



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Câmpus de Presidente Prudente

Alexandre Antonio Gonçalves

**Uso de metodologias ativas para o ensino de programação
orientada a objetos**

Presidente Prudente
2022

Alexandre Antonio Gonçalves

**Uso de metodologias ativas para o ensino de programação
orientada a objetos**

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de Presidente Prudente.

Orientador: Profº. Dr. Danilo Medeiros Eler
Coorientador: Profº. Dr. Maurício Araújo Dias

Presidente Prudente
2022

G635u	Gonçalves, Alexandre Antonio Uso de metodologias ativas para o ensino de programação orientada a objetos / Alexandre Antonio Gonçalves. -- Presidente Prudente, 2022 98 p. Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências e Tecnologia, Presidente Prudente Orientador: Danilo Medeiros Eler Coorientador: Maurício Araújo Dias 1. Programação orientada a objetos (Computação). 2. Metodologia de ensino. 3. Aprendizagem baseada em problemas. 4. Software educacional. I. Título.
-------	---

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Ciências e Tecnologia, Presidente Prudente. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Uso de metodologias ativas para o ensino de programação orientada a objetos

AUTOR: ALEXANDRE ANTONIO GONÇALVES

ORIENTADOR: DANILO MEDEIROS ELER

COORIENTADOR: MAURÍCIO ARAÚJO DIAS

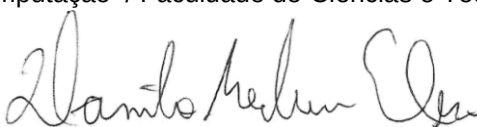
Aprovado como parte das exigências para obtenção do Título de Mestre em Ciência da Computação, área:
Computação Aplicada pela Comissão Examinadora:

Prof. Dr. DANILO MEDEIROS ELER (Participação Virtual)
Departamento de Matemática e Computação / Faculdade de Ciências e Tecnologia de Presidente Prudente

Prof. Dr. JORGE MARQUES PRATES (Participação Virtual) Universidade
Estadual de Mato Grosso do Sul

Prof. Dr. CELSO OLIVETE JUNIOR (Participação Virtual)
Departamento de Matemática, Estatística e Computação / Faculdade de Ciências e Tecnologia de Presidente Prudente

Presidente Prudente, 10 de março de 2022



Dedico este trabalho para minha esposa Viviane, meus filhos Luiza e Pedro, aos meus pais Aurélio e Maribel, ao meu avô Aurélio (*in memoriam*) e a todos que acreditaram que eu seria capaz de atingir meus objetivos.

AGRADECIMENTOS

Agradeço a Deus pela vida.

À minha esposa Viviane e meus filhos Luiza e Pedro pela paciência e ausência de minha presença e atenção, mas que sempre me apoiaram.

Aos meus pais Aurélio e Maribel que torceram e acreditaram em mim.

Ao meu orientador Danilo que teve paciência magistral comigo e sempre esteve me apoiando e auxiliando no desenvolvimento do trabalho com ideias e conhecimentos formidáveis.

Ao meu coorientador Maurício que com seus apontamentos e sabedoria me norteou e enriqueceu meus conhecimentos (entendimentos e aprendizados), me proporcionando novos horizontes.

RESUMO

Docentes e discentes devem possuir atitudes de corresponsabilidade e autogestão no processo de aprendizagem. Assim, instituições de ensino superior devem investir em estudos que aprimorem os métodos didáticos, tornando a aprendizagem algo significativo, estando num movimento constante de “aprender a aprender”. Buscando contribuir com essa necessidade do ensino superior, foi realizada uma revisão sistemática sobre as metodologias de aprendizagem que visam melhorar o processo de aprendizagem de programação orientada a objetos. Baseado nesses documentos e resultados, este trabalho aborda o processo de aprendizagem baseado na Taxonomia de Bloom, algumas metodologias ativas e apresenta um projeto de trabalho para a disciplina de Programação Orientada a Objetos, utilizando metodologias de aprendizagem ativa, com foco principal na aprendizagem baseada em problemas (ABP) e o RoboCode como ferramenta de apoio ao aprendizado. Com isso, foi possível constatar que a elaboração de materiais adequados, com exemplos e atividades baseados nos problemas recorrentes na rotina diária, trabalhos em equipe e o estímulo através da competição entre os alunos, aumenta o aprendizado dos conteúdos abordados, mantém a motivação pelos estudos e a busca de novos conhecimentos. Assim, este trabalho contribui com pesquisadores que buscam respostas para alcançarem melhores resultados no ensino de programação orientada a objetos.

Palavras-chave: Metodologia ativa. Programação orientada a objetos. RoboCode.

ABSTRACT

Teachers and students must have attitudes of co-responsibility and self-management in the learning process. Thus, higher education institutions should invest in studies that improve didactic methods, making learning something meaningful, being in a constant movement of “learning to learn”. Seeking to contribute to this need for higher education, a systematic review was carried out on the learning methodologies that aim to improve the object-oriented programming learning process. Based on these documents and results, this work addresses the learning process based on Bloom's Taxonomy, some active methodologies and presents a work project for the Object Oriented Programming discipline, using active learning methodologies, with a main focus on learning based on problems (ABP) and RoboCode as a learning support tool. With this, it was possible to verify that the elaboration of adequate materials, with examples and activities based on the recurring problems in the daily routine, teamwork and the stimulus through competition among students, increases the learning of the covered contents, maintains the motivation for studies. and the search for new knowledge. Thus, this work contributes to researchers who seek answers to achieve better results in teaching object-oriented programming.

Keywords: Active methodology. Object-oriented programming. RoboCode.

LISTA DE ILUSTRAÇÕES

Figura 2.1: Aprendizagem receptiva x por descoberta (NOVAK, 1984)	22
Figura 2.2: Categorias do domínio cognitivo (FERRAZ e BELHOT, 2010).	24
Quadro 2.1: Divisões dos níveis da taxonomia de Bloom – reproduzido de (FERRAZ e BELHOT, 2010).....	24
Figura 2.3: Categorização atual da Taxonomia de Bloom proposta por Anderson (FERRAZ e BELHOT, 2010).	25
Quadro 2.2. Estrutura do processo cognitivo na taxonomia de Bloom (revisada) – reproduzida de (FERRAZ e BELHOT, 2010).....	25
Figura 2.4: Modelo de sala de aula invertida (COSTA et al., 2017).	34
Figura 3.1: Materiais disponibilizados para a disciplina de POO no Classroom (Autor, 2021).	38
Figura 3.2: Materiais disponibilizados para a disciplina de POO no github (Autor, 2021).	39
Figura 3.3: RoboCode (ROBOCODE, 2021).....	40
Figura 3.4: Campo de batalha (Autor, 2021).	41
Figura 3.5: Etapas das atividades do RoboCode (Autor, 2021).	43
Figura 3.6: Codificando um robô (Autor, 2021).	44
Figura 3.7: Criação de uma batalha (Autor, 2021).	44

LISTA DE TABELAS

Tabela 3.1: Resultado quantitativo das respostas do primeiro questionário (Autor, 2021).	47
Tabela 3.2: Resultado quantitativo das respostas do segundo questionário (Autor, 2021).	49
Tabela 3.3: Comparativo entre as respostas do primeiro e segundo questionário (Autor, 2021).	50

LISTA DE ABREVIATURAS E SIGLAS

ABP	Aprendizagem Baseada em Problemas
IBM	International Business Machines Corporation
JRE	Java Runtime Environment
OO	Orientação a Objetos
PBL	Problem Based Learning
POO	Programação Orientada a Objetos
IDE	Integrated Development Environment

SUMÁRIO

1. INTRODUÇÃO	11
1.1. Motivação	12
1.2. Objetivo	13
2. FUNDAMENTAÇÃO	15
2.1. Programação orientada a objetos	15
2.1.1. Dificuldades em aprender POO	18
2.2. Processo de aprendizagem	19
2.3. Métodos e metodologias de aprendizagem	26
2.4. Metodologias ativas	27
2.4.1. Aprendizagem baseada em problemas (ABP)	29
2.4.2. Aprendizagem baseada em projetos	31
2.4.3. Aprendizagem entre pares	32
2.4.4. Aprendizagem colaborativa	32
2.4.5. Sala de aula invertida	34
2.4.6. Gamificação	35
2.5. Considerações finais	36
3. METODOLOGIA PARA APOIAR O ENSINO DE POO	37
3.1. Materiais	37
3.1.1. Aulas	37
3.1.2. RoboCode	40
3.1.3. Questionários	41
3.1.4. Participantes	42
3.2. Procedimentos	42
4. RESULTADOS	46
4.1. Aplicação da metodologia	46
4.2. Análise do projeto	47
5. CONCLUSÕES E TRABALHOS FUTUROS	53
5.1. Conclusões Gerais	53
5.2. Trabalhos futuros	55
REFERÊNCIAS	57
APÊNDICE A – REVISÃO SISTEMÁTICA	62
APÊNDICE B – TRABALHOS RELACIONADOS	68

1. INTRODUÇÃO

A programação orientada a objetos (POO) tornou-se, nos últimos anos, um paradigma de programação muito influente. A POO é amplamente utilizada na educação e na indústria, e quase todas as universidades ensinam orientação a objetos em seus currículos (KÖLLING, 1999).

Conforme KÖLLING (1999), a comunidade de software apoia a ideia de ensinar programação orientada a objetos, pois, suporta de forma elegante os conceitos que são ensinados por muitos anos, como programação bem estruturada, modularização e design de programas. Também apoia técnicas de abordagem de problemas que só mais recentemente fizeram parte do currículo: programação em equipes, manutenção de grandes sistemas e reutilização de software. Resumidamente, a programação orientada a objetos parece ser uma boa ferramenta para ensinar as metodologias de programação consideradas importantes. No entanto, o ensino de programação orientada a objetos apresenta questionamentos como: “Porque é tão difícil ensinar POO?” ou “Porque parece ser mais difícil ensinar POO do que programação procedimental?”. Por isso, na tentativa de melhorar o ensino de POO, muitas técnicas e metodologias de aprendizagem tem sido testadas para conseguirem atingir esse objetivo e um tema que está em grande repercussão no momento são as metodologias ativas.

Segundo JUNHUA et al. (2013), ocorreram avanços de significados expressivos no desenvolvimento de abordagens de aprendizado ativo em diversas áreas. Esses esforços buscaram a transformação do conteúdo de ambientes de salas de aulas tradicionais e passivos para modelos em que os alunos assumem um papel muito mais dinâmico e participativo. Por exemplo, com a aprendizagem baseada em projetos (ABP) os alunos aprendem em um ambiente com apoio de aplicativos, normalmente em equipes, direcionado por problemas em aberto, não existindo somente uma resposta correta. A abordagem de aprendizado ativo ajuda o aluno a se envolver em atividades e discussões, podendo proporcionar uma experiência de aprendizado potencialmente enriquecedora e contribuir para reduzir evasão, reprova e desmotivação.

1.1. Motivação

Algumas disciplinas apresentam elevados índices de evasão e reprova, dificultando ou impedindo que os alunos continuem o curso. Um exemplo é a disciplina de POO. Segundo RIBEIRO et al. (2018), dentre possíveis fatores que contribuem para esta situação, pode-se destacar:

- i) baixa capacidade de abstrair conceitos e raciocínio lógico para desenvolver algoritmos;
- ii) desmotivação do estudante, que, na maioria das vezes, encara a disciplina como um grande obstáculo a ser superado; e
- iii) metodologia de ensino, que não desperta o interesse do estudante.

Nos primeiros dias de aula, alunos dos cursos de Ciência da Computação e áreas afins, normalmente, estão com grande entusiasmo, mas com o decorrer das atividades, vão perdendo o interesse e a motivação (JUNHUA et al., 2013). A complexidade dos conceitos, linguagens e ferramentas que precisam ser entendidos num período curto, são destacadas como os maiores problemas encontrados pelos alunos, agravando-se ainda mais com as metodologias de aprendizagem que são centralizadas no professor (RIBEIRO, BITTENCOURT e SANTANA, 2018).

A programação é uma habilidade básica que todos os alunos de Ciência da Computação precisam desenvolver. Porém, geralmente, as disciplinas de programação são consideradas difíceis e tendem a apresentar as maiores taxas de abandono e reprova. Segundo DIJKSTRA (1982 apud OLIVEIRA et al., 2017), durante o processo de ensino e de aprendizagem de programação, muitos estudantes apresentam dificuldades em raciocinar e assimilar as abstrações necessárias na criação e compreensão dos algoritmos.

Neste contexto, os problemas informados com o raciocínio lógico, que são fundamentais para programar, tem como consequência a desmotivação, podendo ser a causa decisiva para as reprovações e evasões (RODRIGUES, 2002 apud OLIVEIRA et al., 2017). Por isso, a importância em buscar métodos que despertem o interesse e envolvam os alunos de forma consistente tem se tornado cada vez maior, a fim de aprofundar os fundamentos, habilidades técnicas, habilidades de inovação e habilidades de aprendizado. Abordagem de aprendizado ativo ajuda o

aluno a se envolver em atividades e discussões, podendo proporcionar uma experiência de aprendizado potencialmente enriquecedora (JUNHUA et al., 2013).

Existem algumas disciplinas na área computacional que apresentam muitas dificuldades de ensino, porém, destacaremos a programação orientada a objetos (POO) que se tornou, nos últimos anos, o paradigma de programação mais influente na área da computação e na indústria.

1.2. Objetivo

O principal objetivo deste trabalho é colaborar com o processo de ensino de POO por meio de uso de técnicas e metodologias de aprendizagem eficazes, que sejam capazes de aumentar o aprendizado. Também, este trabalho observa como o uso dessas metodologias colaboram com o desenvolvimento de habilidades técnicas e de autoavaliação, bem como no engajamento do estudante. Assim, responder às seguintes questões de pesquisa:

- Quais estratégias pedagógicas poderiam ser utilizadas para desenvolver habilidades técnicas e de autoavaliação nos alunos de POO?
- Quais estratégias pedagógicas poderiam ser utilizadas para contribuir com o engajamento (motivação) dos estudantes de POO?

Assim, este trabalho poderá contribuir com a área acadêmica que constantemente busca melhores resultados no processo ensino aprendizagem, principalmente para os cursos que ministram a disciplina de POO.

O restante deste trabalho está organizado da seguinte maneira: no Capítulo 2 são descritos alguns fundamentos utilizados como base para o esclarecimento sobre programação, apontando algumas dificuldades encontradas nessa área e algumas metodologias de aprendizagem. O Capítulo 3 apresenta a metodologia que foi adotada, indicando como foi conduzido a realização do projeto. No Capítulo 4 são apresentados os resultados e a análise dos resultados. Por fim, no Capítulo 5 são apresentadas as conclusões obtidas com base no projeto realizado e a proposta de trabalho futuro para que possa aprimorar o ensino de POO. Também consta como parte deste trabalho o Apêndice A que trata sobre uma revisão sistemática realizada

antes do projeto e o Apêndice B que apresenta alguns trabalhos relacionados ao tema do projeto.

2. FUNDAMENTAÇÃO

Este capítulo aborda alguns conceitos que são a base do nosso estudo, sendo primordial para o leitor que não tem conhecimento do assunto. Trataremos sobre programação orientada a objetos, processo de aprendizagem, métodos e metodologias de aprendizagem que podem ser aplicados nos cursos de todos os níveis de ensino.

2.1. Programação orientada a objetos

Para PAGE-JONES (2000), o desenvolvimento de software começou simplesmente com a programação procedural¹. À medida que o tamanho dos sistemas e a experiência humana aumentaram, as pessoas compreenderam que escrever meramente códigos para uma aplicação em uma torrente de consciência já não era o bastante.

Durante toda a história do software, as pessoas tentaram atingir a reutilização. Infelizmente, na maioria das unidades procedimentais de código, não são suficientemente independentes para serem reutilizadas independentemente. Mas, com a programação orientada a objetos é possível efetuar reuso de código (PAGE-JONES, 2000).

A Orientação a Objetos (OO) é um termo que agrupa as ideias de muitos pesquisadores de software. Algumas pessoas dizem que OO é uma completa ruptura com o método procedimental, outras dizem que é só uma mudança superficial, diferenciando um pouco com uma aparência atraente em relação ao método procedimental (PAGE-JONES, 2000).

FARINELLI (2007) diz que a orientação a objetos é uma técnica que visualiza os sistemas como sendo coleção de objetos integrantes, sendo possível melhorar a reusabilidade e extensibilidade dos softwares. Seus fundamentos, abrange os princípios da abstração, hierarquização, encapsulamento, classificação, modularização, relacionamento, simultaneidade e persistência.

¹ Programação Procedural foi o primeiro método de programação utilizado, que identifica uma subtarefa que faz parte de um algoritmo sequencial.

RUMBAUGH e BLAHA (2004) complementam dizendo que o software é organizado como uma coleção de objetos distintos, incorporando estrutura de dados e comportamentos, diferentemente de outras técnicas de programação que as estruturas de dados e o comportamento possuem baixa relação.

A programação orientada a objetos (POO) surgiu porque algumas linguagens procedimentais não conseguiram ser eficientes em sistemas de grande porte. A POO permite que objetos do mundo real que compartilham propriedades (ou atributos) e comportamentos comuns sejam agrupados por classe (ZIVIANI, 2015).

A POO possui alguns recursos e características que serão explicados a seguir:

- **Objetos:** surgiu pela necessidade de resolver alguns problemas encontrados em linguagens procedimentais. Ele possui métodos (funções e procedimentos) e variáveis que representam seus campos de dados (atributos). Um objeto não só cria uma aproximação bem grande com o mundo real, como também exclui o problema gerado por variáveis globais (ZIVIANI, 2015).
 - **Atributos:** os objetos possuem características que por sua vez, tem seus valores, definindo o estado do objeto. Esses valores só podem ser mudados por meio de disparos de eventos que fazem a mudança desses estados do objeto (FARINELLI, 2007).
 - **Métodos:** são procedimentos ou funções que realizam as ações próprias do objeto, ou seja, são as ações que os objetos realizam. Todas as ações dos objetos são feitas por meio de métodos, que é onde o objeto se manifesta e interage com outros objetos (FARINELLI, 2007).
- **Classes:** esse conceito surgiu pela necessidade de se criar vários objetos do mesmo tipo. Assim, uma classe é uma especificação para um ou mais objetos, que também pode-se dizer que é uma instância de uma classe. As instâncias de uma classe terão os mesmos atributos e métodos, porém, com valores diferentes (COX, 1986).

- **Abstração:** é a visão geral dos aspectos essenciais de uma aplicação, sem se preocupar com os detalhes, tendo o foco no que um objeto é e faz antes de implementá-lo (RUMBAUGH e BLAHA, 2004).
- **Encapsulamento:** também chamado de ocultamento de informações, é a separação dos aspectos externo de um objeto (que acaba sendo acessado por outros objetos) dos internos (escondidos dos outros objetos), evitando que partes da aplicação possam ser tão independentes que possam gerar enormes efeitos em cascata (RUMBAUGH e BLAHA, 2004). O objeto fica com toda a sua funcionalidade disponível sem que você precise saber como ele funciona internamente, nem como armazena internamente os dados que você recupera (FARINELLI, 2007).
- **Herança:** nome dado ao processo de se criar uma classe a partir de outra. A classe estendida de uma classe base tem todas suas características originais, como também podem ser adicionadas outras. A classe base é chamada de classe pai ou superclasse, enquanto a classe estendida é chamada de subclasse ou derivada. A herança é um recurso que proporciona mais facilidade na reutilização de classes a partir de pequenos ajustes na classe base (SEBESTA, 2010).
- **Polimorfismo:** é a capacidade de uma operação ou nome de atributo ser referenciado em mais de uma classe e executar procedimentos diferentes em cada uma delas. Ocorre quando um método que já foi definido numa superclasse tem uma redefinição na subclasse ou herdeiro com um comportamento diferente (PAGE-JONES, 2000).
- **Sobrecarga de Métodos:** ocorre quando um objeto tem comportamentos diferentes ao utilizarmos métodos com o mesmo nome. É um tipo de polimorfismo *ad hoc*, porém, as assinaturas (parâmetros) são diferentes (ZIVIANI, 2015).

A orientação a objetos é um paradigma subjacente que molda toda maneira de pensar sobre como mapear um problema em um modelo algorítmico. Ele determina de maneiras fundamentais a estrutura até de programas simples. Não pode ser “adicionado” a outras construções de linguagem; em vez disso, substitui a estrutura fundamental da programação procedural (KÖLLING, 1999).

2.1.1. Dificuldades em aprender POO

O ensino de POO é muito discutido no meio acadêmico pois várias tentativas e formas de ensino foram aplicadas gerando uma grande lista de problemas com variados aspectos. Muitas perguntas são lançadas, mas a mais questionada é: “Por que o ensino da programação orientada a objetos parece ser mais difícil do que o ensino de programação procedimental?” (KÖLLING, 1999).

Mas antes dessa pergunta ser respondida, devemos nos atentar e nos preocupar em identificar quando deve ser aplicado o ensino de POO. Para KÖLLING (1999), RAGONIS e BEN-ARI (2005) e MAZAITIS (1993) quanto mais precoce for o contato com POO melhor, portanto, as instituições de ensino devem primeiramente ensinar POO ao invés de Programação Processual.

Do modo que está ocorrendo o ensino (primeiro processual e depois orientado a objetos), os alunos aprendem uma forma de programar e depois tem que desaprender, pois os conceitos de POO são totalmente diferentes e normalmente os estudantes tentam aplicar a lógica processual em POO (KÖLLING, 1999).

Com base em estudos e análises de vários documentos foram detectados alguns problemas (KÖLLING, 1999; RAGONIS e BEN-ARI, 2005; MAZAITIS, 1993):

- Mudança de paradigma (do procedimental para o orientado a objetos);
- Dificuldade de entender o ambiente de desenvolvimento;
- Dificuldade em discernir objetos;

A mudança de paradigma envolve muita sobrecarga de aprendizado pois deve ocorrer alterações na mentalidade sobre programação e todo esse processo deve ser realizado com muita cautela (MAZAITIS, 1993).

Os estudantes ainda não têm proficiência num paradigma e de repente são forçados a fazer uma mudança da abordagem de design tradicional, da análise estruturada, do método de decomposição funcional, para uma nova estratégia (MAZAITIS, 1993).

A orientação de objetos é um paradigma subjacente que molda toda nossa maneira de pensar sobre como mapear um problema em um modelo algorítmico.

Não pode ser "adicionado" a outras construções linguísticas; em vez disso, substitui a estrutura fundamental da programação procedimental (KÖLLING, 1999).

Outro problema é o ambiente de desenvolvimento e para KÖLLING (1999), ambientes para o ensino e os requisitos para esses ambientes são muito menos discutidos na literatura do que as linguagens de programação. Para ele os docentes não se preocupam em descobrir os benefícios ou desvantagens do ambiente de programação.

Segundo KÖLLING (1999), antigamente, os cursos introdutórios se concentraram no desenvolvimento de algoritmos em linguagens processuais ou funcionais. Para isso, um editor e um compilador eram tudo o que era necessário para a parte prática do trabalho. Os cursos modernos agora usam linguagens orientadas a objetos e o material de assunto ensinado inclui testes, depuração e reutilização de código. O autor ainda aborda que para dar a um aluno iniciante a chance de lidar com esse aumento da complexidade, é necessário um melhor apoio ao meio ambiente.

O último problema que é sobre o discernimento de objetos, YAN (2009) diz que conceitos abstratos e pensamento podem ser outra dificuldade que os alunos enfrentarão na aprendizagem da programação orientada a objetos. É bastante difícil encontrar a equivalência dos conceitos de classe e objeto na vida real. Os alunos, portanto, lutam para compreender conceitos ainda mais complexos, como herança e polimorfismo, se não conseguem compreender a classe básica e a maneira de pensar sobre objetos.

Muitos outros problemas poderiam ser destacados, porém, não é nosso intuito abordar todos, mas identificar alguns e buscar uma metodologia que possa ajudar a facilitar o aprendizado de POO. Dentre os problemas, focaremos em evitar ferramenta com IDE (Integrated Development Environment – Ambiente de Desenvolvimento Integrado) complexa e mudar o paradigma de ensino.

2.2. Processo de aprendizagem

O aprendizado ocorre constantemente por meio de um processo de reconstruções, permitindo diferentes modos de relações entre fatos e objetos com

foco na utilização dos saberes em situações diversificadas. O aluno deve estar envolvido de tal maneira que o aprendizado seja por meio de uma metodologia que o transforme em protagonista de sua aprendizagem, ajudando no desenvolver do seu senso crítico, como também, competências para relacionar esses conhecimentos ao mundo real (KOEHLER et al., 2012).

O modo de ensinar com o uso das tecnologias, os desafios impostos aos professores e as diferentes formas e meios que surgiram para melhorar esse processo, exigem mudanças nos métodos de ensino. Com a transformação da sociedade, o método tradicional de ensinar precisa ser modificado. As práticas e os saberes docentes devem ser aprimorados constantemente (BRIGHENTI et al., 2015).

A aprendizagem abrange o desenvolvimento intelectual da pessoa em todas as operações mentais: capacidade de pensar, refletir, analisar, comparar, criticar, justificar, argumentar, inferir conclusões, generalizar, buscar e processar informações, compará-las, criticá-las, organizá-las, produzir conhecimentos, descobrir, pesquisar, criar, inventar, imaginar. Se todo esse processo for limitado a somente ouvir algumas informações e assimilá-las, a aprendizagem estará comprometida tendo um efeito muito pequeno em sua evolução (GAETA e MASETTO, 2010).

Nos últimos anos houve um aumento nas discussões sobre os saberes docentes, sendo objeto de estudo mundialmente por causa da profissionalização do ensino e dos docentes. Estes estudos observam não somente para a falha na transmissão do conhecimento aos alunos, mas ao conjunto de fatores que são construídos e adquiridos com a formação, como também, nas vivências, experiências e habilidades específicas adquiridas com o tempo (BRIGHENTI et al., 2015).

BRIGHENTI et al. (2015) expõem que os alguns métodos e metodologias de aprendizagem aplicados no Brasil ainda são os mesmos aplicados nas escolas jesuítas, que focavam em aulas expositivas, memorização de conteúdos, resolução de exercícios e sistemas rígidos de avaliação e conduta. O autor reforça que o professor não deve ter somente o papel de ensinar o conteúdo, mas que uma mudança em sua didática deve ocorrer, assumindo o papel de mentor e facilitador,

priorizando ser o intermediador entre o aluno e a informação. Para isso, constantemente suas técnicas, métodos e metodologias de aprendizagem precisam ser aprimoradas, de acordo com as necessidades que vão surgindo.

PELIZZARI et. al. (2001) diz que a aprendizagem é muito mais significativa conforme os novos conteúdos são incorporados às estruturas de conhecimento do aluno, a partir da relação com seu conhecimento prévio. Senão, ela se torna mecânica ou repetitiva e o novo conteúdo será armazenado de forma isolada ou através de associações arbitrárias na estrutura cognitiva.

AUSUBEL (1973) apud SILVA e SCHIRLO (2014) explica que a Aprendizagem Significativa é o processo de integrar um novo conhecimento à estrutura cognitiva do aluno de forma não arbitrária e não literal, em que o conhecimento prévio do estudante se relaciona com o novo conhecimento, modificando sua estrutura cognitiva. Já a Aprendizagem Mecânica é quando o novo conteúdo se relaciona com pouca ou nenhuma informação prévia da estrutura cognitiva do aluno, resultando em fórmulas e leis decoradas sem nenhuma interação com conceitos relevantes existentes, sendo esquecidos num período curto de tempo.

AUSUBEL, NOVAK e HANESIAN (1980) afirmam que, existe dois tipos básicos de aprendizagem: por recepção e por descoberta, e isso independe se é na aprendizagem significativa ou mecânica (Figura 2.1).

A aprendizagem por recepção é identificada quando ocorre sob a forma de aprendizagem mecânica ou de aprendizagem significativa quando o professor expõe todo o conteúdo e essa é a única forma de aprendizagem, não ocorrendo nenhuma descoberta por parte do aluno. Então, ele deve simplesmente armazenar o conteúdo do material exposto, reproduzindo o mesmo no futuro, caso necessite. A aprendizagem por descoberta, podendo ser mecânica ou significativa, o aluno precisa fazer uma junção das informações com sua estrutura cognitiva e, através de todo um processo de reorganização e transformação, obtenha o conhecimento desejado ou que esclareça algo que estava duvidoso ou indefinido, tornando-se significativo na aprendizagem por recepção (AUSUBEL, NOVAK e HANESIAN, 1980).

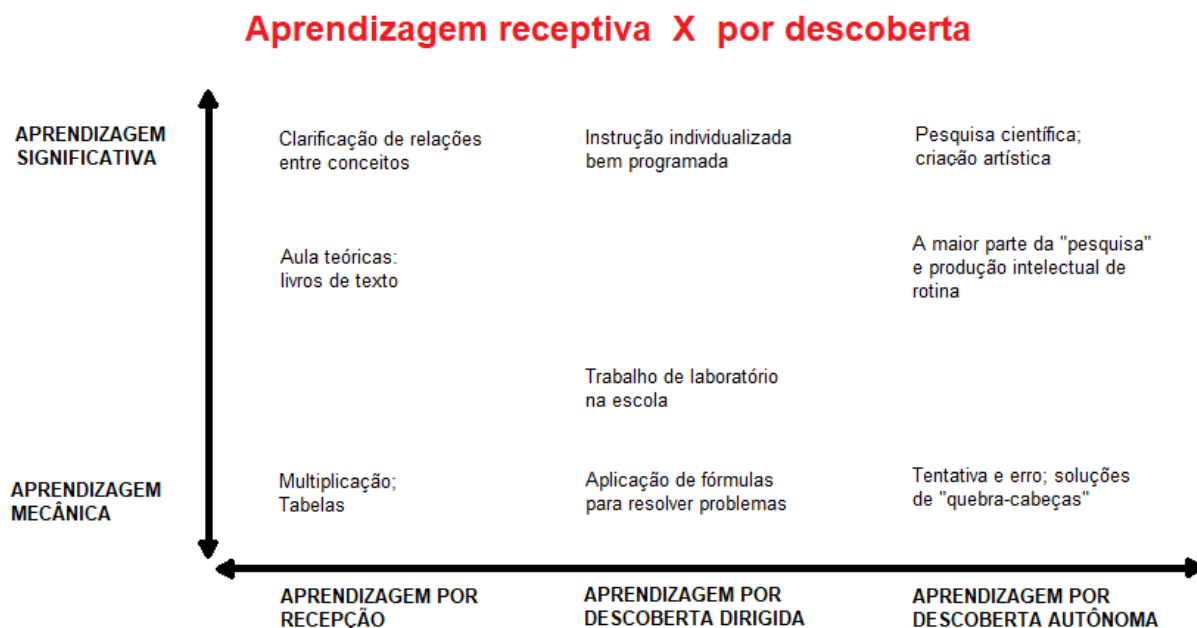


Figura 2.1: Aprendizagem receptiva x por descoberta (NOVAK, 1984)

De acordo com MOREIRA e MASINI (2006) apud SILVA e SCHIRLO (2014), com o decorrer da aprendizagem significativa, os conceitos são desenvolvidos, diferenciados e elaborados por meio de interações sucessivas, conduzindo à diferenciação progressiva, como também à reconciliação integrativa.

Na diferenciação progressiva deve ser preparado um conteúdo de maneira que primeiro são apresentados os conceitos básicos da disciplina e de forma progressiva, as diferenças por meio de conceitos específicos. Assim, na diferenciação progressiva, a aprendizagem significativa acontece num processo contínuo, em que, à medida que novas relações são estabelecidas entre os conceitos estudados, os estudantes adquirem conhecimentos mais significativos (Moreira e Masini (2006) apud SILVA e SCHIRLO, 2014).

FERRAZ e BELHOT (2010), diz que, para estruturar o processo educacional, é necessário definir objetivos de aprendizagem de forma consciente, dando condições para mudar pensamentos, condutas e ações.

Os objetivos instrucionais devem ter uma definição clara e estruturada de acordo com o conhecimento e as competências que se enquadram no perfil do profissional que se deseja formar, orientando o processo de ensino, dando a possibilidade de selecionar as estratégias, procedimentos, delimitar o conteúdo

específico e instrumentos de avaliação, levando para uma aprendizagem consistente e duradoura (FERRAZ e BELHOT, 2010).

De acordo com BRANSFORD e COCKING (2002), as pessoas devem assumir o controle da sua própria aprendizagem, aprendendo a reconhecer quando entendem e quando precisam de mais informações.

Mas é difícil essa percepção e a maioria das vezes precisa de uma orientação de educadores que tenham conhecimento no processo de ensino baseado em algum princípio e com bons resultados. Apesar de existir muitos estudos nesse campo, a Taxonomia de Bloom sempre teve boa aceitação na área educacional.

A Taxonomia² de Bloom é uma ferramenta de classificações estruturadas e orientadas com a intenção de organizar um processo de desenvolvimento cognitivo, afetivo e psicomotor, por meio da criação de métodos de ensino. Entre os três domínios, o cognitivo é o mais conhecido e utilizado (FERRAZ e BELHOT, 2010).

BLOOM et al. (1956) diz que o processo de aprender, está relacionado ao desenvolvimento cognitivo, que é formado pelas habilidades, atitudes, desenvolvimento intelectual, além do novo conhecimento em si. Inclui também, a identificação de fatos específicos, procedimentos padrões e conceitos que incentivam constantemente o desenvolvimento intelectual.

Os objetivos desse domínio, foram agrupados em seis categorias, apresentados hierarquicamente com base na sua complexidade e dependência, seguindo a ordem do mais simples para o mais complexo, detalhados no Quadro 2.1, que, além de indicar as categorias e suas definições, apresenta os verbos que podem ser utilizados para representá-los (BLOOM et al., 1956).

Para progredir para uma nova categoria, precisa obter um desempenho mínimo na anterior, pois cada dependência é formada pelas capacidades adquiridas nos níveis anteriores (BLOOM et al., 1956).

A Figura 2.2 ilustra as categorias do domínio cognitivo.

² Taxonomia é um termo bastante usado em diferentes áreas e, segundo a Wikipédia (2006) apud FERRAZ e BELHOT (2010), é a ciência de classificação, denominação e organização de um sistema pré-determinado e que tem como resultante um framework conceitual para discussões, análises e/ou recuperação de informação.

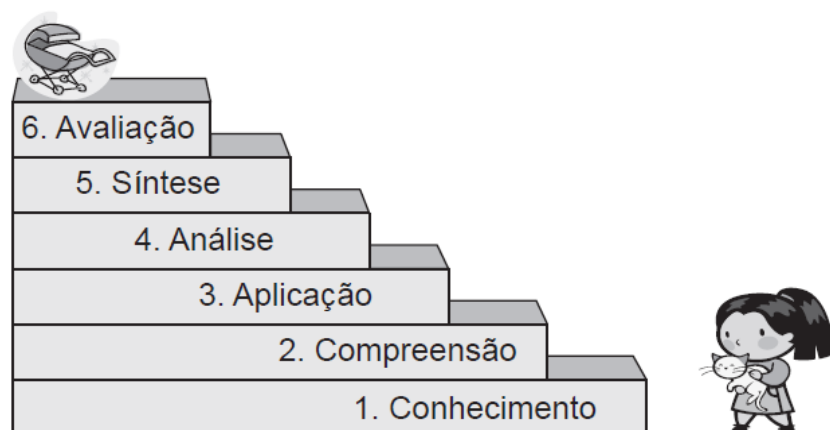


Figura 2.2: Categorias do domínio cognitivo (FERRAZ e BELHOT, 2010).

Quadro 2.1: Divisões dos níveis da taxonomia de Bloom – reproduzido de (FERRAZ e BELHOT, 2010)

	Nível	Definição	Amostra de Verbos
1	Conhecimento	O aluno irá recordar ou reconhecer informações, ideias e princípios na forma em que foram aprendidos.	Escreva, Liste, Rotule, Nomeie, Diga e Defina.
2	Compreensão	O aluno compreende ou interpreta informação com base em conhecimento prévio ou novo.	Explique, Resuma, Parafraseie, Descreva.
3	Aplicação	O aluno seleciona, transfere e usa dados e princípios para completar um problema ou tarefa com um mínimo de supervisão.	Use, Compute, Resolva, Demonstre, Aplique, Construa.
4	Análise	O aluno distingue, classifica e relaciona pressupostos, hipóteses, evidências ou estruturas de uma declaração ou questão.	Análise, Categorize, Compare, Contraste e Separe.
5	Síntese	O aluno cria, integra e combinam ideias em um produto, plano ou proposta, novos para ele.	Crie, Planeje, Elabore hipótese(s), Invente e Desenvolva.
6	Avaliação	O aluno aprecia, avalia ou critica com base em padrões e critérios específicos.	Julgue, Recomende, Critique e Justifique.

Alguns anos depois da publicação, após discussões de especialistas, ocorreram algumas mudanças e, na atual Taxonomia de Bloom, a base das categorias foi mantida, ainda são seis categorias, o nome também não mudou (podendo ocorrer a expressão “revisada” adicionada a ele), mas, fizeram mudanças com a separação conceitual de conhecimento e processo cognitivo, que estão detalhadas abaixo (KRATHWOHL, 2002 apud FERRAZ e BELHOT, 2010):

- A categoria Conhecimento foi renomeada para Lembrar, mas manteve os aspectos verbais utilizados na categoria; Compreensão foi renomeada para Entender; e Aplicação, Análise, Síntese e Avaliação, foram alteradas para a forma verbal Aplicar, Analisar, Sintetizar e Criar, pelo fato da ação pretendida ficar mais clara, como também o resultado esperado na categoria;

- A ordem das categorias Avaliação e Síntese (avaliar e criar) foram invertidas; e
- Os nomes das subcategorias existentes foram alterados para verbos no gerúndio.

A Figura 2.3, mostra as classificações das categorias após as mudanças e o Quadro 2.2, apresenta uma explicação de como deve ser utilizado cada categoria e os verbos adequados a elas.

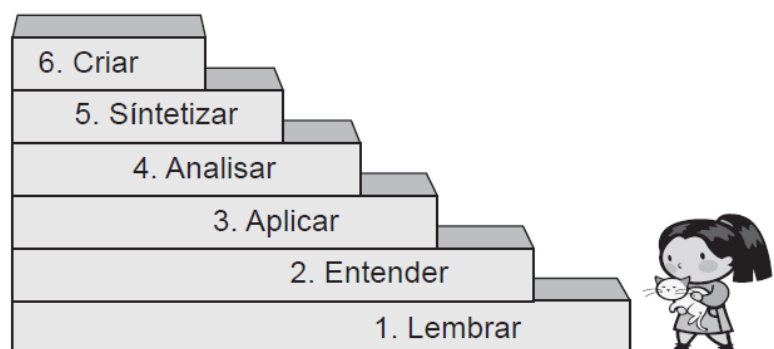


Figura 2.3: Categorização atual da Taxonomia de Bloom proposta por Anderson (FERRAZ e BELHOT, 2010).

Quadro 2.2. Estrutura do processo cognitivo na taxonomia de Bloom (revisada) – reproduzida de (FERRAZ e BELHOT, 2010).

- | |
|--|
| <p>1. Lembrar: Relacionado a reconhecer e reproduzir ideias e conteúdos. Reconhecer requer distinguir e selecionar uma determinada informação e reproduzir ou recordar está mais relacionado à busca por uma informação relevante memorizada. Representado pelos seguintes verbos no gerúndio: Reconhecendo e Reproduzindo.</p> <p>2. Entender: Relacionado a estabelecer uma conexão entre o novo e o conhecimento previamente adquirido. A informação é entendida quando o aprendiz consegue reproduzi-la com suas “próprias palavras”. Representado pelos seguintes verbos no gerúndio: Interpretando, Exemplificando, Classificando, Resumindo, Inferindo, Comparando e Explicando.</p> <p>3. Aplicar: Relacionado a executar ou usar um procedimento numa situação específica e pode também abordar a aplicação de um conhecimento numa situação nova. Representado pelos seguintes verbos no gerúndio: Executando e</p> |
|--|

Implementando.

4. Analisar: Relacionado a dividir a informação em partes relevantes e irrelevantes, importantes e menos importantes e entender a inter-relação existente entre as partes. Representado pelos seguintes verbos no gerúndio: Diferenciando, Organizando, Atribuindo e Concluindo.

5. Avaliar: Relacionado a realizar julgamentos baseados em critérios e padrões qualitativos e quantitativos ou de eficiência e eficácia. Representado pelos seguintes verbos no gerúndio: Checando e Criticando.

6. Criar: Significa colocar elementos junto com o objetivo de criar uma nova visão, uma nova solução, estrutura ou modelo utilizando conhecimentos e habilidades previamente adquiridos. Envolve o desenvolvimento de ideias novas e originais, produtos e métodos por meio da percepção da interdisciplinaridade e da interdependência de conceitos. Representado pelos seguintes verbos no gerúndio: Generalizando, Planejando e Produzindo.

2.3. Métodos e metodologias de aprendizagem

No processo educativo, a proposta de trabalho dos docentes precisa ser planejada de modo que as metodologias utilizadas consigam atingir seus objetivos. A metodologia de ensino pode ser descrita como um “conjunto de procedimentos didáticos, representados por seus métodos e técnicas de ensino”. Esses métodos têm a intenção de atingir a máxima eficácia dos objetivos de ensino e de aprendizagem, obtendo o máximo de rendimento (BRIGHENTI et al., 2015).

GEMIGNANI (2012) baseia-se na ideia de que a prática educativa tem a necessidade de mudar os conteúdos e o modo de avaliar, buscando atingir as finalidades do ensino. A proposta curricular deve se basear em projetos que sejam próximos da realidade educativa, profissionalizante, propostas simples com recursos variados, conseguindo unificar os planejamentos de aprendizagem elaborados pelos professores. Essas mudanças apontam para a migração da disciplinaridade para a interdisciplinaridade, como também, novas técnicas de ensino-aprendizagem, como as metodologias ativas, que é um grande desafio na formação de professores do futuro.

GEMIGNANI (2012), destaca a necessidade de formação de professores que saibam correlacionar teoria e prática, que consigam, de maneira criativa, resolver problemas do cotidiano da sociedade e da escola, que ele não seja somente um dominador de conhecimentos, mas que consigam melhorar suas práticas de ensino, pois o modelo tradicional não está tendo mesma a eficácia que antes, surgindo a

necessidade em aumentar o acesso escolar e cultural da população menos preparada, causada pelo avanço tecnológico e científico.

O aluno contemporâneo deve ser capaz, diante de tantas competências exigidas (éticas, políticas e técnicas), de autogerenciar seu processo de formação. Pesquisadores da contemporaneidade tentam criar metodologias que consigam transformar o aluno em autônomo, conseguindo se autogerenciar e ter corresponsabilidade pelo seu próprio processo de formação. Tais metodologias são chamadas de metodologias ativas de aprendizagem.

2.4. Metodologias ativas

Para NAGAI e IZEKI (2013), uma metodologia ativa baseia-se em formas de desenvolver o processo de aprender, baseado em problemas reais ou simulados, que possam ser solucionados por meio de desafios e atividades básicas da prática social, em diferentes contextos. Utiliza a problematização como estratégia de ensino-aprendizagem para motivar o aluno, que foca no problema, examina, reflete e começa e realizar descobertas.

GEMIGNANI (2012), diz que metodologia ativa é uma concepção educativa que estimula processos construtivos de ação-reflexão-ação, em que o aluno é um agente ativo em seu processo de aprendizado numa situação prática de experiências, por meio de problemas que geram desafios e possibilitam pesquisar e descobrir soluções que possam ser aplicadas na realidade.

Assim, a metodologia ativa proporciona um entendimento e intervenção consistente sobre a realidade, valorizando todos os participantes e seus conhecimentos, no processo de construção coletiva, além de permitir a liberdade no processo de pensar e no trabalho em equipe. Promove a interatividade, estimulando a autoaprendizagem, em que o professor atua como orientador e o aluno como sendo o agente central do processo (MOURÃO, 2017).

Conforme MOURÃO (2017), os princípios que norteiam esta metodologia, estão relacionados com a aprendizagem:

- Construtiva
- Colaborativa

- Interdisciplinar
- Contextualizada
- Reflexiva
- Crítica
- Investigativa
- Humanista
- Motivadora
- Desafiadora

As metodologias ativas podem ser classificadas de diversas maneiras ou métodos, tais como:

- Problematização
- Estudo de Caso
- Discussão em Classe
- Aprendizagem Baseada em Problemas
- Aprendizagem Baseada em Projetos
- Seminários
- Mapas Mentais e Conceituais
- Aprendizagem Baseada em Grupos e Simulação
- Sala de Aula Invertida
- Gamificação

Diante deste contexto, conforme KOEHLER et al. (2012), o estudante precisa ter atitudes cada vez mais ativas, deixando de ser um simples receptor de conteúdos e buscar informações referentes aos problemas em questão. Ser criativo, ter curiosidade científica, pensamento crítico, conseguir fazer autoavaliação, cooperar no trabalho em equipe, ter senso de responsabilidade e ser ético são qualidades fundamentais a serem desenvolvidas em seu perfil.

Os estudantes e a sociedade passaram e continuam passando por significativas e grandes mudanças em seus modelos de vida e, com isso, as metodologias tradicionais já não conseguem ser eficientes como antes, sendo necessário, que as práticas docentes sejam aprimoradas (BRIGHENTI et al., 2015).

O planejamento é muito importante para o ensino, pois por meio dele é possível definir os métodos que serão utilizados durante as atividades. Nesse contexto, BRIGHENTI et al. (2015) tentam entender como que os professores planejam seus cursos sem utilizar um pouco de criatividade, sempre utilizando os mesmos métodos de ensino e formas de avaliação, sem tentar adequar esses procedimentos de acordo com as mudanças e evoluções que vem ocorrendo, que simplesmente discorrem sobre os assuntos de um livro, sem refletir o que realmente é importante os alunos aprenderem.

Diante de todos esses apontamentos, foi realizado um estudo de algumas metodologias (apresentadas nas seções subsequentes) que são utilizadas na área de tecnologia, mais especificamente, em programação. Não serão abordadas todas as metodologias de aprendizagem ativa, pois a quantidade de opções é muito grande, tornando-se muito custosa, mas também não sendo necessária, porque nosso objetivo é encontrar metodologias eficazes na aprendizagem de programação orientada a objetos.

2.4.1. Aprendizagem baseada em problemas (ABP)

Também chamada de *Problem Based Learning* (PBL), essa é uma metodologia ativa de ensino-aprendizagem que teve origem nos anos 1960 no Canadá, mais especificamente na escola de Medicina da Universidade McMaster (MOURÃO, 2017).

O modelo médico foi ajustado para várias áreas da educação, inclusive em engenharia e computação. No Brasil, foi adotado inicialmente em cursos de medicina, começando a serem praticadas na computação, em meados de 2010. (BITTENCOURT, 2012).

A ABP é um processo de aprendizagem que se baseia numa construção significativa, e não no processo passivo de recepção de informações, presente nas mais variadas pesquisas da psicologia cognitiva. No início de um processo instrucional é inserido um problema relevante, usado para criar um contexto e gerar motivação, reconstruindo a maneira que se relaciona com o mundo, ultrapassando sua vontade de aprender. Na aprendizagem, o sujeito que aprende é o centro do

processo, assim, existe a necessidade de mudar a maneira de pensar e do seu relacionamento com o mundo (MOURÃO, 2017).

A diferença entre a ABP e o modelo tradicional de ensino é a utilização de problemas reais, dando início ao processo de aprendizagem, na tentativa de estimular o desenvolvimento de habilidades mediante resolução de problemas envolvendo os conteúdos apresentados até aquele momento, partindo de um problema simples e de modo gradual, aumenta-se o nível de dificuldade e complexidade. Esta aprendizagem constitui-se em explicar termos e expressões do texto, entender o contexto por meio de questionamentos, analisar o problema, identificar e formular hipóteses e, caso tenha necessidade, buscar informações para responder o problema e discutir conhecimentos (MOURÃO, 2017).

Conforme NAGAI e IZEKI (2013), existem alguns passos essenciais para utilização da ABP. Primeiro, precisa escolher um ou mais problemas reais para estimular o aprendizado, proporcionando o envolvimento dos estudantes e que eles se sintam desafiados a resolver os problemas com habilidades, atitudes e conhecimentos que possuem. Os alunos precisam enxergar que esta é uma oportunidade para construir seu conhecimento por meio de interações e pesquisas colaborativas. Mas, no primeiro contato com a ABP, eles precisam ser orientados, preparados e motivados, pois não possuem a habilidade necessária para a ABP, sendo que sempre participaram de ambientes de sala de aula tradicional.

De acordo com NAGAI e IZEKI (2013), a ABP é adequada para as aulas introdutórias, pois acaba ajudando os alunos a desenvolverem confiança e habilidades para resolverem problemas que são totalmente diferentes dos vistos até o momento. Os autores ainda destacam alguns pontos positivos da ABP:

- Estudantes com atitudes positivas;
- Possibilidade de uma abordagem agradável, motivadora e desafiadora para a aprendizagem;
- O conhecimento adquirido é armazenado por muito mais tempo se comparado com o ensino tradicional;
- O aprendizado se torna profundo, desenvolvendo habilidades para a resolução de problemas;

Para GEMIGNANI (2012), existe sete passos no processo de ensino, quando utiliza-se a Aprendizagem Baseada em Problemas:

1. Apresentação do problema (leitura pelo grupo).
2. Esclarecimento de alguns termos conceituais pouco conhecidos e de dúvidas sobre o problema.
3. Definição e síntese do problema em discussão, com identificação das áreas ou pontos relevantes.
4. Análise do problema utilizando os conhecimentos prévios (tempestade de ideias - brainstorming)
5. Desenvolvimento de hipóteses para explicar o problema e identificação de lacunas de conhecimento.
6. Definição dos objetivos de aprendizagem e identificação dos recursos de aprendizagem apropriados.
7. Busca de informação e estudo individual.

Na Aprendizagem Baseada em Problemas, o professor/tutor tem um papel importante, que é estimular o aluno a entender o problema de forma independente e autônoma, exigindo esforços por parte dos professores que devem criar cenários de ensino que favoreçam a maneira de trabalhar e o processo de aprendizagem independente do grau de complexidade e relevância do assunto. (GEMIGNANI, 2012).

2.4.2. Aprendizagem baseada em projetos

Do mesmo modo que na aprendizagem baseado em problemas, a aprendizagem baseada em projetos (project-based learning), o aluno deve assumir uma atitude ativa no processo de construção do conhecimento. Os dois métodos são muito parecidos, pois seguem os mesmos princípios, com a diferença de que na aprendizagem baseada em projeto pode-se realizar algo na prática, como o desenvolvimento de alguma programação de uma solução fornecida, e pelo fato de que ao iniciar um projeto, normalmente já possuem um produto final para ser criado (Donnelly & Fitzmaurice, 2005 apud AFONSO, 2013).

Segundo Thomas (2000) apud Afonso (2013), esta metodologia de ensino agrega vários benefícios ao processo de ensino-aprendizagem como o aumento da autoconfiança e melhora das atitudes em relação à aprendizagem, devido as participações nas atividades.

2.4.3. Aprendizagem entre pares

A metodologia do “peer instruction” (Instrução entre Pares), destaca a importância do entendimento e aplicação dos conceitos por meio da discussão entre os estudantes. Nessa metodologia, o primeiro passo é a compreensão e o entendimento dos conceitos. O próximo passo seria desenvolver suas habilidades com aplicações práticas, sendo o mínimo esperado pelo profissional e sua atuação (KOEHLER et al., 2012).

A metodologia tende a envolver, mantendo o comprometimento dos alunos que ficam atentos durante a aula com o apoio de atividades, que exigem a aplicação de conceitos fundamentais que cada aluno absorveu nas aulas apresentadas, tendo esses conceitos explicados posteriormente aos seus colegas. Ao contrário da metodologia tradicional, em que ocorre perguntas que normalmente poucos alunos (os mais motivados) participam, neste caso, todos estão participando ativamente, com questionamentos mais estruturados (KOEHLER et al., 2012, apud MAZUR, 2007, p.5).

A linguagem utilizada numa discussão entre os alunos, proporciona um entendimento mais rápido dos conceitos, do que a explicação técnica do professor. Segundos as pesquisas de Mazur, com a utilização da metodologia, o aproveitamento dos conceitos subiu de 20% para 60%, comparado com as metodologias tradicionais (KOEHLER et al., 2012).

2.4.4. Aprendizagem colaborativa

O processo de transferir o conhecimento de uma cabeça para outra, é algo que muitas pessoas acreditam dar certo. No entanto, a aprendizagem colaborativa é fundamentada na ideia de que o conhecimento é formado por um conjunto de conhecimentos de uma comunidade de pessoas, que mediante conversas ou pelo

simples fato de trabalharem juntas direta ou indiretamente (ex: resolução de problemas, elaboração de projetos, estudos de caso, etc) chegam a um entendimento. É um modo de incentivar a participação do estudante no processo de aprendizagem, tornando-a um processo ativo e efetivo. São vários métodos educacionais nomeados de aprendizagem cooperativa ou aprendizagem em grupo pequeno (TORRES et al., 2004).

Conforme TORRES et al. (2004), a Aprendizagem Colaborativa efetivamente tem aumentado o grau de conhecimento dos alunos e o desenvolvimento de habilidades de trabalho em grupo. A Associação Americana para o Avanço da Ciência (1989) elaborou um documento advertindo que o trabalho em grupo deveria ser mais utilizado para a elaboração de trabalhos científicos e tecnológicos, pois os engenheiros e cientistas, normalmente, trabalham em conjunto ao invés de ficarem isolados, fato que deveria ocorrer com os estudantes, para adquirirem experiências por meio de troca de conhecimentos e responsabilidades. Os estudantes que aprenderam em pequenos grupos tiveram melhor aproveitamento dos conteúdos do que os que não participaram de trabalho cooperativo ou colaborativo. Numa aprendizagem colaborativa, coletivamente os estudantes formam seu conhecimento por meio de trocas constantes de informações, de modos diferentes de enxergar as coisas, de elaborar questionamentos, de realizar avaliações e resoluções de questões. Com isso, obtém-se um grupo coerente e uniforme, pois todos compartilham suas atividades, tornando uma ferramenta ou instrumento a serviço da construção coletiva do saber.

As atividades realizadas em grupo são os motivos de ajudar na união e participação do grupo, além de proporcionar a dinâmica entre os integrantes. É na fase de gerenciar estas atividades que os participantes se organizam, dividem as obrigações de cada um, discutem ideias e suas opiniões, interagem entre si e estipulam subtarefas, através de um projeto elaborado, definido e negociado de forma coletiva. As estratégias pedagógicas estão focadas na construção do conhecimento e na colaboração entre pares, que não busca atingir o mesmo nível de conhecimento, pois respeita o princípio de que cada aluno tem suas diferenças e que na heterogeneidade, produzem e crescem juntos, estabelecendo novas formas de relações entre pares. O desenvolvimento de atividades em grupo requer o gerenciamento de conflitos sócio-cognitivos, possibilidades de mudanças ou opções,

rever conceitos, esclarecer pontos de vista, dividir cargas cognitivas, reelaborar ideias, dividir autorias, negociar e constantemente ter auto e mútua-regulação (TORRES et al., 2004).

2.4.5. Sala de aula invertida

A Sala de Aula Invertida, ou *Flipped Classroom*, é uma derivação do ensino híbrido (semi-presencial), ou também chamado de *Blended Learning*, podendo ser classificado como um modelo de ensino-aprendizagem que agrupa o ensino presencial (tradicional) e o ensino *online* (*e-learning*) (COSTA et al., 2017).

Nessa metodologia os alunos estudam os conteúdos de modo *online* (podendo ser na escola, em casa, ou em qualquer outro lugar), fora do horário de aula. Na sala de aula, o tempo é específico para o aprendizado com o acompanhamento do professor (COSTA et al., 2017).

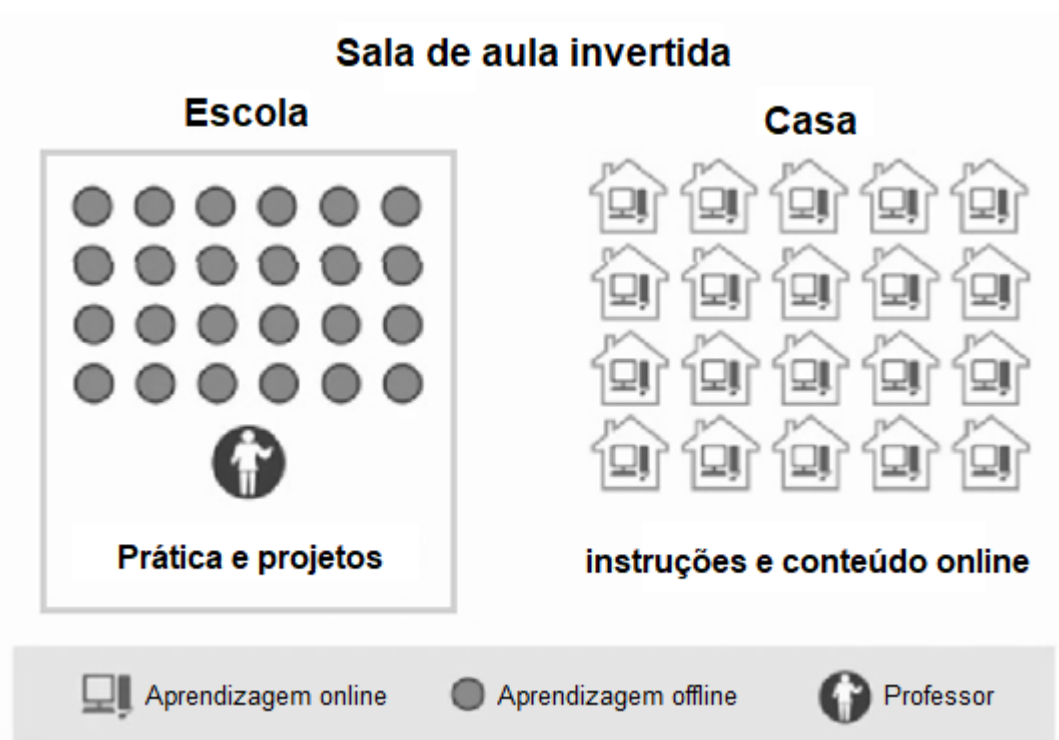


Figura 2.4: Modelo de sala de aula invertida (COSTA et al., 2017).

A Figura 2.4 retrata o ensino híbrido no modelo sala de aula invertida, em que do lado esquerdo (School) são trabalhadas as atividades em sala de aula juntamente com o professor, realizando tarefas, discutindo, montando projetos, entre

outras atividades. Do lado direito (Home), a representação do ambiente online, em que os alunos acessam uma plataforma virtual de aprendizagem, em locais e horários que lhe for mais conveniente, para estudar e realizar as atividades (COSTA et al., 2017).

2.4.6. Gamificação

Devido à popularização dos jogos, principalmente os eletrônicos, surgiu um fenômeno moderno chamado de gamificação, que está em constante crescimento no meio educacional (COSTA et al., 2017).

A gamificação é definida pelo uso de elementos de jogos, que não esteja num ambiente de jogo, possibilitando que os alunos se envolvam como processo mediante o ganho de pontos ou o ranqueamento do usuário em relação aos outros (SOUZA e LEITE, 2020); normalmente ela emprega esses princípios e *design* com o objetivo de aumentar o envolvimento do usuário, tanto relacionado à sua produtividade, aprendizado e realização de atividades motoras, quanto em promover mudanças comportamentais. A palavra gamificação foi criada em 2002 pelo programador britânico Nick Pelling, mas o conceito de gamificação já existe há bastante tempo (COSTA et al., 2017).

Com a intenção de aumentar a participação, prender a atenção, criar engajamento, comprometimento e deixar contextos diversos não relacionados a jogos mais ricos, a gamificação utiliza mecanismos de recompensas que são dadas ao jogador como prêmio por ter finalizado alguma tarefa ou por se comportar de maneira adequada no processo de aprendizagem. A maioria dos sistemas gamificados trabalham com métodos de recompensa na intenção de estimular os jogadores, atribuindo a eles, pontos, aumento na barra de progresso, prêmios, moedas virtuais e, quando possível, recompensas não virtuais (COSTA et al., 2017).

Na área da educação, essa metodologia é usada para aumentar a satisfação e o comprometimento do estudante durante o processo de ensino-aprendizagem, tentando mantê-lo interessado e inspirado a continuar a aprendizagem. Nota-se que a utilização dessa técnica está aumentando em salas de aula ou em plataformas educacionais, que buscam prender a atenção dos estudantes, mostrando que é uma

tendência o seu uso e, quando bem elaborada, tende a ter resultados positivos no ensino, que pode ser por causa do *feedback*, do maior nível de diversão durante a aprendizagem ou por tornar a aprendizagem atrativa por causa da constante interação com as novas tecnologias digitais (COSTA et al., 2017).

2.5. Considerações finais

Neste capítulo foram apresentados conceitos básicos sobre nosso objeto de estudo (POO), métodos e metodologias de aprendizagem, explanando algumas práticas que são destaques na área acadêmica e referências mundiais no quesito aprendizagem. Também foram descritas algumas metodologias ativas de ensino, abordando como podem ser utilizadas e quais os ganhos que podem ocorrer no processo ensino-aprendizagem.

Muitas características podem ser percebidas em cada uma das metodologias, sendo perceptível o grande ganho de qualidade quando comparadas à metodologia de ensino tradicional, centrada no professor, não deixando o aluno ser livre para buscar o conhecimento e experiência.

O próximo capítulo, descreve a metodologia utilizada, apresentando a ferramenta RoboCode, todas as etapas do projeto e análise dele.

3. METODOLOGIA PARA APOIAR O ENSINO DE POO

Este capítulo apresenta a metodologia utilizada para a elaboração deste trabalho. Na Seção 3.1 é apresentada a forma de como foram conduzidas as aulas, a ferramenta que foi utilizada no projeto prático com os alunos e o questionário utilizado após algumas fases do projeto com o intuito de obter a percepção e compreensão dos alunos. A Seção 3.2 informa quais foram as etapas e os procedimentos realizados durante o projeto e na Seção 3.3 é feita uma análise dos resultados obtidos.

3.1. Materiais

3.1.1. Aulas

As aulas foram trabalhadas de modo remoto devido à pandemia da COVID-19. Os materiais com os conteúdos da disciplina com os conceitos de POO (os conteúdos sobre classe, objetos, herança, polimorfismo, entre outras) foram disponibilizados na plataforma *Classroom*³ da instituição conforme a Figura 3.1. Os materiais também foram disponibilizados na página do *github*⁴ do professor da disciplina (Figura 3.2).

O aluno, além de participar da aula de modo remoto juntamente com o professor que explanava os conteúdos e tirava dúvidas, também tinha os materiais e vídeos gravados para poder assistir em outros momentos, sendo uma maneira de reforçar seu aprendizado.

Também foram propostos exercícios para que pusessem em prática todo o conteúdo apresentado e caso o aluno encontrasse alguma dificuldade, o professor resolvia o exercício, deixava gravado na plataforma para poder assistir, além de disponibilizar os materiais utilizados na resolução.

³ Google Classroom é um sistema de gerenciamento de conteúdo para escolas que procuram simplificar a criação, a distribuição e a avaliação de trabalhos. Ele é um recurso do Google Apps redirecionado à área de educação. (Wikipédia).

⁴ GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo. <https://github.com/>.

Programação Orientada a Objetos I

CIC037MAT3T00

Mural

Atividades

Pessoas

Notas

Aulas

	Aula 12 - Interface Gráfica - Partes III, IV e V	Item postado em 12 de jul. d...
	Aula 11 - Interface Gráfica - Partes I e II	Item postado em 10 de jul. d...
	Aula 10 - Programação em Camadas	Item postado em 30 de jun. d...
	Aula 09 - Métodos e Atributos Static -- Con...	Item postado em 23 de jun. d...
	Aula 08 - Resolução dos Exercícios	Última edição: 18 de jun. de ...
	Aula 08 - Polimorfismo	Item postado em 16 de jun. d...
	Aula 07 - Classes e Métodos Abstratos	Item postado em 9 de jun. de...
	Aula 06 - Herança - Resolução dos Exercícios	Item postado em 26 de mai. ...
	Aula 06 - Herança	Item postado em 19 de mai. ...
	Aula 05 - Vetores de Objetos - Resolução d...	Item postado em 6 de mai. d...
	Aula05 - Vetores de Objetos	Item postado em 5 de mai. d...
	Aula 04 - Resolução dos Exercícios	Última edição: 30 de abr. de ...
	Aula04 - Construtores	Última edição: 28 de abr. de ...
	Aula03 - Classes e Objetos com Java	Item postado em 23 de abr. d...
	Aula 02 - Classes e Objetos	Última edição: 16 de abr. de ...
	Aula01 - Video da resolução dos exercicios	Item postado em 14 de abr. d...
	Aula 01	Item postado em 9 de abr. de...

Figura 3.1: Materiais disponibilizados para a disciplina de POO no *Classroom* (Autor, 2021).

Universidade Estadual Paulista Júlio de Mesquita Filho (Unesp)

Faculdade de Ciências e Tecnologia de Presidente Prudente (FCT/Unesp)

Departamento de Matemática e Computação (DMC)

Programação Orientada a Objetos -- Turma 2021

Prof. Dr. Danilo Medeiros Eler

Atividades

1. [Introdução à Orientação a Objetos \(vídeo\)](#)
 - a. [Exercícios de Abstração \(pdf\)](#)
 - b. [Resolução de Exercícios de Abstração \(vídeo\)](#)
 - c. [Resolução de Exercícios de Abstração \(pdf\)](#)
 - d. [Slides da aula \(pdf\)](#)
2. [Classes e Objetos \(vídeo\)](#)
 - a. [Classes e Objetos com Java - Exercícios 1 a 3 \(vídeo\)](#)
 - b. [Projeto do Netbeans - Exemplos - resolução dos exercícios da Aula 01 \(zip\)](#)
 - c. [Classes e Objetos com Java - Exercícios 4 e 5 \(vídeo\)](#)
 - d. [Introdução ao Java - Sintaxe \(vídeo\)](#)
 - e. [Projeto do Netbeans - Sintaxe \(zip\)](#)
3. [Classes e Objetos - Exercícios \(pdf\)](#)
 - a. [Resolução dos exercícios 1, 2 e 3 \(vídeo\)](#)
 - b. [Projeto do Netbeans - Resolução dos exercícios 1, 2 e 3 \(zip\)](#)
 - c. [Construtores \(vídeo\)](#)
 - d. [Resolução dos exercícios 4, 5 e 6 \(vídeo\)](#)
 - e. [Resolução dos exercícios 7 e 8 \(vídeo\)](#)
 - f. [Projeto do Netbeans com Resolução dos exercícios 1 a 8 \(zip\)](#)
4. [Classes e Objetos - Vetores \(vídeo\)](#)
 - a. [Classes e Objetos - Vetores - slides da aula \(pdf\)](#)
 - b. [Prática - Exemplo de implementação: vetores, classe sistema e menu \(vídeo\)](#)
 - c. [Projeto do Netbeans com códigos usados na aula prática \(zip\)](#)
 - d. [Lista de Exercícios \(pdf\)](#)
 - e. [Resolução do Exercício 1 \(vídeo\)](#)
 - f. [Projeto do Netbeans com resolução do Exercício 1 \(zip\)](#)
 - g. [Resolução do Exercício 2 \(vídeo\)](#)
 - h. [Resolução do Exercício 3 \(vídeo\)](#)
 - i. [Projeto do Netbeans com Resolução dos exercícios 1 a 3 \(zip\)](#)
5. [Herança](#)
 - a. [Herança - Parte 1 \(vídeo\)](#)
 - b. [Herança - Parte 1 - Prática \(vídeo\)](#)
 - c. [Herança - Parte 1 - slides da aula \(pdf\)](#)
 - d. [Herança - Parte 2 \(vídeo\)](#)
 - e. [Herança - Parte 2 - Prática \(vídeo\)](#)
 - f. [Herança - Parte 2 - slides da aula \(pdf\)](#)
 - g. [Projeto do Netbeans com os exemplos da aula \(zip\)](#)
 - h. [Lista de Exercícios \(pdf\)](#)
 - i. [Resolução do Exercício 1 \(vídeo\)](#)
 - j. [Resolução do Exercício 2 \(vídeo\)](#)
 - k. [Resolução do Exercício 3 - Parte 1 \(vídeo\)](#)
 - l. [Resolução do Exercício 3 - Parte 2 \(vídeo\)](#)
 - m. [Resolução do Exercício 4 \(vídeo\)](#)
 - n. [Projeto do Netbeans com resolução dos Exercícios 1, 2, 3 e 4 \(zip\)](#)
6. [Classes Abstratas e Polimorfismo](#)
 - a. [Classes e Métodos Abstratos \(vídeo\)](#)
 - b. [Classes e Métodos Abstratos - slides da aula \(pdf\)](#)
 - c. [Polimorfismo \(vídeo\)](#)
 - d. [Polimorfismo - slides da aula \(pdf\)](#)
 - e. [Projeto do Netbeans com códigos usados na parte prática das aulas \(zip\)](#)
 - f. [Lista de Exercícios \(pdf\)](#)
 - g. [Resolução do Exercício 1 \(vídeo\)](#)
 - h. [Resolução do Exercício 2 \(vídeo\)](#)
 - i. [Projeto do Netbeans com resolução dos Exercícios 1 e 2 \(zip\)](#)

Figura 3.2: Materiais disponibilizados para a disciplina de POO no *github* (Autor, 2021).

3.1.2. RoboCode

O RoboCode⁵ é um jogo criado por Mathew A. Nelson e lançado pela IBM em 2001, tornando-se *open source* em 2005. A IBM viu uma oportunidade de promover o RoboCode como uma maneira divertida de começar a aprender programar em Java (mas outras linguagens como Kotlin e Scala também são possíveis), e a IBM queria promover Java também (ROBOCODE, 2021).



Figura 3.3: RoboCode (ROBOCODE, 2021).

RoboCode é um jogo de programação (e ambiente de desenvolvimento) que ensina Java de forma divertida e gratificante. O ambiente do jogo consiste em um pequeno campo de batalha retangular povoado com robôs-tanques. Estes robôs-tanques programados inteligentemente manobram, pesquisam e destroem outros robôs no campo de batalha (O'KELLY e GIBSON, 2006).

Por meio da classe Robo e de alguns exemplos disponibilizados, os alunos podem conferir o funcionamento e as ações, reutilizar os códigos de diversos robôs e fazer as modificações que desejarem. Logo após poderão testar o desempenho do seu novo robô na arena de batalha.

Robocode oferece um ambiente de desenvolvimento completo e vem com um instalador, editor de robô embutido e compilador Java. O Robocode requer apenas que um JRE (Java Runtime Environment) já exista no sistema operacional (Windows, Linux, macOS, Unix) no qual o Robocode está instalado. Portanto, tudo que um desenvolvedor Robocode precisa para começar é fornecido com o arquivo de distribuição principal do Robocode (robocode-xxx-setup.jar). Robocode também

⁵ RoboCode – <https://robocode.sourceforge.io/>

suporta o desenvolvimento de robôs usando IDEs externos como, por exemplo (ROBOCODE, 2021):

- IntelliJ IDEA
- Eclipse
- NetBeans
- Visual Studio Code

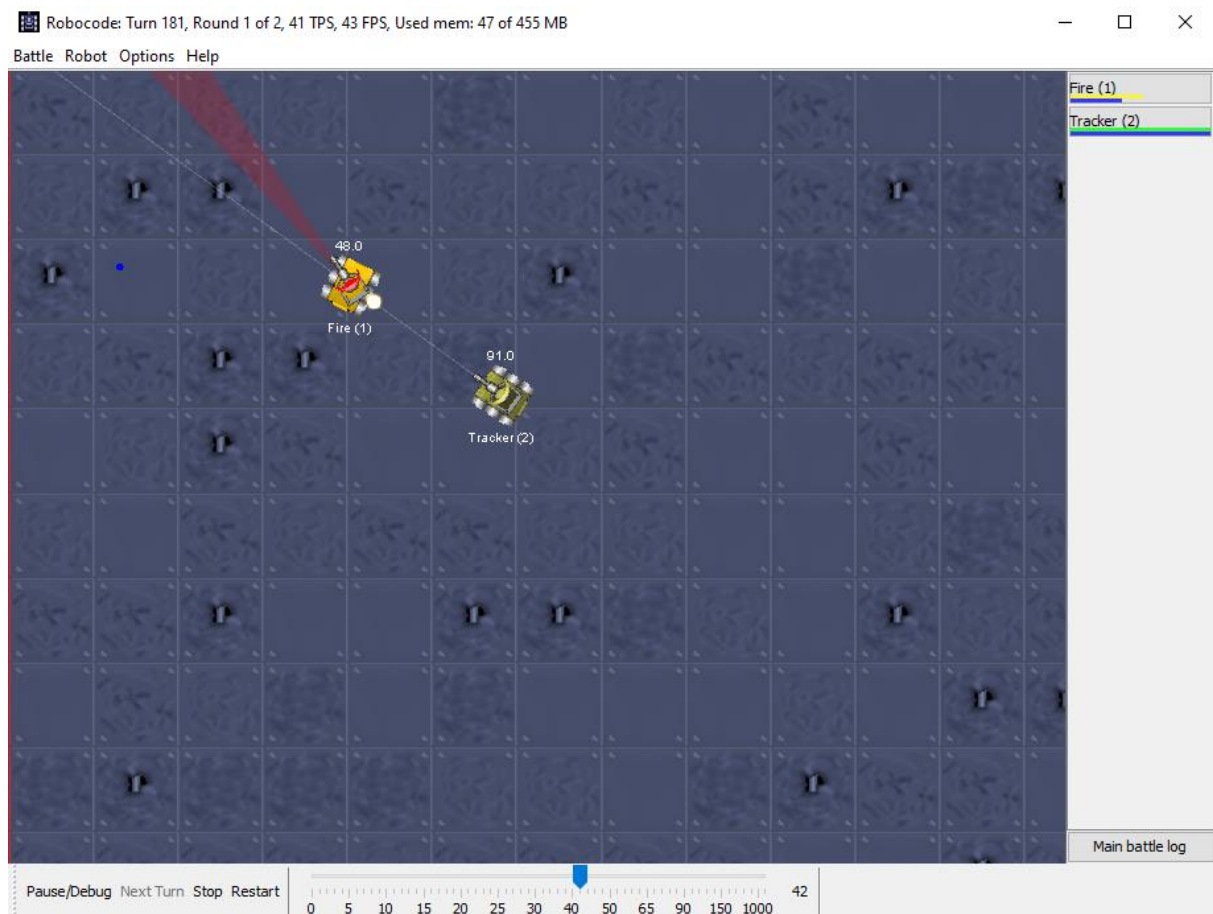


Figura 3.4: Campo de batalha (Autor, 2021).

3.1.3. Questionários

Foram aplicados dois questionários auto avaliativos durante o projeto, com a intenção de avaliar de forma quantitativa o nível de percepção de aprendizado dos alunos. Os questionários foram formulados na escala Likert com cinco conceitos

como respostas: Ótimo, Bom, Regular, Ruim e Péssimo. Os questionários foram desenvolvidos utilizando a plataforma do Google Forms⁶.

O primeiro questionário foi aplicado após os alunos terem estudado os conceitos básicos de POO. Foram sete perguntas para verificar o nível de aprendizado percebido pelos alunos quanto aos conteúdos que foram estudados por meio da metodologia de ensino tradicional.

O segundo questionário foi aplicado após a utilização do RoboCode como ferramenta de apoio, que focou em algumas perguntas que se relacionam com as do primeiro questionário e abordou outros assuntos que pudessem contribuir para a melhoria de todo o projeto.

Esses questionários também foram realizados para verificar, juntamente com as atividades que os alunos realizaram, se as respostas lançadas por eles estavam de acordo com o resultado das atividades, podendo assim, medir o nível de autoavaliação dos mesmos.

3.1.4. Participantes

O projeto foi desenvolvido com 19 alunos do 3º Módulo do curso de Ciência da Computação da UNESP - Universidade Estadual Paulista Júlio de Mesquita Filho, campus de Presidente Prudente, na disciplina de Programação Orientada a Objetos – Turma 2021.

3.2. Procedimentos

Com base nas dificuldades encontradas para ensinar e aprender POO, foram selecionadas algumas metodologias ativas e uma ferramenta de apoio para tentar facilitar o aprendizado dessa disciplina que é responsável por muitas reprovações e desistências de curso.

Primeiramente, foram disponibilizadas na internet (conforme a Seção 3.1.1) os materiais sobre os conceitos de POO. Esses materiais foram lançados

⁶ Google Forms é um aplicativo de gerenciamento de pesquisas lançado pelo Google. Os usuários podem usar o Google Forms para pesquisar e coletar informações sobre outras pessoas e também podem ser usados para questionários e formulários de registro. (Wikipédia).

gradativamente conforme o assunto para que não houvesse confusão no momento de estudar. O professor da disciplina realizava a aula remotamente no formato tradicional, abordando os conteúdos e realizando alguns exercícios de exemplo. O próximo passo foi disponibilizar exercícios para os alunos resolverem e esclarecerem as dúvidas.

A fase seguinte, foi um planejamento de seis etapas para utilização do Robocode conforme a Figura 3.5.

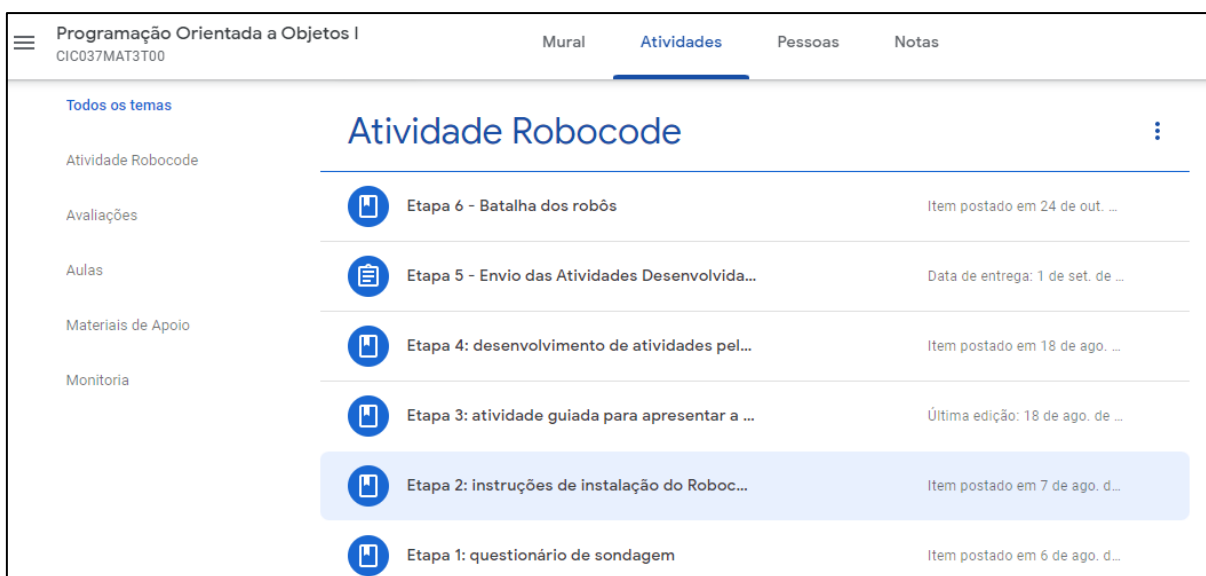


Figura 3.5: Etapas das atividades do RoboCode (Autor, 2021).

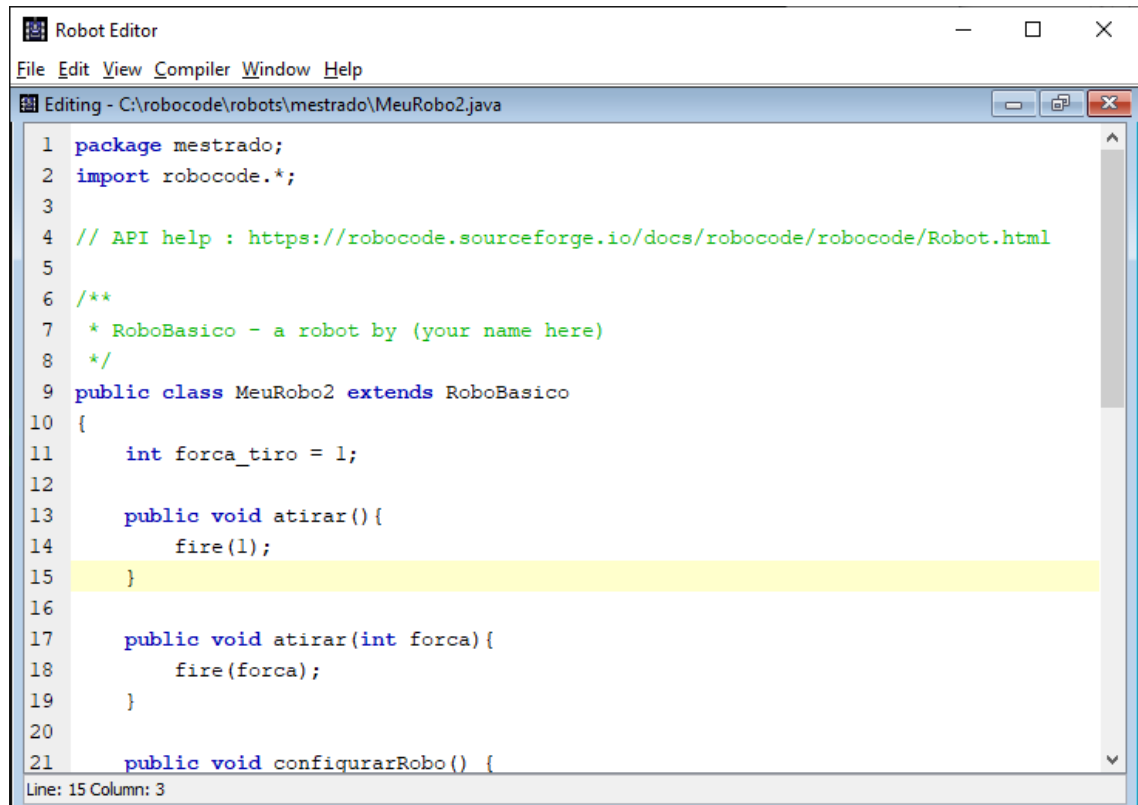
A Etapa 1 foi a aplicação do primeiro questionário de sondagem conforme informado na Seção 3.1.3, em que os dados foram armazenados para análise.

O próximo passo (Etapa 2) foi a disponibilização de materiais para a instalação do RoboCode. Os alunos poderiam instalar a ferramenta RoboCode que possui interface gráfica, porém, se fosse de interesse deles, era possível adicionar a biblioteca do RoboCode no NetBeans⁷ para codificação e mapeamento dos projetos no RoboCode, para que ele reconheça o diretório onde estão os robôs.

Na Etapa 3 foi realizada uma aula síncrona para apresentar o Robocode, algumas funcionalidades como manusear a ferramenta de codificação (Figura 3.6), a criação de alguns robôs (Figura 3.7) e como poderia ser feita a batalha entre os

⁷ O NetBeans IDE é um ambiente de desenvolvimento integrado gratuito e de código aberto para desenvolvedores de software nas linguagens Java, JavaScript, HTML5, PHP, C/C++, Groovy, Ruby, entre outras. O IDE é executado em muitas plataformas, como Windows, Linux, Solaris e MacOS. (Wikipédia).

exemplos de robôs e os que poderiam ser criados pelos alunos. Neste momento, também foram realizadas algumas comparações e dicas de como poderia ser utilizada a POO no RoboCode, fazendo com que a programação pudesse ser mais divertida e instigante.



```

1 package mestrado;
2 import robocode.*;
3
4 // API help : https://robocode.sourceforge.io/docs/robocode/robocode/Robot.html
5
6 /**
7  * RoboBasico - a robot by (your name here)
8  */
9 public class MeuRobo2 extends RoboBasico
10 {
11     int forca_tiro = 1;
12
13     public void atirar(){
14         fire(1);
15     }
16
17     public void atirar(int forca){
18         fire(forca);
19     }
20
21     public void configurarRobo() {

```

Figura 3.6: Codificando um robô (Autor, 2021).

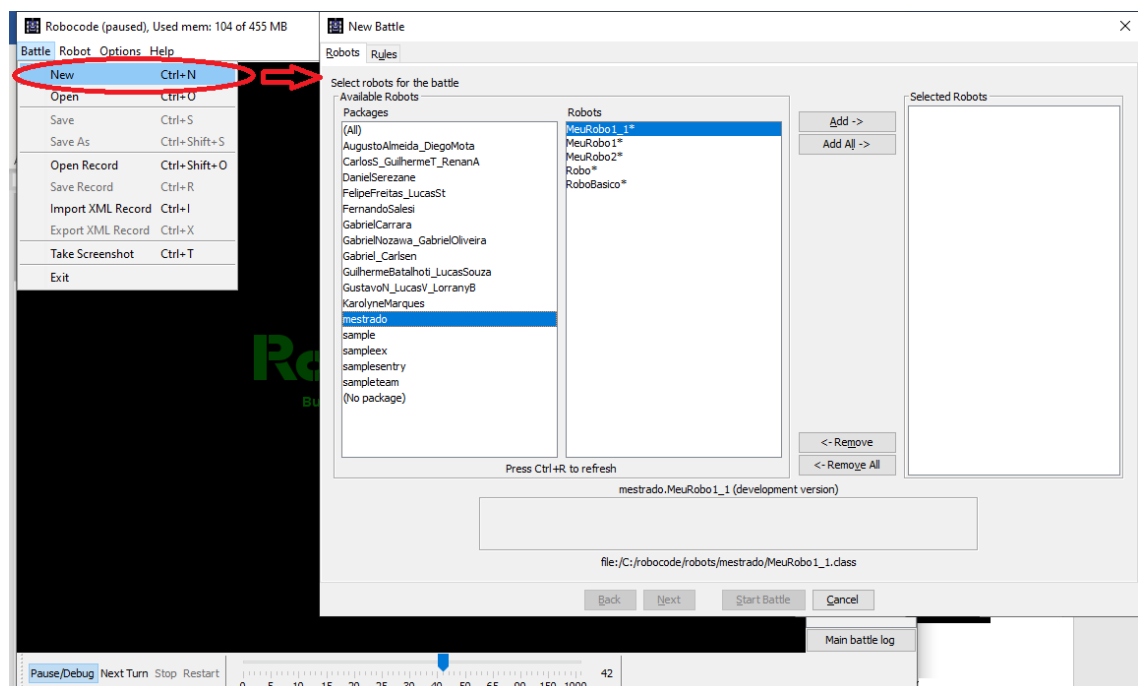


Figura 3.7: Criação de uma batalha (Autor, 2021).

Depois de apresentados todos esses conteúdos, na Etapa 4 foi o momento dos alunos praticarem seus aprendizados e habilidades que adquiriram. Por meio das metodologias ativas Aprendizagem Baseada em Problemas (ABP), Colaborativa e em Grupo, dois projetos foram propostos para serem desenvolvidos em até três alunos e eles deveriam criar dois robôs conforme especificações e regras estabelecidas pelo professor. Além desses dois projetos ocorreu um terceiro que tinha o objetivo de se criar um robô de acordo com a vontade do grupo. Não existia regra, eles poderiam montar seus robôs com as características e funcionalidades que desejassem (Etapa 5). Esses robôs participariam de uma batalha entre os robôs dos outros grupos.

Na Etapa 6, foi realizada a batalha entre os robôs. Foram feitas duas competições, a primeira foi uma batalha geral em que todos os robôs participaram dos combates. Na segunda competição os robôs foram divididos em três grupos de quatro robôs cada. Os dois melhores de cada grupo foram selecionados para a batalha final. Foi um momento de tensão e entusiasmo.

Após essas etapas, o segundo questionário foi realizado para verificar se houve melhoria no aprendizado de POO após a utilização do RoboCode e também as opiniões dos alunos referentes a todo o projeto.

Para o fechamento do projeto, foram feitas várias análises com base nas respostas dos questionários e nos códigos dos projetos desenvolvidos pelos alunos para verificar se realmente as respostas eram condizentes e válidas.

4. RESULTADOS

4.1. Aplicação da metodologia

A aplicação do projeto com as metodologias ativas ocorreu entre agosto e outubro de 2021 e foram utilizadas as seguintes metodologias ativas:

- Aprendizagem Baseada em Problemas (ABP)
- Colaborativa
- Grupo

De acordo com alguns projetos pesquisados (MOURÃO (2017), SANTOS et al. (2015), BOUDIA et al. (2019), COSTA et al. (2017), O'KELLY e GIBSON (2006) e KAILA et al. (2016)) as metodologias ativas demonstram resultados de aprendizagem superiores aos da metodologia tradicional.

Neste trabalho o foco da metodologia ABP foi estimular o estudante a tentar superar suas dificuldades e trabalhar de maneira organizada no desenvolvimento do projeto que exigia o aprendizado do processo e habilidades para resolver problemas reais. A metodologia em Grupo teve o papel de fazer os alunos conseguirem ser uma equipe, trabalhando de forma cooperativa com troca de saberes e habilidades. E para complementar todo esse processo de aprendizagem a metodologia Colaborativa para gerar a motivação necessária para o bom aproveitamento dos estudos, não deixando o desinteresse se apossar da mente do aluno.

No primeiro projeto, exigiu-se o aprendizado dos conceitos básicos de objeto e classe para que fosse possível colocar em prática o desenvolvimento dos robôs juntamente com os assuntos de herança e polimorfismo, desenvolvendo os níveis de aplicação e análise da Taxonomia de Bloom.

O segundo projeto além de trabalhar os mesmos conceitos do projeto anterior, teve o objetivo de fazer com que o aluno desenvolvesse a habilidade de sintetizar outros trabalhos baseados no seu aprendizado e que conseguisse criar soluções próprias.

4.2. Análise do projeto

O primeiro questionário levou em consideração os temas relacionados à programação orientada a objetos. Foram realizadas as seguintes perguntas:

- 1- Como você considera seu aprendizado sobre Classes e Objetos?
- 2- Como você considera seu aprendizado sobre Abstração?
- 3- Como você considera seu aprendizado sobre Encapsulamento?
- 4- Como você considera seu aprendizado sobre Herança?
- 5- Como você considera seu aprendizado sobre Polimorfismo?
- 6- Como você considera seu aprendizado sobre Sobrescrita (*override*) de Métodos?
- 7- Com relação ao aprendizado e prática dos conceitos de Programação Orientada a Objetos (POO), como você avalia os conteúdos e materiais apresentados na disciplina Programação Orientada a Objetos I (POO I)?

Também foi feito um questionamento (pergunta 7) sobre os conteúdos e materiais disponibilizados para verificar possíveis mudanças no intuito de melhorar o processo de aprendizagem.

Como resultado do primeiro questionário aplicado antes da metodologia proposta, obtivemos um levantamento quantitativo com base na percepção de cada aluno, conforme a Tabela 3.1, constatando que os alunos conseguiram aprender os conceitos aplicados.

Questão	Ótimo	Bom	Regular	Ruim	Péssimo
1	8	9	2	0	0
2	1	11	7	0	0
3	3	9	7	0	0
4	5	12	2	0	0
5	2	10	7	0	0
6	7	7	4	1	0
7	14	5	0	0	0

Tabela 3.1: Resultado quantitativo das respostas do primeiro questionário (Autor, 2021).

Por meio das atividades realizadas pelos estudantes na primeira fase de estudos é possível comprovar que as informações coletadas pelo questionário estão condizentes. Alguns alunos ainda apresentam dificuldades em lembrar e entender (1º e 2º nível da Taxonomia Revisa de Bloom) os conceitos estudados.

De acordo com o resultado, percebe-se que os conceitos sobre Classes e Objetos (questão 1) e Herança (questão 4) foram os que tiveram melhor aprendizado dentre os assuntos e que os conceitos sobre Sobrescrita (*override*) de Métodos (questão 6) ficaram com menor aprovação, porém, atingiu nossa expectativa, apresentando valores aceitáveis no processo de aprendizagem.

Os outros assuntos tiveram uma avaliação satisfatória, deixando a sensação de que, apesar dos resultados não terem sido ruins, existe margem para melhoria. Por fim, os conteúdos e materiais disponibilizados na primeira etapa confirmaram que estão abordando de maneira adequada os temas trabalhados.

O segundo questionário foi aplicado após o desenvolvimento das atividades com o RoboCode e teve as seguintes questões:

1- Como você considera o material disponibilizado sobre a Instalação e Configuração do Robocode?

2- Como você considera a orientação disponibilizada sobre a utilização e manuseio do RoboCode?

3- Como você considera o seu aprendizado ao praticar POO com o desenvolvimento de robôs com Robocode?

4- Como você considera o Robocode sendo utilizado como uma ferramenta de aprendizado?

5- Como você classificaria os exemplos de robôs apresentados para exemplificar alguns conceitos de POO?

6- Após a utilização do Robocode, como você considera seu aprendizado sobre Classes e Objetos?

7- Após a utilização do Robocode, como você considera seu aprendizado sobre Abstração?

8- Após a utilização do Robocode, como você considera seu aprendizado sobre Encapsulamento?

9- Após a utilização do Robocode, como você considera seu aprendizado sobre Herança?

10- Após a utilização do Robocode, como você considera seu aprendizado sobre Polimorfismo?

11- Após a utilização do Robocode, como você considera seu aprendizado sobre Sobrescrita (ou override) de Métodos?

12- Considerando as atividades desenvolvidas com o Robocode, qual é a sua avaliação sobre a seguinte afirmação: "Melhorei meu aprendizado sobre os conceitos de POO após desenvolver atividades com o Robocode".

13- Considerando as atividades desenvolvidas com o Robocode, qual é a sua avaliação sobre a seguinte afirmação: "Melhorei meu aprendizado sobre a aplicação de POO após desenvolver atividades com o Robocode".

14- O que poderia ser feito para melhorar seu aprendizado sobre POO?

15- Cite pontos positivos que merecem destaque em todo o processo de ensino de POO.

16- Cite pontos negativos (que devem ser melhorados ou utilizados) para o processo de ensino de POO.

17- Se desejar, adicione outros comentários, sugestões ou contribuições.

A Tabela 3.2 apresenta o resultado das respostas com dados também quantitativos até a questão 13, pois a partir da 14, foram respostas dissertativas.

Questão	Ótimo	Bom	Regular	Ruim	Péssimo
1	13	5	1	0	0
2	8	10	1	0	0
3	6	7	6	0	0
4	13	4	2	0	0
5	8	9	2	0	0
6	11	8	0	0	0
7	5	13	1	0	0
8	9	8	2	0	0
9	11	8	0	0	0
10	8	11	0	0	0
11	11	8	0	0	0
12	6	6	6	1	0
13	6	8	4	1	0

Tabela 3.2: Resultado quantitativo das respostas do segundo questionário (Autor, 2021).

Os materiais sobre a instalação e utilização do RoboCode (questão 1 e 2) foram bem classificados e foi possível confirmar esse resultado, pois quase não foi preciso realizar apoio nesta etapa com os alunos, ou seja, conseguiram fazer a instalação e configuração sozinhos (somente com o material disponibilizado).

As questões 3, 4 e 5 abordaram a utilização do RoboCode e seus aprendizados com base nos exemplos e também teve boas indicações.

Entre as questões 6 e 11 estão focadas as perguntas que se relacionam com as questões de 1 a 6 do primeiro questionário. Para essas perguntas, foi feito um comparativo que pode ser observado na Tabela 3.3. A metodologia tradicional corresponde ao primeiro questionário e as metodologias ativas e RoboCode ao segundo questionário.

Metodologia	Ótimo	Bom	Regular	Ruim	Péssimo
Classes e Objetos					
Tradicional	8	9	2	0	0
RoboCode	11	8	0	0	0
Abstração					
Tradicional	1	11	7	0	0
RoboCode	5	13	1	0	0
Encapsulamento					
Tradicional	3	9	7	0	0
RoboCode	9	8	2	0	0
Herança					
Tradicional	5	12	2	0	0
RoboCode	11	8	0	0	0
Polimorfismo					
Tradicional	2	10	7	0	0
RoboCode	8	11	0	0	0
Sobrescrita (ou override) de Métodos					
Tradicional	7	7	4	1	0
RoboCode	11	8	0	0	0

Tabela 3.3: Comparativo entre as respostas do primeiro e segundo questionário (Autor, 2021).

Neste comparativo nota-se a melhora da percepção dos participantes (alunos) para todos os conceitos em todos os assuntos estudados. Destaque para a Sobrescrita (*override*) de Métodos que teve um baixo rendimento no início no projeto e após esse trabalho teve grande evolução entre os alunos. Esse conceito foi bem trabalhado nas atividades desenvolvidas com o RoboCode.

As respostas apresentadas não poderiam ser válidas sem realmente ter uma constatação, pois os alunos poderiam responder o que fosse de seu interesse, então, as respostas foram comparadas com os trabalhos práticos que foram entregues, comprovando que os resultados e as respostas são equivalentes, garantindo que a autoavaliação dos alunos foi verdadeira.

Quanto ao aprendizado de POO utilizando metodologias ativas e o RoboCode (questões 12 e 13) os alunos indicaram que tiveram bons rendimentos e que puderam compreender os conceitos e práticas de POO.

Observando essas análises podemos responder a primeira pergunta levantada no projeto que é: “Quais estratégias pedagógicas poderiam ser utilizadas para desenvolver habilidades técnicas e de autoavaliação nos alunos de POO?”. Pode ser por meio de uma metodologia de aprendizagem bem planejada com prazos coerentes, seleção de materiais adequados, exemplos que compreendam fatores e qualidades de atividades reais sobre o assunto estudado e uma ferramenta de aprendizado que seja fácil e simples de ser utilizada que favoreça o aprendizado dos conceitos estudados e não de uma determinada linguagem ou da própria ferramenta em si. Também é importante deixar os alunos explorarem suas lacunas (sempre direcionado pelo professor), podendo assim, perceber seus pontos fracos e obter um *feedback* por meio de questionários que os ajudam a refletir sobre suas habilidades e aprendizados.

Para complementar a qualidade do projeto e melhorar o processo de ensino foram aplicadas as questões de 14 a 17 que de forma dissertativa, fizeram apontamentos que contribuíram com o projeto e vão auxiliar em projetos futuros. As questões foram:

14- O que poderia ser feito para melhorar seu aprendizado sobre POO?

15- Cite pontos positivos que merecem destaque em todo o processo de ensino de POO.

16- Cite pontos negativos (que devem ser melhorados ou utilizados) para o processo de ensino de POO.

17- Se desejar, adicione outros comentários, sugestões ou contribuições.

Esses questionamentos mostraram que os alunos ficaram felizes com o desenvolvimento do projeto. Que muitas atividades enriqueceram seus aprendizados e puderam melhorar o aprendizado sobre os assuntos estudados. Suas respostas contribuíram com sugestões ou apontamentos para deixar os projetos futuros mais completos e aperfeiçoar a metodologia aplicada.

Alguns alunos destacaram que a utilização do RoboCode poderia ser feita desde o início do projeto antes de algumas atividades práticas, pois o momento em que foi apresentada reduziu o aproveitamento da ferramenta. Mesmo assim, o RoboCode, na maioria dos casos, ajudou muito o aprendizado de POO. Um outro fator destacado foi a falta do uso do laboratório com o auxílio do professor, que em alguns momentos, favoreceria muito o aprendizado, e que devido à pandemia não foi possível, mas, pelo fato da ferramenta ser intuitiva e divertida, não ficaram desmotivados em progredir nos estudos.

Elogiaram o fato das aulas serem gravadas e disponibilizadas no YouTube para que possam rever as explicações quando ficam com dúvidas ou para realizar alguma atividade e que os exercícios práticos foram de extrema importância para o aprendizado, dizendo ainda que poderiam ter mais exemplos diversificando as funcionalidades ou até demonstrando casos reais de rotinas de trabalho.

Assim, a segunda questão levantada no projeto que foi: “Quais estratégias pedagógicas poderiam ser utilizadas para contribuir com o engajamento (motivação) dos estudantes de POO?” pode ser respondida com base no trabalho em equipe, as atividades com desafios reais e o processo de fazer com que os alunos se envolvessem nas atividades como participantes fundamentais no projeto, sempre auxiliado pelo professor, foi algo que ajudou bastante para manter o entusiasmo e interesse durante todos os trabalhos.

De forma geral, o projeto apresentou resultados favoráveis com alguns detalhes para serem corrigidos, mas agradou os alunos, que, diante das atividades práticas demonstraram ter aprendido os conceitos de POO com mais propriedade.

Todas as etapas realizadas no projeto foram consideradas importantes e necessárias para a percepção da necessidade de correções na metodologia de ensino.

5. CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo apresenta as conclusões que puderam ser extraídas do trabalho aplicado e os trabalhos futuros que poderão ser explorados com base nas experiências adquiridas.

5.1. Conclusões Gerais

Com base nas técnicas e metodologias estudadas nos documentos da revisão sistemática, algumas delas foram utilizadas para a realização do projeto. Assim, foram elaboradas abordagens para cada um dos principais conceitos de programação orientada a objetos, conforme segue, com indicação do nível da hierarquia dos objetivos educacionais da Taxonomia de Bloom Revisada apresentado por ANDERSON e KRATHWOHL (2001):

- **Lembrar e Entender – Classes e Objetos:** essa abordagem utilizou o conhecimento prévio (lembrar) como andaime para auxiliar na aprendizagem de novos conceitos da disciplina de POO, no caso, os principais conceitos de classes e objetos. Para tanto, o conhecimento de diferentes funcionalidades de elementos de jogos de computador foi utilizado para que fossem realizadas associações entre tais elementos e como poderiam ser modelados com conceitos de POO. Assim, o aluno abordou os conceitos de classes, objetos e atributos por meio dessa analogia;
- **Aplicar:** utilizando o ambiente de aprendizagem RoboCode o aluno conseguiu aplicar o que aprendeu.
- **Analisar – Herança e Polimorfismo:** baseado na metodologia de aprendizado por grupo com a aprendizagem baseada em problemas o aluno desenvolveu outras atividades baseadas em herança e polimorfismo;
- **Sintetizar – Aplicação de vários conceitos:** por meio de duas atividades com RoboCode os alunos desenvolveram robôs conforme exigências propostas pelo professor e um último robô que foi desenvolvido conforme a criatividade da equipe, onde os alunos puderam aplicar todos os conceitos praticados.

Após todas essas aplicações os alunos adquiriram as competências e habilidades necessárias para o bom desenvolvimento de POO. Competência pode ser compreendida como um conjunto de três princípios: O saber, a habilidade de saber fazer e o interesse de querer fazer (NAKAO et al, 2012, p. 32).

Também foi possível perceber algumas habilidades durante o projeto:

- Autoavaliação: os alunos demonstraram através das respostas dos questionários que conseguem fazer autoanálise de seus conhecimentos e habilidades;
- Pesquisa: a todo momento eles estavam em constante busca para adquirir mais informações sobre os conteúdos e atividades;
- Habilidades técnicas: por meio da utilização correta dos conceitos nas atividades práticas;
- Soluções para problemas reais: a batalha dos robôs foi um grande desafio que conseguiram superar pois puderam enxergar possíveis problemas e planejar estratégias em seus projetos;
- Engajamento: permaneceram motivados e dedicados durante todos os projetos. Isso pode ser comprovado pela observância de que todos os alunos fizeram todas as atividades, não ocorrendo desistência ou desinteresse pela disciplina.

A aula tradicional quando bem aplicada pode gerar aprendizados significativos e atrair a atenção dos alunos com algumas limitações. Porém, a utilização de metodologias de aprendizagem ativa pode transformar esse cenário proporcionando resultados superiores de aprendizado sem diminuir a motivação.

Para alcançar altos índices de aprendizado é necessário um conjunto de fatores bem organizados e planejados, que não sejam cansativos e monótonos mantendo o engajamento e a vontade de explorar cada vez mais sobre os assuntos estudados.

A preparação do material e como ele será disponibilizado é um fator muito importante e que muitas vezes não é dado o devido valor para essa fase. Cada aluno tem seu tempo e modo de estudar e a disponibilização desses conteúdos devem estar em diversas modalidades, não sendo um fator limitador de

aprendizagem. Mas é necessário revisar alguns exemplos utilizados no projeto e proporcionar mais atividades e de menores proporções para conseguir resultados em períodos mais curtos.

O cronograma também deve ser bem planejado. Como no caso deste trabalho, verificou-se que a utilização do RoboCode deveria ocorrer desde os primeiros contatos com os conteúdos e não somente na etapa final.

A utilização da metodologia em grupo e a aprendizagem baseada em problemas em conjunto com o RoboCode também trouxeram benefícios. As discussões do grupo são momentos de ampliar e compartilhar seus aprendizados. Que os desafios podem ser superados com mais facilidade quando a equipe está em harmonia e buscando alcançar o mesmo objetivo. Apesar da distância (por consequência da pandemia) entre os alunos e os professores ter desfavorecido a oportunidade de compartilharem mais as ideias e diminuir as dúvidas surgidas, ainda assim foi possível atingir níveis de conhecimento muito favorável.

O RoboCode também foi fator fundamental para enriquecer e praticar os conteúdos, sendo simples e prático, não exigindo estudos aprofundados e sem layouts complexos.

Portanto, o projeto demonstrou que a utilização de ferramentas tecnológicas, juntamente com metodologias de aprendizagem ativa, aumenta a capacidade de absorver melhor os conteúdos e dinamizam as atividades práticas, gerando ambientes de trabalhos mais “leves” e menos exaustivos, diminuindo o stress e a desmotivação. Não foi medido, e nem era nosso objetivo, o percentual de aumento de aprendizagem, pois o tempo foi curto e esse foi o primeiro projeto executado, não sendo possível fazer comparativos, porém, será um campo que pode ser explorado em trabalhos futuros.

5.2. Trabalhos futuros

Para futuros trabalhos, propõe-se que sejam testadas outras metodologias de aprendizagem ativa para poder identificar maneiras variadas de evoluir o processo de ensino e aprendizagem de POO.

Também pode ser explorada com mais detalhes a metodologia baseada em problemas, abordando atividades mais diversificadas com exemplos reais nos mais variados ramos de trabalho.

Para aumentar o entusiasmo e sempre pensando na motivação dos alunos, pode-se adotar a metodologia ativa de gamificação, com etapas de conquistas onde cada fase aborde temas específicos e conforme as evoluções dos estudos os alunos recebem titulações de acordo com as habilidades adquiridas.

REFERÊNCIAS

- AFONSO, N. M. M. Da tarefa ao projeto – Uma visão construtivista do ensino da Programação Orientada a Objetos. Universidade do Minho Instituto de Educação, 2013.
- ANDERSON, L. W., KRATHWOHL, D. R. Bloom's Taxonomy Revised - Understanding the New Version of Bloom's Taxonomy. Leslie Owen Wilson, 2001.
- AUSUBEL, D. P., NOVAK, J. D., HANESIAN, H. Psicologia educacional. 2ª edição. Trad. Eva Nick et al. Rio de Janeiro: Interamericana, 1980.
- BERNARDES, C. E., MEDEIROS, L. A. Programação e robótica aplicadas a novos métodos de ensino. Instituto Federal De Santa Catarina (IFSC). Itajaí, 2019.
- BEZERRA, E. V., GOMES, A. S., SOUZA, F. V. C., ALVES, S. V. L. Análise da prática docente com rede social educativa na disciplina de Programação Orientada a Objeto. II Congresso Brasileiro de Informática na Educação (CBIE 2013).
- BITTENCOURT, R. A., RODRIGUES, C. A., CRUZ, D. S. S. Uma Experiência Integrada de Programação Orientada a Objetos, Estruturas de Dados e Projeto de Sistemas com PBL. Universidade Estadual de Feira de Santana (UEFS), 2012.
- BLOMM, B. S., ENGLEHART, M.D., FURST, E. J., HILL, W. H., KRATHWOHL, D. R. Taxonomy of educational objectives. Londen: LongMans, 1956.
- BOUDIA, C., BENGUEDDACH, A., HAFFAF, H. Collaborative Strategy for Teaching and Learning Object-Oriented Programming Course: A Case Study at Mostafa Stambouli Mascara University, Algeria. Informatica: an Internation Journal of Computing and Informatics. Vol. 43, number 1, March 2019. Disponível em: <http://www.informatica.si/index.php/informatica/article/view/2335>. Acesso em: 23/04/2019.
- BRANSFORD, J. D., COCKING, R. R. How people learn: brain, mind, experience, and school. National Academies Press; Expanded ed., 2002.
- BRIGHENTI, J., Biavatti, V. T., SOUZA, T. R. Metodologias de aprendizagem-aprendizagem: uma abordagem sob a percepção dos alunos. Revista GUAL, v. 8, n. 3, p. 281-304. Florianópolis, 2015.
- COSTA, A. F. F., MELO, A. F. M. F., MOREIRA, G. G., CARVALHO, M. A., LIMA, M. V. A., CHAVES, J. O. M. Aplicação de sala invertida e elementos de gamificação para melhoria do ensino-aprendizagem em programação orientada a objetos. Nuevas Ideas en Informática Educativa, vol. 13, p. 223 – 232. Santiago de Chile, 2017.
- COX, B. J. Object-Oriented Programming - An Evolutionary Approach. Addison Wesley Publishing Co., Inc, 1986.

DEITEL, H. M. Java How to Program. 6ª Edition. Prentice Hall, 2004.

DERMEVAL, D., COELHO, J. A. P. de M., BITTENCOURT, I. I. Mapeamento Sistemático e Revisão Sistemática da Literatura em Informática na Educação. In: JAQUES, Patrícia Augustin; PIMENTEL, Mariano; SIQUEIRA, Sean; BITTENCOURT, Ig. (Org.) Metodologia de Pesquisa em Informática na Educação: Abordagem Quantitativa de Pesquisa. Porto Alegre: SBC, 2019. (Série Metodologia de Pesquisa em Informática na Educação, v. 2). Disponível em: <https://metodologia.ceie-br.org/livro-2>. Acesso em: 23/04/2019.

ESTEVES, M., MENDES, A. J. A Simulation Tool To Help Learning Of Object Oriented Programming Basics. 34th ASEE/IEEE Frontiers in Education Conference. October 20 – 23, 2004, Savannah, GA.

EZENWOYE, O. What Language? - The Choice of an Introductory Programming Language. IEEE Frontiers in Education Conference (FIE), 2018. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8658592>.

FARINELLI, F. Conceitos básicos de programação orientada a objetos. Publicação: 2007. Disponível em: <https://www.researchgate.net/publication/268362210>. Acessado em 15/04/2021.

FELIZARDO, K. R., NAKAGAWA, E. Y., FABBRI, S. C.P. F., FERRARI, F. C. Revisão sistemática da literatura em Engenharia de Software: teoria e prática. Rio de Janeiro: Elsevier, 2017.

FERRAZ, A. P. C. M., BELHOT, R. V. Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. Gest. Prod., São Carlos, v. 17, n. 2, p. 421-431, 2010. Disponível em: <https://www.scielo.br/pdf/gp/v17n2/a15v17n2.pdf>

GAETA, C., MASETTO, M. Metodologias Ativas e o Processo de Aprendizagem na Perspectiva da Inovação. PBL 2010 Congresso Internacional. São Paulo, 2010. Disponível em: <http://each.uspnet.usp.br/pbl2010/trabs/trabalhos/TC0287-1.pdf>. Acesso em: 22/10/2019.

GEMIGNANI, E. Y. M. Y. Formação de Professores e Metodologias Ativas de Ensino-Aprendizagem: Ensinar Para a Compreensão. Revista Fronteira das Educação [online], Recife, v. 1, n. 2, 2012. Disponível em: <http://www.fronteirasdaeducacao.org/index.php/fronteiras/article/view/14>. Acesso em: 23/04/2019.

JUNHUA, L. et al. Innovations in Software Engineering Education: An Experimental Study of Integrating Active Learning and Design-based Learning. 120th ASEE Annual Conference & Exposition. Atlanta, 2013.

KAILA, E., KURVINEN, E., LOKKILA, E., LAA, M. Redesigning an Object-Oriented Programming Course. ACM Journals, Transactions on Computing Education. Vol. 16, No. 4. Agosto, 2016.

KITCHENHAM, BARBARA A.; DYBA, TORE; JORGENSEN, MAGNE. Evidence-based software engineering. In: Proceedings of the 26th international conference on software engineering. IEEE Computer Society, 2004. p. 273-281.

KOEHLER, S. M. F., PINTO, A. S. S., BUENO, M. R. P., SILVA, M. A. F. A., SELLMANN, M. Z. Inovação Didática - Projeto de Reflexão e Aplicação de Metodologias Ativas de Aprendizagem no Ensino Superior: uma experiência com "peer instruction". Janus, Lorena, ano 6, n. 15, 1jan./jul., 2012. Disponível em: <http://unifatea.com.br/seer3/index.php/Janus/article/view/289>. Acesso em 23/04/2019.

KÖLLING, M. The Problem of Teaching Object-Oriented Programming, Part 1: Languages. Journal of Object-Oriented Programming, 11 (8). pp. 8-15. 1999.

MARÇAL, I. Mineração de Interesses Transversais: Técnicas para a identificação de interesses transversais em sistemas de software. Presidente Prudente: UNESP, 2014.

MAZAITIS, D. The Object-Oriented Paradigm in the Undergraduate Curriculum: A Survey of Implementations and Issues. SIGCSE BULLETIN. Vol. 25 No. 3 Sept. 1993.

MOURÃO, A. B. Uma proposta da eficiência do uso da Metodologia Ativa Baseada em Problemas, utilizando Dojo de Programação, aplicada na disciplina de Lógica de Programação. VI Congresso Brasileiro de Informática na Educação (CBIE 2017). Disponível em: <https://www.br-ie.org/pub/index.php/wie/article/view/7285/5083>. Acesso em: 23/04/2019.

NAGAI, W. A., IZEKI, C. A. Relato de experiência com metodologia ativa de aprendizagem em uma disciplina de programação básica com ingressantes dos cursos de Engenharia da Computação, Engenharia de Controle e Automação e Engenharia Elétrica. RETEC – Revista de Exatas e Tecnológicas. Volume 4. Rondonópolis, 2013.

NAKAO, O. S., BORGES, M. N., SOUZA, E. P., GRIMONI, J. A. B. Mapeamento de Competências dos Formandos da Escola Politécnica da USP. Revista de Ensino de Engenharia, v. 31, p. 31-39, 2012.

NOVAK, J. D. A. Learning how to learn. Cambridge University Press 1984.

O'KELLY, J., GIBSON, J. P. RoboCode & Problem-Based Learning: A nonprescriptive approach to teaching programming. ACM SIGCSE Bulletin. Vol. 38, No. 3. Junho, 2006.

OLIVEIRA, M. G., NEVES, A., LOPES, M. F. S., MEDEIROS, H. F., ANDRADE, M. B., REBLIN, L. L. Um Curso de Programação a Distância com Metodologias Ativas e Análise de Aprendizagem por Métricas de Software. CINTED-UFRGS, V. 15 Nº 1, julho, 2017.

PAGE-JONES, M. Fundamentals of Object-Oriented Design in UML. Person Education, 2000.

PELIZZARI, A., KRIEGL, M. L., BARON, M. P., FINCK, N. T. L., DOROCINSKI, S. I. Teoria da aprendizagem significativa segundo Ausubel. Rev. PEC, Curitiba, v.2, n.1, p.37-42, jul. 2001-jul. 2002.

RAGONIS, N., BEN-ARI, M. A Long-Term Investigation of the Comprehension of OOP Concepts by Novices. Computer Science Education Vol. 15, No. 3, September 2005, pp. 203 – 221.

RIBEIRO, A. L., BITTENCOURT, R. A., SANTANA, B. L. Análise da Motivação em um Estudo Integrado de Programação Baseado em PBL. UEFS – Universidade Estadual de Feira de Santana. Feira de Santana, 2018.

ROBOCODE. ReadMe for Robocode. Disponível em: <https://robocode.sourceforge.io/docs/ReadMe.html>. Acessado em: 17/10/2021.

ROCHA, A. S., SILVA, J. F. J., CARNEIRO, D. A., SANTOS, J. M. J., SANTANA, B. L., BITTENCOURT, R. A. Utilização do scratch como ferramenta de auxílio à aprendizagem de programação. XLI Congresso Brasileiro de Educação em Engenharia – COBENGE 2013. 23 a 26 de setembro/2013, Gramado/RS.

RUMBAUGH, J., BLAHA, M. Object-Oriented Modeling and Design with UML. Editora Pearson. 2ª Revisão, 2004.

SANTOS, C. S., SANTOS, A. H. M., SOUZA, S. M., SANTOS, D. M. B., BITTENCOURT, R. A. Aprendendo a programação orientada a objetos com uma abordagem lúdica baseada em Greenfoot e Robocode. COBENGE 2015 – XLIII Congresso Brasileiro de Educação em Engenharia. 2015.

SEBESTA, R. W. Concepts of Programming Languages. Editora Person. 9ª Edição, 2010.

SILVA, S. C. R., SCHIRLO, A. C. Teoria da aprendizagem significativa de Ausubel: reflexões para o ensino de física ante a nova realidade social. Imagens da Educação, v. 4, n. 1, p. 36-42, 2014.

SOUZA, C. M. S., BRITO, R. X., XIMENES, J. N. S., SILVA, P; H. A. Estudo de Caso com a Ferramenta Greenfoot Para o Ensino da Programação Orientada a Objetos de Forma Lúdica na Escola Profissional de Viçosa do Ceará. Disponível em: <https://sol.sbc.org.br/index.php/eripi/article/view/5152>. Acesso em: 23/04/2019.

SOUZA, R. R., LEITE, F. T. Usando gamificação no ensino de programação introdutória. Brazilian Journal of Development. DOI:10.34117/bjdv6n6-043, 2020.

TORRES, P. L., ALCANTARA, P. R., IRALA, E. A. F. Grupos de consenso: uma proposta de aprendizagem colaborativa para o processo de ensino-aprendizagem. Revista Diálogo Educacional, Curitiba, v. 4, n.13, p.129-145, set./dez. 2004.

WIKIPEDIA. Programming language. Disponível em: https://en.wikipedia.org/wiki/Programming_language. Acessado em 15/07/2021.

YAN, L. Teaching Object-Oriented Programming with Games. Sixth International Conference on Information Technology: New Generations, 2009.

ZIVIANI, N. Projetos de Algoritmos: com implementações em Java e C++. Editora Cengage. São Paulo, 2015.

APÊNDICE A

A. REVISÃO SISTEMÁTICA

Desde a década de 1990, existem discussões entre comunidades acadêmicas sobre a aprendizagem de programação orientada a objetos (POO). Alguns autores apontam que, com os avanços na indústria de tecnologia da informação, o sofisticado paradigma orientado a objetos, gerou a necessidade de ensinar conceitos muito abstratos à estudantes que estão iniciando na área de desenvolvimento de sistemas. Questões como herança, encapsulamento, polimorfismo, implementação versus interfaces, que eram estudadas depois de conteúdos básicos, tornaram-se fatores essenciais de disciplinas introdutórias à programação. Além disso, normalmente, muitas instituições de nível superior ensinam a programação estruturada antes da orientada a objetos, fazendo com que muitos estudantes confundam a codificação quando utilizam linguagem orientada a objetos (SANTOS et al., 2015).

Existem evidências de que, em média, os estudantes demoram entre 6 e 18 meses para migrar entre os dois paradigmas (KÖLLING, 1999 apud SANTOS et al., 2015). Os estudantes afirmam que os novos conteúdos e a interpretação entre os dois paradigmas abordados por meio da metodologia de ensino-aprendizagem tradicional não são suficientes para compreenderem o básico de POO e manter os alunos motivados, pois, além das linguagens de programação serem muito complexas, os ambientes de desenvolvimento desorientam os estudantes que estão entrando nessa área (SANTOS et al., 2015).

O modelo tradicional de ensino com aulas expositivas, que realça o empoderamento do professor sobre o aluno, surte bons resultados quando o objetivo é transferir conhecimentos iniciais básicos, mas quando se busca o pensamento crítico ou a resolução de problemas, mestres e aprendizes devem mudar suas ações e responsabilidades tradicionais. Assim, os alunos devem ter um envolvimento mais ativo em seu próprio aprendizado, buscando por descobertas a partir de suas ações, ao invés de aceitar tudo que é falado ou lido. Este tipo de abordagem pedagógica é denominado aprendizagem ativa (NAGAI e IZEKI, 2013).

Porém, trabalhar com novas metodologias de aprendizagem está sendo um grande desafio, principalmente para os professores que foram alfabetizados pelo processo tradicional. Assim, é fundamental o estudo aprofundado em metodologias de aprendizagem abordadas na literatura, para obter uma seleção de qualidade e, posteriormente, fazer a seleção desses conteúdos. Por isso, a necessidade de realizar uma revisão sistemática é de suma importância para este trabalho e assim, conseguir atingir o objetivo deste trabalho que é identificar quais os métodos de ensino que são mais eficazes na aprendizagem de Programação Orientada a Objetos (POO).

A.1. Estratégias de Busca

A busca por materiais de qualidade é essencial para que se realize um estudo adequado e com conteúdo bem embasado para a geração de uma revisão sistemática completa, abrangendo o máximo de informações possíveis nas literaturas, para que, no momento do refinamento das pesquisas, obtenha-se resultados precisos.

O objetivo desta revisão sistemática é identificar a(s) metodologia(s) ativa de ensino que tem melhor rendimento para o aprendizado de programação orientada a objetos, além de buscar uma resposta para a seguinte pergunta: Qual é a melhoria de aprendizado que os alunos podem obter ao utilizar metodologia ativa de ensino, quando comparadas com as metodologias tradicionais?

Para a realização da revisão sistemática vários fatores fizeram parte deste processo. Para o fator escopo, deve-se estabelecer um intervalo de busca dos materiais em estudo, em que foram levadas em considerações os trabalhos entre 2000 e 2019.

O fator método refere-se aos meios de busca que serão utilizados na revisão sistemática, que neste caso, foram buscas automáticas nas seguintes bases de dados:

- Scielo
- ACM Library
- Google Scholar

Como termo de busca, foi utilizada a seguinte *string* de busca nos títulos dos artigos:

((metodologia and ativa) or (active and methodology)) or ((programação and orientada and objeto) or (object and oriented and programming))

Dependendo da base utilizada, foi necessário fazer algumas adaptações para obter um retorno mais preciso, como a utilização de termos somente em inglês e limitar as datas das publicações,.

A.2. Seleção

Para a seleção dos materiais encontrados, levou-se em consideração aqueles que explorassem metodologias ativas no ensino técnico e/ou superior, mais especificamente na área de computação e com abordagens em programação orientada a objetos.

Nesses trabalhos, foram analisados se existia o local da instituição, qual metodologia aplicada e em qual disciplina, verificando como foram conduzidas as atividades e se existiu alguma comparação com métodos tradicionais de ensino.

Para a inclusão ou exclusão dos trabalhos, foi verificado se as seguintes perguntas tinham uma resposta positiva:

- 1- O conteúdo do trabalho tem discussões sobre metodologias ativas na área de computação?
- 2- O trabalho teve alguma discussão ou contribuição em pesquisa sobre a metodologia ativa para o ensino de programação orientada a objetos?

A.3. Avaliação da qualidade

A avaliação de cada material selecionado foi realizada por meio de pontuações atribuídas de acordo com as questões criadas no início da revisão sistemática, a fim de encontrar os melhores materiais para o desenvolvimento da pesquisa.

A seguir, são apresentadas as questões que foram utilizadas na revisão sistemática:

- 1- Existe alguma descrição clara sobre metodologias de aprendizagem de programação orientada a objetos?
- 2- O trabalho apresenta aplicações teóricas/práticas sobre metodologias ativas em programação orientada a objetos?
- 3- Foi apresentado um conteúdo coerente sobre o estudo?
- 4- O trabalho apresenta resultados descritos claramente?
- 5- Houve indicações de pontos positivos ou negativos sobre alguma metodologia?

Baseando-se nas perguntas, as respostas foram classificadas como:

Tabela A.1: Pontuação para os documentos selecionados

Classificação	Pontos
Sim	1
Não	0

Cada resposta terá sua pontuação somada para verificar a qualidade do documento selecionado. Ao final da soma, o total obtido seguiu o seguinte critério:

Tabela A.2: Classificação dos resultados

Pontuação	Resultado
0 a 1 ponto	Regular
2 a 3 pontos	Bom
4 a 5 pontos	Muito Bom

Os documentos com resultados classificados como regular, foram descartados, permanecendo na revisão sistemática aqueles que tiveram aceitação como Bom e Muito Bom.

A.4. Extração de dados

Os dados recuperados dos documentos foram essenciais para esclarecer as perguntas apresentadas na revisão. Assim, os dados extraídos correspondes aos questionamentos que está sendo considerado no trabalho de revisão.

Durante a realização desta fase, foram extraídas as informações necessárias para a realização do trabalho, com base no material selecionado na revisão, como:

- Metodologia ativa de ensino utilizada;

- Local ou instituição onde foi aplicado a metodologia;
- Disciplina utilizada para as atividades;
- Como foram conduzidas as atividades;
- Resultado obtido;

A.5. Síntese dos dados

Atividade de síntese é fundamental em uma revisão sistemática, tendo como objetivo principal ajustar os dados extraídos de cada um dos estudos primários considerados. Diversos métodos podem ser aplicados para estabelecer uma estrutura lógica entre os estudos primários e as conclusões geradas pela revisão sistemática (FELIZARDO, 2017).

A partir do material selecionado, foram extraídas algumas informações para que seja necessário responder as questões da revisão sistemática. A seguir, estão as informações extraídas:

- 1- Identificação do estudo
- 2- Resumo
- 3- Referência bibliográfica
- 4- Tipo de estudo (qualitativo/quantitativo)
- 5- Objetivo do estudo
- 6- Modo de aplicação da técnica
- 7- Resultados e discussões

A.6. Considerações Finais

De acordo com o planejado no protocolo deste estudo, foi realizado a busca por documentos que proporcionasse a realização desta revisão sistemática. Inicialmente foram encontrados 1030 documentos que através da análise dos títulos, 112 artigos foram selecionados, mas após a leitura dos resumos/abstract, somente 42 resultaram para a análise geral do conteúdo (Tabela A.3).

Tabela A.3: Quantidade de documentos por fase

Fase	Documentos Excluídos	Quantidade de Documentos Resultantes
Inicial	-	1030
Análise dos títulos	918	112
Leitura dos resumos/abstract	70	42

Após a avaliação dos documentos selecionados para análise do conteúdo, resultaram em 13 documentos classificados como Bom e Muito Bom, conforme apresentado na Tabela A.4 e esses capítulos são apresentados no próximo capítulo.

Tabela A.4: Classificação dos documentos analisados

Classificação	Quantidade de Documentos
Regular	29
Bom	7
Muito Bom	6
Total	42

APÊNDICE B

B. TRABALHOS RELACIONADOS

Com base nos conteúdos apresentados no Anexo A, foi possível identificar técnicas e metodologias de aprendizagem variadas que foram utilizadas em algumas disciplinas na área de computação. Neste apêndice serão abordados detalhes de como as técnicas foram trabalhadas por algumas instituições, podendo assim, ampliar as ideias de utilização das metodologias de aprendizagem.

Conforme o resultado da revisão sistemática detalhada no Apêndice A, abaixo estão listados os documentos que foram selecionados e suas respectivas classificações:

Tabela B.1: Classificação, disciplina e metodologia dos documentos conforme revisão sistemática

Documento	Classificação	Disciplina	Metodologia
MOURÃO (2017)	Muito Bom	Lógica de Programação	Aprendizagem Baseada em Problemas
			Aprendizagem entre Pares
SANTOS et al. (2015)	Muito Bom	Programação Orientada a Objetos	Aprendizagem Baseada em Projetos
			Gamificação
BOUDIA et al. (2019)	Muito Bom	Programação de Algoritmos	Aprendizagem Baseada em Problemas
			Aprendizagem Colaborativa
COSTA et al. (2017)	Muito Bom	Programação Orientada a Objetos	Aprendizagem entre Pares
			Sala de Aula Invertida
			Gamificação
O'KELLY e GIBSON (2006)	Muito Bom	Programação Orientada a Objetos	Aprendizagem Baseada em Problemas
			Aprendizagem Colaborativa

Documento	Classificação	Disciplina	Metodologia
KAILA et al. (2016)	Muito Bom	Programação Orientada a Objetos	Aprendizagem Colaborativa
OLIVEIRA et al. (2017)	Bom	Programação de Algoritmos	Aprendizagem Baseada em Problemas
BEZERRA et al. (2013)	Bom	Programação Orientada a Objetos	Aprendizagem Colaborativa
			Sala de Aula Invertida
BITTENCOURT (2012)	Bom	Programação Orientada a Objetos	Aprendizagem Baseada em Problemas
			Aprendizagem Baseada em Projetos
NAGAI e IZEKI (2013)	Bom	Programação de Algoritmos	Aprendizagem Baseada em Problemas
			Aprendizagem Baseada em Projetos
			Aprendizagem Colaborativa
SOUZA et al. (2017)	Bom	Programação Orientada a Objetos	Aprendizagem Colaborativa
ESTEVES e MENDES (2004)	Bom	Programação Orientada a Objetos	Aprendizagem Baseada em Problemas
ROCHA et al. (2013)	Bom	Programação de Algoritmos	Aprendizagem Colaborativa

Os trabalhos também foram agrupados de acordo com a metodologia ativa que foram aplicadas pelos autores dos documentos selecionados (Tabela B.2), para se analisar com mais clareza e facilidade o quão benéfico cada técnica pode ser no processo ensino-aprendizagem. Alguns trabalhos apresentaram a utilização de mais de uma técnica, com isso, ela foi enquadrada na metodologia de maior enfoque. Assim, com base nos resultados dos trabalhos selecionados, espera-se conseguir

responder a pergunta do nosso objetivo e verificar qual técnica poderá ser utilizada em nosso estudo.

Tabela B.2: Agrupamento dos trabalhos por metodologia ativa

Documento	Metodologia Ativa
BOUDIA et al. (2019) NAGAI e IZEKI (2013) BEZERRA et al. (2013) SOUZA et al. (2017) O'KELLY e GIBSON (2006) KAILA et al. (2016) ROCHA et al. (2013)	Aprendizagem Colaborativa
MOURÃO (2017) BOUDIA et al. (2019) NAGAI e IZEKI (2013) BITTENCOURT (2012) ESTEVES e MENDES (2004) O'KELLY e GIBSON (2006) OLIVEIRA et al. (2017)	Aprendizagem Baseada em Problemas
SANTOS et al. (2015) NAGAI e IZEKI (2013) BITTENCOURT (2012)	Aprendizagem Baseada em Projetos
MOURÃO (2017) COSTA et al. (2017)	Aprendizagem entre Pares
BEZERRA et al. (2013) KAILA et al. (2016) COSTA et al. (2017)	Sala de aula invertida
SANTOS et al. (2015) COSTA et al. (2017)	Gamificação

B.1. Aprendizagem entre pares

MOURÃO (2017) utilizando a técnica de Instrução entre pares com algumas características de Aprendizagem Baseada em Problemas (ABP), aplicou-as em duas instituições de ensino superior: Centro Universitário do Norte - UNINORTE (particular) e Universidade do Estado do Amazonas - UEA (pública). A metodologia foi aplicada em turmas de cursos de Computação, na disciplina de Lógica de Programação.

MOURÃO (2017) diz que a programação entre pares é uma técnica cujas características envolve o trabalho colaborativo entre dois alunos que compartilham o mesmo computador para desenvolver a solução. Nessa abordagem, acontece diálogos e programação síncrona entre eles, com o objetivo de resolver um problema.

Por meio dessa técnica, os alunos mantêm a atenção na atividade, esclarecem ideias, e quando um deles não consegue evoluir, o outro assume o comando e ambos continuam com os pensamentos alinhados com as práticas (MOURÃO, 2017).

No final da atividade, os alunos explicam para os colegas de sala o que foi desenvolvido, as dificuldades encontradas, estratégias utilizadas e a experiência vivida (MOURÃO, 2017).

A Figura 4.1 apresenta o resultado em valores percentuais do desempenho das turmas de ambas as instituições de ensino (MOURÃO, 2017).

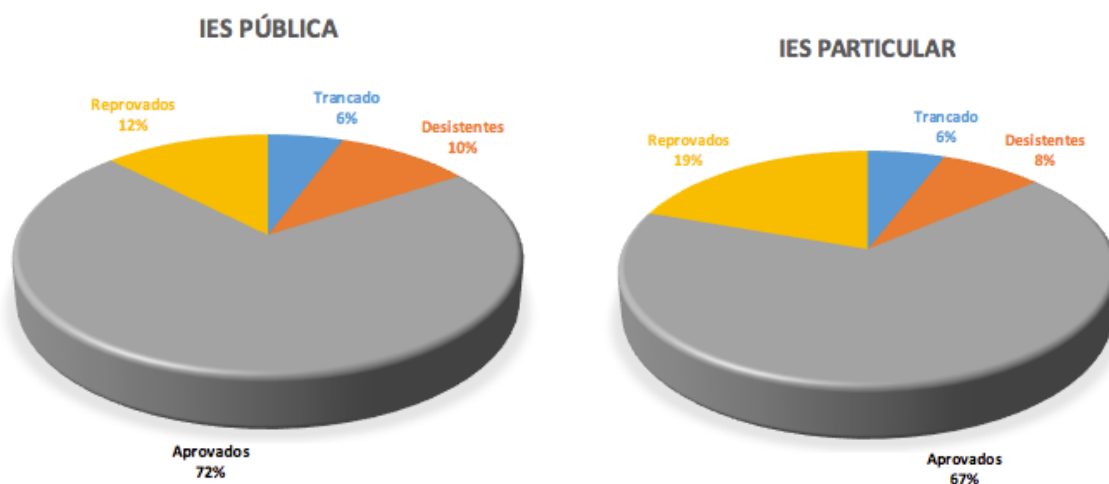


Figura B.1: Percentual de Aprovação com o uso da Metodologia (MOURÃO, 2017)

MOURÃO (2017) apresentou que o desempenho dos alunos está melhorando gradativamente na disciplina de Lógica de Programação, principalmente se for comparar com os alunos que não tiveram contato com nenhuma metodologia (Tabela B.3).

Tabela B.3: Desempenho dos alunos – reproduzido de (MOURÃO, 2017)

Metodologia	Aprovação
Tradicional	40 a 45%
Aprendizado entre Pares	67 a 72%

B.2. Sala de aula invertida e gamificação

COSTA et al. (2017) apresentaram um trabalho realizado por um professor do Campus da Universidade Federal do Ceará (UFC) na cidade de Russas, que analisou o desempenho e comportamento dos alunos no contexto de ensino-aprendizagem na disciplina de Programação Orientada a Objetos (POO).

COSTA et al. (2017) informam que a instituição, buscando melhorar o processo de ensino-aprendizagem, replanejou suas metodologias, criando diversos conceitos e estimulação de estudo. Os materiais de apoio variavam entre textos, vídeos, lista de exercícios, páginas de internet, tudo isso, disponibilizado num Ambiente Virtual de Aprendizagem (AVA), o Moodle⁸. A partir desse ambiente, criou-se o Ambiente Multimeios de Ensino-Aprendizagem (AME), com o qual, por meio de um estudo híbrido (presencial na escola e virtual em casa), os alunos poderiam melhorar seus conhecimentos e habilidades, fazendo um estudo prévio dos materiais disponibilizados no ambiente *online* em qualquer dispositivo e local, juntamente com o método de ensino Instrução entre Pares.

COSTA et al. (2017) explicaram que para deixar o aprendizado mais centrado no estudante do que no instrumento avaliativo, o professor optou por não aplicar provas, como era realizado na metodologia anterior. Ao invés disso, ele estabeleceu diversas atividades para serem realizadas no ambiente *online* e/ou nos encontros presenciais. Além disso, diversos elementos de jogos (Gamificação) foram inseridos no AME desde a sua concepção, como:

⁸ Moodle: <https://moodle.org/>

- **Pontos de Experiência (XP):** um valor numérico que representa o progresso do aluno durante a disciplina.
- **Níveis de Emblemas:** de acordo com o valor acumulado de XP (experiência), o aluno pode ser “promovido” a um nível mais elevado (Figura B.2).
- **Emblemas Especiais:** distribuídos aos alunos que realizam algumas atividades diferenciadas (Figura B.3).
- **Atividades Colaborativas:** cada aluno pertence a um grupo (ou clã) e, durante a disciplina, diversas atividades são propostas para serem realizadas por eles, induzindo a um trabalho cooperativo.
- **Ranking:** há dois tipos de ranking no AME, um individual e outro coletivo. O individual é baseado no XP de cada aluno, apresentando sua posição que ele se encontra em relação aos demais. Já o ranking de grupos é baseado na soma do XP individual de cada membro do grupo.
- **Dificuldade crescente das atividades:** ao passo que a disciplina progride, a quantidade e a dificuldade das atividades aumentam para que os alunos mantenham a motivação e continuem estudando, sem perder o foco.



Figura B.2: Níveis iniciais e seus emblemas (COSTA et al., 2017)



Figura 2.3: Emblemas Especiais (COSTA et al., 2017)

Nesse estudo, verificou-se que a aplicação do método alcançou resultados significativos pois o objetivo inicial era encontrar métodos para que os alunos pudessem absorver com eficiência o conteúdo e ter resultados positivos durante o estudo da disciplina. Isso foi comprovado conforme mostra os dados obtidos das reprovações e aprovações (Figura B.4) (COSTA et al., 2017).

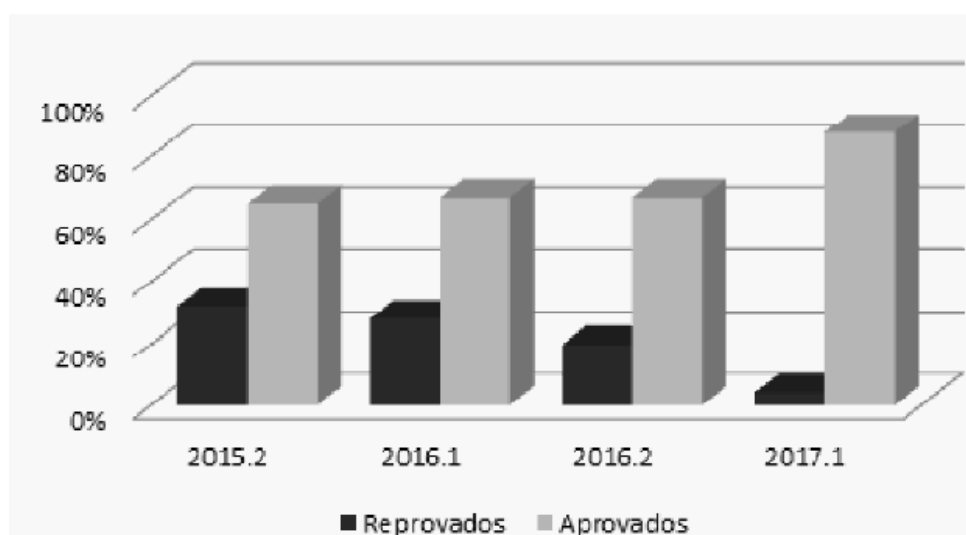


Figura B.4: Índice de aprovações e reprovações em turmas de programação orientada a objetos na Universidade Federal do Ceará campus de Russas (COSTA et al., 2017)

A média das notas teve um considerável aumento de 45% em relação aos semestres anteriores e os resultados obtidos são apresentados na Figura B.5 (COSTA et al., 2017).

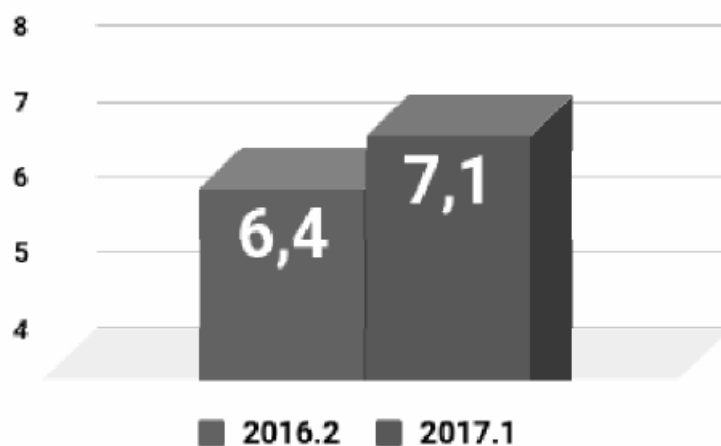


Figura B.5: Média das notas das turmas após a aplicação das metodologias (COSTA et al., 2017)

B.3. Aprendizagem baseada em projeto

Em um trabalho realizado na Universidade Estadual de Feira de Santana, BITTENCOURT (2012) utilizou a metodologia de aprendizagem baseada em projetos (ABP) juntamente com um Estudo Integrador (EI), buscando melhorar o processo de ensino-aprendizagem das disciplinas de Programação Orientada a Objetos, Estrutura de Dados e Projeto de Sistemas no curso de Engenharia da Computação, tendo como objetivo capacitar o aluno a projetar e desenvolver sistemas orientados a objetos em Java, demonstrando domínio dos conceitos de algoritmos e estrutura de dados e ferramentas utilizadas.

O planejamento do EI de Programação segue o princípio de evolução do ciclo em espiral com níveis de dificuldade crescente, exercitando as habilidades juntamente com os conhecimentos até alcançarem as competências necessárias. Propuseram problemas práticos baseados em situações reais, com tempo apropriado que permita a absorção de conhecimentos e a prática de habilidades de programação (BITTENCOURT, 2012).

Ao final, o tutor repassa um *feedback* aos estudantes do comportamento de cada um e recebe o projeto final (código-fonte e documentação), juntamente com um artigo, para que os alunos criem o hábito de escrever com coerência seus trabalhos e que possam distinguir suas informações em referências bibliográficas apropriadas (BITTENCOURT, 2012).

As principais lições aprendidas foram (BITTENCOURT, 2012):

- Ao agregar conhecimento é possível adquirir experiências verdadeiras e práticas de produção de software mais disciplinadas;
- A metodologia ABP possibilita adquirir competências mais amplas de comunicação, trabalho em equipe e autodidatismo;
- A grande quantidade de problemas a ser corrigido num software gera desmotivação, devendo ser evitados;
- A abordagem de novos conceitos nos problemas propostos deve seguir um processo gradativo de acordo com capacidade de assimilação dos alunos.

Em outro trabalho identificado na revisão sistemática, SANTOS et al. (2015) que, juntamente com alunos do curso de Engenharia da Computação da Universidade Estadual de Feira de Santana (UEFS), utilizando a metodologia de Aprendizagem Baseada em Projeto (ABP), trabalharam em uma oficina de aprendizagem de Programação Orientada a Objetos (POO).

Durante as atividades, SANTOS et al. (2015), optaram por utilizar como apoio a ferramenta *Greenfoot* que é uma plataforma de desenvolvimento de aplicativos com foco no aprendizado de programação orientada a objetos, possuindo ferramentas de codificação, máquina virtual (VM), compilador, entre outras. Importantes conceitos de POO como classes, encapsulamento e níveis de abstração podem ser explorados na ferramenta. Assim, optaram por utilizar a linguagem Java (SANTOS et al., 2015).

Também fizeram uso do *Robocode* que é outra plataforma de desenvolvimento integrado para o ensino de programação por meio de programação orientada a eventos. Num pequeno espaço virtual, o aluno cria uma classe de um tanque de guerra utilizando a linguagem Java ou C#, que batalharão com outros robôs em tempo real (SANTOS et al., 2015).

O processo de ensino-aprendizagem combinou alguns tutoriais sobre conceitos de POO, instruções de uso da ferramenta e uma metodologia de ensino utilizando jogos e competições. Progressivamente, os professores sugeriam atividades aos alunos, que resolviam utilizando um dos ambientes lúdicos. Conforme as atividades eram executadas, os tutores ou monitores, esclareciam as dúvidas que surgiam (SANTOS et al., 2015).

A aprendizagem foi centralizada no estudante, deixando o aluno mais a vontade, podendo assim, cada um aprender de modo particular a resolução dos problemas no seu tempo, tendo a intervenção dos monitores, quando realmente era necessário, para tirar dúvidas ou evitar a impossibilidade de continuidade (SANTOS et al., 2015). Os autores destacaram algumas lições da experiência que vivenciaram, como:

Exemplos motivantes: Quando a atividade era longa, tinha a necessidade de motivá-los para manter a atenção constante.

Atenção dividida: Perceberam que em alguns momentos, alguns alunos se dispersaram, principalmente quando era dado algum tempo para analisar e modificar sozinhos o código-fonte.

Alunos e professores com bom nível de relacionamento: Utilização de exemplos descontraídos, criaram atividades divertidas, gerando um ambiente saudável para desenvolver as atividades.

Desafios e competições: Competição feita no *Robocode*, fez com que os alunos se dedicassem no desenvolvimento dos robôs, envolvendo todos na hora do combate, sendo um dos motivos da escolha do *Robocode* como a ferramenta de maior aceitação do que o *Greenfoot*.

Em uma análise mais geral, SANTOS et al. (2015), perceberam que, por meio de uma avaliação usando questionários, os participantes avaliaram a oficina de forma positiva, com média acima de 4, exceto o aspecto cansativa/leve, talvez pelo fato da oficina ser bastante intensiva.

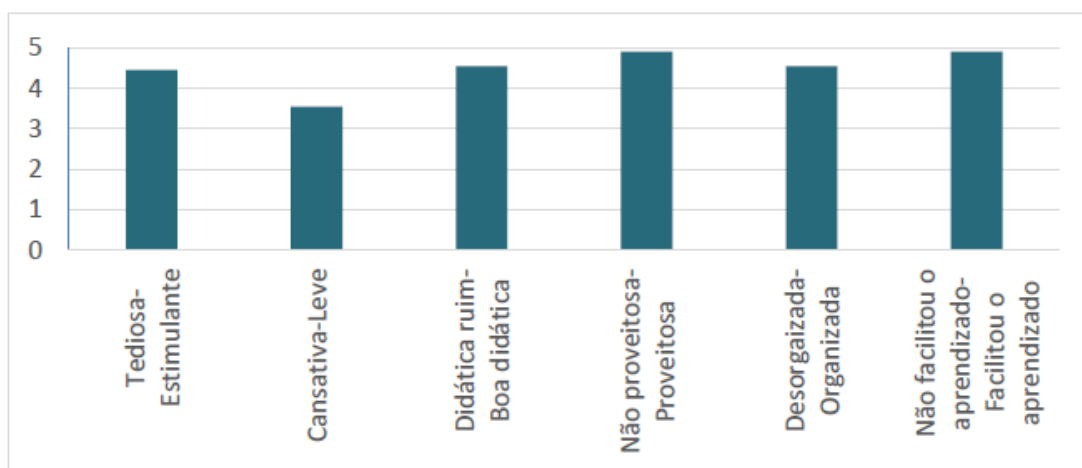


Figura B.6: Avaliação da oficina (SANTOS et al., 2015)

B.4. Aprendizagem baseada em problema

O trabalho de NAGAI e IZEKI (2013) na Universidade Federal de Itajubá (UNIFEI) - Campus Itabira, foi realizado nos cursos de graduação em Engenharia da Computação e Engenharia de Controle e Produção, que cursavam a disciplina de programação de algoritmos utilizando a linguagem C/C++ em um ambiente integrado de desenvolvimento chamado *CodeBlocks*. Os autores propuseram aos discentes das aulas teóricas e práticas alguns problemas em forma de projetos. Cada um destes problemas possuía objetivos diferenciados com o intuito de trabalhar tópicos da disciplina.

Segundo NAGAI e IZEKI (2013), foi montado um cronograma de realização das atividades, incluindo tópicos de conteúdo, projetos, desafios, avaliações individuais e o exame. Os projetos propostos na disciplina possuíam um cenário de pequenas simulações de problemas reais da área de Computação. Os estudantes poderiam se preparar fazendo questionamentos sobre os tópicos das disciplinas antes dos desafios práticos propostos, em que deveriam entregar o código-fonte e um relatório textual do projeto desenvolvido, além de, um integrante da equipe, ter que participar em uma entrevista por escrito sobre o projeto.

NAGAI e IZEKI (2013) relatam que ficaram satisfeitos com o resultado obtido, pois se tratou de uma primeira experiência com aprendizagem ativa baseada em problemas com um caráter cooperativo (Tabela B.4).

Tabela B.4: Taxa de aprovação – reproduzido de (NAGAI e IZEKI, 2013)

Metodologia	Aprovação
Tradicional	45%
ABP	68%

ESTEVES e MENDES (2004) apresentaram um trabalho em que desenvolveram uma ferramenta para auxiliar os alunos no processo ensino-aprendizagem de programação orientada a objetos chamada OOP-Anim.

A principal decisão inicial foi definir quais atividades do aluno o ambiente deveria apoiar. Afirmam que a programação é essencialmente uma atividade de solução de problemas e acreditam que a atividade de aprendizagem mais importante para os alunos é a participação ativa na solução de problemas de programação (ESTEVES e MENDES, 2004).

ESTEVES e MENDES (2004) dizem que demonstrações e explicações são importantes, especialmente nos estágios iniciais de aprendizagem, mas tentar resolver problemas, corrigir erros e melhorar a solução é a atividade de aprendizado mais importante nesse campo.

Consequentemente, a decisão que tiveram foi criar um ambiente flexível que pudesse apoiar os alunos nessas atividades. Não disponibilizaram conteúdo teórico para os alunos lerem, mas o ambiente OOP-Anim permite que insiram sua solução para um problema, simulem e animem para visualizar como funciona, corrigem se necessário e simulam novamente até que uma solução de trabalho seja alcançada, utilizando os conceitos da linguagem Java (ESTEVES e MENDES, 2004).

O ambiente foi desenvolvido essencialmente para apoiar o aprendizado básico de POO. No entanto, para dar algum apoio aos alunos mais avançados, incluíram a possibilidade de animar qualquer método pertencente a qualquer uma das aulas do programa. Isso permite que os estudantes usem o OOP-Anim ao criar soluções maiores e mais complexas. Eles podem se concentrar em qualquer método de forma independente, para que possam se concentrar nas partes mais críticas de suas soluções ou em qualquer método que não funcione conforme o esperado (ESTEVES e MENDES, 2004).

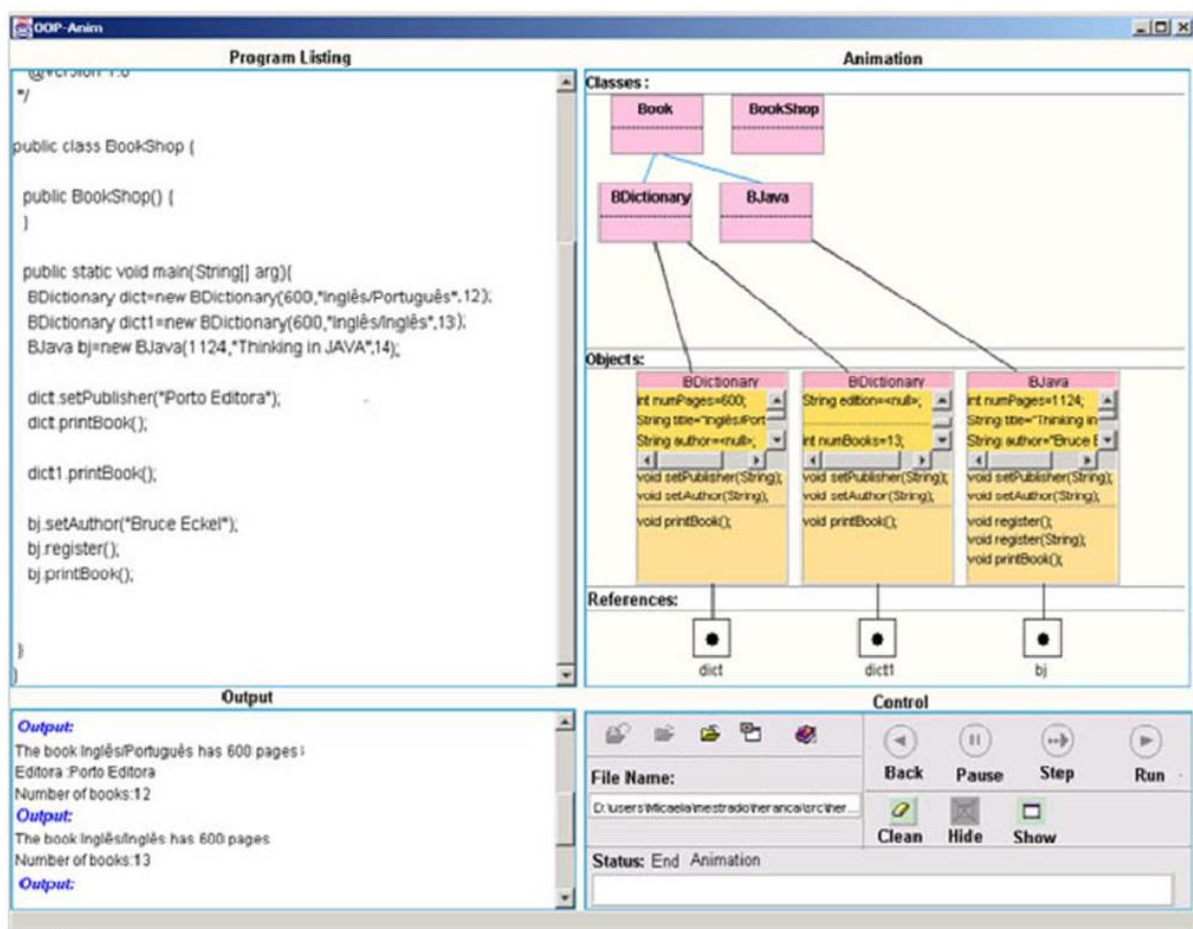


Figura B.7: Interface OOP-Anim (ESTEVEES e MENDES, 2004)

ESTEVEES e MENDES (2004) finalizam dizendo que o OOP-Anim é um ambiente educacional que pretende ajudar o aluno a aprender e dominar os conceitos básicos de programação orientada a objetos. O aluno pode criar programas para resolver problemas de programação possivelmente propostos por um professor, permitindo que o aluno acompanhe de perto a execução. Esse ambiente pode ser usado por estudantes sozinhos ou em grupo, nas aulas ou no auto estudo. Isso pode possibilita que os alunos mais fracos, muitas vezes com dificuldades para lidar com o ritmo da aula, melhorem seus conhecimentos e experiências.

Em outro trabalho, apresentado por OLIVEIRA et al. (2017), foi proposto o uso de metodologias de aprendizagem ativa orientada a problemas, composta por uma turma bem heterogênea, envolvendo alunos de ensino médio, de graduação e de pós-graduação. A turma foi composta por 25 alunos do Curso a Distância de Programação C Essencial e Avançada no Centro de Referência em Formação e Educação a Distância do Instituto Federal do Espírito Santo (IFES).

Os autores utilizaram a estratégia de preparar os alunos para construir programas a partir de atividades envolvendo as principais habilidades de programação como a compreensão, a observação de detalhes, a análise, a sequenciação e a abstração, apoiando-se fortemente nos fóruns que envolviam todos os alunos e nos materiais e atividades do curso para ambientes *online* (OLIVEIRA et al., 2017).

Inicialmente, foram realizados alguns experimentos que tinham o objetivo de diagnosticar como os alunos programavam, identificando dificuldades, boas práticas e até plágios de programação, por meio de análises dos códigos-fonte que eles desenvolveram (OLIVEIRA et al., 2017).

Optaram por escolher exercícios que utilizassem estruturas de repetição, de estruturas condicionais e, principalmente, de expressões lógicas. Segundo seus critérios, uma boa solução desse exercício seria utilizar no máximo de três a cinco comparações, enquanto uma solução ruim utilizaria muitas comparações, mesmo que o programa funcionasse corretamente (OLIVEIRA et al., 2017).

OLIVEIRA et al. (2017), destacaram como pontos positivos, de acordo com os relatos dos alunos:

- fácil introdução dos conteúdos, mesmo para os alunos inexperientes em programação;
- as atividades;
- as discussões colaborativas;
- os exercícios individuais práticos e teóricos.

Em relação aos pontos negativos, apontaram (OLIVEIRA et al., 2017):

- algumas críticas em relação aos prazos das disciplinas;
- do cronograma do curso;
- em relação ao material instrucional, mas praticamente não houve críticas em relação às metodologias utilizadas nas atividades do curso.

O'KELY e GIBSON (2006) buscaram em seu trabalho, realizado na Universidade Nacional da Irlanda em Maynooth (NUIM), implementar uma

abordagem de aprendizagem baseada em problemas para os alunos do primeiro ano do curso de Ciências da Computação no ano acadêmico de 2003-2004.

Como o problema é um componente tão crítico do processo de aprendizagem, é imperativo que se use bons problemas. Sendo assim, foram identificadas cinco características do que torna um problema de ABP bom (O'KELY e GIBSON, 2006):

1. Os problemas efetivos devem envolver o interesse dos alunos e motivá-los a buscar uma compreensão mais profunda dos conceitos que estão sendo introduzidos.
2. Os problemas de ABP devem ser projetados com vários estágios.
3. Os problemas devem ser complexos o suficiente para que a cooperação de todos os membros do grupo seja necessária para que eles trabalhem efetivamente em direção a uma solução.
4. O problema deve ser aberto.
5. Os objetivos do conteúdo do curso devem ser incorporados aos problemas.

Um dos principais obstáculos à implementação da ABP em qualquer disciplina é a falta de um bom conjunto de problemas. No entanto, com exceção de algumas disciplinas, bons problemas de ABP geralmente não aparecem nos livros didáticos. As câmaras de compensação fornecem uma via para permitir o compartilhamento de problemas, mas, infelizmente, faltam problemas de ciência da computação. Isso pode ser parcialmente explicado pelo pequeno número de praticantes relatados de ABP na disciplina de ciência da computação (O'KELY e GIBSON, 2006).

A fim de melhorar a retenção na universidade, tentaram criar um ambiente divertido e educacional, sendo escolhido uma competição de *Robocode*. A atividade passou a ser uma "competição divertida", tornando-se desafiador e difícil (O'KELY e GIBSON, 2006).

Para O'KELY e GIBSON (2006), um dos recursos do *RoboCode* é que os resultados dos esforços podem ser vistos instantaneamente, essa "gratificação instantânea" acelera o ciclo de aprendizado do desenvolvimento e é análoga ao "modelo de prototipagem" usados no desenvolvimento de software.

O processo começa com a coleta de requisitos, em que o desenvolvedor e o cliente atendem e definem os objetivos gerais do software. Um planejamento rápido resulta na implementação de um protótipo, que é avaliado e ajustado pelo usuário até obter o software desejado. Essa estrutura de processo é análoga à maneira como as equipes do *RoboCode* trabalham para desenvolver seu “produto” final. Os alunos ficam encantados com sua capacidade de tornar o tanque-robô “divertido”, “estranho”, etc (O’KELY e GIBSON, 2006).

Os problemas devem ser complexos o suficiente para que a cooperação de todos os membros do grupo seja necessária. Um problema não apenas incorpora explicitamente os objetivos do conteúdo do curso, mas também desafia os alunos a ir além deles. O problema ativa o conhecimento prévio - o conhecimento de experiências anteriores, que todos trazem para a sua aprendizagem, e o conhecimento da programação que os alunos abordaram no 1º semestre, que para alguns alunos é “frágil”. Mas, os alunos “fortes”, podem explicar quais estratégias eles usaram para resolver um problema e porque eles os usaram. A habilidade de trabalhar de maneira eficaz em um ambiente de equipe é um requisito na economia global de hoje. Uma das principais vantagens desse problema é a coesão que ele traz para a equipe, o que aumenta a qualidade do seu trabalho (O’KELY e GIBSON, 2006).

O’KELY e GIBSON (2006), concluem dizendo que a competição é uma ocorrência cotidiana no mundo real e problemas efetivos na ABP enfatizam esse aspecto do mundo real e, o problema apresentado na competição do *Robocode*, combina elementos de diversão, programação, jogos, IA e competição, transformando o conhecimento frágil em uma habilidade transferível concreta que pode ser aplicada em novas situações.

Na ABP, o foco é deslocado do ensino para o aprendizado, e essa mudança em conjunto com um bom problema (*RoboCode*) fornece a cada aluno a liberdade de pensar por si mesmo, ativar seu conhecimento prévio e adquirir novos conhecimentos de maneira exploratória e criativa (O’KELY e GIBSON, 2006).

B.5. Aprendizagem colaborativa

No estudo realizado por BOUDIA et al. (2019) na Mostafa Stambouli Mascara University, na Argélia, os objetivos foram demonstrar o impacto e a eficácia da

implementação do grupo colaborativo na conquista da aprendizagem. Caracterizar a natureza do comportamento do aluno por meio de diferentes funções na programação de aprendizagem, se as técnicas de ensino são úteis para os alunos no ensino presencial e a distância, e realizar um estudo comparativo em diferentes formas de aprendizagem, com o envolvimento de professores e alunos.

Inicialmente, 108 alunos foram divididos aleatoriamente em 3 grupos e que após alguns testes, cada aluno recebeu um papel a ser executado dentro da atividade. Determinaram uma variedade de papéis que os alunos deveriam conhecer como programadores e como membros do grupo. As funções não são atribuídas, mas são executadas após a discussão (BOUDIA et al., 2019).

As funções dos alunos são a peça principal no projeto de aprendizado colaborativo, porque quando todos os membros do grupo discutem e realizam uma tarefa, o aluno assume seu trabalho com determinação. As funções deveriam alternar no final do exercício, a fim de permitir que todos os alunos pudessem desempenhar e experimentar todas as funções, para que todos se revezem e, conseqüentemente, todos os grupos devam dominar. Os alunos poderiam concluir mais de uma função por vez, se necessário (BOUDIA et al., 2019).

Os alunos devem resolver cada atividade de acordo com as seguintes estratégias de resolução:

- Fase 1 (trabalho individual): os alunos tentam ler e entender a declaração do exercício, após que o professor explicar a finalidade do exercício, fornecer alguns detalhes e ajustar o cronômetro por 20 minutos.
- Fase 2 (trabalho individual): os alunos adotam uma estratégia de resolução e tentam resolver o problema parcialmente ou totalmente.
- Fase 3 (trabalho colaborativo): envolve 4 etapas:
 - 3.1 Discussão (análise de problemas)
 - 3.2 Escolha da estratégia de resolução e algoritmos de escrita (concepção)
 - 3.3 Implementação (codificação)
 - 3.4 Depurando e testando o programa final.

- Fase 4: o professor interrompe o trabalho colaborativo, seleciona um código de qualquer grupo e pede ao treinador do grupo que o escreva ou apenas fragmentos no quadro.
- Fase 5: o professor inicia uma discussão com os alunos sobre o código, encontra bugs e modifica-os juntos, depois pede aos alunos que digitem novamente na máquina para exibir os resultados.
- Fase 6 (Trabalho individual): escrevendo notas.

Para BOUDIA et al. (2019), os resultados obtidos permitem afirmar que o design de grupo, o cenário pedagógico e, principalmente, a divisão de papéis combinados com as ferramentas tecnológicas são a metodologia pedagógica produzida nos alunos. Além disso, o engajamento, o comprometimento e a motivação dos alunos aumentaram em cada fase do processo de programação.

A assistência do professor foi muito importante, ajudando os grupos de estudantes a aceitar *feedback* crítico e construtivo, reforçando sua aprendizagem interna e externamente fora da sala de sessões do laboratório. Além disso, dividir tarefas e papéis tornou o aprendizado menos difícil e mais agradável; os alunos com alta determinação continuam a trabalhar em grupos em outros cursos, pois constroem confiança e são membros sólidos (BOUDIA et al., 2019).

No trabalho realizado por BEZERRA et al. (2013), foi adotado uma abordagem qualitativa por meio da metodologia de aprendizagem colaborativa, com base num estudo de caso com professores da disciplina de Programação Orientada a Objeto do curso de Licenciatura em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins - campus Porto Nacional, em 2012.

Utilizando a Rede Social Educacional (REDU), que é uma rede social educativa desenvolvida para ser um apoio no processo de colaboração e aprendizagem. O ambiente expõe uma nova visão de plataforma de ensino, utilizando o poder e o conjunto de características das redes sociais, buscando deixar o mais parecido possível com o modo que as pessoas utilizam, tendo um visual simples, intuitivo e fácil de usar (BEZERRA et al., 2013).

Com propósito educativo, o REDU, possui ferramentas com possibilidade de medir a comunicação e a colaboração em rede, dispondo ainda, de soluções para elaboração de atividades, coordenação, compartilhamento de materiais didáticos, (auto)avaliação e (auto)regulação da aprendizagem (BEZERRA et al., 2013).

Os professores utilizaram no REDU um recurso educacional, apresentado na Figura 4.8, chamado *Compilr*, que é um editor de código online interativo baseado na web, permitindo que se trabalhe em linguagens interpretadas populares como PHP, Ruby, Javascript e Python, bem como em linguagens compiladas como C++, C#, Java, .Net e Objective-C. Através deste recurso o estudante tem a possibilidade de visualizar, estudar o conteúdo da disciplina e programar sem precisar utilizar ou instalar uma IDE (Interface de Desenvolvimento) específica em seu dispositivo (BEZERRA, 2013).

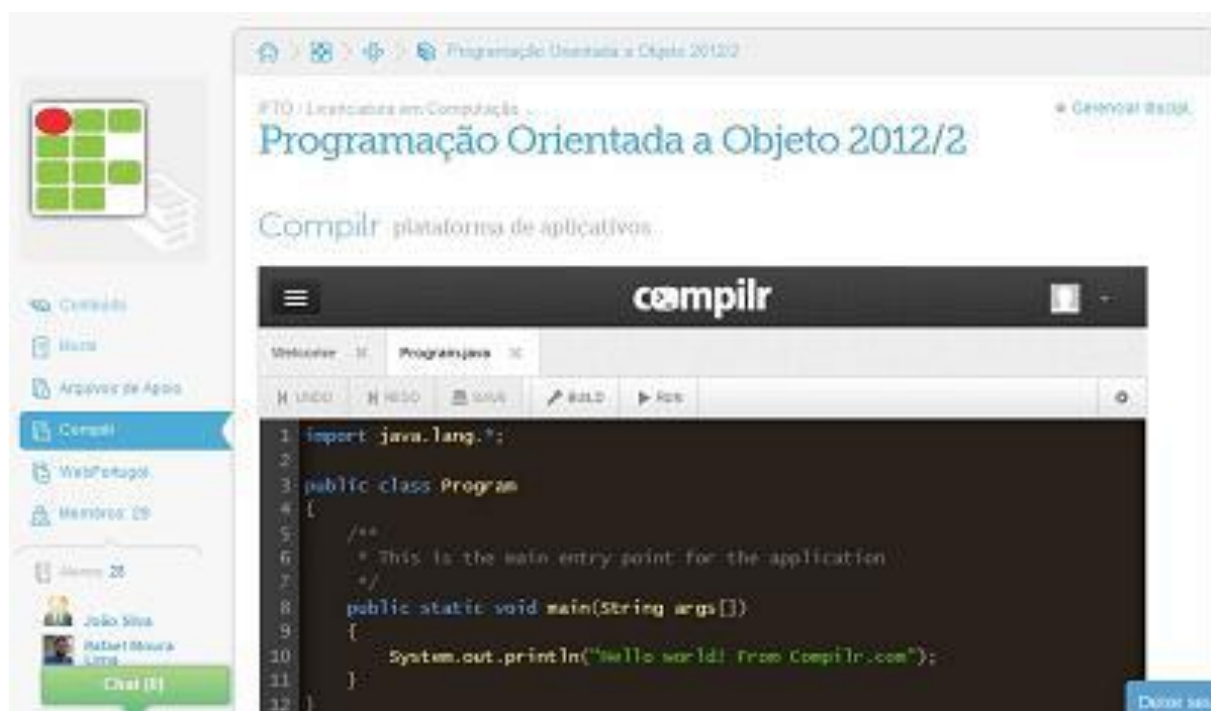


Figura B.8: Editor de programação Java disponibilizado através da API no REDU (BEZERRA, 2013)

No decorrer da fase de planejamento da disciplina, os docentes reestruturaram as atividades, reformularam os exercícios de aprendizagem, desenvolveram e disponibilizaram materiais didáticos para serem utilizados em suas aulas, que ocorreram na modalidade presencial e a distância (BEZERRA, 2013).

Os docentes relataram que a utilização do REDU favoreceu o planejamento de suas atividades, pois puderam compartilhar materiais, experiências e ideias com outros professores, abrindo a oportunidade de trabalharem outras propostas de ensino praticadas por outros docentes (BEZERRA, 2013).

BEZERRA (2013) diz que como resultado, foram apontados vários panoramas que envolveram as atividades do professor, como:

- Os recursos utilizados;
- O monitoramento;
- O planejamento e;
- A avaliação no processo de ensino-aprendizagem.

BEZERRA (2013) conclui que foi uma boa opção utilizar uma rede social educativa no processo de planejamento, ensino, avaliação e relacionamento dos conteúdos programáticos de programação orientada a objeto. Um dos motivos desse sucesso pode estar relacionado à transparência na comunicação e ao possibilitar apoio recíproco, favorecendo a aprendizagem colaborativa.

Ao criar um ambiente virtual para o ensino de POO, com o apoio de uma rede social educacional, em que professores se tornam mediadores e alunos tem a liberdade de ser o autor da construção do seu conhecimento, acarretando na melhora do processo de ensino e aprendizagem, propaga o conhecimento, incentiva a pesquisa, agrega o recurso tecnológico como material de ensino, avança sobre as limitações da tradicional sala de aula, valorizando o papel fundamental do professor (BEZERRA, 2013).

Neste outro trabalho, SOUZA et al. (2017) relataram a realização de uma oficina com carga horária de 10h, com o intuito dos participantes compreenderem os conceitos da orientação a objetos e desenvolverem programas aplicando o paradigma da orientação a objetos utilizando a ferramenta *Greenfoot*. O trabalho teve a participação de 30 alunos do curso Técnico em Informática da EEEP (Escola Estadual de Educação Profissional) Juca Fontenele na cidade de Viçosa do Ceará, todos leigos em Programação Orientada a Objetos.

A ferramenta utilizada foi o *Greenfoot* que de acordo com Kölling (2010) apud SOUZA et al. (2017), é um ambiente de desenvolvimento educacional integrado que

tem o objetivo de aprender e ensinar programação, combinando gráficos interativos com programação em Java, uma linguagem de programação orientada a objetos padrão baseada em texto.

Em outros termos Brito et al. (2014) apud SOUZA et al. (2017), destacam que o *Greenfoot* é uma ferramenta computacional que fornece para iniciantes na área tenham prática com programação orientada a objeto, baseado em aplicações gráficas na linguagem de programação Java, com jogos 2D. Assim, ajuda no crescimento do processo educacional e profissional, assegurando uma aprendizagem eficiente e concisa que pode ser utilizando tanto no ensino técnico, tecnológico ou em cursos de nível superior, de forma lúdica e divertida (Kölling, 2009 apud SOUZA et al., 2017).

SOUZA et al. (2017), através de um questionário que os alunos envolvidos responderam, extraíram as seguintes informações:

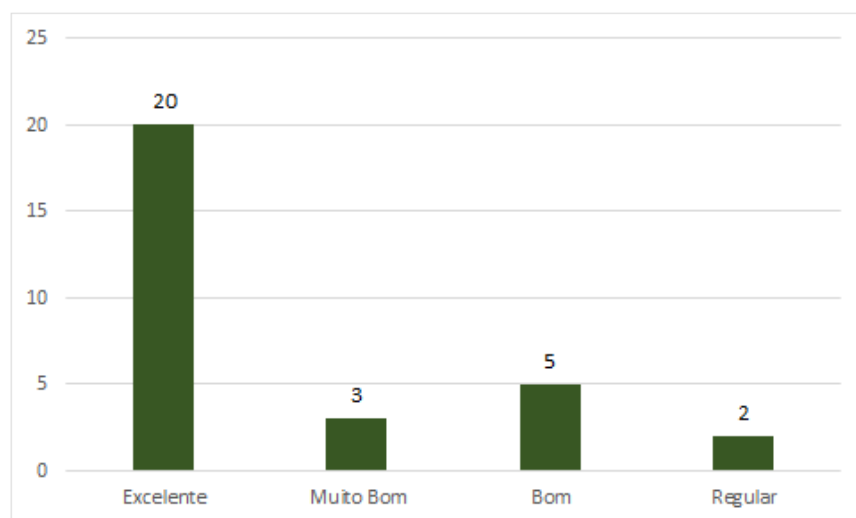


Figura B.9: Assimilação teoria/prática com Greenfoot (SOUZA et al., 2017)

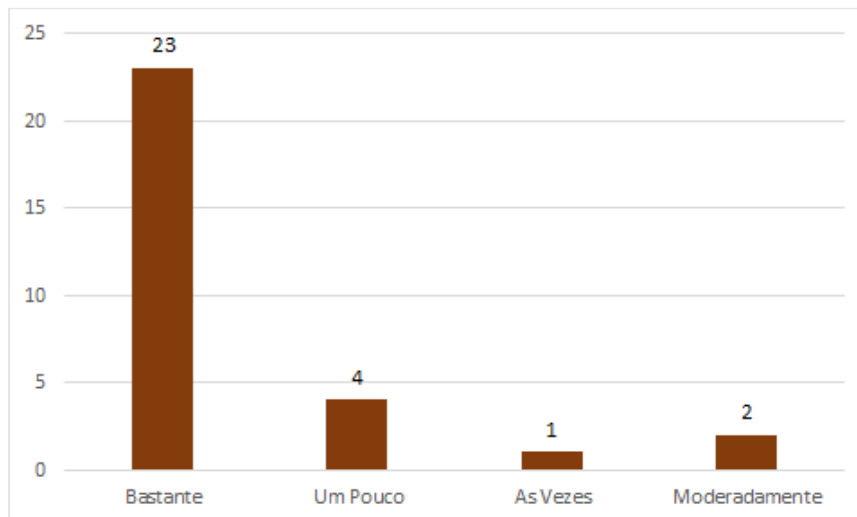


Figura B.10: Auxílio na aprendizagem da P.O.O com Greenfoot (SOUZA et al., 2017)

SOUZA et al. (2017) relataram que ao aplicar o *Greenfoot* à disciplina de Programação Orientada a Objetos como ferramenta de ensino-aprendizagem, foi possível comprovar a eficiência que ferramentas lúdicas tem para o ensino de programação, fazendo com que os alunos se envolvam na atividade, facilitando o aprendizado dos conteúdos com maiores graus de dificuldade, reduzindo as taxas de reprovação e evasão em cursos relacionados a área de Computação.

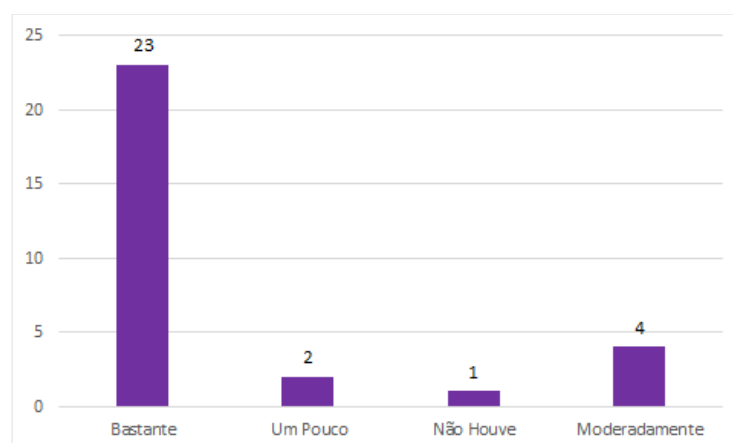


Figura B.11: Eficácia do software Greenfoot para aprender P.O.O (SOUZA et al., 2017)

ROCHA et al. (2013) em um outro trabalho realizado na Universidade Estadual de Feira de Santana foi desenvolvida uma oficina com os alunos do curso de Engenharia da Computação.

Utilizaram uma metodologia em que os alunos tentavam aprender brincando e de forma colaborativa. Os professores criaram desafios e para resolvê-los, os estudantes utilizaram os recursos disponíveis no *Scratch*, que é um ambiente gráfico de programação desenvolvido pelo *Lifelong Kindergarten Group* do MIT⁹ Media Lab, para a criação de aplicativos com o apoio de recursos áudio-visuais de forma intuitiva. Esse ambiente tem a intenção de facilitar o entendimento de conceitos de computação e matemática, estimulando a mente a raciocinar de forma sistemática e crítica, como também, melhorar a colaboração no trabalho (ROCHA et al., 2013).

ROCHA et al., (2013) relataram que o tempo previsto para a realização da oficina era curto e que não seria possível abordar todos os conceitos básico em aulas expositivas, planejaram um aprendizado diferenciado, onde esses conteúdos seriam estudados de maneira indireta.

ROCHA et al., (2013) disseram que, no começo das atividades, mesmo que a maioria dos alunos não tinham conhecimento básico em programação, conseguiram contextualizar os fatores necessário para o jogo.

Muitas dúvidas surgiram no momento de criar *scripts* para os objetos, empregar as estruturas de seleção, controle e repetição, mas na intenção de tornar ativo o papel do aluno no processo de aprendizagem, os tutores deram curtas e pontuais explicações e os monitores, apoiavam somente nas dificuldades referentes ao manuseio da ferramenta, deixando o aluno livre na busca do conhecimento. Com isso, observaram que, mesmo a atividade possuir uma ideia genérica, cada aluno trabalhou de forma tranquila, no seu tempo, sem aquela pressão exercida durante algumas atividades tradicionais (ROCHA et al., 2013).

ROCHA et al. (2013), fazendo um comparativo com o ano anterior que foi realizado o projeto, foi possível perceber a grande melhora dos alunos, que demonstraram adquirir as competências e os conceitos de programação esperados.

⁹ MIT - Massachusetts Institute of Technology

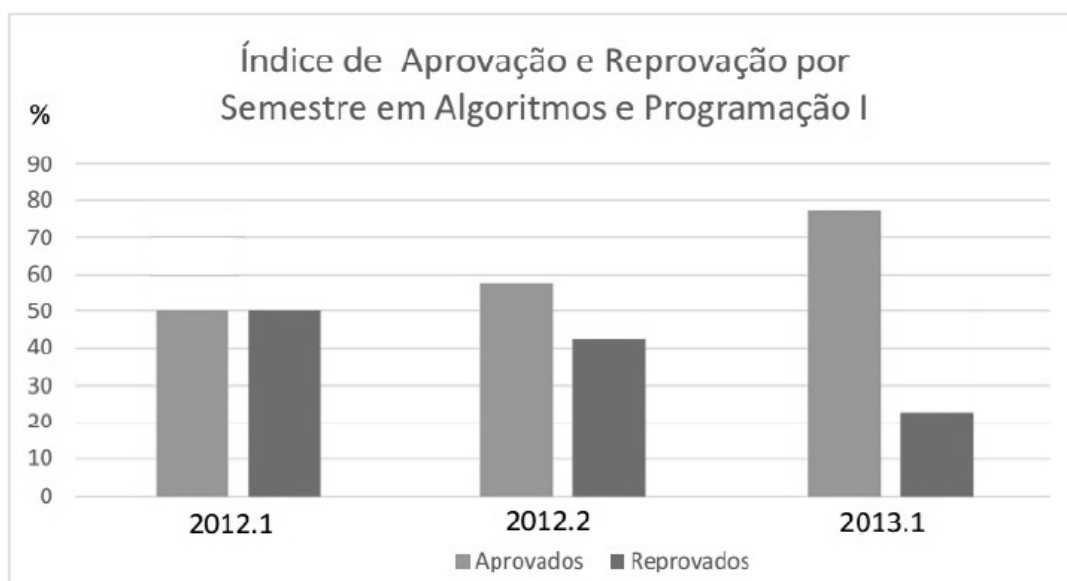


Figura B.12: Gráfico comparativo de aprovação e reprovação de estudantes ingressantes no curso de engenharia de computação (ROCHA et al., 2013)

Tabela B.5: Taxa de aprovação dos alunos (ROCHA et al., 2013)

Metodologia	Aprovação
Tradicional (ano de 2012)	54%
Aprendizagem Colaborativa (2013)	67%

ROCHA et al. (2013) concluiu que com a redução de reprovações, a oficina demonstrou que contribuiu no aprendizado de programação, reduzindo os problemas que ocorriam quando os alunos tinham o primeiro contato com os conceitos de programação, melhorando seus desempenhos.

O trabalho de KAILA et al. (2016) ocorreu na Universidade de Turku na Finlândia, na disciplina de Programação Orientada a Objetos, utilizando a linguagem Java entre 2011 e 2014, onde detectaram uma necessidade iminente de redesenhar os métodos de ensino na educação em tecnologia da informação, contando o apoio com as metodologias ativas de aprendizado colaborativa e sala de aula invertida.

A abordagem escolhida no redesenho foi facilitar o aprendizado ativo, alterando metade das aulas em sessões de aprendizado ativo. Ficou decidido incentivar a colaboração dos alunos, porque vários estudos anteriores provaram que ela tem um efeito altamente positivo no aprendizado. Um terceiro aspecto do

redesenho foi aprimorar a comunicação entre professor e aluno, utilizando pesquisas semanais para coletar as percepções dos alunos nas palestras e nas sessões de aprendizado ativo. Por fim, a metodologia do exame também precisava ser ajustada: Uma abordagem tradicional em papel e caneta era considerada insatisfatória em um curso com muitas tarefas de programação (KAILA et al., 2016).

A refatoração do método de ensino foi baseada no *ViLLE*, uma ferramenta de educação colaborativa desenvolvida no Departamento de Tecnologia da Informação da Universidade de Turku. O *ViLLE* é uma ferramenta de educação colaborativa baseada na Web que suporta diversos tipos de exercícios. A maioria dos exercícios é avaliada automaticamente e fornece feedback imediato quando enviada (KAILA et al., 2016).

KAILA et al. (2016) diz que o objetivo da refatoração era reduzir as taxas de abandono escolar, facilitando a aprendizagem ativa, a colaboração dos alunos e a comunicação entre os alunos e os professores. Ao refatorar, tentaram encontrar as respostas para duas perguntas de pesquisa:

1. A refatoração leva a uma menor taxa de desistência no curso e
2. A refatoração leva a notas mais altas no curso?

Conforme KAILA et al. (2016), os tipos de exercícios do *ViLLE* selecionados para refatoração foram:

- **Exercício de codificação:** avalia automaticamente a solução do aluno em relação à solução do modelo escrita pelo professor. O exercício fornece exceções autênticas do compilador e do tempo de execução, permitindo que os alunos resolvam problemas em potencial antes de reenviar.
- **Exercício de questionário:** fornece perguntas de múltipla escolha e em aberto. Os testes são particularmente úteis para resumos de palestras e para testes de rastreamento de código e habilidades teóricas nos exames.
- **Visualização:** Um exercício de visualização de programa no qual o rastreamento gráfico da execução do código é acompanhado por diferentes tipos de perguntas.

- **Simulação de programa:** um tipo de exercício no qual os alunos precisam simular a execução de programas uma linha de código por vez, criando e manipulando variáveis e métodos.
- **Exercício de classificação:** o *VILLE* suporta quebra-cabeças de Parsons [Parsons e Haden 2006], onde as linhas de código embaralhadas precisam ser ordenadas de acordo com as tarefas especificadas. Além disso, outros tipos de tarefas de classificação (como conectar a definição e o valor da variável) podem ser criados.
- **Pesquisa:** o *VILLE* também suporta pesquisas com incorporação opcional de fotos, áudio e vídeo.

O primeiro passo no redesenho foi implementado, alterando metade das aulas em tutoriais. Os tutoriais, criados no *VILLE*, são uma combinação de material de estudo (como texto, imagens, tabelas e vídeos) e exercícios do *VILLE*. Cada tutorial foi desenvolvido para reforçar os tópicos discutidos na aula da semana. Portanto, a aula e o tutorial formaram um módulo holístico a cada semana, onde a teoria foi oferecida pela primeira vez e, em seguida, o tópico pôde ser ensaiado com exercícios relevantes. Enquanto a maioria dos exercícios era sobre escrever e executar o código do programa, alguns testes e exercícios de classificação também eram usados para manter a tarefa mais variada e interessante. (KAILA et al., 2016).

A segunda etapa foi facilitar a colaboração dos alunos, visto que o desempenho da aprendizagem pode ser significativamente aprimorado se os alunos fizerem os exercícios em colaboração com outros estudantes. Com isso em mente, os tutoriais foram criados para dar suporte à colaboração (KAILA et al., 2016).

KAILA et al. (2016), completa dizendo que o terceiro passo foi facilitar a comunicação entre a equipe do curso e os alunos, onde foram realizadas duas pesquisas a cada semana. A primeira foi usada para coletar opiniões sobre a aula e a segunda sobre os tutoriais. A pesquisa da palestra continha as mesmas três perguntas por semana:

- O que você aprendeu na aula desta semana? - Que coisas ainda não estão claras após a aula desta semana? - Como você melhoraria a aula?

Da mesma forma, após cada tutorial, os alunos responderam as mesmas três perguntas:

- O que você aprendeu neste tutorial? - Que coisas ainda não estão claras após o tutorial? - Como você melhoraria o tutorial?

O feedback foi analisado após cada semana, e os tópicos e tutoriais que pareciam não estar claros, foram corrigidos com base nos problemas relatados pelos alunos (KAILA et al., 2016).

Um exame eletrônico (Figura B.13) foi implementado usando o *ViLLE*. Dessa forma, os alunos podem realmente escrever, compilar e testar seu código e ver os resultados, bem como quaisquer mensagens de erro do compilador na tela. O principal objetivo disso era criar um ambiente em que escrever código em um exame estivesse o mais próximo possível de uma situação autêntica (KAILA et al., 2016).

KAILA et al. (2016), conclui dizendo que a lista de tópicos ensinados permaneceu a mesma após o redesenho, mas o método foi alterado significativamente, onde o aprendizado ativo, a colaboração e a comunicação foram fortemente aprimoradas.

O redesenho foi baseado nas melhores práticas da comunidade de pesquisa de *e-learning* e que a tecnologia educacional pode ser altamente benéfica para a aprendizagem, mas somente se usada com níveis mais altos de envolvimento. A colaboração pode ter um efeito significativo no aprendizado e que os alunos que fazem exercícios em colaboração parecem estar altamente envolvidos na tarefa em questão (KAILA et al., 2016).

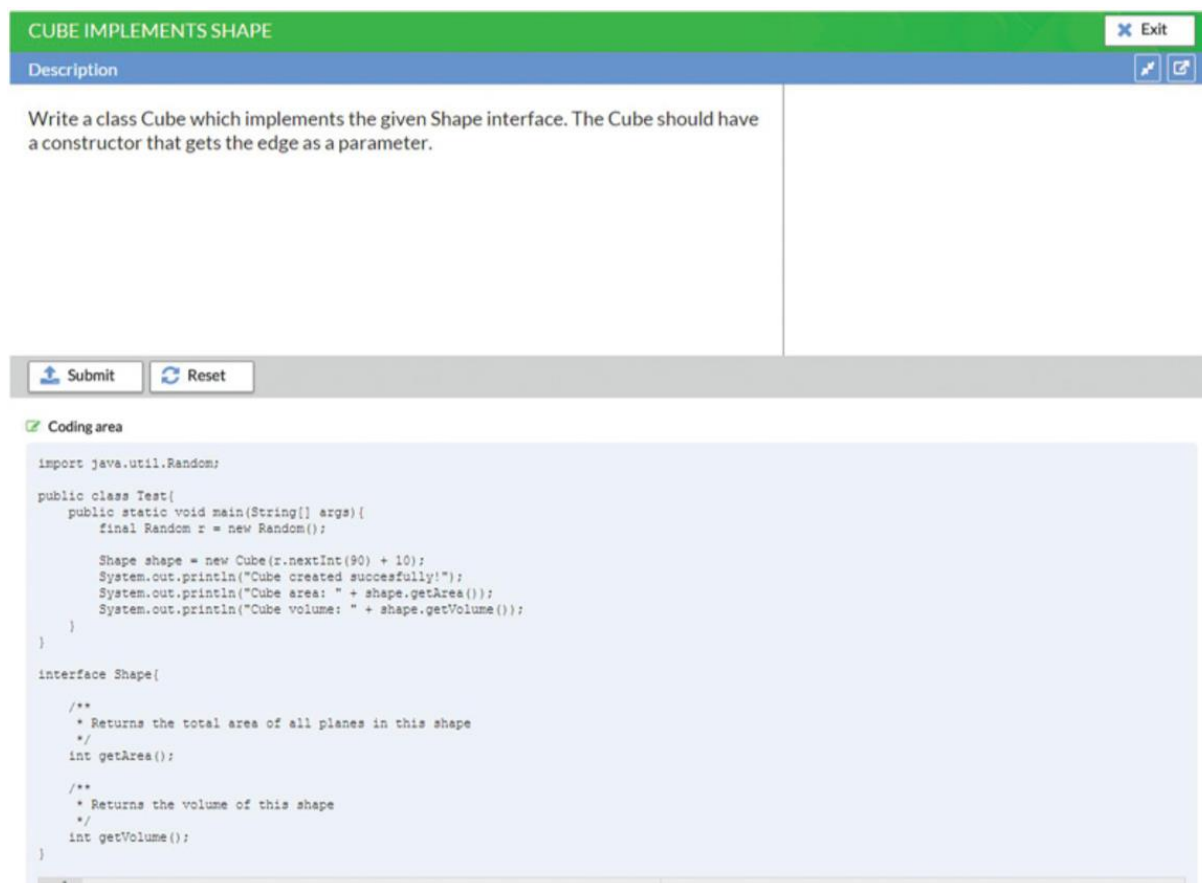


Figura B.13: Exemplo de uma tarefa do exame final (KAILA et al., 2016).

A estrutura nova do curso é exibida na Tabela B.6 e a comparação das metodologias de curso antigas (C1) e novas (C2) é exibida na Tabela B.7.

Tabela B.6: Resumo do Curso Redesenhado – reproduzido de (KAILA et al., 2016).

Componente	Quantidade de horas	Descrição
Aulas	$7 \times 2 = 14$ h	Aulas tradicionais em uma sala de aula
Tutoriais	$7 \times 2 = 14$ h	Sessões de tutorial realizadas em colaboração na sala de aula
Demonstrações	$4 \times 2 = 8$ h	Apresentação de tarefas de programação em frente a classe; feito em grupos menores (geralmente de 20 a 30 alunos). Aprimorado com o trabalho colaborativo no início e no final.
Exercícios adicionais do <i>VILLE</i>	7	Tarefas simples abertas durante a palestra para sublinhar tópicos da palestra
Projeto final	1	Projeto de programação (geralmente um jogo simples ou similar)
Exame final	1 (mas com 3 opções)	7 a 8 tarefas de programação com uma ou duas tarefas adicionais, realizadas eletronicamente no

	para levá-lo)	VILLE.
--	---------------	--------

Tabela B.7: Comparação de metodologias de cursos antigos (C1) e novos (C2) – reproduzido de (KAILA et al., 2016).

Componente	Metodologia antiga	Nova metodologia
Palestras	4h por semana	2h por semana
Tutoriais	-	2h por semana
Demonstrações	4 × 2h	4 × 2h, colaboração
Exercícios adicionais	Nenhum	Aprox. 3 por semana
Projeto final	1	1
Exame final	Caneta e papel	Eletrônico
Feedback coletado	Nenhum	2-3 pesquisas breves por semana + um após o exame
Tutoria	Nenhum	2h por semana

B.6. Considerações finais

Neste capítulo foram detalhadas seis técnicas de metodologia ativa de ensino que tinham por interesse, aumentar o nível de aprendizagem em disciplinas de programação.

É importante ressaltar que normalmente as técnicas são utilizadas em conjunto e que em vários trabalhos, durante a realização das atividades, mesclaram diferentes metodologias. Não existe um padrão de metodologias que trabalham em conjunto, ou seja, pode-se variar muitas possibilidades de técnicas de ensino.

Foi possível observar que algumas ferramentas de apoio acabam tendo preferência e maior aceitação do que outras, tendo resultados diferenciados dependendo de como são trabalhadas e que podem ser utilizadas de diversas maneiras. Essa conduta, pode ser um fator de grande impacto na obtenção dos resultados.