficar em pé (quantidade de horas em que o usuário se levantou e se moveu por pelo menos um minuto). Ao completar esses anéis, os usuários recebem feedback positivo e conquistam conquistas, o que os motiva a manter um estilo de vida ativo.
- **Competições e desafios:** O Apple Watch permite que os usuários desafiem amigos e familiares para competir em atividades físicas por um período de tempo determinado, com recompensas e medalhas virtuais atribuídas com base no desempenho de cada participante.
- **Conquistas e medalhas:** Os usuários podem ganhar conquistas e medalhas virtuais por atingir metas pessoais de saúde e condicionamento físico ou completar desafios específicos, como correr uma determinada distância ou atingir uma sequência de dias completando todos os anéis de atividade.

Conforme Sardi et al. (2021) pesquisas demonstraram que a gamificação pode melhorar os resultados de saúde e aumentar a aderência a intervenções relacionadas à saúde.

Apesar de seus benefícios potenciais, segundo Niemiec e Ryan (2021), a gamificação também apresenta vários desafios e limitações. Uma preocupação é a ênfase excessiva em recompensas extrínsecas, tais como pontos e

distintivos, que podem minar a motivação intrínseca e levar a um envolvimento superficial. Adicionalmente, Sailer et al. (2023) destacam que intervenções gamificadas mal planejadas podem levar a consequências indesejadas, como o estímulo à competição excessiva, práticas de trapaça ou a exploração das regras do sistema. Conforme Alomari et al. (2022) existe o potencial da gamificação exacerbar as desigualdades existentes. Pesquisas demonstraram que os sistemas gamificados podem favorecer certos usuários demográficos, tais como indivíduos com mais conhecimento tecnológico ou com predisposição para o jogo. Isto levanta preocupações sobre a justiça, acessibilidade e inclusividade no projeto e implementação de sistemas gamificados.

Ferri et al. (2020) salientam que a gamificação apresenta diversas questões éticas, especialmente no que diz respeito à privacidade dos dados, autonomia do usuário e manipulação. Os sistemas gamificados segundo Mittelstadt (2021) podem coletar grandes volumes de informações sobre os usuários, que podem ser utilizadas, por exemplo, para criar perfis detalhados e direcionar os usuários com conteúdo personalizado.

No que diz respeito à privacidade dos dados, afirmam que a coleta e armazenamento de informações pessoais e de comportamento dos usuários em sistemas gamificados levantam preocupações sobre a proteção da privacidade e a possibilidade de uso indevido desses dados por terceiros. Além disso, de acordo com Mittelstadt (2021) os usuários podem não estar cientes da quantidade de informações coletadas ou como elas são usadas e compartilhadas

Em relação à autonomia do usuário, a gamificação pode segundo Ferri et al. (2020) influenciar o comportamento dos usuários, incentivando a realização de tarefas específicas e, por vezes, alterando as motivações intrínsecas dos indivíduos. Isso se levanta, de acordo com Seaborn e Fels (2015), questões sobre a autonomia do usuário e até que ponto as pessoas têm controle sobre suas próprias ações e decisões em sistemas gamificados.

Também há a questão da manipulação: segundo Bogost (2015) e Ferri et al. (2020), os sistemas gamificados podem ser projetados para explorar vulnerabilidades psicológicas dos usuários e mantê-los engajados por meio de técnicas de persuasão. Para Seaborn e Fels (2015), isso pode resultar em comportamentos compulsivos ou viciantes e pode ser visto como uma forma de manipulação, especialmente quando os objetivos do sistema gamificado estão alinhados com os interesses comerciais em vez do bem-estar do usuário.

Os avanços na tecnologia, como a realidade virtual, a realidade aumentada e a inteligência artificial, têm o potencial de expandir as possibilidades de gamificação. Estas tecnologias podem melhorar a natureza imersiva e interativa das experiências gamificadas, tornando-as mais envolventes e eficazes conforme Seaborn e Fels (2015). Os futuros sistemas de gamificação podem incorporar abordagens personalizadas e adaptativas, adaptando a experiência a usuários individuais com base em suas preferências, estilos de aprendizagem e desempenho.

Através da análise dos dados do usuário e do uso de algoritmos inteligentes, os sistemas gamificados podem oferecer conteúdo e desafios personalizados, levando em consideração as habilidades, interesses e metas do usuário. Isso pode resultar em uma maior motivação e engajamento, já que os usuários se sentiriam mais envolvidos com o conteúdo e as atividades que são relevantes para eles.

Além disso, os sistemas gamificados podem ser projetados para se adaptarem continuamente ao desempenho e progresso do usuário, ajustando o nível de dificuldade e complexidade das atividades, proporcionando feedback personalizado e oferecendo suporte e recursos apropriados conforme necessário. Isso permite que a experiência de gamificação se ajuste dinamicamente às necessidades do usuário, garantindo que ele seja continuamente desafiado e engajado.

No que diz respeito à aprendizagem adaptativa, a gamificação pode ser usada em contextos educacionais para criar ambientes de aprendizagem adaptativos, onde o conteúdo e as atividades de aprendizagem são ajustados às habilidades e necessidades específicas dos alunos. Isso pode melhorar a eficácia da aprendizagem, já que os alunos receberiam instruções e desafios personalizados que se alinham aos seus estilos de aprendizagem e níveis de habilidade. Isto levaria a experiências mais efetivas e envolventes de gamificação segundo Landers et al. (2018).

Os sistemas gamificados são ferramentas ou plataformas que aplicam elementos de jogos a contextos não relacionados a jogos, como educação, saúde e negócios, com o objetivo de aumentar o engajamento, a motivação e a aprendizagem dos usuários. Chen et al. (2021) destacam que sistemas gamificados avançados podem empregar dados do usuário, algoritmos de aprendizagem de máquinas e inteligência artificial para adaptar as intervenções

de gamificação às necessidades individuais, preferências e estilos de aprendizagem.

Para entender melhor essa abordagem, vamos dividir o processo em três partes principais:

Dados do usuário: Os sistemas gamificados coletam informações sobre os usuários, como seu histórico de interação com a plataforma, desempenho, preferências e comportamentos. Esses dados são essenciais para personalizar a experiência de gamificação de acordo com as características específicas de cada usuário.

Algoritmos de aprendizagem de máquinas: A aprendizagem de máquinas é uma subárea da inteligência artificial que permite que os computadores aprendam e melhorem a partir de dados e experiências sem serem explicitamente programados. Os sistemas gamificados podem utilizar algoritmos de aprendizagem de máquinas para analisar os dados dos usuários e identificar padrões, tendências e insights que possam ser usados para personalizar a experiência de gamificação.

Inteligência artificial: A inteligência artificial (IA) é uma área da ciência da computação que busca desenvolver sistemas capazes de realizar tarefas que normalmente exigiriam inteligência humana, como aprendizado, raciocínio, percepção e compreensão da linguagem natural. No contexto da gamificação, a IA pode ser usada para criar experiências mais adaptativas e personalizadas, ajustando as intervenções de gamificação em tempo real com base nas necessidades e preferências dos usuários.

A gamificação, quando combinada com tecnologias emergentes como realidade virtual (VR), realidade aumentada (AR) e Internet das Coisas (IoT), tem o potencial de transformar a maneira como interagimos e aprendemos em diferentes contextos. Conforme apontado por Morschheuser et al. (2022), essas tecnologias inovadoras podem ser usadas para criar experiências gamificadas imersivas, interativas e adaptáveis, oferecendo oportunidades promissoras para melhorar o engajamento e a eficácia das intervenções.

A Realidade Virtual (VR) permite a criação de ambientes simulados em 3D que imergem os usuários em experiências realistas e envolventes. Na educação e treinamento, por exemplo, a VR pode ser usada para simular situações complexas, como cirurgias médicas, reparos de equipamentos ou

cenários de emergência, proporcionando aos alunos uma oportunidade de praticar habilidades e tomar decisões em um ambiente seguro e controlado.

A Realidade Aumentada (AR) integra elementos digitais ao ambiente físico, enriquecendo a percepção e interação dos usuários com o mundo real. Na educação, a AR pode ser usada para melhorar a compreensão de conceitos complexos, como anatomia humana ou processos químicos, permitindo que os alunos visualizem e interajam com modelos 3D superpostos ao ambiente real. Na saúde, a AR pode ajudar os profissionais a monitorar e visualizar informações em tempo real, como sinais vitais e orientações de tratamento, enquanto realizam procedimentos médicos.

A Internet das Coisas (IoT) refere-se à conexão de dispositivos e sensores a uma rede, permitindo a comunicação e troca de dados entre eles. Em contextos gamificados, a IoT pode ser usada para fornecer feedback e suporte em tempo real para iniciativas de mudança de comportamento, como monitoramento de atividades físicas, uso de energia ou gerenciamento de saúde. Por exemplo, dispositivos como *smartwatches* e rastreadores de atividades, podem coletar e transmitir dados de desempenho e saúde, ajudando os usuários a monitorar seu progresso e ajustar suas metas de acordo.

A gamificação tem sido cada vez mais aplicada em diversos contextos, como educação, saúde, negócios e meio ambiente. No entanto, um desafio crítico enfrentado por esse campo, conforme apontado por Koivisto et al. (2021), é a necessidade de aproximar a pesquisa acadêmica e a aplicação prática das técnicas de gamificação. Para superar esse desafio e garantir que as intervenções gamificadas sejam eficazes e bem-sucedidas, é fundamental que pesquisadores e profissionais colaborem estreitamente.

Onde a gamificação pode funcionar:

A gamificação tem sido eficaz na melhoria do engajamento e aprendizado dos alunos, incentivando a participação ativa, a colaboração e a experimentação. Exemplos de sucesso incluem sistemas de aprendizado adaptativo, jogos educacionais e plataformas de aprendizado baseadas em projetos.

Intervenções gamificadas têm demonstrado ser úteis no estímulo a comportamentos saudáveis, como atividade física, adesão a medicamentos e gerenciamento do estresse. Aplicativos de saúde que recompensam os usuários por alcançar metas pessoais são exemplos de iniciativas bem-sucedidas nesse contexto.

A gamificação tem sido aplicada com sucesso em áreas como treinamento de funcionários, desenvolvimento de habilidades e engajamento do cliente. Programas de fidelidade gamificados e treinamentos baseados em jogos são exemplos de abordagens que têm gerado resultados positivos.

Onde a gamificação pode não funcionar:

Intervenções de gamificação mal planejadas, que não levam em consideração as teorias e os princípios pedagógicos, podem não ser efetivas ou até prejudiciais. Para evitar isso, é crucial que os projetos de gamificação sejam baseados em pesquisas e evidências sólidas.

A gamificação pode não funcionar se a implementação for inadequada ou se os elementos do jogo não estiverem bem integrados ao contexto. É fundamental que os profissionais sejam capacitados e sigam as melhores práticas para garantir a eficácia das intervenções.

A gamificação pode ter efeitos indesejados, como aumento da competição, trapaça ou manipulação do sistema. Para minimizar esses riscos, é necessário monitorar e avaliar continuamente o impacto das intervenções e ajustá-las conforme necessário.

À medida que a gamificação se torna mais difundida em vários domínios, Fogg (2021) aponta que é essencial promover práticas de projeto éticas e sustentáveis que priorizem o bem-estar, a autonomia e a privacidade do usuário. Isto pode envolver o desenvolvimento de estruturas e diretrizes éticas para a gamificação, bem como a adoção de abordagens de design centradas no usuário e participativas que envolvam os usuários no processo de design.

Embora a gamificação já tenha tido um impacto significativo em campos como educação, marketing e saúde, há potencial para uma expansão adicional em vários outros domínios. Por exemplo, a gamificação pode ser aplicada à conservação ambiental, ao envolvimento cívico e à produtividade no local de trabalho de acordo com Dichev e Dicheva (2017). Ao identificar e abordar os desafios e oportunidades únicos associados a estes domínios, os pesquisadores e profissionais da gamificação podem desenvolver intervenções sob medida que promovam resultados positivos em diversos contextos.

O desenvolvimento futuro da gamificação depende fortemente da colaboração e da construção de comunidades entre pesquisadores, profissionais e usuários. Estabelecer uma comunicação eficiente e compartilhar conhecimentos, recursos e melhores práticas são fundamentais para avançar no

campo e garantir o sucesso contínuo das intervenções gamificadas. De acordo com Hamari et al. (2014) e Deterding et al. (2011), isso pode envolver várias abordagens, como:

Plataformas online e fóruns: A criação de plataformas online e fóruns dedicados à gamificação permite a troca de ideias, recursos e experiências entre diferentes partes interessadas. Por exemplo, o *Gamification Research Network* (GRN) é uma plataforma que reúne acadêmicos e profissionais interessados em pesquisar a gamificação.

Grupos de mídia social: Grupos e comunidades em plataformas de mídia social como LinkedIn, Facebook e Reddit ajudam a conectar pessoas interessadas em gamificação, promovendo discussões e compartilhamento de informações. A GamFed, uma federação internacional de profissionais de gamificação, mantém uma presença ativa no LinkedIn, compartilhando notícias e eventos relacionados ao campo.

Encontros regulares e oficinas: A organização de encontros locais, oficinas e eventos de networking em diferentes regiões pode ajudar a reunir pesquisadores, profissionais e entusiastas da gamificação. A *Gamification Europe* é um exemplo de conferência que promove workshops e apresentações de projetos relacionados à gamificação.

Conferências e congressos: A realização de conferências e congressos internacionais sobre gamificação possibilita a reunião de especialistas de diferentes disciplinas e setores para compartilhar conhecimentos e experiências. A CHI PLAY é uma conferência internacional sobre jogos e interação homem-computador que apresenta pesquisas e aplicações de gamificação.

Como a gamificação continua a amadurecer como um campo, há uma necessidade crescente para o desenvolvimento de padrões e diretrizes que informem o projeto, implementação e avaliação de sistemas gamificados. Essas normas e diretrizes podem ajudar a garantir que as intervenções de gamificação sejam baseadas em princípios sólidos, práticas baseadas em evidências e considerações éticas. Koivisto et al. (2021) cita que eles também podem promover consistência e qualidade em diferentes aplicações de gamificação, facilitando a comparação e avaliação da eficácia de várias intervenções por parte de usuários, profissionais e legisladores.

É fundamental promover práticas responsáveis de gamificação que coloquem em primeiro plano o bem-estar, a segurança e a privacidade do

usuário. Segundo Fogg (2021), isso pode ser alcançado ao incluir funcionalidades que incentivem padrões de uso equilibrados e saudáveis. Por exemplo, a implementação de limites de tempo pode ajudar os usuários a gerenciar o tempo que passam interagindo com aplicações gamificadas, evitando o uso excessivo e a dependência.

Além disso, é importante fornecer lembretes regulares para que os usuários façam pausas e se afastem das atividades gamificadas por um tempo. Isso pode ajudar a evitar problemas de saúde relacionados ao uso prolongado de dispositivos eletrônicos, como a fadiga visual e as lesões por esforços repetitivos.

As ferramentas de automonitoramento e autorregulação também podem ser incorporadas aos sistemas gamificados para permitir que os usuários acompanhem seu próprio progresso e ajustem seu comportamento de acordo. Essas ferramentas podem incluir gráficos de desempenho, metas ajustáveis e recursos de *feedback* personalizado.

À medida que a gamificação se torna mais difundida, é crucial garantir que as intervenções gamificadas sejam acessíveis e inclusivas para usuários com diversas habilidades, origens e necessidades. Para Connell et al. (2021) isto pode envolver a adoção de princípios de design universal, que enfatizam a flexibilidade, simplicidade e adaptabilidade no design de produtos e ambientes.

Design universal é um conceito que visa criar produtos, ambientes e sistemas que possam ser usados por todas as pessoas, independentemente de suas habilidades físicas, cognitivas ou sensoriais. Ele considera a diversidade humana em todas as suas formas, desde diferenças de altura e peso até habilidades motoras, visuais, auditivas e cognitivas. Busca atender às necessidades de todos os usuários, independentemente de suas habilidades, idade ou cultura.

De acordo com Steinfeld e Maisel (2012) o conceito de design universal surgiu por meio do movimento pelos direitos dos deficientes. O seu objetivo é garantir que os produtos e ambientes possam ser utilizados por todas as pessoas com o máximo de autonomia possível, sem a necessidade de adaptações ou modificações adicionais. Alguns exemplos de elementos do design universal incluem: rampas de acesso para cadeirantes, fontes de texto grandes e em negrito para pessoas com deficiência visual, botões com alto contraste para pessoas com deficiência cognitiva, entre outros.

As primeiras experiências com o conceito levaram os líderes da área a ir além dessas origens e a identificar conexões com o design para o envelhecimento, a sustentabilidade social e o design centrado no usuário. Ao mesmo tempo, outros defensores do design para melhorar a vida das pessoas estão reconhecendo o valor do paradigma do design universal (STEINFELD e MAISEL, 2012).

Ao incorporar características tais como interfaces personalizáveis, métodos de entrada alternativos e níveis de dificuldade ajustáveis, os projetistas de gamificação podem criar experiências mais inclusivas e equitativas que atendam a uma ampla gama de usuários.

As interfaces personalizáveis permitem que os usuários ajustem a aparência, o layout e as funcionalidades das plataformas gamificadas de acordo com suas preferências e necessidades. Isso inclui a capacidade de modificar cores, fontes, tamanhos de texto e elementos visuais, bem como adicionar ou remover funcionalidades específicas. Essa personalização possibilita que os usuários criem experiências mais confortáveis e envolventes, tornando a gamificação mais acessível a pessoas com diferentes habilidades e preferências.

Os métodos de entrada alternativos, por sua vez, referem-se a diferentes maneiras pelas quais os usuários podem interagir com uma plataforma gamificada, além dos métodos tradicionais, como teclado e mouse. Isso inclui dispositivos de entrada adaptativos, como joysticks, touchscreens, controles por voz e tecnologias de rastreamento ocular. Ao oferecer várias opções de interação, os projetistas de gamificação podem acomodar as necessidades de usuários com diferentes habilidades motoras e cognitivas, garantindo que as experiências gamificadas sejam acessíveis a um público mais amplo.

Já os níveis de dificuldade ajustáveis permitem que os usuários modifiquem a complexidade e o desafio das tarefas ou atividades dentro de uma plataforma gamificada. Isso pode incluir a alteração da velocidade do jogo, a quantidade de informações fornecidas ou a complexidade das metas e objetivos. Ao oferecer níveis de dificuldade ajustáveis, os projetistas de gamificação podem garantir que as experiências sejam desafiadoras e envolventes para usuários com diferentes habilidades e conhecimentos prévios, promovendo uma experiência mais inclusiva e equitativa para todos os participantes.

Dichev e Dicheva (2017), dizem que se pode envolver a realização de pesquisas que examinem as implicações da gamificação para os resultados

sociais, econômicos e ambientais, bem como a exploração do potencial da gamificação para contribuir para os desafios globais, tais como redução da pobreza, mudança climática e desenvolvimento sustentável.

A gamificação pode influenciar os resultados sociais ao incentivar a colaboração, a comunicação e a construção de relacionamentos entre os usuários. Por exemplo, aplicações gamificadas em educação podem promover a aprendizagem cooperativa e a troca de conhecimento entre estudantes, melhorando o engajamento e o desempenho acadêmico. Além disso, a gamificação em ambientes de trabalho pode facilitar o trabalho em equipe e a coesão, levando a uma maior satisfação e produtividade dos funcionários.

No que diz respeito aos resultados econômicos, a gamificação pode gerar valor e oportunidades de negócios em várias indústrias. Por exemplo, a gamificação do marketing pode aumentar a fidelidade e a retenção do cliente, enquanto a gamificação em treinamento corporativo pode melhorar a eficiência e a eficácia dos programas de desenvolvimento de funcionários. Também ressaltam o potencial da gamificação para impulsionar a inovação e a competitividade, incentivando a colaboração e a geração de ideias entre os participantes.

Ela pode ser usada para promover a conscientização e a mudança de comportamento em relação aos problemas ambientais. As intervenções gamificadas podem ser utilizadas para incentivar ações sustentáveis, como o uso eficiente de recursos, a redução do consumo de energia e a promoção de práticas de reciclagem. Por exemplo, aplicativos gamificados que recompensam os usuários por adotarem hábitos ecologicamente corretos podem ajudar a criar uma maior consciência ambiental e a impulsionar mudanças positivas.

A gamificação também tem o potencial de abordar questões globais, como a redução da pobreza, as mudanças climáticas e o desenvolvimento sustentável. Os sistemas gamificados podem ser projetados para envolver e mobilizar os cidadãos na solução de problemas complexos, facilitando a coleta de dados, a tomada de decisões e a implementação de ações efetivas. Exemplos incluem projetos gamificados que incentivam a educação financeira e a geração de renda em comunidades carentes ou iniciativas que promovam ações de conservação e adaptação às mudanças climáticas.

Ao enfrentar os desafios e limitações relacionados à gamificação, aprimorando a acessibilidade e a inclusão e incentivando a conscientização e a

educação digital, o campo da gamificação pode potencializar seus efeitos positivos e minimizar riscos potenciais. Além disso, é fundamental avaliar o impacto social da gamificação, promover a ciência aberta e a colaboração entre diferentes áreas e desenvolver padrões e diretrizes para assegurar o sucesso contínuo e a sustentabilidade da gamificação na era digital.

Uma abordagem mais detalhada pode incluir:

- Trabalhar para criar experiências gamificadas que considerem as necessidades de diversos grupos de usuários, incluindo pessoas com deficiências físicas, cognitivas ou sensoriais, e adaptando o design e a interação para garantir a acessibilidade.
- Promover a conscientização sobre o uso responsável e ético da gamificação e desenvolver programas educacionais que capacitem os indivíduos a interagir de forma eficiente e segura com sistemas gamificados.
- Conduzir estudos e pesquisas que analisem o impacto da gamificação em questões sociais, econômicas e ambientais e investigar como a gamificação pode contribuir para a solução de desafios globais, como redução da pobreza, mudança climática e desenvolvimento sustentável.
- Incentivar a troca de conhecimentos, recursos e melhores práticas entre pesquisadores, profissionais e usuários de gamificação, por meio de plataformas online, fóruns, grupos de mídia social e eventos presenciais.
- Estabelecer diretrizes e padrões para o design e a implementação de intervenções gamificadas, que garantam a qualidade, a ética e a eficácia das soluções gamificadas, bem como a proteção da privacidade e autonomia dos usuários.

À medida que a gamificação segue transformando a maneira como aprendemos, trabalhamos e nos conectamos com o mundo digital, é fundamental manter-se atento e adaptar-se às necessidades e expectativas em constante mudança dos usuários e demais envolvidos conforme Deterding et al., (2011). Para isso, é necessário cultivar uma cultura de inovação, responsabilidade e inclusão, que traga benefícios para todos. Isso pode ser alcançado por meio de várias estratégias, incluindo:

Pesquisa e desenvolvimento contínuos: Hamari et al. (2014) apontam que investir em pesquisa e desenvolvimento para identificar as melhores práticas e inovações emergentes em gamificação, garantindo que as soluções sejam baseadas em evidências e alinhadas com as necessidades dos usuários.

Estabelecer canais de comunicação abertos e eficazes entre desenvolvedores, usuários e partes interessadas, permitindo o compartilhamento de feedback e ideias para aprimorar continuamente as experiências gamificadas.

Para Seaborn e Fels (2015) garantir que o design e a implementação das soluções gamificadas estejam centrados no usuário, levando em consideração suas necessidades, preferências e objetivos individuais.

De acordo com Kapp (2012) oferecer oportunidades de capacitação e educação para desenvolvedores, profissionais e usuários de gamificação, a fim de ampliar o conhecimento e a compreensão sobre os princípios e práticas responsáveis de gamificação.

Promover a diversidade e a representatividade em todos os aspectos da gamificação, desde a concepção e desenvolvimento até a implementação e avaliação, para garantir que as experiências sejam inclusivas e reflitam as diversas perspectivas dos usuários segundo Steinfeld e Maisel (2012).

Para Sailer et al. (2023) implementar processos de monitoramento e avaliação para medir o impacto e a eficácia das soluções gamificadas, identificar áreas de melhoria e garantir a responsabilidade das partes envolvidas segundo.

A gamificação na educação oferece uma série de oportunidades para melhorar e enriquecer as experiências de aprendizagem tradicionais. Conforme mencionado por Sailer et al. (2023), ao combinar a gamificação com abordagens pedagógicas tradicionais, é possível criar experiências de aprendizagem combinadas que atendam a diversas preferências e necessidades de aprendizagem dos alunos. Detalhes de como a gamificação pode ser aplicada na educação.

Segundo Hamari et al. (2014) a gamificação na educação pode aumentar a motivação e o engajamento dos alunos ao incorporar elementos de jogo, como pontos, medalhas, placares e narrativas, em atividades e tarefas de aprendizagem. Isso pode tornar o aprendizado mais divertido e envolvente, incentivando os alunos a participar ativamente e a se comprometer com o processo de aprendizagem.

Ela também pode ser usada para criar experiências de aprendizagem personalizadas e adaptativas, que levam em consideração as habilidades, interesses e necessidades individuais dos alunos para Chen et al. (2021). Isso pode ser alcançado através do uso de algoritmos e técnicas de inteligência artificial, conforme Morschheuser et al. (2022), que permitem que os sistemas gamificados ajustem a dificuldade das tarefas e forneçam feedback personalizado com base no desempenho e progresso do aluno.

Para Dichev e Dicheva (2017) a gamificação pode apoiar o desenvolvimento de habilidades socioemocionais, como colaboração, comunicação, empatia e resolução de problemas. Ao incorporar atividades de equipe e cenários baseados em problemas nos sistemas gamificados, os alunos podem aprender a trabalhar juntos, compartilhar ideias e resolver desafios de forma eficaz.

A gamificação na educação pode fornecer feedback imediato e contínuo aos alunos, permitindo que eles monitorem seu progresso e ajustem suas estratégias de aprendizagem conforme necessário. Isso pode ajudar a criar uma cultura de aprendizagem autônoma, onde os alunos assumem a responsabilidade pelo próprio desenvolvimento e são encorajados a buscar melhorias contínuas.

Já Deterding et al. (2011) salientam que a gamificação pode ser integrada à aprendizagem baseada em projetos e à resolução de problemas, incentivando os alunos a aplicar seus conhecimentos e habilidades em contextos do mundo real. Isso pode promover a criatividade, o pensamento crítico e a capacidade dos alunos de transferir o aprendizado para situações fora da sala de aula.

A gamificação na educação pode ajudar a tornar o aprendizado mais inclusivo e acessível para uma ampla gama de alunos, incluindo aqueles com necessidades especiais ou dificuldades de aprendizagem segundo Steinfeld & Maisel (2012). Ao oferecer múltiplas formas de representação, expressão e engajamento, os sistemas gamificados podem ser projetados para acomodar as necessidades e preferências individuais dos alunos. Por exemplo, através de interfaces personalizáveis, métodos de entrada alternativos e ajustes nos níveis de dificuldade, os projetistas de gamificação podem garantir que todos os alunos possam participar e se beneficiar das experiências de aprendizagem gamificadas.

Para Morschheuser et al. (2022) a gamificação na educação pode ser combinada com tecnologias emergentes, como realidade virtual (VR), realidade aumentada (AR) e Internet das Coisas (IoT), para criar experiências de aprendizagem imersivas e inovadoras. Por exemplo, a VR e a AR podem ser usadas para criar simulações e ambientes de aprendizagem realistas e envolventes, enquanto dispositivos IoT podem fornecer feedback e suporte em tempo real para atividades de aprendizagem baseadas em projetos.

A gamificação na educação também pode ser usada para promover o desenvolvimento de competências digitais, como alfabetização digital, pensamento computacional e programação conforme Caponetto et al. (2014). Ao introduzir atividades e desafios relacionados à tecnologia nos sistemas gamificados, os alunos podem adquirir habilidades valiosas que serão cada vez mais importantes na sociedade e no mercado de trabalho do século XXI.

Da mesma forma, para Cugelman (2023) na saúde, a gamificação pode complementar os métodos convencionais de tratamento, tais como medicação e terapia, para fornecer aos pacientes apoio adicional e motivação para mudança de comportamento.

Hamari et al. (2014) destacam a importância da integração estratégica da gamificação às práticas educacionais tradicionais, a fim de maximizar os benefícios potenciais das intervenções gamificadas, ao mesmo tempo em que se preserva o valor e a integridade dos métodos estabelecidos. Para isso, é crucial identificar áreas prioritárias de pesquisa e abordá-las adequadamente. Algumas dessas áreas incluem:

É essencial investigar como a gamificação afeta o aprendizado e o engajamento dos alunos a longo prazo. Isso ajudará os educadores e pesquisadores a entender melhor os efeitos duradouros da gamificação na motivação, retenção de conhecimento e desenvolvimento de habilidades dos alunos.

O papel das diferenças individuais nos resultados da gamificação: As diferenças individuais, como personalidade, habilidades cognitivas e estilos de aprendizagem, podem afetar a eficácia da gamificação na educação. Pesquisar o papel dessas diferenças pode ajudar a projetar sistemas gamificados mais personalizados e adaptativos, que atendam às necessidades específicas de cada aluno.

Potencial da gamificação para enfrentar desafios globais: A gamificação pode ser usada para abordar problemas e desafios globais, como redução da pobreza, mudança climática e desenvolvimento sustentável conforme Dichev & Dicheva (2017). Estudar como a gamificação pode ser aplicada nesses contextos pode ajudar a identificar oportunidades e estratégias para promover a conscientização, a ação e a mudança em escala global.

Há uma necessidade de metodologias de pesquisa mais rigorosas e robustas que possam efetivamente avaliar o impacto das intervenções gamificadas. Sailer et al. (2023) destacam a necessidade de desenvolver novas ferramentas e métricas de avaliação para aprimorar a eficácia e o impacto da gamificação na educação. Essas abordagens podem incluir:

O desenvolvimento de questionários validados e confiáveis pode ajudar a avaliar a percepção dos alunos, o engajamento e a satisfação com as experiências gamificadas. Essas avaliações podem fornecer informações valiosas sobre como os alunos estão respondendo às intervenções gamificadas e podem ser usadas para adaptar e aprimorar o design dos sistemas gamificados.

Os indicadores de desempenho, como taxas de conclusão de tarefas, pontuações e progresso, podem ser usados para medir o sucesso das intervenções gamificadas. Esses indicadores podem ajudar a identificar áreas que precisam ser aprimoradas e a informar ajustes no design e na implementação da gamificação.

Para Sailer et al. (2017) Ensaios Controlados Aleatórios (ECAs) podem ser usados para comparar a eficácia de intervenções gamificadas com outras abordagens educacionais. Ao atribuir aleatoriamente os participantes a diferentes condições, os pesquisadores podem isolar os efeitos da gamificação e determinar seu impacto específico no aprendizado e no engajamento.

Estudos longitudinais, que acompanham os alunos ao longo do tempo, podem ajudar a entender os efeitos a longo prazo da gamificação na aprendizagem e no desenvolvimento de habilidades. Esses estudos podem fornecer informações importantes sobre como a gamificação afeta a motivação, a retenção de conhecimento e a transferência de habilidades para outros contextos.

A combinação de métodos quantitativos e qualitativos pode fornecer uma visão mais abrangente da eficácia da gamificação. Por exemplo, pesquisadores

podem usar análise de dados quantitativos para medir o desempenho e o engajamento dos alunos, enquanto realizam entrevistas ou grupos focais para obter insights mais aprofundados sobre suas experiências e percepções.

Ao avançar as metodologias de gamificação, os pesquisadores podem gerar evidências mais confiáveis e válidas sobre a eficácia da gamificação, contribuindo para o desenvolvimento de práticas e políticas baseadas em evidências em vários domínios.

Para sustentar o crescimento e desenvolvimento da gamificação, é crucial capacitar a próxima geração de profissionais da gamificação com o conhecimento, habilidades e experiência necessários para projetar, implementar e avaliar intervenções gamificadas efetivamente. De acordo com Koivisto et al. (2021), isso pode ser alcançado através do desenvolvimento de cursos, oficinas e programas de treinamento especializados em gamificação, bem como a criação de oportunidades de mentoria e caminhos de carreira para aspirantes a profissionais da gamificação.

Além disso, Sailer et al. (2023) enfatizam a importância de adotar metodologias ágeis de desenvolvimento, cultivar uma mentalidade de crescimento e promover uma cultura de experimentação e tomada de riscos dentro da comunidade de gamificação. Essas abordagens permitem a rápida adaptação e inovação em resposta às mudanças nas necessidades e expectativas dos usuários e da sociedade.

Outro aspecto importante é a criação de redes de colaboração entre profissionais da gamificação, acadêmicos, educadores e outros interessados. Essa colaboração pode impulsionar a inovação, compartilhando conhecimentos, recursos e melhores práticas entre os profissionais do campo.

Para alcançar sucesso a longo prazo no campo da gamificação, é fundamental investir na educação e no desenvolvimento profissional dos futuros profissionais da área. Ao fazer isso, as partes interessadas podem garantir que o campo continue a prosperar e inovar em resposta à evolução das necessidades e expectativas dos usuários e da sociedade como um todo.

Além disso, é importante estabelecer parcerias entre instituições acadêmicas, indústria e organizações governamentais para incentivar o desenvolvimento e a aplicação da gamificação em diversos contextos. Ao promover o diálogo e a cooperação entre esses diferentes setores, os

profissionais de gamificação podem identificar oportunidades promissoras e enfrentar os desafios emergentes no campo.

O estudo contínuo das motivações, comportamentos e necessidades dos usuários é crucial para aprimorar a eficácia e relevância da gamificação, assim como a adoção de abordagens éticas e responsáveis, garantindo privacidade, segurança, inclusão e acessibilidade. A gamificação prospera com investimento em educação, desenvolvimento profissional, colaboração entre setores, adaptação a tendências tecnológicas emergentes e compreensão das motivações e necessidades dos usuários.

A natureza interdisciplinar da gamificação proporciona oportunidades de colaboração e troca de conhecimento em vários campos. Ao se envolverem com especialistas de disciplinas como psicologia, sociologia, ciência da computação, educação e design. Hamari et al. (2014) aponta que pesquisadores e profissionais da gamificação podem desenvolver uma compreensão mais abrangente dos fatores que contribuem para o sucesso e eficácia das intervenções gamificadas.

Ela se baseia em teorias e princípios psicológicos para entender a motivação humana e projetar experiências que sejam atraentes e gratificantes para os usuários segundo Ryan e Deci (2000). Conceitos como motivação intrínseca e extrínseca, teorias de aprendizagem e mecanismos de recompensa estão no cerne do design de experiências gamificadas.

A gamificação também incorpora aspectos sociológicos, como a interação social, a colaboração e a competição entre os usuários. Segundo Deterding et al. (2011) a análise das dinâmicas sociais e culturais pode ajudar a projetar soluções gamificadas que sejam mais relevantes e atraentes para os usuários em diferentes contextos.

A implementação de soluções gamificadas geralmente envolve o uso de tecnologias digitais, como aplicativos móveis, plataformas online e sistemas de realidade virtual e aumentada. Para Dichev e Dicheva (2017) a ciência da computação fornece o conhecimento técnico necessário para desenvolver e integrar essas tecnologias de forma eficiente e eficaz.

A gamificação é amplamente aplicada no campo da educação para melhorar o engajamento dos alunos e facilitar a aprendizagem. Kapp (2012) aponta que a teoria educacional fornece informações valiosas sobre como

projetar experiências gamificadas que sejam pedagogicamente sólidas e alinhadas com os objetivos de aprendizagem.

O design é um componente essencial da gamificação, pois envolve a criação de experiências esteticamente agradáveis e funcionalmente eficientes. Zichermann e Cunningham, 2011 dizem que princípios de design de jogos, design de interação e design de interface do usuário são fundamentais para o desenvolvimento de soluções gamificadas que sejam envolventes e fáceis de usar.

De acordo com Kapp (2012), esta abordagem interdisciplinar pode fomentar o desenvolvimento de soluções inovadoras de gamificação que se baseiam em uma gama diversificada de perspectivas, teorias e metodologias, permitindo que o campo enfrente problemas e desafios complexos, tais como

Compreender e integrar múltiplas perspectivas: Para Zichermann e Cunningham (2011), o envolvimento de especialistas de várias disciplinas traz consigo uma diversidade de opiniões e teorias que precisam ser consideradas e integradas ao desenvolver soluções gamificadas. A síntese de diferentes abordagens e a criação de um entendimento comum entre os participantes é fundamental para o sucesso das intervenções gamificadas.

Dichev e Dicheva (2017) salientam a rápida evolução das tecnologias digitais impõe um desafio constante à gamificação. Para Landers et al. (2018), pesquisadores e profissionais devem se manter atualizados sobre as últimas tendências e desenvolvimentos tecnológicos para criar soluções que aproveitem ao máximo as possibilidades oferecidas pela tecnologia.

A avaliação do impacto e da eficácia das soluções gamificadas é um desafio significativo, uma vez que envolve a consideração de variáveis multifacetadas e a análise de dados qualitativos e quantitativos segundo Hamari et al. (2014). Já para Sailer et al. (2017) identificar as métricas adequadas e desenvolver métodos robustos de avaliação é crucial para determinar o sucesso das intervenções.

Um dos principais objetivos da gamificação é aumentar a motivação dos usuários, seja através de recompensas extrínsecas, como pontos e insígnias, ou através do incentivo à motivação intrínseca, relacionada ao interesse e à satisfação. De acordo com Mekler et al. (2013) encontrar o equilíbrio adequado entre esses dois tipos de motivação é um desafio que requer uma compreensão profunda das necessidades e preferências dos usuários.

Para Dunn et al. (2016) A criação de soluções gamificadas que sejam acessíveis e inclusivas para uma ampla variedade de usuários é um desafio complexo que deve ser abordado. Isso envolve considerar fatores como diversidade cultural, habilidades e limitações físicas, e garantir que as intervenções gamificadas sejam projetadas para atender às necessidades de todos os usuários.

Embora os projetos de gamificação em larga escala recebam frequentemente atenção e recursos significativos, é crucial reconhecer e apoiar iniciativas de gamificação de pequena escala e de base que podem ter um impacto mais localizado. A seguir, destacamos alguns exemplos:

- Duolingo é um aplicativo popular de aprendizado de idiomas que utiliza gamificação para motivar os usuários a aprenderem novas línguas. Com um sistema de pontos, conquistas, níveis e competições, o Duolingo mantém os usuários engajados e focados em seus objetivos de aprendizado.
- ECO Coins é um projeto que visa criar uma moeda sustentável para recompensar ações ecológicas positivas. Os participantes podem ganhar ECO Coins por meio de atividades como reciclagem, conservação de energia e participação em eventos ecológicos. Eles podem trocar suas moedas por produtos e serviços sustentáveis oferecidos por parceiros do projeto.
- Foldit é um projeto de ciência cidadã que utiliza a gamificação para envolver os jogadores na resolução de problemas complexos de dobramento de proteínas. Ao transformar a pesquisa científica em um jogo, o Foldit conseguiu atrair milhares de participantes e gerar resultados científicos significativos.
- Recyclebank é uma plataforma online que recompensa os usuários por aprender sobre sustentabilidade e realizar ações ecológicas, como reciclar e economizar energia. Os participantes ganham pontos que podem ser trocados por descontos e ofertas especiais em várias lojas e serviços.
- Salesforce, uma das maiores empresas de software de gestão de relacionamento com o cliente (CRM), integrou elementos de gamificação em sua plataforma para melhorar o engajamento e a produtividade de seus usuários. Através de competições,

recompensas e incentivos, a Salesforce incentiva os membros da equipe de vendas a alcançar e superar seus objetivos de desempenho.

- Waze (adquirida pelo Google em 2013) é um aplicativo de navegação e tráfego em tempo real que utiliza elementos de gamificação para encorajar os usuários a contribuírem com informações de tráfego e rotas. Os usuários ganham pontos e recompensas por relatar acidentes, engarrafamentos e outros eventos de trânsito, ajudando a construir um mapa de tráfego em tempo real mais preciso e útil.

Estas iniciativas podem fornecer valiosos insights sobre os desafios e oportunidades únicas associadas à gamificação em contextos e comunidades específicas, bem como servir como campo de teste para abordagens e estratégias inovadoras de gamificação conforme apontado por Koivisto et al. (2021).

Em suma, a gamificação surgiu como uma estratégia promissora para aumentar o envolvimento dos usuários, a motivação e a aprendizagem em ambientes como educação e trabalho, mas não somente limitado a eles. Ao incorporar elementos de projeto de jogos, apelando para a motivação do usuário e equilibrando a motivação intrínseca e extrínseca, ela tem o potencial de transformar a aprendizagem tradicional e as experiências de trabalho.

Zichermann e Cunningham (2011) categorizam a motivação individual em dois tipos: intrínseca, originada internamente no sujeito, e extrínseca, externa e baseada no ambiente circundante. A gamificação busca estimular a motivação intrínseca dos alunos, despertando o interesse em realizar atividades por vontade própria. Ao longo deste trabalho, detalharemos como a gamificação pode potencializar a motivação intrínseca dos estudantes.

A motivação intrínseca e extrínseca desempenha papéis importantes na eficácia da gamificação. A motivação intrínseca refere-se ao interesse e entusiasmo natural do indivíduo por uma atividade, sendo um poderoso impulsionador do engajamento e aprendizado. Já a motivação extrínseca é impulsionada por fatores externos, como recompensas, elogios ou a obtenção de um status específico.

A gamificação em um ambiente de aprendizagem deve, portanto, atuar com o intuito de aumentar a motivação extrínseca e intrínseca dos alunos,

procurando despertar diferentes tipos de emoções e sentimentos que aumentem seu interesse em realizar as atividades propostas e contribuir para a aquisição do conhecimento (SOBREIRO, 2017).

A gamificação busca harmonizar essas duas formas de motivação, oferecendo elementos lúdicos e desafiadores que estimulam a curiosidade e o interesse intrínseco, ao mesmo tempo que utiliza sistemas de recompensa, como pontos, insígnias e rankings, que atendem à motivação extrínseca. Para que a gamificação seja bem-sucedida em ambientes como educação e trabalho, é fundamental equilibrar cuidadosamente essas duas formas de motivação, ajustá-las às necessidades e preferências individuais dos usuários, a fim de promover experiências de aprendizagem e trabalho mais envolventes e significativas.

Entretanto, é crucial abordar as limitações e implicações éticas associadas à gamificação para garantir sua aplicação eficaz e responsável. À medida que a tecnologia avança e a pesquisa continua, as perspectivas futuras da gamificação são promissoras, com potencial para experiências mais imersivas, personalizadas e adaptativas.

5 DESENVOLVENDO O APRENDIZADO DE PROGRAMAÇÃO ORIENTADA A OBJETOS COM GAMIFICAÇÃO – PRODUTO DA DISSERTAÇÃO

Programação orientada a objetos é um paradigma de programação amplamente utilizado em várias linguagens de programação. Ensinar esse paradigma aos estudantes é crucial para prepará-los para carreiras de sucesso na área de desenvolvimento de software. No entanto, muitos alunos enfrentam desafios ao aprender esses conceitos, pois podem ser abstratos e difíceis de entender, sem exemplos concretos e práticos. Neste capítulo, apresentaremos uma roda de conversa realizada com profissionais da área para embasar o porquê da utilização da linguagem de programação Java no produto desta dissertação, uma aula de programação orientada a objetos gamificada. Em seguida, a aula foi desenvolvida para abordar os conceitos fundamentais de orientação a objetos, como objetos e classes, herança e polimorfismo, encapsulamento de forma mais envolvente, facilitando a compreensão dos alunos e com demonstrações práticas de como utilizamos e trabalhamos com a linguagem Java.

5.1 RODA DE CONVERSA

Para situar a composição da aula com as práticas realizadas por profissionais da área de engenharia de software foi realizada uma roda de conversa. Na composição do grupo foram considerados, para a participação, critérios que permitissem relatos de práticas de programadores iniciantes, experientes e seniores, observando a inclusão de diversidade de gênero masculino e feminino. A roda de conversa se assemelha a um grupo de discussão que aborda os pontos fundamentais das práticas de profissionais coletivamente. Para Weller (2010, p.247), esses tipos de coleta de dados é “um instrumento por meio do qual o pesquisador estabelece uma via de acesso que permite a reconstrução dos diferentes meios sociais e do *habitus* coletivo do grupo”. A autora ressalta que “seu objetivo principal é a análise dos epifenômenos (subproduto ocasional de outro) relacionados ao meio social, ao contexto geracional, às experiências de exclusão social (WELLER, 2010, p.247). Na obtenção dos dados provenientes das experiências se evidencia a contextualização da plataforma a ser produzida cotejadas com as práticas dos profissionais programadores. Isso permite nas análises a identificação do que importante no tratamento do tema.

Além disso, a roda de conversa é “uma forma de coleta de dados em que o pesquisador se insere como sujeito da pesquisa pela participação na conversa” (MOURA; LIMA, 2014, p. 25). Esses autores ressaltam que a roda de conversa é um processo de coleta de dados que permite compartilhar experiências e o desenvolvimento de reflexões sobre um assunto, que os participantes expressam considerações sobre suas práticas. “Rodas de Conversa, quando utilizada como instrumento de pesquisa, uma conversa em um ambiente propício para o diálogo”. (MOURA; LIMA, 2014, p. 25). Trata-se de uma roda de conversa porque não segue todas as regras de um grupo focal. A base desse tipo de coleta de dados toma como pressuposto as argumentações de Weller (2010).

Ressalta Weller (2010, p. 246) que “Os grupos de discussão, como método de pesquisa, passaram a ser utilizados a partir da década de 1980, sobretudo nas pesquisas sobre juventude”. A autora destaca que nesta técnica, a organização da coleta de dados se dá em fases, conforme Weller (2010, p.251): “Seleção das passagens relevantes para a pesquisa; transcrição da passagem inicial, das passagens de foco e daqueles relevantes para a pesquisa; reconstrução da estrutura temática da passagem a ser analisada, que também poderá ser dividida em temas e subtemas”. Esta técnica exige a transcrição completa das falas realizadas durante a roda de conversa, e recomenda uma análise do tipo documentário de interpretação (WELLER, 2010, p. 251), no qual

Esse processo compreende dois momentos: interpretação formulada e interpretação refletida. Durante a interpretação formulada, busca-se compreender o sentido imanente das discussões e decodificar o vocabulário coloquial.

Com efeito, o pesquisador faz o registro escrito do que foi dito pelos participantes evidenciando “o conteúdo dessas falas para uma linguagem que também poderá ser compreendida por aqueles que não pertencem ao meio social pesquisado”. (WELLER, 2010, p. 251).

Na finalização das análises Weller (2010, p. 252) afirma que “Toda interpretação somente ganhará forma e conteúdo quando realizada e fundamentada na comparação com outros casos empíricos” e em seguida colocada em discussão com os autores que fundamentaram o estudo, favorecendo elaborar apontamentos que se referem às situações típicas de um determinado grupo social e não restritas ao grupo participante.

Assim, para a roda de conversa foi feita a consulta da possibilidade de participação, com agenda definida em dia e hora. A roda de conversa foi gravada

e por isso foi escolhida o aplicativo Teams da Microsoft. Como os participantes são diferentes países foi necessário realizar com cuidado o agendamento pela diversidade de fusos horários.

Na organização da coleta de dados um planejamento das questões foi feito. Na definição dos itens foi feita consulta a outras pesquisas que abordam composição plataformas. Foram definidos como Tópicos orientadores da conversa:

- Introdução
- Pontos chave do aprendizado
- Dificuldades
- Qual a contribuição do Java na opinião de vocês?
- O que mais vocês (convidados) teriam para acrescentar?

Para a seleção dos participantes foram observadas as recomendações de Weller (2010) que recomenda incluir pessoas experientes com o assunto tratado, e mesmo que o próprio pesquisador deve ter alguma informação sobre a pertinência dos incluídos em relação ao assunto abordado. No caso, os integrantes pertencem aos mesmos grupos sociais de programadores. Essa pertinência favorece o levantamento de pontos pertinente durante a conversa. Moura e Lima (2014) indicam a importância da realização de convite para a participação, e que o convite deve ser feito de modo a conciliar a disponibilidade de todos em relação à agenda estabelecida. Os convites foram feitos para:

- José¹² – Engenheiro de Software, reside no exterior.
- Paula – Engenheira Elétrica de formação e está mudando de área, mais especificamente na área de Engenharia de Dados, morou no exterior.
- Rogério – Engenheiro de Software com bastante experiência, residiu no exterior.
- Laerte – Engenheiro de Dados, residiu no exterior.
- Aldair – Gerente de Arquitetura de Software em uma multinacional.

Após os convites realizados por meio de WhatsApp, os participantes expressaram seu consentimento em participar da investigação. A roda de

¹² Foram considerados nomes fictícios para preservar o anonimato respeitando as normas éticas em pesquisa com seres humanos.

conversa foi gravada e transcrita. Para a gravação os participantes foram consultados quanto à autorização.

Em relação às questões éticas em pesquisa com seres humanos estão respeitadas as indicações de anonimato dos participantes, o respeito a fidedignidade dos dados consultados. Também são observados os preceitos de confidencialidade dos dados e o consentimento expresso no Termo de Consentimento Livre e Esclarecido, conforme consta do Anexo I. Ainda, observa-se a questão de riscos na coleta de dados não incluindo questões de ordem pessoal para a realização da roda de conversas.

Para análise das falas realizadas durante a roda de conversa foi feito um agrupamento que toma como possibilidade de compreensão dos dados obtidos a análise do conteúdo foi fundamental. Conforme, Bardin (2011) examinar o conteúdo é uma técnica de pesquisa com elementos e características metodológicas que possibilitam analisar os dados com objetividade, a partir da sistematização das falas e contribuições dos participantes. Ressalta Bardin (2011, p. 42), que as técnicas de análise de dizeres ao serem examinadas com procedimentos sistemáticos favorecem fazer submergir os conteúdos presentes nas mensagens contribuindo para apontar indicadores.

Do conjunto das leituras dessas falas, considerando os níveis da primeira leitura, flutuante, até a última leitura interpretativa, realizadas em etapas sucessivas, destacando os pontos fundamentais, se estabelece a releitura compreensiva, a leitura de sistematização destacando os conteúdos importantes para a composição do produto de pesquisa. O processo dessa leitura também toma por referência as etapas propostas por Bardin (2011), para o processo da análise de dados, quais se apresentam como: a descrição analítica, a interpretação referencial dos dados e a análise, realizada pela organização dos conteúdos das falas e pela leitura flutuante, que faculta ao pesquisador elaborar indicativos contribuem para a interpretação dos dados. São atributos de análise, conforme Bardin (2011). Os objetos das falas dos participantes são originários de sua experiência profissional. Essas experiências contêm aspectos psicológicos e comportamentais, relacionados às emoções, pensamentos e ações, como ressalta Bardin (2011). Enfim esses conteúdos incluem propriedades que expressam os espaços e tempos de atuação profissional dos participantes de modo quantitativo e qualitativo, ainda que nessa pesquisa se valorize muito mais a qualidade das proposições e não a sua quantidade.

Com esse conjunto de indicações de procedimentos metodológicos parece ser possível levar a termo essa investigação.

5.1.1 Dados da Roda de Conversa

A roda de conversa, com duração de 1 hora, 55 minutos e 33 segundos, proporcionou um ambiente propício para a troca de ideias e a construção de conhecimento entre os participantes. O grupo apresentou uma diversidade de experiências com linguagens orientadas a objetos, como por exemplo Java, além de outras linguagens de programação, incluindo profissionais que trabalham com Java desde o início dos anos 2000 e aqueles que ingressaram recentemente na área. Em muitos casos, Java foi a primeira ou segunda linguagem aprendida durante a carreira, embora tenha sido a primeira para a grande maioria. Python e NodeJS também foram mencionadas como linguagens adquiridas posteriormente pelos colegas, com alguns deles atuando principalmente com essas linguagens ou, em alguns casos, trabalhando com múltiplas linguagens, dependendo da atividade a ser realizada.

Na formação acadêmica, quase todos os participantes tiveram contato com Java como a primeira linguagem de programação aprendida em sala de aula. A riqueza da documentação, a quantidade de materiais disponíveis para aprendizado e a facilidade de encontrar esses materiais foram destacadas, assim como a dimensão e qualidade da comunidade, a quantidade e qualidade de bibliotecas e frameworks e o fato de algumas dessas bibliotecas serem incorporadas pela linguagem e se tornarem padrão. Além disso, foi mencionada a excelente capacidade de integração do Java com outros componentes, bem como a quantidade de livros traduzidos para o português. Todos os participantes, sem exceção, passaram a consumir mais conteúdo em inglês, o que facilitou bastante o aprendizado, pois aumentou a quantidade e qualidade do material disponível. No entanto, todos comentaram a dificuldade no aprendizado da língua inglesa. No início da carreira, a maioria dos participantes seguiu um aprendizado de maneira desestruturada.

A evolução do Java e a adoção de melhorias de outras linguagens de programação ao longo do tempo, como *Generics*, *Lambda* e *Closure*, também foram discutidas. A velocidade das atualizações da linguagem é um tema polêmico, pois, apesar de trazer novos recursos e capacidades, pode ser um desafio manter a última versão atualizada para algumas empresas e

colaboradores, especialmente quando se possui um grande número de servidores. A disponibilidade de múltiplos *Java Development Kits* (JDKs) e *Java Virtual Machines* (JVMs) foi vista como positiva pelos participantes.

A orientação a objetos foi discutida como a abordagem dominante no início dos anos 2000, e o Java foi lembrado como a linguagem que atraiu muitos programadores devido às inúmeras oportunidades de emprego disponíveis nas empresas. Recursos de aprendizado, como o fórum GUJ e a Java Magazine, foram mencionados com apreço como pilares do aprendizado. Os participantes concordaram que é essencial praticar gradualmente o que se aprende para solidificar o conhecimento. Alguns participantes aprenderam outras linguagens ao longo do tempo, como Scala, Elixir e Closure, e relataram dificuldades com todas elas. Ter parcerias e colaborações durante o aprendizado foi apontado como extremamente útil, assim como a importância de ter objetivos tangíveis e construir projetos passo a passo para alcançar esses objetivos. A leitura e compreensão de código, mesmo sem ter amplo domínio da linguagem, foi destacada como um exercício válido para o aprendizado.

A partir da segunda linguagem adquirida, a curva de aprendizado tende a diminuir. A flexibilidade, ou falta dela, em linguagens de programação foi discutida. Java é vista como uma linguagem mais restrita, o que pode ser desafiador no início do aprendizado, mas, eventualmente, é considerada uma qualidade. Linguagens mais permissivas, como JavaScript, requerem uma base mais sólida para escrever códigos eficientes, devido à maior flexibilidade e ao risco de encontrar armadilhas no código ou seguir caminhos menos eficientes e mais complicados.

Houve discussões sobre o equilíbrio entre o aprofundamento dos conhecimentos em linguagem, estruturas de dados, algoritmos, padrões de projeto e o pragmatismo necessário para ensinar o mínimo indispensável para que alguém possa ingressar no mercado de trabalho. Além disso, foi enfatizado que o conhecimento em linguagem de programação, por si só, não é suficiente. Hoje em dia, é preciso dominar noções de implantação de código em produção, ferramentas de suporte, computação em nuvem, escalabilidade, testes, transações, DevOps e outros aspectos relacionados à área de tecnologia.

É inegável que o contexto tecnológico e a demanda do mercado continuam evoluindo rapidamente. Por isso, os profissionais devem ser capazes de aprender e se adaptar às mudanças na indústria. Desenvolver habilidades de

aprendizagem ao longo da vida, como autodidatismo, resolução de problemas e pensamento crítico, torna-se cada vez mais importante.

A troca de experiências e conhecimentos entre profissionais da área também é fundamental, seja por meio de fóruns, redes sociais, grupos de estudo, workshops ou eventos. Esses espaços de interação proporcionam um ambiente onde os participantes podem compartilhar desafios, soluções e melhores práticas, bem como se manter atualizados sobre as últimas tendências e avanços tecnológicos.

Outra perspectiva levantada foi a importância da diversidade de conhecimentos, indo além do domínio de uma única linguagem de programação. A familiaridade com múltiplas linguagens e tecnologias é benéfica tanto para o crescimento profissional quanto para a resolução de problemas complexos e multifacetados.

Além disso, o mercado atual exige uma compreensão cada vez maior dos aspectos éticos e sociais relacionados ao desenvolvimento e implementação de soluções tecnológicas. Os profissionais de tecnologia da informação devem estar cientes das implicações e consequências de suas ações e decisões no contexto mais amplo da sociedade.

Neste cenário, a roda de conversa corrobora as evidências encontradas na literatura, destacando que o Java ainda é uma linguagem de programação altamente atrativa para o aprendizado atualmente, especialmente como primeira linguagem. Obviamente, não se trata de uma solução perfeita para todas as questões de negócios ou tecnologias. O ideal é utilizar a ferramenta mais adequada para cada tarefa específica, o que sempre representa um desafio. Entretanto, é consenso que o Java é uma linguagem respeitada e com um amplo mercado global.

O contexto geográfico e social pós-pandêmico também impulsiona a relevância do aprendizado de linguagens de programação, não apenas do Java, mas de outras linguagens igualmente, visto que a possibilidade de trabalhar remotamente de qualquer parte do mundo amplia significativamente as oportunidades profissionais. Agora, é possível trabalhar em projetos, colaborar com colegas, atuar em empresas e enfrentar desafios internacionais sem sair de casa. Embora a barreira linguística possa ser um obstáculo em alguns casos, dependendo do cliente, da empresa ou do país, essa questão poderia ser abordada em uma discussão separada.

Dessa forma, a roda de conversa serve como um importante espaço de reflexão e troca de experiências sobre o papel do Java e outras linguagens de programação, no contexto atual e futuro da indústria tecnológica. Ela também enfatiza a necessidade de se adaptar às mudanças e desafios do mercado de trabalho, buscando o aprimoramento contínuo das habilidades e competências necessárias para ter sucesso nesse ambiente dinâmico e globalizado.

Em suma, o debate sobre a melhor primeira linguagem de programação a ser aprendida é complexo e multifacetado, envolvendo considerações pedagógicas, mercadológicas, técnicas e pessoais. Independentemente da linguagem escolhida, é fundamental investir no desenvolvimento contínuo das habilidades e competências que permitam aos profissionais enfrentar os desafios do mercado e adaptar-se às constantes mudanças no campo da tecnologia da informação, tendo sempre em mente a responsabilidade ética e social inerente à atuação na área.

5.2 AULA GAMIFICADA DE PROGRAMAÇÃO ORIENTADA A OBJETOS

Esta dissertação propõe uma aula interativa de 140 minutos (recomendável com um intervalo de 10 minutos), com o objetivo de promover o ensino para a aprendizagem dos conceitos fundamentais de programação orientada a objetos, empregando a gamificação como meio de aumentar o engajamento e a compreensão dos estudantes. A linguagem de programação Java será utilizada como base para as atividades práticas. A aula proposta toma por referência os fundamentos teóricos da programação orientada a objetos desenvolvidos nessa pesquisa e dos dados coletados na roda de conversa. Portanto, cinge-se a uma proposta, não aplicada. A sua aplicação poderá gerar um novo estudo.

A aula pode ser amparada por recursos auxiliares como slides, quadro branco, códigos-exemplo, um quadro de líderes e um sistema de recompensas. Essa estrutura pode ser facilmente adaptada para um ambiente online, utilizando soluções digitais como alternativas ao quadro branco ou lousa.

Troféu e medalhas podem ser de ouro, prata e bronze, sendo 50 pontos para a o troféu que somente será presenteada em ocasiões especiais, 10 pontos para a medalha de ouro, 6 para a medalha de prata e 3 para a medalha de bronze.

O troféu e medalhas serão confeccionados em uma imagem JPG e será distribuído formalmente pelo professor, com menção a todos os participantes da aula, destacando que todos conseguiram e que, independente da dificuldade, todos estão aptos.



Fonte: Canva¹³

O quadro de liderança ficará exposto para todos os participantes da aula, mas sempre buscando incentivar os que estão abaixo na tabela, mostrando como a diferença em relação aos vencedores não é inatingível.

A fim de equalizar os participantes da aula, recompensas devem ser dadas observando não apenas os resultados práticos, mas também os esforços. Por isso, o professor poderá distribuir recompensas não aos líderes, mas sim aos que mais se esforçaram, como incentivo:

- Ganhará 10 pontos e uma medalha de ouro, o participante da aula que concluir a atividade ou desafio sem ajuda, 6 pontos e uma medalha de prata os que conseguirem com a ajuda de algum colega e 3 pontos e uma medalha de bronze aquele que conseguir com a ajuda do professor.
- Conclusão de todas as tarefas sem ajuda dos colegas ou professor: Medalha de ouro e 10 pontos
- Alunos que auxiliem um colega recebem uma medalha de bronze adicional e mais 3 pontos. Alunos que auxiliem três colegas recebem uma medalha adicional de prata e mais 6 pontos. Alunos que auxiliarem cinco colegas ou mais recebem uma medalha adicional de ouro e mais 10 pontos para cada cinco colegas auxiliados

¹³ <http://www.canva.com>

- O participante da aula que tiver auxiliado o maior número de colegas recebe um troféu e 50 pontos

Este esquema se aplica para cada tópico ou módulo da aula, sendo que cada tópico ou módulo pode possuir desafios adicionais.

O plano de aula a seguir apresenta detalhes sobre o conteúdo a ser ministrado e a aplicação da gamificação como metodologia de ensino, além de sugestões práticas para o uso da linguagem Java durante a atividade.

Agenda da aula:

- Introdução
- Configuração do ambiente de desenvolvimento Java
- Classes e objetos
- Herança e polimorfismo
- Encapsulamento
- Conclusão e recompensas

Este projeto Java será escrito com o auxílio da biblioteca Maven (para vias de simplificação na criação do projeto Java, detalhes não serão cobertos pela aula) e ferramenta IDE NetBeans, ambos são projetos da fundação Apache.

Apache Maven é uma ferramenta de automação de compilação e gerenciamento de projetos de software, amplamente utilizada em projetos Java. Ele ajuda a padronizar o processo de construção, facilitando a compilação, teste, empacotamento, implantação e distribuição de projetos de software.

NetBeans é um ambiente de desenvolvimento integrado (IDE - *Integrated Development Environment*) de código aberto e gratuito, desenvolvido principalmente para a linguagem de programação Java, mas também suporta outras linguagens. Foi inicialmente lançado em 1997.

5.2.1 Introdução

Tempo: 15 minutos

Apresentar brevemente o tema da aula que é a programação orientada a objetos explicando conceitos como classes, objetos, herança, polimorfismo e encapsulamento.

Introduzir a gamificação como uma abordagem para melhorar o envolvimento e a aprendizagem, discutindo elementos como pontos, troféus e medalhas, quadros de liderança e recompensas.

5.2.2 Configuração do Ambiente de Desenvolvimento Java

Tempo: 25 minutos

Orientar os estudantes através da configuração de um ambiente de desenvolvimento Java ou *Java Development Kit*, instalação da IDE Apache NetBeans e criação de um projeto Java.

Instalar o OpenJDK:

- a. Acesse o site do Adoptium: <https://adoptium.net/>.
- b. Selecione a versão mais recente do OpenJDK adequada ao seu sistema operacional (Windows, macOS ou Linux).
- c. Faça o download do instalador e execute-o.
- d. Siga as instruções na tela para concluir a instalação.

Quem instalar o OpenJDK primeiro ganha uma medalha de bronze e 3 pontos.

Instalar o Apache NetBeans:

- a. Acesse o site oficial do Apache NetBeans através do seguinte link: <https://netbeans.apache.org/download/index.html>.
- b. Localize e selecione a versão apropriada do NetBeans para o seu sistema operacional (Windows, macOS ou Linux).
- c. Faça o download do arquivo de instalação.
- d. Execute o arquivo baixado e siga as instruções na tela para concluir a instalação, isso pode ser geralmente feito através de duplo clique no ícone do instalador.

Após a instalação bem-sucedida do Apache NetBeans, inicie o programa clicando no ícone do NetBeans no seu computador.

Quem instalar o Apache NetBeans primeiro ganha uma medalha de bronze e 3 pontos.

Para confirmar se o Java está instalado no seu computador, siga estas instruções para Windows e macOS:

Windows: Pressione as teclas *Win* + *R* no seu teclado para abrir a janela “Executar”. Digite *cmd* e pressione *Enter* para abrir o “Prompt de Comando”. Na

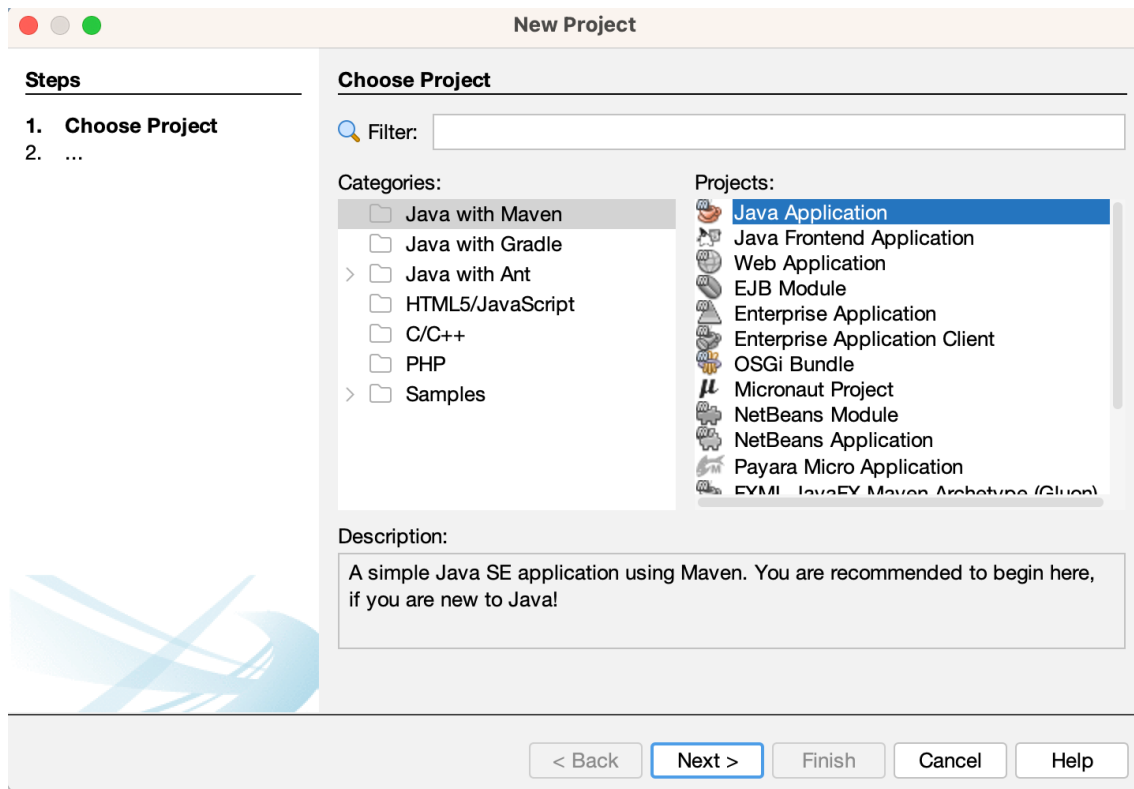
janela do “Prompt de Comando”, digite o seguinte comando e pressione Enter: “*java -version*”. Se o Java estiver instalado, você verá informações sobre a versão do Java, como “*openjdk version*” ou “*java version*”. Se não estiver instalado, você verá uma mensagem indicando que o comando não foi encontrado.

macOS: Abra o aplicativo “Terminal” (você pode encontrá-lo no Launchpad ou usando a busca no Spotlight). Na janela do “Terminal”, digite o seguinte comando e pressione Enter: “*java -version*”. Se o Java estiver instalado, você verá informações sobre a versão do Java, como “*openjdk version*” ou “*Java version*”. Se não estiver instalado, você verá uma mensagem indicando que o comando não foi encontrado.

Exemplo de instalação bem sucedida do JDK:

```
$ java -version
openjdk version "20" 2023-03-21
OpenJDK Runtime Environment Homebrew (build 20)
OpenJDK 64-Bit Server VM Homebrew (build 20, mixed mode, sharing)
```

Para criar um projeto Maven no Apache NetBeans, siga as instruções: Abra o Apache NetBeans. No menu superior, clique em Arquivo (ou “*File*”, caso esteja em inglês) e selecione “Novo Projeto...” (ou “*New Project...*”). Na janela “Novo Projeto” que se abre, expanda a categoria Java com Maven (ou “*Java with Maven*”, caso esteja em inglês). Selecione Aplicação Java (ou “*Java Application*”) e clique em Próximo (ou *Next*).



Agora você verá a tela “Nome e Localização do Projeto” (ou “*Project Name and Location*”). Aqui, você pode configurar o nome do projeto, o local onde ele será salvo e o pacote base. Preencha os campos conforme desejado e clique em Finalizar (ou “*Finish*”). Neste caso iremos digitar no campo *Project Name* ou Nome do Projeto “oop” e no campo Group Id: “com.uninter.oop”

O Apache NetBeans criará o projeto Maven e abrirá a estrutura de diretórios na janela “Projetos” (ou “*Projects*”). Você pode começar a escrever seu código no arquivo, neste caso Oop.java, Mavenproject1.java caso você utilize a opção padrão, ou o nome que você tenha escolhido. O arquivo está localizado dentro do pacote (*package*) especificado anteriormente.

Opção padrão da tela.

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

Artifact Id:

Group Id:

Version:

Package: (Optional)

< Back Next > **Finish** Cancel Help

Após a customização desejada.

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

Artifact Id:

Group Id:

Version:

Package: (Optional)

< Back Next > **Finish** Cancel Help

Atribuir pontos para a conclusão bem sucedida da configuração e atualizar o quadro de liderança com a inclusão de um desafio extra: o participante da aula que for mais criativo e conseguir explorar novas classes e objetos no fim da tarefa receberá adicionalmente uma medalha de prata e 6 pontos.

Quem criar o projeto com o nome mais criativo ganha uma medalha de bronze e 3 pontos. Quem criar o projeto com o nome menos criativo também ganha uma medalha de bronze e 3 pontos.

Quem criar o pacote com o nome mais criativo ganha uma medalha de bronze e 3 pontos. Quem criar o pacote com o nome menos criativo também ganha uma medalha de bronze e 3 pontos.

Os participantes da aula vão escolher os vencedores com a ajuda do professor.

5.2.3 Classes e Objetos Java

Tempo: 25 minutos

Ensinar os participantes da aula a criar classes e objetos em Java.

Uma classe em Java é um bloco de construção fundamental na programação orientada a objetos (OOP) usando a linguagem Java. Uma classe é um modelo ou um plano que define a estrutura e o comportamento dos objetos que serão criados com base nessa classe. Em outras palavras, uma classe é um conjunto de especificações que descrevem as propriedades (também chamadas de atributos) e os métodos (ações ou comportamentos) que os objetos dessa classe possuem.

A classe serve como um modelo a partir do qual os objetos são criados ou instanciados. Cada objeto criado a partir de uma classe é chamado de “instância” da classe e possui seu próprio conjunto de valores para as propriedades e pode executar os métodos definidos na classe.

Em Java, as classes são definidas usando a palavra-chave *class*, seguida pelo nome da classe e um bloco de código que contém os atributos e métodos da classe.

Um *package* ou pacote é uma forma de organizar classes e interfaces relacionadas em um único *container* ou diretório. Pacotes facilitam a modularização, reutilização e manutenção do código, além de evitar conflitos de nomes entre classes com o mesmo nome em diferentes partes do projeto. Eles também fornecem um mecanismo para controlar o acesso a membros de classes e interfaces por meio de modificadores de acesso.

Um pacote é criado declarando-se uma instrução de pacote no início de um arquivo de código-fonte Java, antes de qualquer declaração de classe ou

interface. A instrução do pacote é seguida pelo nome do pacote, que geralmente segue uma convenção de nomenclatura baseada no domínio reverso da organização ou desenvolvedor responsável pelo código, garantindo a unicidade dos nomes de pacotes.

Objetos são a representação de entidades do mundo real, que possuem características (atributos) e comportamentos (métodos). Um objeto é uma instância de uma classe, que é a estrutura ou modelo que define os atributos e métodos que um objeto pode ter.

Na prática para criar uma nova classe Java no Apache NetBeans, siga as etapas abaixo:

Abra o Apache NetBeans e certifique-se de que o seu projeto está aberto na janela “Projetos” (ou “Projects”).

Expanda os diretórios do seu projeto até encontrar o pacote onde deseja adicionar a nova classe. Os pacotes geralmente estão localizados dentro do diretório `src/main/java`. Ou dentro da estrutura de pacote customizada, neste caso `src/main/java/com/uninter/oop`. Esta estrutura se encontra dentro do diretório do projeto criado.

Clique com o botão direito do mouse no pacote onde deseja criar a nova classe e selecione **Novo Arquivo** (ou **New File**) > **Classe Java...** (ou **Java Class...**).

Na janela “Nova Classe Java” (ou “New Java Class”) que aparece, insira o nome da classe no campo “Nome da Classe” (ou “Class Name”). É importante seguir as convenções de nomenclatura, ou seja, começar com uma letra maiúscula e usar o formato CamelCase.

(Opcional) Se desejar alterar o pacote, você pode fazê-lo no campo “Pacote” (ou “Package”).

Clique em **Finalizar** (ou **Finish**) para criar a nova classe.

Quem criar a classe com o nome mais criativo ganha uma medalha de bronze e 3 pontos. Quem criar a classe com o nome menos criativo também ganha uma medalha de bronze e 3 pontos.

Neste exemplo criamos uma classe Carro da seguinte forma (caso queira copiar e colar):

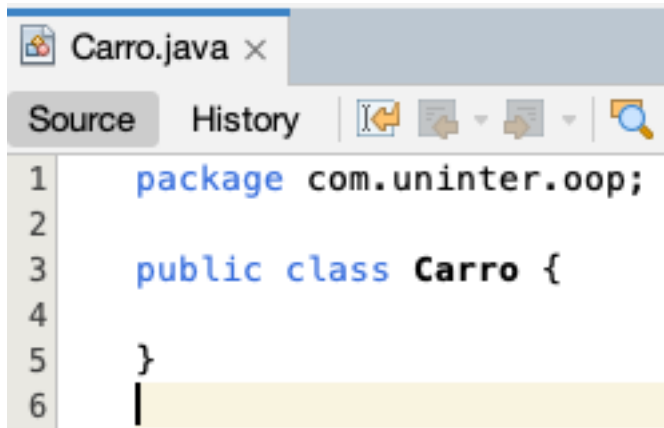
Classe Carro.java e seu conteúdo

```
package com.uninter.oop;
```

```
public class Carro {
```

```
}
```

Imagem da classe Carro no NetBeans.



Quem criar outra classe com o nome mais criativo ganha uma medalha de bronze e 3 pontos. Quem criar outra classe com o nome menos criativo também ganha uma medalha de bronze e 3 pontos.

Neste exemplo editamos a classe Oop.java (ou outro nome caso tenha escolhido anteriormente) e instanciamos a classe Carro armazenando-a no objeto *carro*. Não vamos nos ater aos detalhes do “public static void main” agora em Java, por hora basta saber que devemos colocar as instruções de código dentro deste bloco para sua execução.

Classe Oop.java e seu conteúdo

```
package com.uninter.oop;
```

```
public class Oop {
```

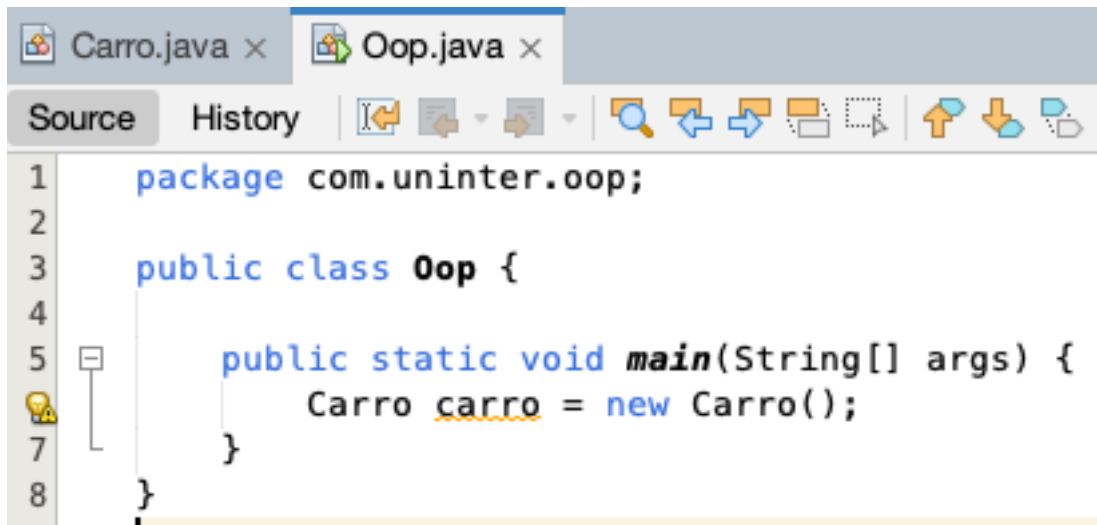
```
    public static void main(String[] args) {
```

```
        Carro carro = new Carro();
```

```
    }
```

```
}
```

Imagem da classe Oop no NetBeans.



Logo neste caso a classe é Carro, e o objeto é carro também. Para objetivo de clareza. Poderíamos instanciar várias vezes a classe Carro, cada uma em um objeto distinto, como por exemplo:

Classe Oop.java e seu conteúdo

```
package com.uninter.oop;
```

```
public class Oop {
```

```
    public static void main(String[] args) {
```

```
        Carro carro = new Carro();
```

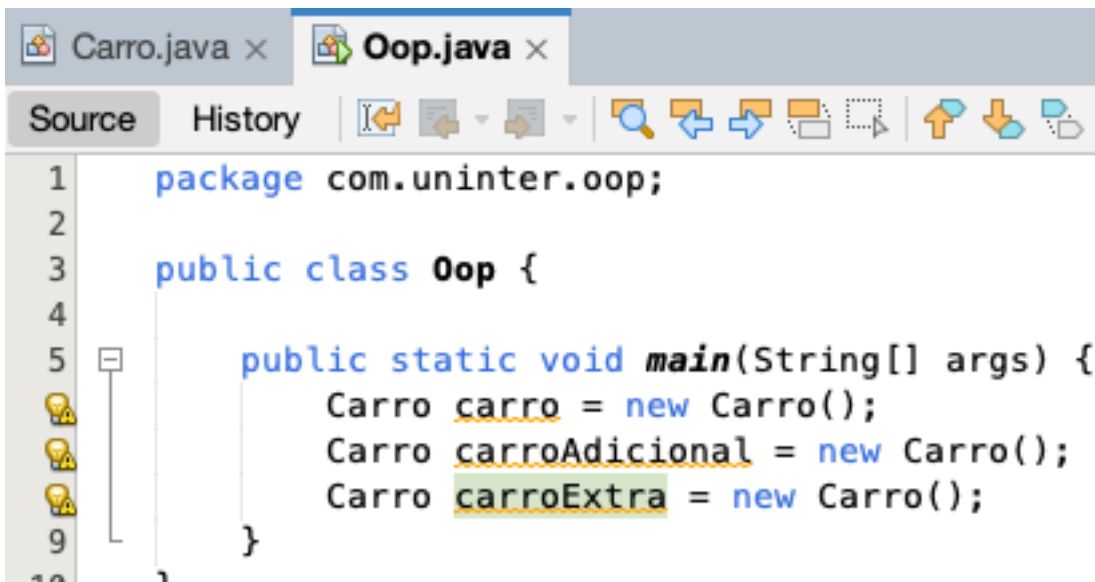
```
        Carro carroAdicional = new Carro();
```

```
        Carro carroExtra = new Carro();
```

```
    }
```

```
}
```

Imagem da classe Oop no NetBeans após as novas instancias de Carro.



```

1  package com.uninter.oop;
2
3  public class Oop {
4
5      public static void main(String[] args) {
6          Carro carro = new Carro();
7          Carro carroAdicional = new Carro();
8          Carro carroExtra = new Carro();
9      }
10

```

Atribuir pontos para a conclusão bem sucedida da tarefa e atualizar o quadro de liderança com a inclusão de um desafio extra: o aluno que for mais criativo e conseguir explorar novas classes e objetos (como tipos de combustíveis, veículos etc.) no fim da tarefa receberá adicionalmente uma medalha de prata e 6 pontos. Os participantes da aula vão escolher o vencedor com a ajuda do professor.

5.2.4 Herança e Polimorfismo

Tempo: 25 minutos

Ensinar aos participantes da aula sobre herança e polimorfismo em Java, usando exemplos do desafio de codificação anterior.

Herança é um mecanismo que permite a criação de uma nova classe a partir de uma classe existente, herdando seus atributos e métodos. A nova classe é chamada de subclasse e a classe existente de superclasse. Uma superclasse é uma classe da qual outras classes podem herdar atributos e métodos. Em outras palavras, é uma classe pai ou classe base de uma ou mais subclasses.

Em Java, a palavra-chave *extends* é usada para implementar a herança. A herança promove a reutilização de código e a organização hierárquica das classes, ou que o mesmo bloco de código seja reutilizado de forma que com o

aumento da quantidade de código escrito não necessariamente o código cresce de maneira diretamente proporcional.

Polimorfismo é a capacidade de uma classe assumir várias formas, permitindo que objetos de diferentes classes sejam tratados como objetos de uma classe comum. O termo “polimorfismo” deriva das palavras gregas “*poli*” (muitos) e “*morf*” (formas), sugerindo que uma única classe ou interface pode assumir múltiplas formas. Em Java, o polimorfismo é implementado através de herança e interfaces. O polimorfismo permite a reutilização de código e flexibilidade no design do software. Em Java pode ser implementado de duas maneiras: polimorfismo de subtipos (também conhecido como polimorfismo de tempo de execução) e polimorfismo paramétrico (também conhecido como polimorfismo de tempo de compilação).

Vamos demonstrar na prática este conceito criando a classe Combustivel (sem acento mesmo pois por padrão classes Java devem ser simplificadas, ou sem caracteres especiais). Crie também a classe Etanol seguindo os passos anteriormente demonstrados para a criação da classe Carro, só mude o nome para Combustivel e Etanol respectivamente.

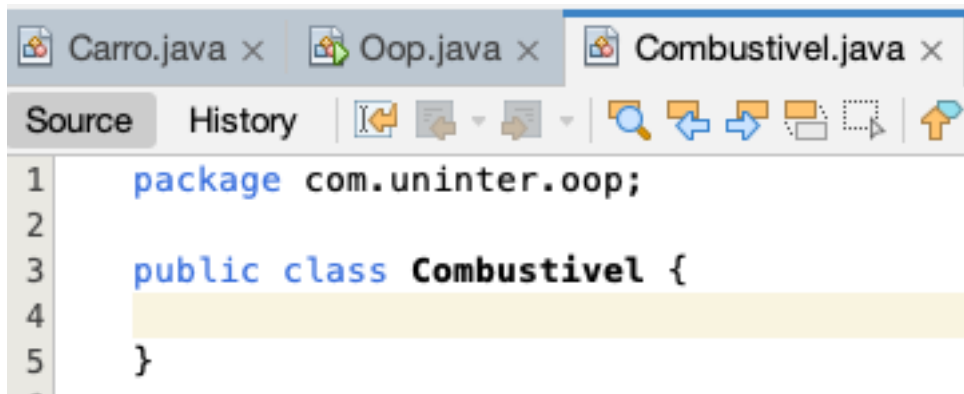
Classe Combustivel.java e seu conteúdo

```
package com.uninter.oop;  
  
public class Combustivel {  
  
}
```

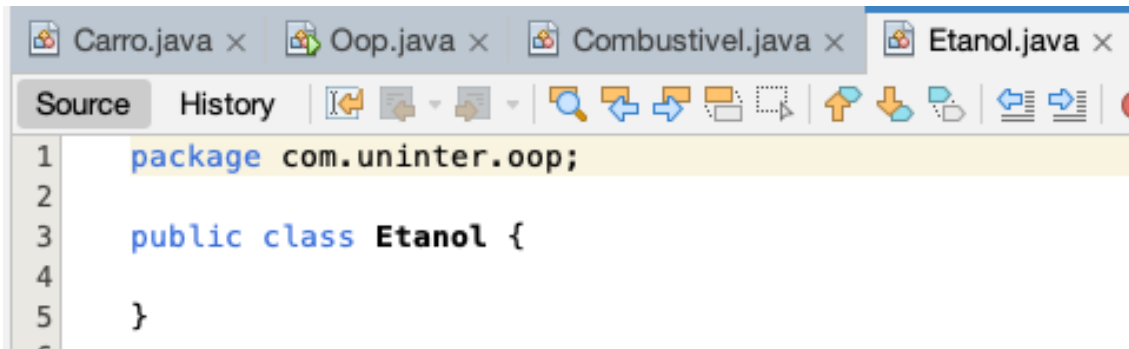
Classe Etanol.java e seu conteúdo

```
package com.uninter.oop;  
  
public class Etanol {  
  
}
```

Imagem das classes Combustivel e Etanol no NetBeans.



```
1 package com.uninter.oop;  
2  
3 public class Combustivel {  
4  
5 }
```



```
1 package com.uninter.oop;  
2  
3 public class Etanol {  
4  
5 }
```

Para demonstrar herança, iremos alterar a classe Etanol de forma que ela agora estenda a classe Combustivel com a adição da palavra *extends* e a classe que desejamos, da seguinte forma:

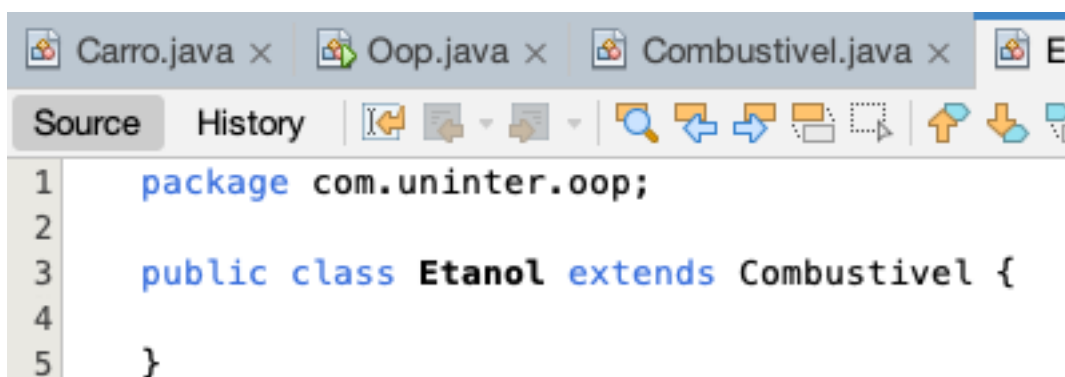
Classe Etanol.java e seu conteúdo agora com herança

```
package com.uninter.oop;
```

```
public class Etanol extends Combustivel {
```

```
}
```

Exemplo da classe Etanol estendendo a classe Combustível como demonstração de herança.



```
1 package com.uninter.oop;  
2  
3 public class Etanol extends Combustivel {  
4  
5 }
```

Iremos agora trabalhar o polimorfismo demonstrando a Classe Etanol sendo instanciada em um objeto Combustivel.

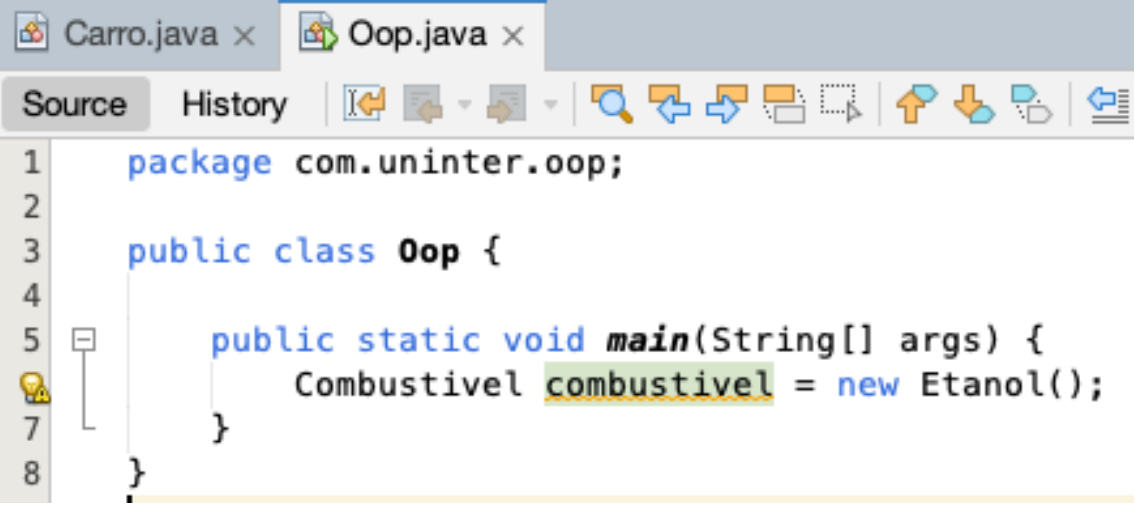
Classe Oop.java demonstrando polimorfismo

```
package com.uninter.oop;

public class Oop {

    public static void main(String[] args) {
        Combustivel combustivel = new Etanol();
    }
}
```

Imagem da class Oop com exemplo de polimorfismo instanciando a classe Etanol no objeto Combustivel.



```
1 package com.uninter.oop;
2
3 public class Oop {
4
5     public static void main(String[] args) {
6         Combustivel combustivel = new Etanol();
7     }
8 }
```

Este conceito nos permite tratar diversas classes quando existem similaridades, por exemplo poderíamos criar classes de Gasolina. E estender Gasolina em classes de GasolinaComum ou GasolinaAditivada. Poderíamos utilizar este conceito para se trabalhar com um motor Total Flex por exemplo, o qual seria abastecido com um combustível. E este combustível poderia ser gasolina comum, gasolina aditivada, etanol etc.

Atribuir pontos para a conclusão bem sucedida da tarefa e atualizar o quadro de liderança com a inclusão de um desafio extra, o desafio de

implementações elegantes e uso extensivo de herança e polimorfismo: o aluno que for mais criativo e conseguir explorar outras classes e herança no fim da tarefa receberá adicionalmente uma medalha de prata e 6 pontos. Os participantes da aula vão escolher o vencedor com a ajuda do professor.

5.2.5 Encapsulamento

Tempo: 25 minutos

Ensinar aos participantes da aula sobre encapsulamento e a importância de esconder dados em Java.

Atribuir pontos para a conclusão bem sucedida da tarefa e atualizar o quadro de liderança com a inclusão de um desafio extra, o desafio de criatividade: o aluno que for mais criativo e conseguir explorar o encapsulamento no fim da tarefa receberá adicionalmente uma medalha de prata e 6 pontos. Os participantes da aula vão escolher o vencedor com a ajuda do professor.

Encapsulamento é o mecanismo que protege os atributos e métodos de uma classe, restringindo o acesso a eles. Em Java, *“private”*, *“protected”* e *“public”* são palavras-chave que definem os diferentes níveis de acesso aos membros de uma classe:

- *“private”*: membros de uma classe definidos como *“private”* só podem ser acessados por métodos dentro da mesma classe. Isso significa que esses membros são encapsulados e não podem ser acessados diretamente de fora da classe.
- *“protected”*: membros de uma classe definidos como *“protected”* podem ser acessados por métodos dentro da mesma classe, bem como por subclasses da classe e por outras classes no mesmo pacote.
- *“public”*: membros de uma classe definidos como *“public”* podem ser acessados por qualquer classe em qualquer pacote, tornando-os completamente visíveis e acessíveis.

Esses níveis de acesso são importantes para garantir a segurança e a integridade do código em um sistema. Usar *“private”* para membros que não devem ser acessados de fora da classe ajuda a evitar erros de programação e limita o acesso ao código que deve ser protegido. Usar *“protected”* para membros que precisam ser acessados por subclasses ou outras classes no mesmo pacote

permite uma maior flexibilidade de uso, enquanto ainda protege o código contra acesso não autorizado. Finalmente, usar “public” para membros que precisam ser acessados por outras classes permite a interoperabilidade do código entre diferentes partes de um sistema.

O encapsulamento garante a integridade dos dados e a separação de responsabilidades entre as classes.

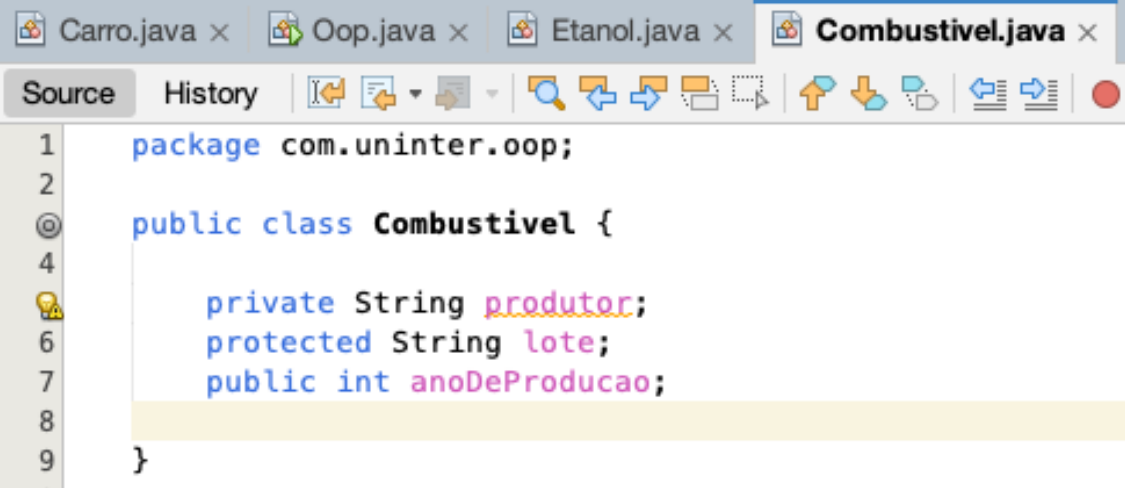
Classe Combustivel.java com demonstração de “private”, “protected” e “public”.

```
package com.uninter.oop;

public class Combustivel {

    private String produtor;
    protected String lote;
    public int anoDeProducao;

}
```



```
Carro.java x Oop.java x Etanol.java x Combustivel.java x
Source History
1 package com.uninter.oop;
2
3
4 public class Combustivel {
5
6     private String produtor;
7     protected String lote;
8     public int anoDeProducao;
9
10 }
```

“Getters” e “setters” são métodos em uma classe que permitem o acesso e a manipulação dos atributos de um objeto, garantindo o princípio do encapsulamento na Programação Orientada a Objetos (POO). Eles servem como uma interface para acessar e modificar os atributos de um objeto de maneira controlada, sem expor diretamente os atributos, protegendo assim a integridade dos dados.

“Getters:” Um “getter” é um método que retorna o valor de um atributo específico do objeto. Geralmente, o nome do método começa com “get” seguido pelo nome do atributo com a primeira letra em maiúsculo. Os métodos “getters” não recebem parâmetros e retornam o valor do atributo associado.

“Setters:” Um “setter” é um método que define o valor de um atributo específico do objeto. Geralmente, o nome do método começa com “set” seguido pelo nome do atributo com a primeira letra em maiúsculo. Os métodos “setters” recebem um parâmetro do mesmo tipo do atributo e não retornam nenhum valor.

No exemplo abaixo, produtor é “private” mas será exposto através de “set” e “get”.

Classe Combustivel.java com exemplo de “set” e “get”.

```
package com.uninter.oop;
```

```
public class Combustivel {  
  
    private String produtor;  
    protected String lote;  
    public int anoDeProducao;  
  
    public String getProdutor() {  
        return produtor;  
    }  
  
    public void setProdutor(String produtor) {  
        this.produtor = produtor;  
    }  
  
}
```

Imagem da classe Combustivel com exemplos de “set” e “get”.

```

1  package com.uninter.oop;
2
3  public class Combustivel {
4
5      private String produtor;
6      protected String lote;
7      public int anoDeProducao;
8
9      public String getProdutor() {
10         return produtor;
11     }
12
13     public void setProdutor(String produtor) {
14         this.produtor = produtor;
15     }
16
17 }

```

Atribuir pontos para a conclusão bem sucedida da tarefa e atualizar o quadro de liderança com a inclusão de um desafio extra, o desafio de encapsulamento: o aluno que for mais criativo e conseguir criar outras classes de forma a explorar o encapsulamento no fim da tarefa receberá adicionalmente uma medalha de prata e 6 pontos. Os alunos vão escolher o vencedor com a ajuda do professor.

5.2.6 Conclusão e Recompensas

Tempo: 25 minutos

Recapitular os principais pontos de aprendizagem e conceitos da lição.

Classe Oop.java demonstrando herança, polimorfismo e encapsulamento.

```
package com.uninter.oop;
```

```
public class Oop {
```

```
    public static void main(String[] args) {
```

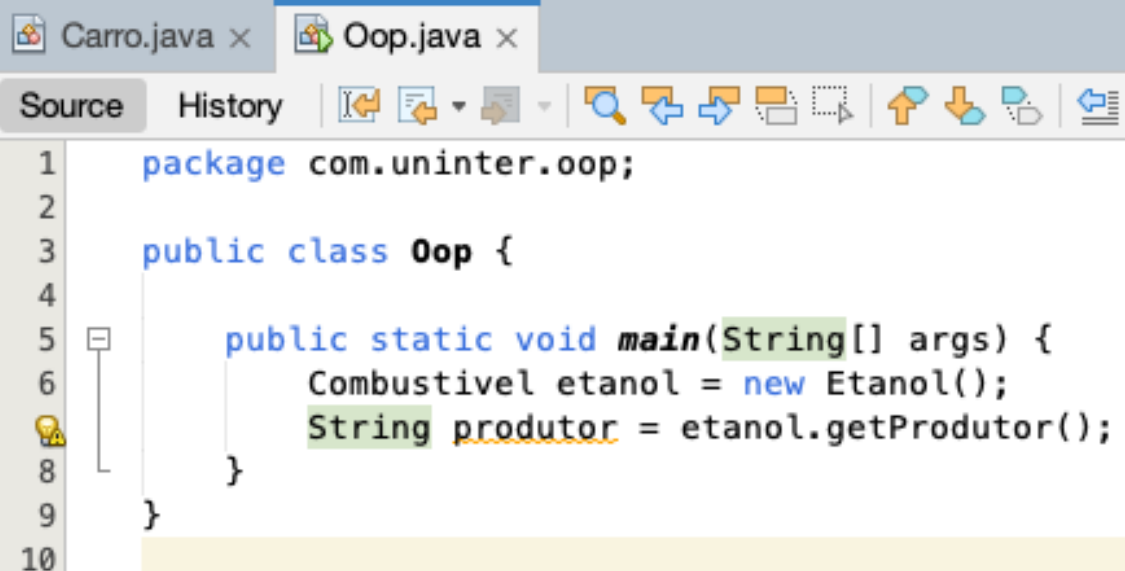
```
        Combustivel etanol = new Etanol();
```

```

        String produtor = etanol.getProdutor();
    }
}

```

Imagem da classe Oop demonstrando herança, polimorfismo e encapsulamento no NetBeans.



```

1  package com.uninter.oop;
2
3  public class Oop {
4
5      public static void main(String[] args) {
6          Combustivel etanol = new Etanol();
7          String produtor = etanol.getProdutor();
8      }
9  }
10

```

Lembre que a classe Etanol não possui propriedade nenhuma, mas ao estender a classe Combustivel, passa a ter acesso a todas as propriedades da classe. Ao instanciar a classe Etanol num objeto Combustivel demonstramos polimorfismo e ao utilizar o método *getProdutor()* demonstramos encapsulamento e herança.

Ilustra-se assim a eficácia da programação orientada a objetos, especialmente em softwares compostos por dezenas, centenas ou até mesmo milhares de classes. A manutenção e a legibilidade do código são significativamente aprimoradas por meio dessa abordagem.

Indagar os participantes da aula com relação à qualidade da aula e eventuais conhecimentos que algum aluno possa ter que seja relevante ao tema, oferecendo recompensa as menções, como por exemplo UML.

Anunciar a classificação final do quadro de líderes e distribuir troféus e medalhas de sucesso aos melhores desempenhos.

O aluno com menor pontuação ganha um troféu e 50 pontos como recompensa pela perseverança.

Oferecer recompensas a estudantes de alto nível, tais como recursos adicionais, pequenos prêmios ou pontos de bônus para futuras lições.

Os elementos de gamificação incluídos neste plano de aula são pontos, troféus e medalhas, tabelas de liderança e recompensas. Incentivar a competição amigável e a colaboração entre os estudantes, o que pode melhorar ainda mais sua experiência de aprendizagem.

Para encerrar, o esquema de pontuação padrão aplicado para todos os temas ou módulos da aula podem ser representados como orientação a objetos. A gamificação seria o primeiro objeto. A aula estende gamificação. O módulo ou tema estende a aula.

Guia rápido de recompensas da aula:

Medalha	Pontuação	Descrição da atividade
Troféu	50 pontos	No término da aula auxiliou o maior número de colegas
Troféu	50 pontos	No término da aula com menor pontuação: perseverança
Ouro	10 pontos	Concluiu todas as atividades sem ajuda
Ouro	10 pontos	Concluiu a atividade ou desafio sem ajuda
Ouro	10 pontos	No término da aula auxiliou 5 ou mais colegas
Prata	6 pontos	Concluiu a atividade ou desafio com ajuda de um colega
Prata	6 pontos	No término da aula auxiliou 3 colegas
Bronze	3 pontos	Concluiu a atividade ou desafio com ajuda do professor
Bronze	3 pontos	No término da aula auxiliou 1 colega
Bronze	3 pontos	Criou uma classe, objeto, pacote mais criativo
Bronze	3 pontos	Criou uma classe, objeto, pacote menos criativo
Bronze	3 pontos	Instalou OpenJDK primeiro
Bronze	3 pontos	Instalou Apache NetBeans primeiro

6 CONSIDERAÇÕES FINAIS

Essa pesquisa teve como objetivo geral desenvolver uma aula gamificada sobre programação orientada a objetos (POO), com base nos princípios e nas boas práticas identificadas na literatura e de profissionais da área. Adaptaram-se os elementos e mecânicas de jogos às características e necessidades dos estudantes, bem como aos objetivos pedagógicos e conteúdos a serem ensinados, resultando em um produto.

Para o desenvolvimento do estudo, estabeleceram-se os objetivos específicos: revisar a literatura sobre a gamificação na educação e sua aplicação no ensino de programação orientada a objetos, identificando as principais abordagens, elementos e mecânicas de jogos utilizadas, bem como os resultados e impactos na aprendizagem. Os resultados dessa revisão bibliográfica compõem o capítulo teórico. Para realizá-lo, foram feitas consultas às referências da área de ciências da computação quanto à linguagem Java, abordada desde a sua criação, examinando sua evolução, aplicabilidade e vantagens.

Outro assunto examinado foi referente à programação orientada a objetos, focalizando conceitos e abordagens existentes na literatura, ampliando a compreensão deste tipo de programação.

Para a realização do objetivo de desenvolver uma aula sobre POO gamificada, com base nos princípios e nas boas práticas identificadas na literatura, adaptando os elementos e mecânicas de jogos às características e necessidades dos estudantes, bem como aos objetivos pedagógicos e conteúdos a serem ensinados, foram necessários diversos estudos. A esse objetivo se associa a aplicabilidade da aula gamificada em um contexto educacional real, seja na educação formal, não-formal ou informal, focando na experiência dos estudantes, seu desempenho e motivação, e os fatores que influenciam a implementação e o sucesso da abordagem gamificada, além de identificar os fatores que contribuem para o seu sucesso.

Assim, um dos estudos foi direcionado à compreensão do ensino tanto na educação formal como na não-formal, uma vez que a programação de uma aula exige esses entendimentos. Foi realizado um estudo na perspectiva das teorias de aprendizagem, assim como a abordagem da aula adota metodologias inovadoras. A revisão de literatura também inclui esses conhecimentos para situar a gamificação no contexto dessas abordagens. Essa discussão abrangeu

ainda o próprio entendimento de gamificação com base na literatura que aborda o assunto.

Na composição da aula gamificada, para trazer elementos da prática de programação orientada a objetos, foi realizada uma investigação empírica em uma roda de conversa com profissionais da área, o que resultou em indicações que confirmam muitas das proposições indicadas na literatura.

Por fim, o objetivo foi discutir as implicações dos resultados para a prática pedagógica, as limitações do estudo e as recomendações para pesquisas futuras, fornecendo insights valiosos para educadores e pesquisadores interessados em desenvolver e implementar estratégias de ensino inovadoras e eficazes para a aprendizagem de programação orientada a objetos.

Nesse sentido, o estudo indica que a aplicação da gamificação como metodologia no ensino de programação orientada a objetos demonstra ser uma abordagem promissora. Ao incorporar elementos lúdicos e de competição no processo educacional, é possível aumentar significativamente o engajamento e a motivação dos estudantes, levando a um melhor entendimento dos conceitos fundamentais.

A combinação da gamificação com a linguagem Java permite a exploração de cenários e desafios práticos do mundo real, permitindo aos participantes da aula não só adquirir habilidades teóricas, mas também aplicá-las em contextos reais de desenvolvimento de software.

A adaptação desta abordagem para ambientes presenciais e online garante a flexibilidade necessária para atender às diversas necessidades dos estudantes e às restrições educacionais contemporâneas. Dessa forma, o ensino de orientação a objetos com gamificação como metodologia e Java como linguagem de programação torna-se uma ferramenta valiosa para melhorar a qualidade do ensino, seja por meio da educação formal, não-formal ou informal.

O mercado de trabalho internacional desempenha um papel fundamental no foco do produto. Ao considerar a programação, diversos aspectos são levados em conta, como a história da linguagem, sua evolução, o uso por grandes e renomadas multinacionais ao longo do tempo e as principais necessidades, tanto em termos operacionais quanto em desafios tecnológicos, como nuvem, escala global e a escassez de mão de obra qualificada em todo o mundo.

As empresas buscam e necessitam de profissionais com qualificações específicas, de modo que os aprendizes possam adquirir conhecimentos essenciais para se tornarem autônomos e, em seguida, facilitarem o aprofundamento em áreas específicas do conteúdo. Portanto, é crucial que os indivíduos desenvolvam habilidades e competências relevantes para se destacarem no mercado de trabalho internacional e atenderem às demandas das organizações.

Além disso, é importante salientar a necessidade de adaptação constante, pois a tecnologia e as demandas do mercado estão sempre evoluindo. Os profissionais devem manter esforços e buscar atualizações acerca das novas tendências e desenvolvimentos no campo da programação, a fim de aumentar suas chances de sucesso e atender às expectativas das empresas que procuram mão de obra qualificada.

Na aprendizagem das ciências computacionais, os profissionais precisam desenvolver habilidades e competências relevantes, adaptar-se continuamente às mudanças no campo da programação e aprofundar seu conhecimento nessas áreas específicas. Com esse conjunto de habilidades, os indivíduos estarão melhor preparados para enfrentar os desafios tecnológicos e operacionais que as organizações enfrentam atualmente, além de compreender os fundamentos dessas linguagens e seus usos.

Destaca-se que no desenvolvimento dessa investigação, na perspectiva do investigador, esses estudos permitiram aprendizagens na organização de argumentação teórica, na compreensão dos fundamentos da gamificação e na proposição de uma aula gamificada considerando os preceitos que a fundamentam.

Dentre os desafios para futuras pesquisas, segue como aplicar a gamificação no ensino de outros temas dentro da programação e engenharia de software, além da integração com outras abordagens pedagógicas: investigar a integração da gamificação com outras metodologias para proporcionar uma abordagem holística, abrangente e que, em última instância, contribua para a empregabilidade e o sucesso dos estudantes no mercado de trabalho.

7 REFERÊNCIAS

ALOMARI, M.; SHARADGAR, T. & XU, W. Gamificação para todos? Explorando os efeitos das diferenças individuais nos efeitos da gamificação. **Computers in Human Behavior**, v. 126, p. 107-119, 2022.

ALVES, M. M. & TEIXEIRA, O. Gamificação e objetos de aprendizagem: elementos da gamificação no design de objetos de aprendizagem. In: FADEL, L. M. et al. (Org.) *São Paulo: Pimenta Cultural*, 2014.

AMBRÓSIO, A. P. & COSTA, F. M. O uso de PBL para o ensino de algoritmos e programação de computadores. **PBL 2010: International Conference**. São Paulo - SP.

ARAUJO, J. Da metodologia ativa à metodologia participativa. In: Ilma Passos Alencastro Veiga. (Org.). **Metodologia participativa e as técnicas de ensino-aprendizagem**. 1 ed. Curitiba: PR: CRV, v. 1, p. 17-54, 2017

ARAUJO, J. Para uma análise das representações sobre as técnicas de ensino. In: Ilma Passos Alencastro Veiga. (Org.). **Técnicas de Ensino: por que não?**. 14 ed. Campinas, SP: Editora Papirus, p. 11-34, 2011

ARMSTRONG, D. J. Os quarks do desenvolvimento orientado a objetos. **Communications of the ACM**, v. 49, n. 2, p. 123-128, 2006.

BARDIN, L. **Análise de conteúdo**. São Paulo: Edições 70, 2011

BAYLISS, J. D. Aprendizagem baseada em jogos para ensino de programação introdutória. In **Anais do 44º Simpósio Técnico da ACM sobre Educação em Ciência da Computação (SIGCSE '13)**, pp. 499-504, 2013.

BECK, K.; SOUSA, R.; FAVERO, M. **Test driven development: teste e designno mundo real com PHP**. São Paulo: Casa do Código, 2015.

BECK, K. **Extreme Programming Explained: Abraçando a mudança**. 2. ed. Addison-Wesley, 2021.

BENDRATH, E. A. **A educação não-formal a partir dos relatórios da UNESCO**. 2014. 310 f. Tese (doutorado) - Universidade Estadual Paulista Júlio de Mesquita Filho, Faculdade de Ciências e Tecnologia, 2014. Disponível em: <http://hdl.handle.net/11449/123334>.

BEREITER, C. SCARDAMALIA, M. **Aprendizagem profunda: Aprenda a aprender e a desenvolver a inteligência**. Porto Alegre: Artmed, 2014.

BIBEAU, A. Desembaraçando código espaguete: reescrevendo aplicativos para clareza, desempenho e facilidade de manutenção. **IEEE Software**, v. 32, n. 1, p. 68-74, 2015.

BLOCH, J. & GAFTER, N. **Java efetivo**: melhores práticas para a plataforma Java. 3. ed. Rio de Janeiro: Alta Books, 2018.

BOGOST, I. **Why gamification is bullshit**. In S. P. Walz & S. Deterding (Eds.), *The gameful world: approaches, issues, applications* (pp. 65-80). Cambridge, MA: MIT Press, 2015

BOSCH, J. **Velocidade, dados e ecossistemas**: prosperando em um mundo orientado por software. London: CRC Press, 2019.

BRANDÃO, C. R. **O que é educação**. São Paulo: Brasiliense, 1981.

BRASIL. **Constituição**. Constituição da República Federativa do Brasil. Brasília, DF: Senado Federal: Centro Gráfico, 1988.

BRASIL. INEP. **Sinopse estatísticas**. MEC/INEP, 2022. Disponível em: <https://www.gov.br/inep/pt-br/areas-de-atuacao/pesquisas-estatisticas-e-indicadores/censo-da-educacao-superior>

BRASIL. **Lei de Diretrizes e Bases da Educação Nacional**. Lei n. 9.394,20 de dezembro de 1996.

BRITISH COUNCIL BRASIL. **Demandas de aprendizagem de Inglês no Brasil**. 1 ed. São Paulo, British Council. (Elaborado com exclusividade para o British Council pelo Instituto de Pesquisa Data Popular), 2014.

BUDD, T. A. **Programação orientada a objetos**: uma abordagem evolutiva. São Paulo: Pearson, 2019.

BURKE, B. **Gamificar: como a gamificação motiva as pessoas a fazerem coisas extraordinárias**. São Paulo: DVS Editora, 2015.

CAPONETTO, I.; EARP, J.; OTT, M. Gamification and education: a literature review. **Proceedings of the 8th European Conference on Games-Based Learning**, p. 50-57, 2014.

CATUNDA, M. A. & LOPES, M. P. Uma análise da ferramenta Alice no ensino da lógica de programação. In Ernane Rosa Martins. **A ciência da computação e o desenvolvimento de conteúdo tecnológico relevante para a sociedade**, Ponta Grossa, Atena, 2020.

CHEN, Y.; LANDERS, R. N. & GURSOY, D. The state of gamification in the 21st century: an update and future research agenda. **Computers in Human Behavior**, v. 116, 106639, 2021.

CHOU, Y. **Gamificação acionável: além de pontos, crachás e painéis de líderes**. Octalysis Media, 2015.

COBO, C. & MORAVEC, J.W. **Aprendizagem invisível**: como um crescente ecossistema de conhecimento além da escola está expandindo a

aprendizagem e o desenvolvimento de habilidades para o século 21. Barcelona: Laboratori de Mitjans Interactius, 2011.

COELHO, M. A. Conectivismo: uma nova teoria da aprendizagem para uma sociedade conectada. **SAPIENS - Revista de divulgação Científica**, 2019. Recuperado de <https://revista.uemg.br/index.php/sps/article/view/3433>

COOMBS, P. H. **The world educational crisis: a systems analysis**. Oxford: University Press, 1968.

CONCEIÇÃO, S. Silva da; DIAS LEAO, G. A. de A. & SCHNEIDER, H. N. Inovação curricular: processos de aprendizagem gamificados mediados pelas tecnologias digitais. **E-Curriculum**, São Paulo, v.19, n. 2, p. 564-587, jul. 2021. Disponível em http://educa.fcc.org.br/scielo.php?script=sci_arttext&pid=S1809-38762021000200564&lng=pt&nrm=iso.

CONNELL, B.R.; JONES, M.; MACE, R.; MUELLER, J.; MULLICK, A.; OSTROFF, E.; SANFORD, J.; STEINFELD, E.; STORY, M. & VANDERHEIDEN, G. **Os princípios do design universal**. Carolina do Norte: NC State University, 2021.

CORMIER, D.; VO, T. Online learning platforms: the future of Java education. **International Journal of Computer Science and Education in Computer Science**, v. 1, n. 1, p. 35-47, 2020.

COSTA, F. T. da. **Uma análise do cenário de ensino de linguagens de programação nos cursos de computação do estado do Ceará**. Mestrado em Informática Aplicada da Universidade de Fortaleza (UNIFOR), Dissertação Mestrado, 2018.

CUGELMAN, B. **Gamificação: o que é e por que é importante para desenvolvedores de mudança de comportamento em saúde digital**. JMIR Serious Games, 2023.

CUMBLEY, R.t; CHURCH, A. Game on: envolvendo clientes e funcionários por meio da gamificação. **Richmond Journal of Law and Technology**, v. 20, n. 2, 2013, p. 1-22.

CUNHA, M. I. da; ZANCHET, B. B. & RIBEIRO, G. M. Qualidade do ensino de graduação: culturas, valores e seleção de professores. **Práxis Educativa**, v. 08, n. 01, 2013, p. 219-241. Disponível em: http://educa.fcc.org.br/scielo.php?script=sci_arttext&pid=S1809-43092013000100010.

DEITEL, P. & DEITEL, H. **Java: como programar**. 10 ed. São Paulo: Pearson ~~Unicamp~~, 2016.

DETERDING, S.; DIXON, D.; KHALED, R. & NACKE, L. De elementos de design de jogos a *gamefulness*: definindo "gamificação". **Anais do 15º Congresso Acadêmico Internacional MindTrek: Visualizando Ambientes Futuros de Mídia**, ACM, 2011, p. 9-15.

DICHEV, C.& DICHEVA, D. Gamificando a educação: o que se sabe, o que se acredita e o que ainda é incerto: uma revisão crítica. **International Journal of Educational Technology in Higher Education**, 2017.

DUNN, T. J.; BAGULEY, T.& BRUNSDEN, V. From alpha to omega: a practical solution to the pervasive problem of internal consistency estimation. **British Journal of Psychology**, 2016.

ECKEL, B. **Thinking in Java**. 5 ed. Prentice Hall, 2021.

EDUCATION FIRST. EF **English Proficiency Index**. 12 ed. [S.l.]: Education First, 2022. Disponível em: <https://www.ef.co.uk/assetscdn/WIBlwq6RdJvcD9bc8RMd/cefcom-epi-site/reports/2022/ef-epi-2022-english.pdf>.

EL-SHEIKH, J. W.; COFFEY, L.; WHITE, J. **Exploring Technologies, Materials, and Methods for an Online Foundational Programming Course**, 2008. Disponível em https://www.academia.edu/22493708/Exploring_Technologies_Materials_and_Methods_for_an_Online_Foundational_Programming_Course.

FELLEISEN, M.; FINDLER, R. B.; FLATT, M. & KRISHNAMURTHI, S. A estrutura e interpretação do currículo de ciência da computação. **Journal of Functional Programming**, v. 18, n. 6, p. 655-698, 2008.

FERRI, G.; D'ERRICO, F.; D'AMICO, R. Ethical reflections on gamification: a systematic literature review. **Behaviour & Information Technology**, v. 39, n. 10, p. 1041-1057, 2020.

FOGG, B. J. **Hábitos minúsculos**: as pequenas mudanças que mudam tudo. Penguin, 2021.

FOWLER, M. **Refatoração**: aperfeiçoando o design de códigos existentes. 2. ed. São Paulo: Novatec, 2022.

FREIRE, P. **Direitos humanos e educação libertadora**: gestão democrática da educação pública na cidade de São Paulo. São Paulo: Paz e Terra, 2019.

FREIRE, P. **Pedagogia da autonomia**: saberes necessários à prática educativa. São Paulo: Paz e Terra, 1996.

FREIRE, P. **Pedagogia do oprimido**. Rio de Janeiro: Paz e Terra, 1987.

FREIRE, P. SHOR, I. **Medo e ousadia**: o cotidiano do professor. 13. ed. Rio de Janeiro: Paz e Terra, 2011.

FURGERI, S. **Java 8 ensino didático**: desenvolvimento e implementação de aplicações. São Paulo: Saraiva. 2018.

GIUSTA, A. da S. Concepções de aprendizagem e práticas pedagógicas. **Educação em Revista**, Belo Horizonte, v. 29, n. 01, p. 17-36, mar. 2013. Disponível em: <https://www.scielo.br/j/edur/a/y9JvZV8HZRFN3XtvJ8vf9Rk/?lang=pt>

GOHN, M. da G. **Educação não-formal e cultura política**: impactos sobre associativismo no Terceiro Setor. 5.ed. São Paulo: Cortez, 2006.

GOLDBERG, A. & ROBSON, D. **Smalltalk-80**: the language and its implementation. Massachusetts: Addison-Wesley, 1983.

GOODRICH, M. T.; TAMASSIA, R. & GOLDWASSER, M. H. **Estruturas de dados e algoritmos em Python**. 2. ed. Rio de Janeiro: LTC, 2021

GOSLING, J.; SHERIDAN, M. & NAUGHTON, P. **A história e evolução da linguagem Java**. 1. ed. [S. I.]: Editora Exemplo, 2021.

GUIMARÃES, P. & FARIA-FORTECOËF, C. A complementaridade entre educação (formal, não-formal e informal) e (auto, hetero e eco) formação: uma discussão a partir de autobiografias. **Educação**, v. 46, e32/1–22, 2021. Disponível em <https://doi.org/10.5902/1984644461311>.

GUZDIAL, M. & ERICSON, B. **Computer science in K-12**: an a-to-z handbook on teaching programming. Stylus Publishing, LLC., 2021.

HADJERROUIT, S. Towards a blended learning model for teaching and learning computer programming: a case study. **Informatics in Education**, v. 7, n. 2, p. 181-210, 2008.

HAMARI, J.; KOIVISTO, J. & SARSA, H. A gamificação funciona? uma revisão da literatura de estudos empíricos sobre gamificação, **47th Hawaii International Conference on System Sciences**, 2014.

HEW, K. F.; LOH, W.Y. & CHENG, K. W. Melhorando o engajamento na aprendizagem on-line: o que os alunos querem? **Journal of Educational Technology & Society**, v. 19, n. 3, p. 142-152, 2016.

HODGES, C.; MOORE, S.; LOCKEE, B.; TRUST, T. & BOND, A. The difference between emergency remote teaching and online learning. **Educase Review**. Mar. 2020. Disponível em: <https://er.educause.edu/articles/2020/3/the-difference-between-emergency-remote-teaching-and-online-learning>.

HORN, M. B. & TAKER, H. **Blended**: usando a inovação disruptiva para aprimorar a educação. Porto Alegre: Penso, 2015.

HORSTMANN, C. S. **Conceitos de computação com Java**: uma abordagem de aprendizagem baseada em objetos. Rio de Janeiro: LTC, 2013.

HORSTMANN, C. S. **Java SE8 para os realmente impacientes**: um curso rápido sobre o básico. Massachusetts: Addison-Wesley Professional, 2016.

HUNICKE, R.; LEBLANC, M. & ZUBEK, R. MDA: Uma abordagem formal para o design de jogos e pesquisa de jogos. In: **Proceedings of the AAAI Workshop on Challenges in Game AI**, v. 4, p. 1-5, 2004.

IBANEZ, M. B.; DI-SERIO, A. & DELGADO-KLOS, C. Gamificação para envolver estudantes de ciência da computação em atividades de aprendizagem: um estudo de caso. **IEEE Transactions on Learning Technologies**, v.7, n.3, p. 291-301, 2014.

ILLERIS, K. Uma compreensão abrangente sobre a aprendizagem humana. **Teorias contemporâneas da aprendizagem**. Porto Alegre: Penso, 2013.

IVANOVIĆ, M.; BUDIMAC, Z.; MISHEV, A.; BOTHE, K. & JURCA, I. Java Across Different Curricula, Courses and Countries Using a Common Pool of Teaching Material. **Informatics in Education**, v. 12, n. 2, p. 153–179, 2013.

KAPP, K. **The gamification of learning and instruction: game-based methods and strategies for training and education**. San Francisco: Pfeiffer, 2012.

KIILI, K.; DE FREITAS, S.; ARNAB, S. & LAINEMA, T. **The design, experience And practice of networked learning**. Springer, 2014.

KOIVISTO, J.; NIEMINEN, S. & MULTISILTA, J. Uma taxonomia de elementos de gamificação para variar objetivos pedagógicos. In: **Proceedings of the 13th european conference on game-based learning** (pp. 304-312). Academic Conferences International Limited, 2021.

LANDERS, R. N.; BAILEY, J.; CALLAN, R. C.; DASTECHCHI, A. & SHI, X. Gamificação da aprendizagem desativa a rede de modo padrão. **Frontiers in Psychology**, v. 9, p. 1-10, 2018.

LEITINHO, M. & CARNEIRO, C. Aprendizagem baseada em problemas: uma abordagem pedagógica e curricular. In Ilma Veiga. **Novas tramas para as técnicas de ensino e estudo**. Campinas: Papirus, 2013.

LEUTENEGGER, S. Uma abordagem de jogos para o ensino de programação introdutória. In **Anais do 38º Simpósio Técnico da ACM sobre Educação em Ciência da Computação (SIGCSE '07)**, pp. 115-118. 2007.

LIANG, X.; ZHENG, X.; HUANG, L. Explorando o impacto das estratégias de marketing game-like nas intenções de compra do cliente. **Journal of Business Research**, v. 144, p. 324-335, 2022.

LOPES, R. M. S. & BITTENCOURT, I. I. Design of a gamified environment to foster learner engagement. **International Journal of Human-Computer Studies**, v. 127, p. 23-37, 2019.

LOUDEN, K. C. & LAMBERT, K. A. **Programming languages: principles and practices**. Course Technology, Cengage Learning. Third Edition, 2012.

LUXTON-REILLY, A. et al. Introdução à edição especial sobre educação em ciência da computação. In: **Educação em Ciência da Computação**, v. 28, n. 2, p. 99-100, 2018. DOI: 10.1080/08993408.2018.1509170.

MANZANO, J. A. N. G. & OLIVEIRA, J. F. de; **Algoritmos: lógica para desenvolvimento de programação de computadores**. Ed. 28. São Paulo, Editora Érica Saraiva, 2016.

MARCIAL, E. C. **Megatendências mundiais 2030: o que entidades e personalidades internacionais pensam sobre o futuro do mundo?** Contribuição para um debate de longo prazo ~~no~~ Brasil. Brasília: Ipea, 2015.

MARCZEWSKI, A. **Gamification: a simple introduction & a bit more**. Kindle Edition, 2013.

MARTIN, R. C. **Arquitetura limpa: um guia do artesanato para estrutura e design de software**. Prentice Hall, 2018.

MATTAR, J. Aprendizagem em ambientes virtuais: teorias, conectivismo e MOOCs. **Revista Digital de Tecnologias Cognitivas**, n.7, p. 20-40, 2013.

MATTAR, J. (org.). **Relatos de pesquisas em aprendizagem baseada em games** 1. ed. São Paulo: Artesanato Educacional, 2020.

MEDEIROS, L. F. de. A construção de um ideal tecnocientífico. **Perspectivas em Gestão & Conhecimento**, João Pessoa, v. 2, n. 2, p. 43-56, jul./dez. 2012. Disponível em <http://www.pesquisa.ufrpb.edu.br/revista/revista.php?revista=1&numero=2&pagina=43>

MEDINA, M. & FERTIG, Cristina. **Algoritmos e programação: teoria e prática**. São Paulo, Novatec Editora, 2005.

MEKLER, E. D.; BRÜHLMANN, F.; OPWIS, K. & TUCH, A. N. Disassembling gamification: the effects of points and meaning on user motivation and performance. In: CHI '13 **Extended Abstracts on Human Factors in Computing Systems**, 2013.

MITTELSTADT, B. The ethics of gamification: moral and social values in the design of gamified systems. **Ethics and information technology**, v. 23, n. 2, p. 117-130, 2021.

MORA, A.; RIERA, D.; GONZÁLEZ, C. & ARNEDO-MORENO, J. A literature review of gamification design frameworks. **Games and culture**, v. 14, n. 5, p. 491-519, 2019.

MORAN, J. Mudando a educação com metodologias ativas. In: **Convergências midiáticas, educação e cidadania: aproximações jovens**. 2015. Disponível em: http://www2.eca.usp.br/moran/wp-content/uploads/2013/12/mudando_moran.pdf.

MOREIRA, M. A. **Mapas conceituais e aprendizagem significativa**. Porto Alegre - RS, UFRGS Brasil, 2012. Disponível em: <http://moreira.if.ufrgs.br>

MORSCHHEUSER, B.; ZAGAR, M.; DICK, M.; DE LUISE, Daniele & TARI, A. R. Projetando gamificação inteligente para saúde conectada. **Journal of Medical Systems**, v. 46, n. 1, p. 1-10, 2022.

MOURA, A. B. F.; LIMA, M. DA G. S. B. A reinvenção da roda: roda de conversa, um instrumento metodológico possível. **Interfaces da Educação**, v. 5 n. 15, p.

24–35, 2015. Recuperado de <https://periodicosonline.uems.br/index.php/interfaces/article/view/448>.

NASCIMENTO, E. R.; BRITO, I. P. L. de & PADILHA, M. A. S. Engajamento de docentes na educação superior: implementando ensino híbrido. **Revista e-Curriculum**, v.18, n.2, p. 951-969, abr./jun. 2020.

NIEMIEC, C. P. & RYAN, R. M. Autonomia, competência e relacionamento na sala de aula: aplicando a teoria da autodeterminação à prática educacional. **Theory and Research in Education**, v. 19, n. 2, p. 255-280, 2021.

ODERSKY, M.; SPOON, L. & VENNERS, B. **Programming in scala: a comprehensive step-by-step guide**. walnut creek: Artima, 2020.

PALHARES, J. A. Reflexões sobre o não-escolar na escola e para além dela. **Revista Portuguesa de Educação**, v. 22, n. 2, p. 53-84, 2009.

PAPASTERGIOU, M. (2009). Aprendizagem baseada em jogos digitais no ensino de ciência da computação no ensino médio: impacto na eficácia educacional e motivação do aluno. **Computers & Education**, v. 52, n. 1, p. 1-12, 2009.

PEROZA, J. **Provocações antecipatórias ou a esperança como inédito-viável: a contribuição do pensamento utópico de Paulo Freire para a Formação de Professores**. 2014. Tese de doutorado. Pontifícia Universidade Católica do Paraná, Curitiba, 2014.

PRESTES, Z. R. **Quando não é a mesma coisa: análise de traduções de Lev Semionovitch Vigotski no Brasil: repercussões no campo educacional**. 2010. 295 f. Tese (Doutorado em Educação) Universidade de Brasília, Brasília, 2010.

REIS, S. M. A. de O. Paulo Freire: 100 anos de práxis libertadora. **Práx. Educ.**, Vitória da Conquista, v. 17, n. 47, p. 238-258, ago.2021. Disponível em http://educa.fcc.org.br/scielo.php?script=sci_arttext&pid=S2178-26792021000400238&lng=pt&nrm=iso.

RICHTER, C. J. **Ensino de programação orientada a objetos na educação profissional por meio do desenvolvimento de jogos apoiado pelo ambiente Greenfoot**. Santa Maria, UFMS, 2018. Dissertação de mestrado. Disponível em: https://repositorio.ufsm.br/bitstream/handle/1/20661/DIS_PPGTER_2019_RICHTER_CLEITOM.pdf?sequence=1&isAllowed=y

ROBSON, K.; PLANGGER, K.; KAVANAUGH, S. S.; SUTHERLAND, W. & GUNNARSSON, J. Is it all a game? Understanding the principles of gamification. **Business Horizons**, v. 58, n. 4, p. 411-420, 2015.

RYAN, R. M.; DECI, E. L. Teoria da autodeterminação e a facilitação da motivação intrínseca, desenvolvimento social e bem-estar. **American Psychologist**, v. 55, n. 1, p. 68-78, 2000.

RYAN, R. M.; DECI, E. L. **Self-determination theory: basic psychological needs in motivation, development, and wellness**. Guilford Publications, 2017.

SAILER, M.; HENSE, J. U.; MAYR, S. K.; MANDL, H. How gamification

motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. **Computers in Human Behavior**, v. 69, p. 371-380, 2017.

SAILER, M.; HENSE, J. U.; MANDL, H & KLEVERS, M. Perspectivas psicológicas sobre a motivação através da gamificação. **I-Com: Zeitschrift für interaktive und kooperative Medien**, v. 12, n. 1, p. 17-27, 2023.

SAINI, H.& KAUR, R. **Software engineering: concepts and practices**. CRC Press, 2021.

SANTOS, R. O. dos; MOSER, A. & LIMA, T. **Hipertexto como mediador pedagógico**. Disponível em: [https://www.academia.edu/85771254/Hipertexto como mediador pedagógico](https://www.academia.edu/85771254/Hipertexto_como_mediador_pedag%C3%B3gico)

SARDI, L.; IDRI, A.; FERNÁNDEZ-ALERGUIA, M. J. Uma revisão sistemática da gamificação na e-Health. **Journal of Biomedical Informatics**, v.113, p. 103591, 2021.

SEABORN, K.& FELS, D. I. Gamificação em teoria e ação: uma pesquisa. **International Journal of Human-Computer Studies**, v. 74, p. 14-31,2015.

SEBESTA, W. R. **Conceitos de linguagens de programação**. 11 ed. Porto Alegre: Editora Bookman, 2018.

SEVERINO, A. J. Educação, trabalho e cidadania: a educação brasileira e o desafio da formação humana no atual cenário histórico. **Perspectiva**, v. 14, n.02, p. 65-71, 2001.

SEVERINO, A. J. **Metodologia do trabalho científico**. 24 ed. SãoPaulo: Cortez, 2016.

SHARAN, K. **Expert Java: Master the art of object-oriented programming and build complex applications using Java**. Packt Publishing Ltd., 2021.

SHELLY, G. B.; VERMAAT, M. E. & QUASNEY, R. **Descobrimos a computação: fundamentos e prática**. São Paulo: Cengage Learning, 2013.

SIEMENS, G. **Connectivism: a learning theory for the digital age**. 2004. Disponível em <https://lidtfoundations.pressbooks.com/chapter/connectivism-a-learning-theory-for-the-digital-age/>.

SKIENA, S. **Manual de projeto de algoritmos**. 3. ed. São Paulo: Blucher, 2020.

SOBREIRO, J. A. P. **Proposta de desenvolvimento de instrumento de aplicação de atividades gamificadas para disciplinas do ensino superior**. Mestrado Profissional em Educação e Novas Tecnologias do Centro Universitário Internacional (UNINTER), Dissertação Mestrado, 2017.

SOUZA, D. M.; BATISTA, M. H. da S. & BARBOSA, E. F. Problemas e dificuldades no ensino e na aprendizagem de programação: um mapeamento sistemático. **Revista Brasileira de Informática Educação**, v.24, n.1, p. 39-52, 2016.

STEINFELD, E. & MAISEL, J. **Universal design**: creating inclusive environments. Hoboken, NJ: John Wiley & Sons, 2012.

TRAVASSOS SARINHO, V. **Uma biblioteca de componentes semânticos para especificação de linguagens de programação**. Dissertação (Mestrado). Programa de Pós-Graduação em Ciência da Computação. Universidade Federal de Pernambuco, Recife, 2003.

UNESCO. **Recomendação sobre o desenvolvimento da educação de adultos**. Aprovada pela Conferência Geral da UNESCO na sua décima nona reunião. Nairobi, 26 de novembro de 1976. Braga: Universidade do Minho/Projeto de Educação de Adultos, 1977.

VEIGA, I. P. A. (org.). **Aula**: gênese, dimensões, princípios e práticas. Campinas: Papirus, 2012.

VEIGA, I. P. A. (org.). **Novas tramas para as técnicas de ensino e estudo**. Campinas: Papirus, 2013.

VEIGA, I. P. A. (org.). **Metodologia participativa e as técnicas de ensino-aprendizagem**. Curitiba: CRV, 2017.

VEIGA, I. P. A. & FERNANDES, R. C. de A. (org.). **Por uma didática da educação superior**. 1. ed. Campinas, SP: Autores Associados, 2020.

VIEIRA PINTO, A. **Sete lições sobre educação de adultos**. 8. Ed. São Paulo: Cortez Editora, 1993.

WASSERMAN, A. I. Questões de engenharia de software para desenvolvimento de aplicativos móveis. In: **Anais do Workshop FSE/SDP sobre Futuro da Pesquisa em Engenharia de Software**. p. 397-400. ACM, 2013.

WELLER, W. Grupos de discussão aportes teórico-metodológicos. In WELLER, W.; PFAFF. **Metodologias da pesquisa qualitativa** em educação: teoria e prática. Petrópolis/RJ: Vozes, 2010.

WILDEMEERSCH, D. & STROOBANTS, V. Aprendizagem transicional e facilitação reflexiva. O caso da aprendizagem para o trabalho. In ILLERIS, Knud et al. Uma compreensão abrangente sobre a aprendizagem humana. **Teorias contemporâneas da aprendizagem**. Porto Alegre: Penso, 2013.

ZANETTI, H. & BONACIN, R. Uma metodologia baseada em semiótica para elaboração e análise de práticas de programação com robótica pedagógica. **XXV Simpósio Brasileiro de Informática na Educação**, 2014.

ZICHERMANN, G. & CUNNINGHAM, C. **Gamification by design**. New York: O'Reilly, 2011.

ANEXO I - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO
(para maiores de 18 anos)

Nós, Dr. Rodrigo Otávio dos Santos e Gilberto Antonio Müller Sobrinho, do Centro Universitário Internacional Uninter, convidamos você profissional da área de engenharia de a participar de uma pesquisa intitulada “**Aula Gamificada de Programação Orientada a Objetos**”.

- a) O objetivo desta pesquisa é: desenvolver uma aula gamificada de orientação à objetos na linguagem de programação Java.
- b) Para sua participação nesta pesquisa, será necessário participar de uma roda de conversa online, na plataforma Microsoft Teams.
- c) Para tanto você não precisará deslocar-se de sua residência, basta estar presente no dia e horário combinado para a roda de conversa que terá a duração de aproximadamente 90 minutos.
- d) Ao participar desta pesquisa, poderá ocorrer algum desconforto, principalmente ao tempo de duração da atividade.
- e) Alguns riscos relacionados ao estudo podem ser os eventuais constrangimentos de expor suas opiniões em grupo sobre Java. Nenhuma questão é de caráter pessoal.
- f) Os pesquisadoras, responsáveis por este estudo poderão ser localizadas por meio do endereço eletrônico: Dr. Rodrigo Otávio dos Santos e Gilberto Antonio Müller Sobrinho, gilberto.mueller@outlook.com, para esclarecer eventuais dúvidas que você como participante possa ter e fornecer-lhe as informações que desejar, antes, durante ou depois de encerrado o estudo.
- g) A sua participação neste estudo é voluntária e se não quiser mais fazer parte da pesquisa poderá desistir a qualquer momento e solicitar que lhe devolvam este Termo de Consentimento Livre e Esclarecido assinado.
- h) As informações relacionadas ao estudo poderão ser conhecidas por pessoas autorizadas. No entanto, se qualquer informação for divulgada em relatório ou publicação, isto será feito sob forma codificada, para que **a sua identidade seja preservada e mantida sua confidencialidade.**
- i) O material obtido através da roda de conversa será utilizado unicamente para essa pesquisa e será destruído ao término do estudo, dentro de 1 ano.

- j) Quando os resultados forem publicados, não aparecerá seu nome, e sim um código.
- k) As despesas necessárias para a realização da pesquisa são de sua responsabilidade e o participante não receberá qualquer valor em dinheiro pela sua participação.

Eu, _____, li esse Termo de Consentimento e compreendi a natureza e objetivo do estudo do qual concordei em participar. A explicação que recebi menciona os riscos e benefícios. Eu entendi que sou livre para interromper minha participação a qualquer momento sem justificar minha decisão e sem qualquer prejuízo para mim. Eu concordo voluntariamente em participar deste estudo.

Curitiba/PR, 30 de novembro de 2022.

Nome completo e assinatura do participante da pesquisa

Gilberto Antonio Müller Sobrinho
Pesquisador