



Pós-Graduação em Ciência da Computação

GISLENE ALBUQUERQUE PIRES DA SILVA

FLIPPED CLASSROOM, APRENDIZAGEM COLABORATIVA E GAMIFICATION: CONCEITOS APLICADOS EM UM AMBIENTE COLABORATIVO PARA ENSINO DE PROGRAMAÇÃO



Universidade Federal de Pernambuco

posgraduacao@ufpe.cin.br

www.cin.ufpe.br/~posgraduacao

RECIFE

2017

Gislene Albuquerque Pires da Silva

Flipped Classroom, Aprendizagem Colaborativa e Gamification: Conceitos Aplicados em um Ambiente Colaborativo para Ensino de Programação

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre Profissional em Ciência da Computação.

ORIENTADOR: Prof. Fernando da Fonseca de Souza

RECIFE
2017

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

S586f Silva, Gislene Albuquerque Pires da
Flipped classroom, aprendizagem colaborativa e *gamification*: conceitos aplicados em um ambiente colaborativo para ensino de programação / Gislene Albuquerque Pires da Silva. – 2017.
90 f.: il., fig., tab.

Orientador: Fernando da Fonseca de Souza.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2017.
Inclui referências e apêndices.

1. Engenharia de software. 2. Ambiente virtual de ensino. 3. Ensino-aprendizagem. I. Souza, Fernando da Fonseca de (orientador). II. Título.

005.1

CDD (23. ed.)

UFPE- MEI 2018-019

Gislene Albuquerque Pires da Silva

**Flipped Classroom, Aprendizagem colaborativa e Gamification:
conceitos aplicados em um ambiente colaborativo para ensino de
Programação**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre Profissional em 22 de dezembro de 2017.

Aprovado em: 22 / 12 / 2017.

BANCA EXAMINADORA

Prof. Fernando da Fonseca de Souza
Centro de Informática / UFPE
(Orientador)

Prof. Marizete Silva Santos
Universidade Federal Rural de Pernambuco

Prof. Aida Araújo Ferreira
Instituto Federal de Pernambuco

AGRADECIMENTOS

Ao meu Deus e Senhor agradeço e hei de agradecer todos os dias da minha vida, sem Ti Senhor eu não sou nada, cada passo que dei para chegar até aqui estavas ao meu lado e se hoje estou tendo a oportunidade de me tornar Mestre foi graças a Ti Senhor.

À minha família, minha mãe Maria, meu pai Francisco e meu irmão Jefferson obrigada por todo amor, carinho e dedicação, vocês são a minha vida e a minha alegria. Ao meu esposo Daniel pela paciência, amor e todo o apoio. À minha avó Francisca (in memoriam) pelo carinho e amor que sempre me demonstrou, deixo aqui o meu muito obrigada.

Aos professores Franciéric Araújo e Elanne Cristina quero deixar meu total agradecimento por todo o apoio dado durante a realização deste estudo, agradeço por todas as dicas e orientações que me foram dadas. Ao Instituto Federal do Piauí que, juntamente a UFPE através do CIn, me possibilitou a oportunidade deste Mestrado.

Ao professor Fernando: Obrigada pela paciência e por toda a ajuda que me deu, mesmo à distância não me abandonou em nenhum momento, foi sempre muito compreensivo, prestativo e pronto para as orientações, muito obrigada.

Aos amigos e companheiros de trabalho: Tércio, Thiago, Assunção, Wellington, Cleverton, Alysson e Diego agradeço o carinho e compreensão que sempre demonstraram, por muitas vezes chegava cansada pela manhã devido noites mal dormidas e sempre me ajudavam com as atividades do dia-a-dia.

RESUMO

Na era da globalização, o rápido desenvolvimento das tecnologias e da economia aumentam a demanda por graduados em Tecnologia da Informação. A Programação de Computadores é uma das disciplinas base em cursos de graduação em Computação e Engenharias e por muitas vezes os alunos sentem dificuldade em compreender os seus conceitos abstratos e, infelizmente, os métodos de ensino estáticos e tradicionais fazem pouco para melhorar a capacidade dos alunos a visualizar e compreender os difíceis conceitos de Programação de Computadores. Estudos recentes mostram que universidades têm buscado inovar seus métodos de ensino e usar metodologias ativas, nas quais o aluno assume uma postura mais ativa em relação ao seu aprendizado, o que tem se mostrado eficiente no auxílio ao processo de ensino-aprendizagem. Metodologias ativas como *Flipped Classroom*, *Problem Based Learning*, jogos, entre outros, têm sido utilizadas com o objetivo de aumentar o envolvimento dos alunos na aprendizagem buscando melhorar seu desempenho acadêmico. Diante do exposto, este trabalho apresenta a aplicação das metodologias *Flipped Classroom* ou "sala de aula invertida", Aprendizagem Colaborativa e *Gamification* mediadas por ambientes virtuais de ensino utilizadas como estratégias para o ensino/aprendizagem de Programação de Computadores. Um protótipo foi desenvolvido para a aplicação da Aprendizagem Colaborativa com a finalidade de aumentar a interação social entre os alunos permitindo que aprendessem algo juntos, além de avaliar sua efetividade no que se refere ao ensino/aprendizagem de Programação de Computadores.

Palavras-Chave: Ensino-Aprendizagem. Ambientes Virtuais de Ensino. Programação de Computadores. Metodologias Ativas.

ABSTRACT

In the globalization era, the rapid development of technologies and economy increases the need for graduates in Information Technology. Computer Programming is one of many basic disciplines in Computing and Engineering graduate courses; and frequently the students feel difficulties in understanding the abstract ideas, and unfortunately, the static and traditional teaching methods do little to improve the student's capacity to view and understand the complex concepts of Computer Programming. Recent researches show that universities have looked for innovating their teaching methods and using methodologies that the students assume a more active behavior on their learning. This has been demonstrating to be an efficient aid to the teaching-learning process. Methodologies like Flipped Classroom, Problem Based Learning, games, amongst others, have been used to enhance the student's involvement on learning and improve his or her academic performance. Given the above, this work introduces the application of Flipped Classroom methodology, Collaborative Learning and Gamification mediated by virtual environments used as strategies to the Computer Programming teaching/learning process. A prototype was developed for the application of Collaborative Learning aimed at enhancing the social interaction between students as well as inducing them to learn something together. Also, this work evaluates its effectiveness on the teaching-learning process of Computer Programming.

Keywords: Teaching-Learning. Virtual Learning Environment. Computers Programming. Active Methodologies.

LISTA DE FIGURAS

Figura 2.1 - Quatro Pilares do <i>Flipped Classroom</i>	16
Figura 3.1 - Tela da Ferramenta Codecademy	31
Figura 4.1 - Características da Licença MIT.....	34
Figura 4.2 - Código Utilização Interpretador Python	36
Figura 4.3 - Tela Inicial do Protótipo CodeinPY	38
Figura 4.4 - Tela da Sala Virtual do Protótipo CodeinPY	39
Figura 4.5 - Casos de Uso do protótipo CodeinPY.....	40
Figura 4.6 - Diagrama de Atividades	41
Figura 4.7 - Diagrama de Máquina de Estados.....	42
Figura 4.8 - Diagrama de Sequência.....	42
Figura 5.1 - Experimento "A"	52
Figura 5.2 - Experimento "A" - Utilização do Protótipo CodeinPY.....	53
Figura 5.3 - Experimento "A" - Tela de Grupo resolvendo Algoritmo no Protótipo CodeinPY	53
Figura 5.4 - Experimento "B"	54
Figura 5.6 - Avaliação da Efetividade da Ferramenta Codecademy para o Ensino dos Alunos	59
Figura 5.7 - Observações sobre a Ferramenta Codecademy	59
Figura 5.8 - Avaliação do Protótipo CodeinPY como Ferramenta de Aprendizagem.....	60
Figura 5.9 - Avaliação da Efetividade do Protótipo CodeinPY em Relação ao Ensino / Aprendizagem	60
Figura 5.10 - Observações sobre o Protótipo CodeinPY	61
Figura 5.11 - Aplicação do Método T de <i>Student</i> no Programa R	64

LISTA DE QUADROS

Quadro 2.1 - Comparação entre Metodologia Tradicional e Sala de Aula Invertida	17
Quadro 2.2 - Resumo da Análise dos Trabalhos Relacionados e os Assuntos Contemplados.....	22
Quadro 3.1 - Comparativo de Ferramentas <i>Online</i> para Aprendizagem de Programação de Computadores.....	27
Quadro 3.2 - Comparativo de Ferramentas <i>online</i> para Aprendizagem de Python.....	29
Quadro 3.3 - Ranking de Ferramentas em Relação às 24 Dimensões Pedagógicas e o Uso de <i>Gamification</i>	30
Quadro 4.3 - Arquitetura da Máquina utilizada pelo Protótipo CodeinPY.....	40
Quadro 4.4 - Descrição dos Casos de Uso	41
Quadro 5.1 - Cronograma das etapas da Pesquisa	47
Quadro 5.2 - Questionário sobre Codecademy	56
Quadro 5.3 - Questionário sobre CodeinPY	57
Quadro 5.4 - Corte do Questionário sobre CodeinPY	58
Quadro 5.5 - Média Geral das Notas dos Alunos.....	62
Quadro 5.6 - Quantidade de Alunos por Intervalo de Notas	62
Quadro 5.7 - Desvio Padrão e Variância	63
Quadro 5.8 - Questionário para Avaliação Docente.....	66

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Motivação e Justificativa	11
1.2	Problema e Perguntas da Pesquisa	12
1.3	Objetivos	13
1.4	Caracterização da Pesquisa	13
2	FUNDAMENTAÇÃO CONCEITUAL	15
2.1	<i>Flipped Classroom</i>	15
2.2	Metodologias Ativas	17
2.2.1	Aprendizagem Colaborativa	18
2.2.2	<i>Gamification</i>	19
2.3	Trabalhos Relacionados	20
2.4	Ambiente Virtual de Aprendizagem e Inversão da Sala de Aula	23
2.5	Considerações Finais do Capítulo	25
3	FERRAMENTAS PARA O <i>FLIPPED CLASSROOM</i>	26
3.1	Comparação das Ferramentas	26
3.2	Considerações Finais do Capítulo	32
4	UM PROTÓTIPO PARA APRENDIZAGEM COLABORATIVA	33
4.1	O Protótipo CodeinPY	33
4.2	Tecnologias Utilizadas	35
4.3	Funcionalidades do Protótipo	38
5	EXPERIMENTOS	44
5.1	Metodologia	44
5.2	Participantes	45
5.3	Ferramentas	45
5.4	Procedimentos	46
5.5	Coleta de Dados	47
5.6	Análise de Dados	57
5.7	Considerações Finais do Capítulo	67
6	CONSIDERAÇÕES FINAIS	68

6.1	Principais Contribuições.....	68
6.2	Limitações	69
6.3	Trabalhos Futuros	70
	REFERÊNCIAS	72
	APÊNDICE A - Quadro com Nota dos Alunos Avaliados.....	76
	APÊNDICE B - Avaliação Aplicada - Experimento "B"	77
	APÊNDICE C - Algoritmos Utilizados - Experimento "A"	78
	APÊNDICE D - Figura com Resposta de Simulação - Experimento "A"	80
	APÊNDICE E - Gráficos com Dados Coletados - Questionário CodeinPY.....	81
	APÊNDICE F - Gráficos com Dados Coletados - Questionário Codecademy.....	84
	APÊNDICE G - Respostas do Questionário da Avaliação Docente	88

1 INTRODUÇÃO

A Programação de Computadores é uma das disciplinas núcleo em cursos de graduação em Computação e Engenharias. Na era da globalização, o rápido desenvolvimento das tecnologias e a economia baseada em conhecimento aumentam a demanda por graduados em Tecnologia da Informação.

Desenvolver habilidades em Programação de Computadores normalmente exige uma prática intensiva e, por muitas vezes, programadores iniciantes têm dificuldade em entender conceitos abstratos para resolução de problemas reais.

Infelizmente, os métodos de ensino estáticos e tradicionais fazem pouco para melhorar a capacidade dos alunos a visualizar e compreender os difíceis conceitos de Programação de Computadores (ELGAMAL et al., 2013).

1.1 Motivação e Justificativa

Segundo Panwong & Kemavuthanon (2014), a maneira tradicional de ensino é centrada no professor, na qual este apenas repassa conhecimentos para seus alunos. No entanto, ao final da aula, os alunos geralmente esquecem o que aprenderam e encontram dificuldade para continuar outro assunto que requer informações anteriores.

Os citados autores retratam também a existência de uma metodologia chamada de aprendizagem baseada em problemas (PBL – *Problem Based Learning*), no qual a comunicação se dá em ambas as direções na aprendizagem. Trata-se de processo de aprendizagem no qual há melhor comunicação entre professor e aluno.

Ou seja, os alunos tornam-se capacitados para assumir um papel responsável na aprendizagem e o professor torna-se um facilitador para apoiar e guiar os alunos.

Sobre isso, Valente (2014) afirma que universidades têm buscado inovar seus métodos de ensino por meio de metodologias ativas, nas quais o aluno assume uma postura mais participativa, em oposição à metodologia tradicional, por meio da utilização de jogos, PBL, sala de aula invertida (FC - *Flipped Classroom*), entre outros.

A esse respeito, a tecnologia é muitas vezes utilizada para aumentar o envolvimento dos

alunos na aprendizagem e melhorar seu desempenho acadêmico. Além disso, a motivação para a aprendizagem do aluno também é um habilitador fundamental do sucesso da aprendizagem.

A motivação é relevante no contexto de aprendizagem colaborativa, na qual a interação social¹ desempenha um importante papel (CÁMARA et al., 2014).

Nesse contexto, Moura & Teixeira (2016) afirmam que o uso de metodologias ativas têm se mostrado eficiente no auxílio do processo ensino-aprendizagem, no qual a mudança de postura do professor em sala de aula, associada com atividades bem definidas, pode contribuir no desenvolvimento de ambiente propício à discussão e à cooperação entre pares, o que se mostra eficiente no processo de autoaprendizagem².

Desta maneira, será apresentada, na seção a seguir, o problema e perguntas da pesquisa que trata sobre os problemas envolvendo o ensino/aprendizagem dos conceitos de Programação de Computadores.

1.2 Problema e Perguntas da Pesquisa

Durante décadas, pesquisadores têm tentado criar formas automatizadas de ensino na Web visando reduzir as barreiras do ensino de Programação de Computadores. Um interesse geral é a de desenvolver maneiras de aumentar a probabilidade de interação e facilitar a realização dos objetivos de aprendizagem (LAW et al., 2010).

Ou seja, buscar inovações tecnológicas visando aumentar o envolvimento dos alunos na aprendizagem de Programação de Computadores, criando um ambiente atrativo no qual o estudante saia do papel passivo de receptor e passe a ser responsável pelo seu aprendizado.

Diante disso, as questões da pesquisa são:

- Qual a efetividade da adoção de ambientes virtuais de aprendizagem como apoio à metodologia *Flipped Classroom* na composição das estratégias para ensino-aprendizagem dos conceitos de Programação de Computadores?

¹ Refere-se a todas as ações recíprocas entre dois ou mais indivíduos durante as quais há compartilhamento de informações.

² Processo de aquisição de conhecimentos, habilidades, que a pessoa realiza por sua conta, aprendendo por si mesmo.

- Quais estratégias e inovações tecnológicas podem ser utilizadas para resolver problemas envolvendo a aprendizagem de Programação de Computadores?

1.3 Objetivos

Com base nas questões e problema norteadores da pesquisa, o objetivo geral deste estudo é analisar a efetividade da adoção da metodologia *Flipped Classroom* em conjunto com outras metodologias ativas no processo de ensino-aprendizagem de Programação de Computadores mediado por ambientes virtuais de aprendizagem.

Tendo em vista alcançar esse objetivo geral, os seguintes objetivos específicos são apresentados:

- a) Pesquisar e comparar ferramentas (ambientes virtuais de aprendizagem) que possam ser utilizadas como apoio à metodologia *Flipped Classroom*;
- b) Desenvolver um protótipo de ferramenta para aplicação da aprendizagem colaborativa; e
- c) Avaliar as ferramentas por meio de experimentos e identificar as limitações e dificuldades encontradas.

1.4 Caracterização da Pesquisa

A pesquisa científica pode ser classificada por diversos critérios. Entre eles, é possível diferenciar os seus tipos de acordo com sua natureza, objetivos ou procedimentos técnicos (WAZLAWICK, 2014).

Quanto à natureza, a pesquisa realizada foi Aplicada: "Objetiva gerar conhecimentos para aplicação prática e dirigidos à solução de problemas específicos. Envolve verdades e interesses locais" (SILVA & MENEZES, 2005, p. 20). Buscou assim, por meio da aplicação prática minimizar as dificuldades encontradas no processo de ensino/aprendizagem de Programação de Computadores.

Foi Descritiva sob o ponto de vista de seus objetivos, pois foi "caracterizada pelo levantamento de dados e pela aplicação de entrevistas e questionários" (WAZLAWICK, 2014, p. 22). A pesquisa buscou por meio da aplicação de questionários, entrevistas e observação sistemática, retratar os fatos como são e como estarão após a aplicação da ferramenta e conceitos

propostos.

Em relação aos procedimentos técnicos foi Experimental: "consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto" (Gil, 2002, p. 47).

1.5 Estrutura da Dissertação

Neste capítulo, foi realizada uma breve descrição sobre a situação atual do tema geral deste trabalho: Ensino/Aprendizagem dos conceitos de Programação de Computadores. Foram apresentados o problema e perguntas da pesquisa bem como os objetivos geral e específicos que buscam responder a essas perguntas. Além disso, foram apontadas a justificativa e motivação que levaram à realização desta pesquisa, bem como sua classificação em relação à metodologia utilizada.

Além deste capítulo, este trabalho está organizado da seguinte maneira:

- **Capítulo 2 – Fundamentação Conceitual:** Descreve sobre o *Flipped Classroom* e metodologias ativas (estratégias para aprendizagem) para o Ensino de Programação de Computadores, ambiente virtual de aprendizagem e inversão da sala de aula, além da análise dos trabalhos relacionados.
- **Capítulo 3 - Ferramentas para o *Flipped Classroom*:** Expõe os detalhes do levantamento de ferramentas e os critérios de avaliação e seleção.
- **Capítulo 4 - Um Protótipo para a Aprendizagem Colaborativa:** Descreve os detalhes da implementação do protótipo, licenças e suas funcionalidades.
- **Capítulo 5 – Experimentos:** Descreve como foram realizados os experimentos e apresenta a análise dos resultados.
- **Capítulo 6 - Considerações Finais:** Apresenta as conclusões, limitações e proposta para trabalhos futuros.

Por fim, são apresentadas as referências utilizadas na composição da revisão de literatura e fundamentação conceitual aplicadas na pesquisa.

2 FUNDAMENTAÇÃO CONCEITUAL

Este capítulo expõe as teorias e fundamentação conceitual utilizadas como base de pesquisa deste estudo. A Seção 2.1 apresenta conceitos e histórico sobre a metodologia *Flipped Classroom* e como ela é ou pode ser trabalhada na sala de aula. A Seção 2.2 aborda o *Flipped Classroom* e metodologias ativas que em conjunto possam levar a um impacto positivo na aprendizagem de Programação de Computadores. Os trabalhos relacionados são apresentados na Seção 2.3. A Seção 2.4 mostra a classificação e definição dos tipos de ambientes virtuais de aprendizagem. Por último, a Seção 2.5 apresenta um resumo geral das contribuições apresentadas neste capítulo.

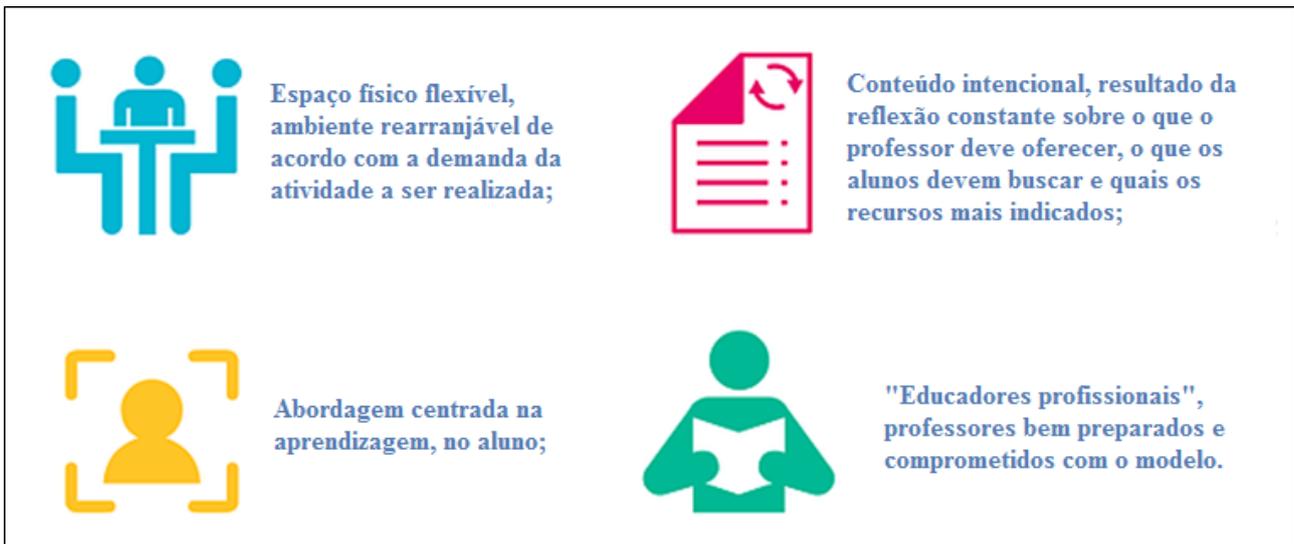
2.1 *Flipped Classroom*

A utilização de inovações tecnológicas vem ganhando espaço no ambiente educacional e conforme anteriormente citado, metodologias ativas como o *Flipped Classroom* ou "sala de aula invertida" estão sendo incorporadas ao processo educativo incentivando a que professores repensem suas práticas pedagógicas e busquem inovar seu processo de ensino.

O *Flipped Classroom* é uma metodologia de ensino no qual há uma realocação das atividades de aprendizagem, ou seja, o que seria tradicionalmente feito durante a aula (exposição do assunto) é deslocada como atividade de casa, isto é, os estudantes assistem a vídeos ou outras mídias do assunto que seriam vistos em sala, e o trabalho que seria de casa é realizado no espaço-tempo da sala de aula com a ajuda da supervisão de instrutores (SOHRABI & IRAJ, 2016).

Rodrigues (2015) conta um pouco da história dessa metodologia que foi criada por dois professores norte-americanos, Jonathan Bergmann e Aaron Sams, quando resolveram gravar vídeos de suas aulas e disponibilizarem para os alunos faltosos. Os alunos assíduos também se interessaram pelo conteúdo e com isso os dois viram a oportunidade de mudar a maneira como utilizavam seu tempo em sala de aula. Rodrigues (2015) cita também os quatro pilares dessa metodologia, conforme mostra a Figura 2.1.

Figura 2.1 - Quatro Pilares do *Flipped Classroom*



Fonte: <http://www.hoper.com.br>

Como dito, Bergmann & Sams (2016) utilizam vídeos *screencast* em sala de aula como ferramenta didática, no entanto, muitos adeptos aos conceitos do *Flipped Classroom* não utilizam vídeos como aparato didático. Ou seja, o que importa para aplicar a metodologia é adotar ferramentas tecnológicas e o ensino assíncrono com uma abordagem voltada para o aluno visando criar um ambiente estimulante à curiosidade, já que a metodologia é replicável, escalável e personalizável para cada professor.

Portanto, o *Flipped Classroom* não define ferramentas ou aparatos obrigatórios, deixando a cargo do professor encontrar meios para inovar e aplicar os conceitos da "sala de aula invertida" buscando estimular os alunos a terem uma melhor experiência durante o processo de aprendizagem. O Quadro 2.1 exibido a seguir apresenta uma comparação entre uma aula utilizando a metodologia tradicional de ensino e outra utilizando a metodologia *Flipped Classroom* relacionando o tempo e as atividades realizadas em cada uma delas.

Assim, apresenta a atividade realizada em cada uma das metodologias e o tempo médio utilizado para a efetuação de cada atividade proposta.

Quadro 2.1 - Comparação entre Metodologia Tradicional e Sala de Aula Invertida

Sala de aula tradicional		Sala de aula invertida	
Atividade	Tempo	Atividade	Tempo
Atividade de aquecimento	5 minutos	Atividade de aquecimento	5 minutos
Repasse do dever de casa da noite anterior	20 minutos	Perguntas e respostas sobre o vídeo	10 minutos
Preleção de novo conteúdo	30-45 minutos	Prática orientada e independente e/ou atividade de laboratório	75 minutos
Prática orientada e independente e/ou atividade de laboratório	20-35 minutos		

Fonte: Bergmann & Sams (2016, p.13)

O que se pode perceber é que na metodologia tradicional de ensino há uma atividade voltada à preleção de novo conteúdo, ou seja, o professor repassa o assunto aos alunos e sempre a cada aula um novo conteúdo deve ser apresentado.

Na sala de aula invertida esse tempo é totalmente reestruturado: o tempo que seria para repassar um novo conteúdo é substituído pela prática orientada, ou seja, o foco da aula é voltado em sua maior parte para a prática, atividade importante para se desenvolver habilidades em Programação de Computadores.

Como o *Flipped Classroom* não define ferramentas ou aparatos obrigatórios, este estudo buscou ferramentas ou ambientes virtuais de ensino que possam ser utilizadas como apoio à aplicação dos seus conceitos e que também possuam, como estratégias para aprendizagem, o uso de metodologias ativas como *Gamification* e aprendizagem colaborativa, conceitos que serão tratados adiante.

2.2 Metodologias Ativas

Flipped Classroom, portanto, permite explorar novas formas de educar e auxiliar o professor a melhor compreender e orientar os alunos, verificando que grupos de alunos precisam mais de ajuda e orientação por meio de ferramentas para a aprendizagem.

Logo, a aplicação da metodologia *Flipped Classroom* no ensino de Programação de Computadores para iniciantes, utilizando um ambiente de ensino colaborativo, o qual vise aumentar o envolvimento dos alunos na aprendizagem, torna-se um ambiente atrativo, no qual o estudante sai do papel passivo de receptor e passa a ser responsável pelo seu aprendizado.

Neste contexto, a busca por estratégias para mudar essa posição de passividade vem crescendo cada vez mais. Assim, metodologias ativas como *Flipped Classroom*, *Gamification*, PBL, aprendizagem colaborativa, entre outras, vêm sendo gradativamente mais aplicadas no processo de ensino/aprendizagem.

Metodologias ativas utilizam métodos de ensino cujo foco do aprendizado concentra-se nos alunos, ou seja, os próprios alunos assumem a responsabilidade pelo seu processo de aprendizagem tornando-se assim mais ativos. São também conhecidas como aprendizagem ativa e vêm apresentando uma melhoria na aprendizagem dos alunos em comparação com metodologias tradicionais, nas quais os alunos são passivos da informação entregue pelo professor (ARBELAITZ et al., 2015).

Neste cenário, as estratégias *Flipped Classroom*, aprendizagem colaborativa e *Gamification*, objetos do presente estudo, sendo aplicadas em conjunto podem potencializar o processo de ensino/aprendizagem dos alunos criando um ambiente atrativo e tornando-os mais ativos e responsáveis pelo seu autoaprendizado.

2.2.1 Aprendizagem Colaborativa

A aprendizagem colaborativa por prover uma maior interação entre os alunos busca motivá-los ao aprendizado, sendo uma situação na qual duas ou mais pessoas aprendem ou tentam aprender alguma coisa juntas: conhecimentos, capacidades, competências e assim por diante. Ou seja, é definida por um conjunto de processos que ajudam as pessoas a interagirem entre si, a fim de realizar um objetivo específico (CÁMARA et al. 2014).

Essa metodologia é focada no aluno e o papel do professor passa de provedor do conhecimento a orientador, norteando suas aulas para que haja sempre uma maior interação e participação ativa de seus alunos, buscando a troca de experiências, engajamento e motivação pelo aprendizado.

Torres & Irala (2014) afirmam que uma das ideias fundamentais da metodologia é que o conhecimento seja construído socialmente, na interação entre pessoas, tendo o professor, o papel de promover uma aprendizagem mais ativa por meio do estímulo: " ao pensamento crítico; ao desenvolvimento de capacidades de interação, negociação de informações e resolução de problemas; ao desenvolvimento da capacidade de autorregulação do processo de ensino-aprendizagem" (TORRES & IRALA, 2014, p.1).

Ou seja, o professor deve atuar na criação de cenários e ambientes adequados para que o aluno possa desenvolver suas habilidades sociais na interação com outros e não apenas pela transferência de conhecimento.

Neste contexto, o protótipo desenvolvido buscou a interação dos alunos para a construção do conhecimento por meio do compartilhamento de código em tempo real e troca de informações através de *Chat*, este foi desenvolvido utilizando o Socket.IO³ que é um *Framework* que permite a comunicação bidirecional baseada em eventos em tempo real, funcionando em todas as plataformas e navegadores, focando-se igualmente na confiabilidade e na velocidade.

2.2.2 Gamification

Outra estratégia que pode ser útil na aplicação do *Flipped Classroom* é o uso de elementos de *Gamification* tais como a utilização de barras de progresso servindo como meio para acompanhar se os alunos estão se preparando antes das aulas e permitindo ao professor verificar o engajamento e o progresso dos seus alunos (COLLIN et al., 2015).

A esse respeito, este trabalho apresenta ferramentas que possam ajudar na "inversão da sala de aula", baseado nas metodologias ativadas mencionadas, com as quais o aluno possa realizar na prática, pequenas atividades para se familiarizar com a escrita de código. Espera-se com isto fazer o usuário aprender se divertindo, utilizando o conceito de *Gamification* que é uma estratégia de interação que utiliza elementos de jogos para envolver e incentivar participantes a comportamentos desejados, sendo identificada como uma técnica promissora para melhorar o engajamento dos alunos podendo ter um impacto positivo na aprendizagem, tornando as atividades mais atraentes (IBÁÑES et al., 2014).

Portanto, *Gamification* não é transformar qualquer prática em um jogo, mas sim utilizar de elementos de jogos para oferecer uma melhor experiência para quem utiliza tornando a atividade mais leve, divertida e engajadora.

No contexto da aprendizagem dos conceitos de Algoritmos e Programação de Computadores, se familiarizar com a escrita de código pode ser um obstáculo para alguns alunos. Sobre isso, Zelle (2002, p. 7) relata que muitos estudantes sentem dificuldade de aprender conceitos básicos de Ciência da Computação e Programação de Computadores devido à complexidade de

³ <https://socket.io/>

algumas Linguagens de Programação e ferramentas utilizadas em cursos introdutórios.

Em relação a essa complexidade, Zelle (2002, p. 7) declara que a linguagem de programação Python⁴ tem características que a tornam uma ótima escolha como primeira linguagem de programação. Sua estrutura básica é simples, limpa e bem projetada, o que permite aos alunos se concentrarem nas habilidades primárias de pensamento algorítmico sem ficar atolado em detalhes da linguagem.

É importante saber que a linguagem de programação Python é *Software* livre, ou seja, não envolve aquisição de licenças de uso, e é multiplataforma, pois pode ser utilizada praticamente em qualquer sistema operacional. Possui vantagens como a legibilidade de código, além de possuir várias bibliotecas existentes, já criadas e testadas por outras pessoas, o que aumenta a produtividade do programador e reduz a quantidade de erros (MENEZES, 2016, p.25).

Levando em consideração os conceitos apresentados, as ferramentas levantadas e comparadas neste estudo, têm como critérios para sua seleção, o ensino dos conceitos de Programação de Computadores utilizando a linguagem de programação Python e que apliquem como práticas pedagógicas o uso dos conceitos de *Gamification* e aprendizagem colaborativa com foco na aplicação dos conceitos da metodologia *Flipped Classroom*.

2.3 Trabalhos Relacionados

No que se refere ao tema *Flipped Classroom*, esse tem sido abordado cada vez mais na literatura científica. É possível elencar diferentes trabalhos que tratam da aplicação da metodologia *Flipped Classroom* e o ensino de Programação de Computadores como mostrado a seguir.

Em uma pesquisa realizada por Colombo et al. (2014) com estudantes de um Curso Técnico em Informática do Instituto Federal do Espírito Santo (IFES), Campus Cachoeiro, foi aplicado a metodologia *Flipped Classroom* com o intuito de tornar as aulas mais atraentes e com foco no aluno. Foi utilizada a ferramenta Wink⁵ para contemplar a ementa da disciplina Técnicas de Programação e os autores buscaram mostrar que o processo de ensino-aprendizado foi melhor que o processo tradicional.

Jones (2016) realizou um estudo que compara a metodologia tradicional de ensino ao *Flipped Classroom* na disciplina de Compiladores. Foram analisadas duas turmas distintas que

⁴ <https://www.python.org/>

⁵ <http://www.debugmode.com/wink/> - *Software* para criação de tutoriais através da captura de imagens.

tiveram aulas com o mesmo professor e foi possível perceber uma grande aceitação da nova metodologia e uma melhora nas notas dos alunos, indicando que houve melhoria na absorção de conteúdos pelos alunos com mais dificuldades.

Já Oliveira et.al. (2014), buscando melhorias do aprendizado dos alunos do Curso Técnico em Informática do Instituto Federal do Norte de Minas Gerais (IFNMG), Campus Arinos, aplicaram a Plataforma Arduino⁶ como uma ferramenta para o ensino de conceitos de Lógica de Programação e, em trabalhos, comprovar que a implantação da metodologia *Flipped Classroom* para a disciplina Técnicas de Programação obteve melhor resultado que o processo tradicional.

Olsson et al. (2015) apresentaram um trabalho no qual discutiram a definição de um canal extra de comunicação entre professor e aluno para prover um controle de aprendizagem para compreensão dos conceitos de programação e o uso do *Gamification* como uma forma de aumentar a motivação em ambientes virtuais de aprendizagem. Os resultados foram discutidos e indicaram que a visualização por barras de progresso é uma boa maneira de melhorar a visão geral dos participantes em ambientes *online* com conteúdo rico e multifacetado.

Na pesquisa de Jr & Cortelazzo (2015) foi apresentada uma proposta de utilização integrada da educação *online*, ambientes de aprendizagem e sala de aula invertida, como ferramenta potencializadora do ensino de algoritmos e linguagens de programação para modalidade presencial de cursos superiores de graduação.

Como visto, existem trabalhos sobre o *Flipped Classroom* em disciplinas de ensino de Programação de Computadores, porém ainda não foi realizado um estudo sobre a aplicação dos conceitos em conjunto com *Gamification* e ambientes *online* para aprendizagem colaborativa.

O Quadro 2.2 expõe um resumo sobre os trabalhos relacionados que têm como foco os assuntos pesquisados e relacionados a seguir:

1. *Flipped Classroom* e Ensino de Programação de Computadores;
2. Ambiente colaborativo de ensino/aprendizagem; e
3. *Gamification*;

⁶ <https://www.arduino.cc> - Plataforma de prototipação baseada em Hardware e Software de fácil utilização para prototipação de projetos interativos utilizando um microcontrolador.

Quadro 2.2 - Resumo da Análise dos Trabalhos Relacionados e os Assuntos Contemplados.

TRABALHOS RELACIONADOS	<i>FLIPPED CLASSROOM</i>	AMBIENTE COLABORATIVO ONLINE	<i>GAMIFICATION</i>	ENSINO DE PROGRAMAÇÃO
O modelo sala de aula invertida aplicado em uma disciplina do curso técnico em informática Colombo et al. (2014)	✓ ⁷	✗ ⁸	✗	✓
Aprendizado Invertido em Compiladores JONES (2016)	✓	✗	✗	✓
Arduino: Uma proposta para o ensino introdutório de programação Oliveira et.al. (2014)	✓	✗	✗	✓
<i>Visualization and Gamification of e-Learning and Programming Education</i> Olsson et al. (2015)	✗	✗	✓	✓
Sala de aula invertida, ambientes de aprendizagem e educação online: a junção de três métodos para potencialização do ensino de algoritmos JR & CORTELAZZO (2015)	✓	✗	✗	✓

Fonte: Elaborado pela Autora (2017)

Diante dos estudos apresentados, é possível verificar o direcionamento de pesquisas no sentido de avaliar a metodologia *Flipped Classroom* e sua efetividade no ensino de Programação de Computadores, e um dos pontos a serem observados é que nenhum dos estudos acima listados apresenta como proposta um ambiente colaborativo *online*. Ou seja, há uma lacuna a ser preenchida e para isso, um protótipo foi desenvolvido com foco na aprendizagem colaborativa, tornando-se um diferencial da presente pesquisa.

7 Sim

8 Não

Como visto, os estudos mostraram que apenas a adoção da metodologia *Flipped Classroom* no ensino/aprendizagem de Programação de Computadores, mesmo que aplicada de forma individualizada, apresenta resultados positivos. Logo, a ação conjunta com ferramentas como estratégia de aprendizagem leva a crer que resulte em melhorias ainda mais significativas.

Portanto, o principal propósito deste trabalho é pesquisar e comparar ferramentas de apoio ao *Flipped Classroom* a fim de colaborar com evidências sobre a real efetividade no ensino da Programação de Computadores, além de propor uma ferramenta, e por meio de um protótipo, aplicar os conceitos de Aprendizagem Colaborativa como estratégia de aprendizagem e complemento para inversão da sala de aula explorando assim novas formas de educar por meio dos conceitos propostos: *Flipped Classroom*, uso de *Gamification* e Aprendizagem Colaborativa.

O protótipo desenvolvido e a ferramenta selecionada para a realização dos experimentos serão apresentados em capítulos mais adiante.

2.4 Ambiente Virtual de Aprendizagem e Inversão da Sala de Aula

Em pesquisa recente, Kim & Ko (2017) avaliaram 30 ferramentas *online* para aprendizagem de Programação de Computadores. Verificaram sua efetividade pedagógica e analisaram o que ensinam e como ensinam. Para tanto, realizaram uma busca tendo como critérios a popularidade dos termos "tutorial de codificação *online*" e "tutorial de codificação" e a quantidade de tráfego na *Web* por cada termo.

Para ajudar na comparação das 30 ferramentas, elas foram categorizadas em cinco gêneros: tutoriais interativos, referências da *Web*, MOOC (Massive Open Online Course), jogos educacionais e plataformas criativas. Foram subcategorizadas em 24 dimensões pedagógicas, nas quais para saber se a ferramenta atendia determinado critério definiu-se "sim" ou "não" onde "sim" seria o atendimento a determinado critério pedagógico.

Pela definição dos gêneros de Kim & Ko (2017), tutoriais interativos são aqueles nos quais os alunos interagem com um editor de texto ou algo semelhante, passando por sucessivas etapas. São exemplos: o Codecademy⁹, Khan Academy¹⁰ e Codeschool.com¹¹.

⁹ <https://www.codecademy.com/pt-BR>

¹⁰ <https://pt.khanacademy.org/>

¹¹ <https://www.codeschool.com/>

Referências *Web* desempenham papel semelhante a "dicionários". Podem possuir interação com editores de texto nos quais os alunos possam praticar seus códigos. São exemplos: o W3Schools¹² ou Learnpython.org¹³.

MOOC são cursos abertos oferecidos por meios de ambientes virtuais de aprendizagem, geralmente de forma gratuita, nos quais milhares de pessoas podem ter acesso livremente. Possuem estrutura hierárquica com etapas passo a passo, nas quais incorpora *quizzes* após a sequência de instruções. São exemplos: Lynda.com¹⁴, edX¹⁵ e Coursera¹⁶.

Jogos educacionais fornecem objetivos e *feedback* imediato além de um ambiente rico visualmente. São exemplos: o Gidget¹⁷, Code Combate¹⁸ e Code Hunt¹⁹.

Por último, as plataformas criativas que fornecem conteúdo e editor com guia de referências, mas sem muitos objetivos explícitos. São exemplos: Scratch²⁰ e Alice²¹.

Para a seleção da ferramenta de apoio ao processo de inversão de sala de aula e aplicação dos conceitos da metodologia *Flipped Classroom* a ser utilizada neste estudo, das 30 ferramentas estudadas, 7 foram pré-selecionadas devido oferecerem a opção do ensino da Linguagem de Programação Python. A partir daí, essas ferramentas foram avaliadas e classificadas em relação às 24 dimensões pedagógicas apresentadas com o adicional da verificação da utilização ou não do uso de *Gamification*.

A ferramenta com melhor pontuação foi selecionada e utilizada em sala de aula para avaliação dos alunos e verificação de sua efetividade em relação ao aprendizado desses alunos. Esse processo será detalhado mais adiante.

¹² <https://www.w3schools.com/>

¹³ <https://www.learnpython.org/>

¹⁴ <https://www.lynda.com/>

¹⁵ <https://www.edx.org/>

¹⁶ <https://www.coursera.org/>

¹⁷ <http://www.helpgidget.org/>

¹⁸ <https://codecombat.com/>

¹⁹ <https://www.codehunt.com/>

²⁰ <https://scratch.mit.edu/>

²¹ <http://www.alice.org>

2.5 Considerações Finais do Capítulo

Este capítulo apresentou uma revisão da literatura a fim de embasar o presente trabalho. Foram apresentadas a conceituação e características de *Flipped Classroom* ou sala de aula invertida, *Gamification* e Aprendizagem Colaborativa, e como estes aplicados em conjunto buscam potencializar e trazer resultados positivos ao ensino/aprendizagem de Programação de Computadores.

O diferencial do presente estudo é apresentar ferramentas/ambientes virtuais de ensino/aprendizagem que utilizem tais estratégias e que aplicadas em conjunto tragam uma maior atratividade e tornem os alunos mais ativos e responsáveis pelo seu processo de aprendizagem.

Os estudos apresentados mostraram que a metodologia *Flipped Classroom* aplicada mesmo que isoladamente trouxe resultados positivos, o que leva a crer que aplicada em conjunto com outras metodologias ativas traga resultados ainda mais satisfatórios.

A revisão da literatura mostrou a relevância e atualidade do tema, e apresentou como esse tema foi abordado sob o ponto de vista de diferentes autores. A maioria desses trabalhos comparou a metodologia da sala de aula invertida com a metodologia tradicional de ensino não se atendo a ferramentas para apoio à metodologia da inversão da sala de aula.

Como contribuição desta pesquisa, um protótipo foi proposto e desenvolvido para a prática da aprendizagem colaborativa funcionando como estratégia e inovação tecnológica visando aumentar a motivação dos alunos no processo de aprendizagem dos conceitos de Programação de Computadores.

O próximo capítulo apresenta um estudo de 30 ferramentas para o ensino/aprendizagem de Programação de Computadores, e a comparação entre sete dessas ferramentas para utilização como ferramenta de apoio ao *Flipped Classroom*. Também descreve o método utilizado para seleção da ferramenta a ser aplicada na pesquisa.

3 FERRAMENTAS PARA O *FLIPPED CLASSROOM*

Este capítulo expõe um estudo realizado sobre ferramentas para o ensino/aprendizagem de Programação de Computadores e o método para seleção da ferramenta a ser utilizada como apoio à metodologia *Flipped Classroom*, apresentando suas características e qualidades. A Seção 3.1 mostra de forma detalhada os quadros 3.1 a 3.3 com as ferramentas estudadas e a comparação realizada entre elas a fim de selecionar a que melhor se encaixe à metodologia da sala de aula invertida. A Seção 3.2 apresenta as considerações finais do capítulo e aponta a ferramenta a ser avaliada.

3.1 Comparação das Ferramentas

Kim & Ko (2017) elaboraram um estudo no qual realizaram uma comparação entre ferramentas para ensino/aprendizagem de Programação de Computadores levando em consideração a relação ao atendimento ou não a determinadas dimensões pedagógicas, como mostra o Quadro 3.1.

Quadro 3.1 - Comparativo de Ferramentas *Online* para Aprendizagem de Programação de Computadores

Tutorials	1. Personalization			2. Utilization	3. Contents								4. Organization		5. Context			6. Actionability	7. Feedback		8. Transfer learning			9. Support
	a. Age	b. Educational status	c. Coding experiences	Subsequent knowledge	a. Variables	b. Arithmetic	c. Logical	d. Conditional	e. loops	f. Arrays	g. Function	h. Object	a. Bottom-up	b. Hierarchical structure	a. Lecture-based	b. Project-based	c. Storyline	Learners write code	a. Feedback	b. Immediate feedback	a. How to use	b. When to use	c. Why to use	Additional materials self-monitoring
Codecademy Python	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CodeSchool	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	both	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Khan Academy	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	top-down	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Online Python Tutor	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Codingbat Python	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Regex Golf	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Regex 101	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
W3School JS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LearnPython	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Learnjavascript	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Funprogramming	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TutorialsPoint Python	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cprogramming	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Codingunit C tutorial	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Wired.com Javascript	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pythonprogramming	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	both	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Stack Overflow	✓	✓	✓	✓	N/A	N/A	N/A	N/A	N/A	N/A	N/A	top-down	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
edX Microsoft JS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	top-down	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Lynda.com	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	top-down	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Coursera Javascript	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	top-down	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Coursera Python	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	both	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Code.org	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	top-down	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Gidget	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	both	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CodeCombat	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	both	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LightBot	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	both	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CodeHunt	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	both	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Code Avengers	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Grok Learning	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Scratch	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	top-down	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Alice	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	top-down	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Fonte: Kim & Ko (2017, p.3)

O Quadro 3.1 apresenta as 30 ferramentas pesquisadas e sua relação com as 24 dimensões pedagógicas, verificando se atende ou não a cada uma delas. Essas dimensões pedagógicas foram subcategorizadas em:

1. **Personalização** - Idade, status educacional, experiência de codificação.

Avalia se as ferramentas possibilitam a oportunidade de personalizar as formas de educar, se realiza classificação da aprendizagem por faixas etárias, se o *status* educacional

indica os níveis de dificuldade (fácil, intermediário, avançado), e se permite aos usuários a experiência em codificação.

2. **Utilização** - Conhecimento subsequente.

Avalia se as ferramentas requerem dos usuários o conhecimento subsequente sobre os assuntos abordados, ou seja, se os usuários necessitam de um acúmulo de conhecimento para avançar nas etapas.

3. **Conteúdos** - Variáveis, aritmética, lógico, condicionais, repetições, arrays, funções e objetos.

Avalia se as ferramentas abordam os principais conteúdos para aprendizagem de Programação de Computadores.

4. **Organização** - De baixo para cima, estrutura hierárquica.

Avalia se as ferramentas possuem uma organização de seus conteúdos do nível mais simples ao mais complexo e se há estrutura hierárquica entre esses assuntos.

5. **Contexto** - Baseado em leitura, baseado em projetos, enredo.

Avalia como as ferramentas se baseiam na construção do conhecimento, se apenas em leitura ou projetos, se possuem um enredo na apresentação dos conteúdos.

6. **Capacidade de ação** - Alunos escrevem código.

Avalia se as ferramentas permitem que os usuários/alunos escrevam seus códigos na própria ferramenta.

7. **Feedback:** Feedback, feedback imediato.

Avalia se as ferramentas retornam respostas imediatas ou respostas posteriores ao código escrito/informado, avaliando se está correto ou não e se apresenta sugestões para resolução.

8. **Transferência de Aprendizagem** - Como utilizar, onde utilizar e por que utilizar.

Avalia se as ferramentas apresentam em seus conteúdos, a sua forma de utilização, como e onde utilizá-la além do porquê de sua utilização.

9. **Suporte:** Material adicional, automonitoramento.

Avalia se as ferramentas possuem material adicional ao conteúdo apresentado, se possuem, por exemplo, a possibilidade de tirar dúvidas por meio de fórum, *quizzes*, entre outros.

Com base no estudo realizado Kim & Ko (2017) e com a finalidade de realizar a seleção da ferramenta de apoio ao *Flipped Classroom* voltada ao ensino/aprendizagem de Programação de Computadores, a pesquisadora definiu um critério de avaliação para que as ferramentas fossem analisadas: apenas as que possuem em comum o ensino da Linguagem de Programação Python, como mostra o Quadro 3.2.

Quadro 3.2 - Comparativo de Ferramentas *online* para Aprendizagem de Python

Tutorials	1. Personalization			2. Utilization	3. Contents								4. Organization		5. Context			6. Actionability	7. Feedback		8. Transfer learning			9. Support
	a. Age	b. Educational status	c. Coding experiences		Subsequent knowledge	a. Variables	b. Arithmetic	c. Logical	d. Conditional	e. loops	f. Arrays	g. Function	h. Object	a. Bottom-up	b. Hierarchical structure	a. Lecture-based	b. Project-based		c. Storyline	Learners write code	a. Feedback	b. Immediate feedback	a. How to use	
Codecademy Python		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	top-down	✓	✓	✓	✓	✓	✓	✓	✓			✓
Online Python Tutor					✓	✓	✓	✓	✓	✓	✓	✓	✓					✓	✓	✓				✓
Codingbat Python				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓				✓
LearnPython					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓				✓
TutorialsPoint Python		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	both					✓	✓	✓	✓	✓		✓
Pythonprogramming					✓	✓	✓	✓	✓	✓	✓	✓	both					✓	✓	✓	✓	✓		✓
Coursera Python				✓	✓	✓	✓	✓	✓	✓	✓	✓	both	✓	✓			✓	✓	✓	✓	✓		✓

Fonte: Adaptado de Kim & Ko (2017, p.3)

Após essa pré-seleção, criou-se um *ranking* no qual cada uma das 24 dimensões pedagógicas avaliadas valeriam um (1) ponto, e as ferramentas melhores ranqueadas foram consideradas as mais completas. Como adicional a essa comparação, foi acrescentado o critério do uso de *Gamification*, tornando-se assim a 25ª dimensão pedagógica, como mostra o Quadro 3.3.

Quadro 3.3 - Ranking de Ferramentas em Relação às 24 Dimensões Pedagógicas e o Uso de *Gamification*

FERRAMENTAS		Idade	Status educacional	Experiência de codificação	Conhecimento subsequente	Variáveis	Aritmética	Lógico	Condicionais	Repetições	Arrays	Funções	Objetos	De baixo para cima	Estrutura hierárquica	Baseado em leitura	Baseado em projetos	Enredo	Alunos escrevem código	Feedback	Feedback imediato	Como utilizar	Onde utilizar	Por que utilizar?	Material adicional	Uso de <i>Gamification</i>	TOTAL
1	Codecademy Python	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	22
2	Coursera Python	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	0	0	1	1	18
3	TutorialsPoint Python	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	13
4	Codingbat Python	0	0	0	1	1	0	1	1	1	1	0	0	1	1	0	0	0	1	1	1	0	0	0	1	1	13
5	LearnPython	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	12
6	Pythonprogramming	0	0	0	0	1	1	0	1	1	1	0	1	0	0	0	0	0	0	1	0	1	0	0	1	0	10
7	Online Python Tutor	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	6

Fonte: Adaptado de Kim & Ko (2017, p.3)

Como visto no Quadro 3.3, dentre as ferramentas comparadas, as mais bem pontuadas foram a Codecademy e Coursera.

Codecademy é um site que oferece acesso gratuito ao aprendizado de várias linguagens de programação como Javascript, Python e Ruby, entre outros. Possui mais de 24 milhões de usuários que concluíram mais de 100 milhões de exercícios (LEE&KO, 2015). Funciona com um tutorial *online* interativo e utiliza elementos de *Gamification* como medalhas e barras de progressão provendo oportunidades de personalização para os aprendizes (MENSING et al., 2013).

Possui cursos em diversos idiomas e o trabalho de tradução para o português do Brasil foi realizado em parceria com a Fundação Lemann²², organização sem fins lucrativos atuante na educação com foco em aprendizado e inovação.

Já a Coursera é uma empresa de empreendedorismo social e oferece cursos gratuitos *online*. Espera que a experiência do aluno seja mais colaborativa e interativa e, para isso, esforça-se para se tornar uma plataforma distinta de outros tipos de MOOC, utilizando conceitos pedagógicos para ajudar os alunos a aprender de forma mais eficiente e completa, como: Aprendizagem *Online*, Recuperação e Teste para a Aprendizagem, Aprendizagem de Domínio, Avaliações por Pares e

²² <http://www.fundacaolemann.org.br/>

Aprendizagem Ativa (AUDSLEY et al., 2013).

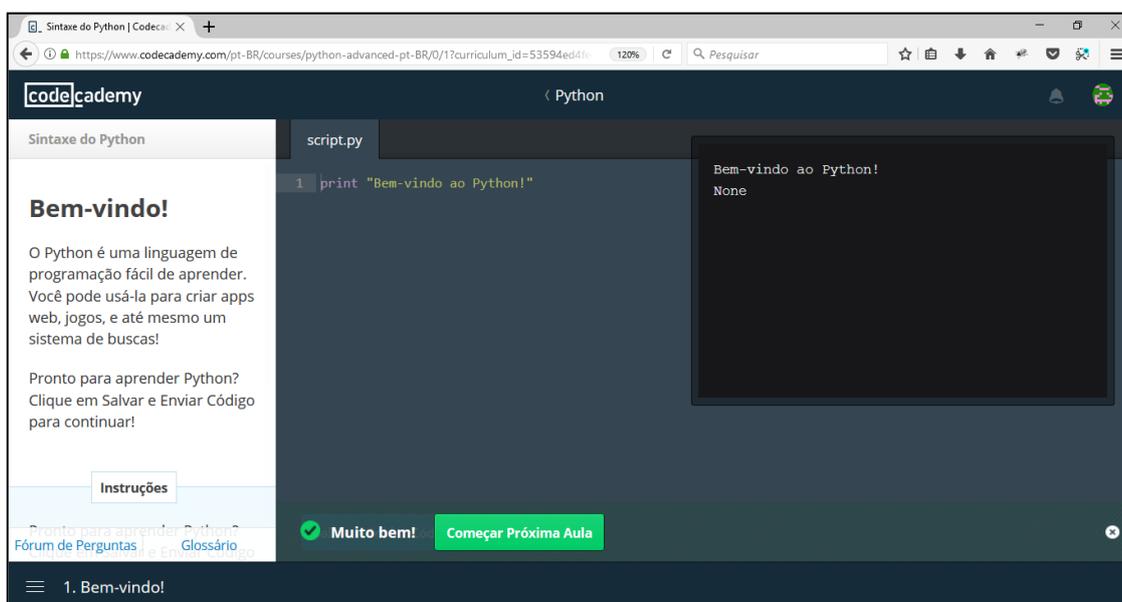
Ambas as ferramentas utilizam elementos de *Gamification*, e buscam uma maior interação e experiência colaborativa. Nelas, uma abordagem utilizada para uma maior interação e colaboração entre os alunos na aprendizagem de Programação é a Programação em Pares (PP) que vem sendo utilizada para promover o pensamento computacional em jovens estudantes (ZHONG et al., 2016).

Em sua pesquisa, Zacharis (2011) afirma que muitos pesquisadores experimentam a implementação de Programação em Pares Distribuídos utilizando aplicativos de compartilhamento ou ambientes colaborativos e o sugerem como uma ferramenta pedagógica eficaz para uma colaboração flexível e uma alternativa aceitável à experiência de programação individual.

Vendo isso, analisando o Quadro 3.3, o que se pode perceber é a ferramenta Codecademy se mostrando mais completa em relação às 24 dimensões pedagógicas, e por isso, foi selecionada e testada a fim de se determinar se a mesma se encaixa à metodologia *Flipped Classroom* e se a mesma é efetiva no ensino-aprendizagem de Programação de Computadores.

Este estudo avaliou a ferramenta Codecademy por meio de experimentos verificando se a mesma foi efetiva no processo de ensino/aprendizagem de Programação de Computadores sendo utilizada como ferramenta de apoio à metodologia *Flipped Classroom*. A seguir a Figura 3.1 apresenta a tela da ferramenta Codecademy do curso de Python disponibilizado em Português.

Figura 3.1 - Tela da Ferramenta Codecademy



Fonte: codecademy.com/pt-BR

A Codecademy disponibiliza um fórum de discussão para que seus usuários consigam interagir e trocar conhecimento, sendo a forma de aprendizagem colaborativa que a mesma disponibiliza. A Programação em Pares Distribuídos utilizando aplicativos de compartilhamento vêm se tornando ferramenta pedagógica eficaz para uma colaboração flexível e para isso, com o intuito de gerar maior interação entre os alunos por meio da codificação em pares, esta pesquisa propôs o protótipo CodeinPy para que os alunos tenham um ambiente para codificação em Python e compartilhamento de seus códigos em tempo real, podendo acessá-la de suas casas e realizarem juntos as atividades propostas pelo professor, colaborando entre si por meio de sala virtual e *Chat*. Este protótipo será apresentado e melhor detalhado mais adiante.

3.2 Considerações Finais do Capítulo

Este capítulo apresentou um estudo sobre 30 ferramentas utilizadas para o ensino/aprendizagem de Programação de Computadores, no qual dentre elas, sete foram pré-selecionadas por terem como conteúdo a Linguagem de Programação Python.

As ferramentas pré-selecionadas foram comparadas em relação a 24 dimensões pedagógicas e foi criado um *ranking* no qual a mais bem pontuada foi a selecionada para utilização neste estudo.

Um novo critério foi adicionado a esse *ranking* que é o uso de *Gamification* adicionando um ponto a mais ao total de pontos possíveis, e após análise desse comparativo verificou-se que a ferramenta Codecademy é a mais completa em relação às 24 dimensões pedagógicas, além de utilizar os conceitos de *Gamification*.

A metodologia *Flipped Classroom* não define ferramentas para o seu uso, logo o levantamento e comparação de ferramentas para apoio a essa metodologia é uma importante contribuição desta pesquisa.

O próximo capítulo apresenta um protótipo de um ambiente *online* desenvolvido para a aprendizagem colaborativa que visa complementar a metodologia da sala de aula invertida trazendo inovações tecnológicas buscando aumentar o envolvimento dos alunos na aprendizagem de Programação de Computadores, permitindo que saiam do papel passivo e se tornem responsáveis pelo seu próprio aprendizado.

4 UM PROTÓTIPO PARA APRENDIZAGEM COLABORATIVA

Este capítulo apresenta como foi realizado o processo de desenvolvimento do protótipo proposto por este estudo, suas principais funcionalidades e as tecnologias utilizadas. A Seção 4.1 apresenta o protótipo CodeinPY. As tecnologias utilizadas em seu desenvolvimento são mostradas na Seção 4.2, suas principais funcionalidades na Seção 4.3 e, por último, na Seção 4.4 as considerações finais do capítulo. As telas do protótipo e as contribuições trazidas para complementar a metodologia *Flipped Classroom* também são apresentadas.

4.1 O Protótipo CodeinPY

Buscando formas de aumentar a interação entre alunos e a motivação pela aprendizagem dos conceitos de Programação de Computadores, uma das estratégias propostas por este estudo foi a utilização da Aprendizagem Colaborativa, a qual é a maneira de uma ou mais pessoas tentarem aprender algo juntas.

Uma das formas para se atingir este fim foi a utilização dos conceitos da Programação em Pares, na qual dois programadores desenvolvem colaborativamente, com um se encarregando de codificar e o outro atuando como observador, sendo este responsável por revisar o código, prevenindo e identificando possíveis erros.

Estácio & Prikladnicki (2015) afirmam que muitos estudos envolvendo a Programação em Pares e o ensino de Programação de Computadores têm sido realizados e os resultados mostram que sua aplicação cria um ambiente que leva os alunos a uma melhor aprendizagem e interação social, além de aumentar o interesse nos temas de Programação.

Apresentam também o conceito de Programação em Pares Distribuídos (PPD) na qual dois ou mais programadores colaboram para desenvolver software usando ferramentas que permitem o compartilhamento de tela e comunicação como áudio, textos ou vídeos. O trabalho também evidenciou que a PPD trouxe benefícios ao ensino de Programação de Computadores, como a qualidade no código e melhoria no desempenho.

Nesse contexto, visando unir os conceitos de Aprendizagem Colaborativa e Programação em Pares Distribuídos, foi desenvolvido neste trabalho um protótipo mesclando essas estratégias, voltado para o ensino/aprendizagem de Programação de Computadores, permitindo aos alunos por meio de uma ferramenta na *web* desenvolver seus códigos ao mesmo tempo em que pudessem

trocar informações e conhecimento com outros alunos, dentro ou fora do ambiente escolar.

Antes de se dar início ao desenvolvimento do protótipo foi realizado um levantamento sobre ferramentas para aprendizagem colaborativa considerando os critérios:

- Codificação em tempo real;
- *Real-time coding tools*;
- *Real-time collaboration coding*; e
- Programação colaborativa.

Para isso, foi realizada uma consulta na plataforma Github²³ que é um repositório de códigos compartilhado pelo mundo todo, na qual uma comunidade com mais de 24 milhões de pessoas aprendem, compartilham e trabalham juntas para criar *software*.

Nessa pesquisa, dos projetos encontrados, foi selecionado apenas o que melhor se encaixava na linha de pensamento desta pesquisa. Foi levada em consideração a licença do projeto, preocupando-se na reutilização do código e possíveis problemas com direitos autorais.

O projeto selecionado foi o Code-in-tune²⁴ que utiliza a licença MIT que permite que seu código seja reaproveitado, modificado e comercializado sem maiores problemas, cujas características são mostradas na Figura 4.1.

Figura 4.1 - Características da Licença MIT



Fonte: <http://escolhaumalicensa.com.br/licencas/mit/>

²³ <https://github.com/>

²⁴ <https://github.com/fadion/Code-in-Tune>

Basta criar um arquivo de texto geralmente denominado de `license.txt` contendo o texto da licença e adicionar à raiz do código-fonte gerado. É proibida a responsabilidade assegurada, logo o código é fornecido sem garantia e o autor do *software* não se responsabilizará por perdas ou danos.

Neste contexto, o protótipo CodeinPY foi desenvolvido reaproveitando o código do projeto Code-in-tune que era basicamente um editor de texto compartilhado, acrescentando-se novas funcionalidades como a execução e compartilhamento de códigos Python em tempo real, além de alterações nas estruturas das páginas, personalização de cores e tradução para português.

Este protótipo²⁵ foi disponibilizado totalmente na *Web*, logo, basta um computador com *Internet* e um navegador para utilizá-lo, o que é uma característica positiva, já que os alunos não precisam se preocupar em instalar ferramentas, voltando totalmente o seu foco para a aprendizagem dos conceitos de Programação de Computadores e o desenvolvimento na linguagem Python.

Após este breve histórico da origem da ferramenta, a seção a seguir irá detalhar as tecnologias utilizadas para o desenvolvimento do protótipo CodeinPY.

4.2 Tecnologias Utilizadas

Como se trata de uma ferramenta utilizada para acesso via *Web*, todas as tecnologias utilizadas no desenvolvimento do protótipo são voltadas para esse ambiente: Node.js²⁶, Express²⁷, Socket.IO²⁸ e o editor Ace.js²⁹. Para a apresentação de suas páginas, as linguagens HTML³⁰, CSS³¹ e JavaScript³² também foram utilizadas.

A principal funcionalidade presente no protótipo CodeinPY que o diferencia do código Code-in-tune foi a inclusão de um interpretador Python, para isso foi utilizada parte da biblioteca Skulpt³³ que permite a execução de códigos fonte Python através do navegador, o script foi adicionado e aplicado em conjunto com o editor Ace.js. A Figura 4.2 apresenta o código fonte

25 <http://200.137.171.4:1337/>

26 <https://nodejs.org/en/>

27 <http://expressjs.com/pt-br/>

28 <https://socket.io/>

29 <https://ace.c9.io/>

30 <https://www.w3.org/html/>

31 <https://www.w3.org/Style/CSS/Overview.en.html>

32 <https://www.javascript.com/>

33 <http://www.skulpt.org/>

utilizado para inclusão do Interpretador no protótipo CodeinPY.

Figura 4.2 - Código Utilização Interpretador Python

```

<script src="/js/app/room.js" type="text/javascript"></script>
<script src="/js/lib/skulpt-kw.js" type="text/javascript"></script>
<script>
    var editor = ace.edit("editor");
    function outf(text){
        var output = document.getElementById("running");
        text = text.replace(/</g, '&lt;');
        output.innerHTML = output.innerHTML + text;
    }
    $('#run').click(function(){
        var prog = editor.getValue();
        var output = document.getElementById("running");
        output.innerHTML = '>>>> ';
        Sk.configure({output:outf});
        try {
            Sk.importMainWithBody("<stdin>", false, prog);
        } catch (e) {
            alert(e);
        }
        return false;
    });

    editor.commands.addCommand({
        name: 'myCommand',
        bindKey: {win: 'Ctrl-R', mac: 'Command-R', lin: 'Ctrl-R'},
        exec: function(editor) {
            var prog = editor.getValue();
            var output = document.getElementById("running");
            output.innerHTML = '>>>> ';
            Sk.configure({output:outf});
            try {
                Sk.importMainWithBody("<stdin>", false, prog);
            } catch (e) {
                alert(e);
            }
        },
        readOnly: true
    });
</script>

```

Fonte: Elaborada pela Autora (2017)

Pereira (2014) define o Node.js como uma plataforma altamente escalável, orientada a eventos, leve, eficiente e ideal para aplicações em tempo real com troca intensa de dados. Utiliza como linguagem o JavaScript e a engine Javascript V8³⁴, a mesma utilizada no navegador Google Chrome³⁵.

Assim como o Javascript, é orientado a eventos, sendo que o Javascript trabalha no lado cliente e o Node.js com eventos de I/O (entrada/saída) do servidor. Tornou-se popular por oferecer bibliotecas de baixo nível que dão suporte a diversos protocolos, tais como o protocolo WebSockets³⁶. Infelizmente nem todos os *browsers* dão suporte a esse recurso, e diante disso, nasceu o Socket.IO³⁷ para resolver essa incompatibilidade.

Socket.IO permite comunicação baseada em eventos bidirecionais em tempo real. É bastante utilizado para criação de *Chat*. Neste projeto foi utilizado em conjunto com o Express³⁸ um *framework* simples e flexível que fornece um conjunto robusto de recursos fundamentais para aplicativos da Web. Foram utilizados para criação do *chat* que viabiliza o meio de comunicação entre alunos no protótipo.

O editor de texto Ace.js é um editor escrito em JavaScript que combina vários recursos e pode ser facilmente incorporado em qualquer página da *Web* ou aplicação JavaScript. É o editor utilizado na IDE Cloud9³⁹ e possui como principais características a possibilidade de mais de 20 temas para personalização, indentação automática de código, linha de comando opcional, funcionalidade de copiar, recortar e colar, entre outras.

A versão do Python reconhecida e utilizada pelo Ace.js foi a versão 2.0, a mesma versão utilizada pelo curso de Python da Codecademy para que os alunos não se confundissem em relação às pequenas diferenças de sintaxe encontradas entre as versões Python 2.0 e 3.0.

Apresentadas as tecnologias utilizadas para o desenvolvimento do protótipo, a seção a seguir lista as principais funcionalidades da ferramenta CodeinPY e os benefícios que a mesma traz tanto para o professor como para a aprendizagem dos alunos. A Figura 4.3 mostra a tela inicial da ferramenta.

34 <https://developers.google.com/v8/>

35 <https://www.google.com/chrome/>

36 <https://www.websocket.org/>

37 <https://socket.io/>

38 <http://expressjs.com/pt-br/>

39 <https://c9.io/>

Figura 4.3 - Tela Inicial do Protótipo CodeinPY



Fonte: Elaborada pela Autora (2017)

4.3 Funcionalidades do Protótipo

O protótipo CodeinPY possui como principal funcionalidade a possibilidade da codificação e execução de forma colaborativa de códigos Python via navegador *Web*. Ou seja, a área onde se codifica pode ser compartilhada em tempo real com outras pessoas possibilitando a codificação por ambas as partes que utilizam e compartilham a mesma sala virtual ao mesmo tempo. Para que todos colaborem no mesmo código, basta que criem uma sala de aula virtual e compartilhem da mesma URL. A partir daí, todos os participantes da sala podem codificar e alterar seus códigos de forma compartilhada.

Essa funcionalidade aplica os conceitos da Aprendizagem Colaborativa no sentido em que por meio dessa ferramenta, os alunos buscam resolver as atividades passadas em sala de aula de forma colaborativa, ou seja, buscam aprender os conceitos de Programação de Computadores juntos, trocando conhecimento entre si.

Para reforçar ainda mais esta ideia, a Programação em Pares também é utilizada, já que na interação entre os pares, geralmente um dos alunos tem mais conhecimento em determinado assunto e fica no papel do piloto enquanto o outro observa a codificação, havendo um revezamento, o que observa passa a ser o piloto e vice-versa. Com isso, ambas as partes participam e aprendem juntas algo novo.

A ferramenta também disponibiliza um *Chat Online* possibilitando assim a conversação entre as partes, separando a área para codificação e a área para comunicação. Por meio desse *chat*,

os alunos podem informar/sanar suas dúvidas, comunicar algum erro de sintaxe no código que a outra parte possivelmente não tenha percebido, ou sugerir algum melhoramento no código compartilhado, entre outras possibilidades.

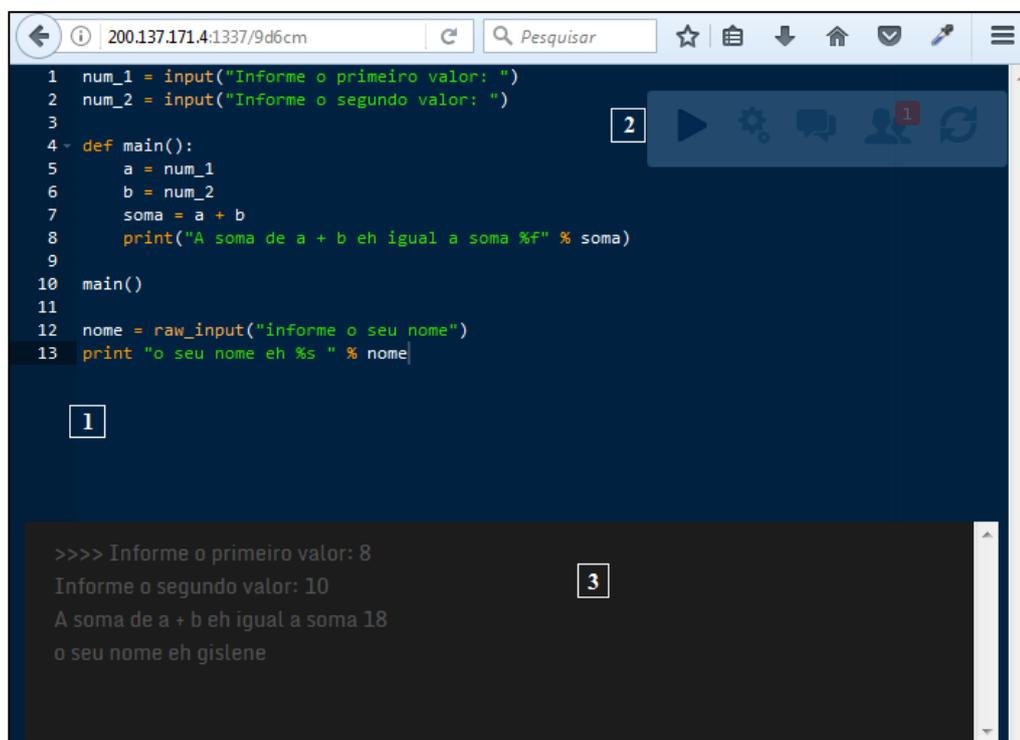
Possui uma área na qual há a informação da quantidade de pessoas participantes da sala virtual, não impondo limitações no código em relação à quantidade de pessoas *online*. Além disso, possibilita personalização, na qual o aluno se identifica pelo nome. Assim, ao trocar mensagens é possível a identificação de quem está *online*.

Além disso, o participante da sala pode personalizar a cor de fundo da área disponibilizada para codificação escolhendo um dos temas disponíveis na ferramenta.

Para o professor, a ferramenta CodeinPY pode funcionar como uma espécie de "lousa digital", pois, permite que o professor codifique exemplos na ferramenta e possibilita que todos os alunos participantes da sala virtual visualizem individualmente, aumentando assim os recursos pedagógicos.

Possibilita também ao professor a oportunidade de inserir novas metodologias para aumentar a motivação na aprendizagem e interação entre os alunos visando incrementar a metodologia de ensino *Flipped Classroom*. A Figura 4.4 apresenta a tela inicial de uma sala virtual do protótipo CodeinPY e a Figura 4.5 os seus Casos de Uso.

Figura 4.4 - Tela da Sala Virtual do Protótipo CodeinPY



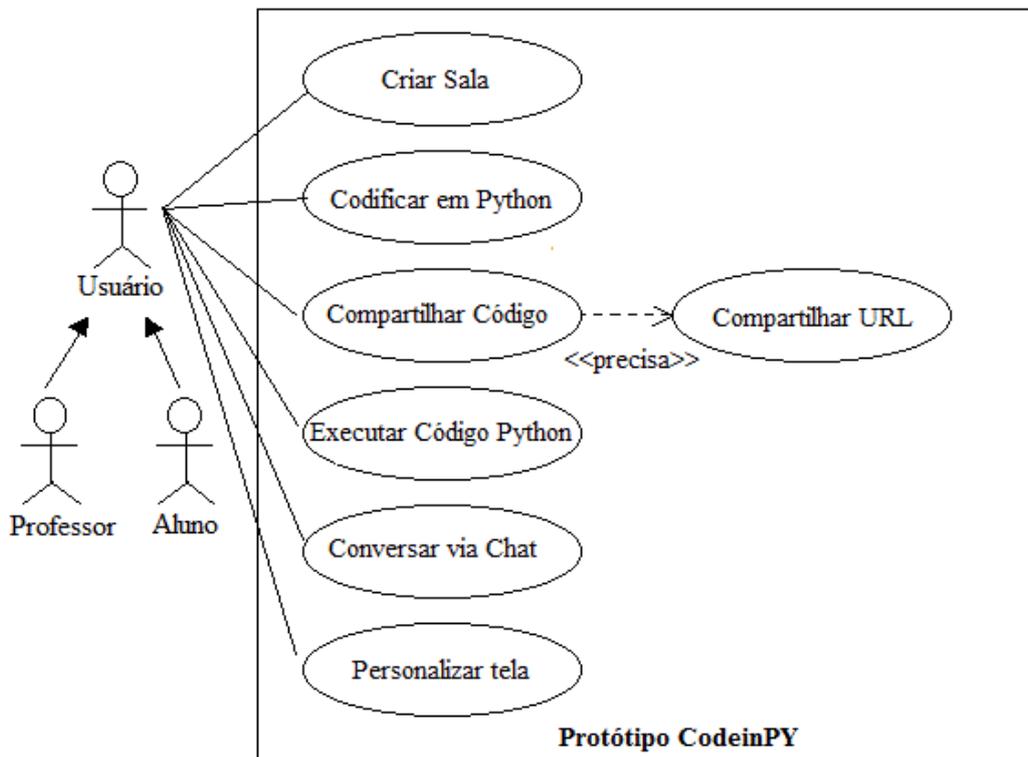
```
1 num_1 = input("Informe o primeiro valor: ")
2 num_2 = input("Informe o segundo valor: ")
3
4 def main():
5     a = num_1
6     b = num_2
7     soma = a + b
8     print("A soma de a + b eh igual a soma %f" % soma)
9
10 main()
11
12 nome = raw_input("informe o seu nome")
13 print "o seu nome eh %s " % nome
```

>>>> Informe o primeiro valor: 8
Informe o segundo valor: 10
A soma de a + b eh igual a soma 18
o seu nome eh gislene

Fonte: Elaborada pela Autora (2017)

A tela inicial da sala virtual do protótipo é composta por três áreas, sendo 1 a área onde os alunos irão escrever seus códigos em Python, Área 2 composta por cinco funcionalidades (Executar código, configurações, *Chat*, usuários *online* e endereço para compartilhamento, respectivamente) e Área 3 onde serão exibidos os resultados da execução do código fonte.

Figura 4.5 - Casos de Uso do protótipo CodeinPY



Fonte: Elaborada pela Autora (2017)

O protótipo é bastante simples e de fácil utilização. Com um mínimo de treinamento qualquer usuário consegue utilizá-lo. A arquitetura usada para implantação do protótipo em produção para a realização dos experimentos é apresentada no Quadro 4.3.

Quadro 4.3 - Arquitetura da Máquina utilizada pelo Protótipo CodeinPY

Arquitetura Máquina	
Sistema Operacional	Linux - Debian 7
Processadores/ Núcleos	4 processadores de 2.6 GHz
Memória RAM	4 GB
Espaço em Disco	120 GB

Fonte: Elaborada pela Autora (2017)

Percebe-se que não se fez necessária a utilização de uma configuração suntuosa para que o protótipo funcionasse sem maiores gargalos, o que permite a quem for utilizá-lo, implantar sem muitos custos. O Quadro 4.4 apresenta uma breve descrição dos casos de uso representados na Figura 4.5.

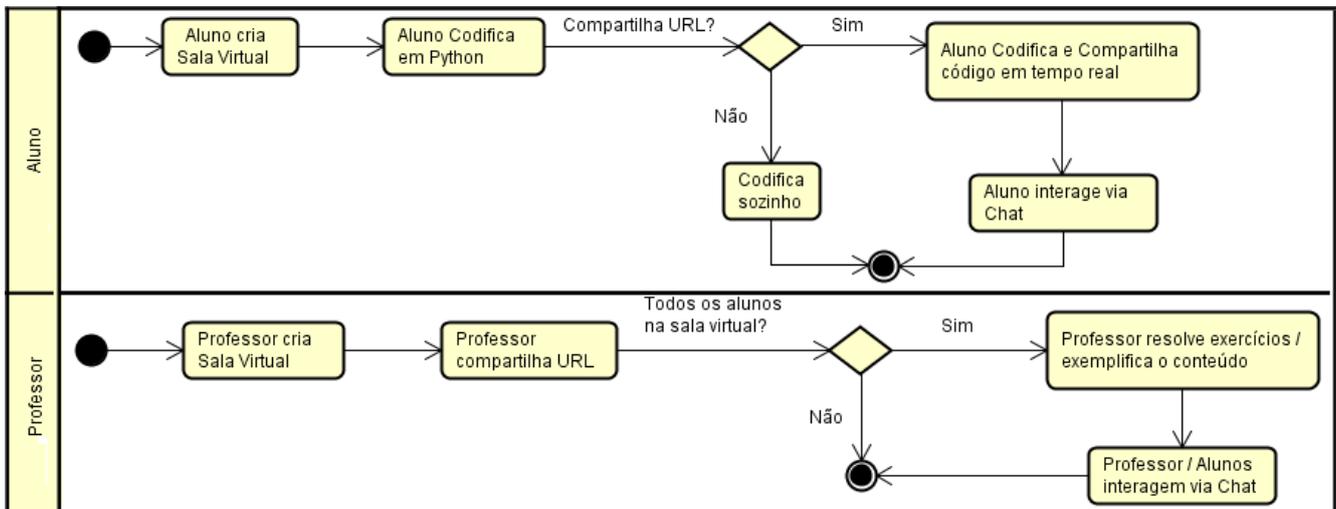
Quadro 4.4 - Descrição dos Casos de Uso

Casos de Uso	
Criar Sala	Professor ou Aluno cria sala de aula virtual.
Codificar em Python	Após sala criada, Aluno ou Professor codifica em Python.
Compartilhar URL	Professor ou Aluno precisa repassar a URL da sala virtual criada para que possa codificar em tempo real de forma colaborativa.
Compartilhar código	Após URL compartilhada, o código é compartilhado em tempo real com demais participantes da sala virtual criada.
Executar código Python	Código é executado e resultado é apresentado no protótipo.
Conversar via Chat	Usuários que utilizam a mesma sala virtual, ou seja, utilizam a mesma URL podem se comunicar via Chat.
Personalizar tela	Usuário pode escolher um dos temas disponíveis no protótipo para personalização da cor de fundo da tela.

Fonte: Elaborada pela Autora (2017)

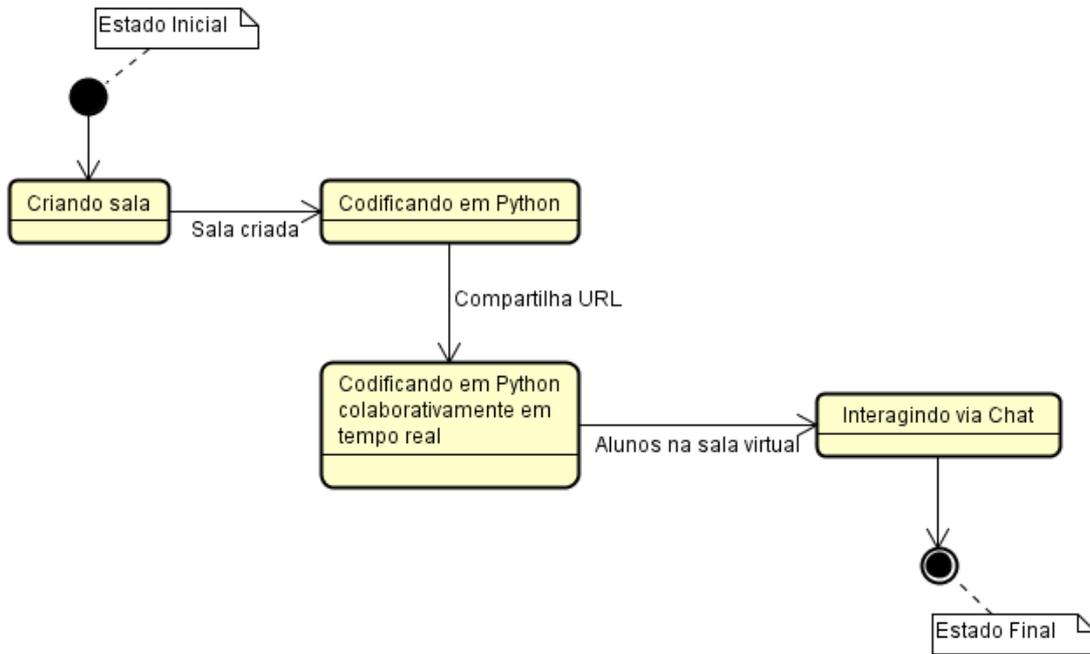
As figuras 4.6, 4.7 e 4.8 mostradas a seguir, apresentam os diagramas de atividades, máquina de estados e de sequência, respectivamente.

Figura 4.6 - Diagrama de Atividades



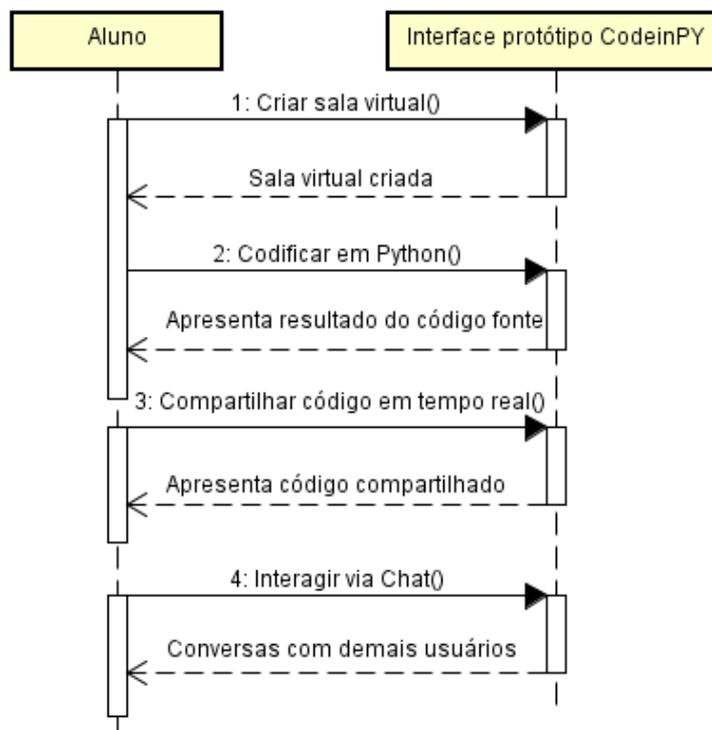
Fonte: Elaborada pela Autora (2017)

Figura 4.7 - Diagrama de Máquina de Estados



Fonte: Elaborada pela Autora (2017)

Figura 4.8 - Diagrama de Sequência



Fonte: Elaborada pela Autora (2017)

4.4 Considerações Finais do Capítulo

Este capítulo evidenciou as características do protótipo CodeinPY apresentando suas funcionalidades e tecnologias aplicadas. Seu desenvolvimento teve a finalidade de aplicar os conceitos da Aprendizagem Colaborativa por meio da Programação em Pares Distribuídos disponibilizando importante contribuição para estimular os alunos no processo de aprendizagem.

Para os professores se apresenta como um inovador aparato tecnológico, pois com apenas um computador e *Internet* eles podem explicar o conteúdo aos seus alunos codificando, estimulando os alunos à prática de programar, além de ser uma ferramenta adaptável já que seu código fonte continuará em código aberto, podendo o professor modificar, melhorar e acrescentar novas funcionalidades, adaptando à sua realidade.

O capítulo a seguir detalhará como foram realizados os experimentos para verificação se os objetivos desta pesquisa foram alcançados, apresentando os meios utilizados para coleta dos dados e a realização da análise dessas informações.

5 EXPERIMENTOS

Neste capítulo, na Seção 5.1, os procedimentos metodológicos para realização dos experimentos deste trabalho são apresentados. A Seção 5.2 descreve os participantes do estudo. Na Seção 5.3 são descritas as ferramentas adotadas na pesquisa. Na Seção 5.4 os procedimentos são apontados. Na Seção 5.5 é apresentada a coleta de dados. Na Seção 5.6, evidencia-se como foi realizada a análise de dados. E, por último, a Seção 5.7 apresenta um resumo geral das contribuições apresentadas neste capítulo.

5.1 Metodologia

A pesquisa visa observar a motivação dos alunos na aprendizagem de Programação e as habilidades desenvolvidas na autoaprendizagem por meio de ambiente *online* de aprendizagem colaborativa baseada na metodologia de ensino *Flipped Classroom*, uso de *Gamification* e Aprendizagem colaborativa.

Para viabilizar os objetivos desse estudo os seguintes passos metodológicos foram aplicados:

- Levantamento bibliográfico inicial;
- Análise de ambientes virtuais de aprendizagem para determinar o que pode ser usado como apoio ao uso das metodologias ativas no processo ensino/aprendizagem;
- Desenvolvimento de um protótipo para a aprendizagem colaborativa;
- Formação dos usuários (alunos e docente) sobre o *Flipped Classroom* e ferramentas propostas;
- Observação das interações e os níveis de comunicação do ambiente colaborativo de aprendizagem verificando a forma como os alunos trabalham em grupo (ex: Programação em Pares);
- Identificação dos problemas relacionados á prática do ensino de Programação de Computadores (Aplicação de questionários, revisão de literatura); e
- Análise da efetividade das práticas de ensino e aprendizagem de Programação de Computadores por meio das ferramentas educativas utilizadas.

5.2 Participantes

A pesquisa tem como sujeitos alunos do 1º ano do Curso Técnico em Informática integrado ao Ensino Médio do Instituto Federal do Piauí (IFPI), com um total de 40 alunos. Os alunos possuem idade entre 14 e 16 anos, sendo 15 do sexo feminino e 25 do sexo masculino, sem conhecimento prévio em Algoritmos e Linguagens de Programação. A professora participante possui doutorado em Computação e experiência de vários anos no ensino de Programação de Computadores.

5.3 Ferramentas

As ferramentas selecionadas para a realização da presente pesquisa foram: Codecademy⁴⁰ para aplicação da metodologia *Flipped Classroom* e conceitos de *Gamification* e o protótipo CodeinPY, desenvolvida para aplicação dos conceitos de Aprendizagem Colaborativa, ambas voltadas para o ensino dos conceitos de Algoritmos e Programação de Computadores utilizando a Linguagem de Programação Python. Logo:

- Codecademy: *Flipped Classroom* + *Gamification*;
- CodeinPY: Aprendizagem Colaborativa.

Com Codecademy os alunos aprendem por meio de pequenos desafios e exemplos, codificando em um editor dentro da própria ferramenta tendo a correção em tempo real da lógica e semântica do seu código.

Também possibilita a aprendizagem colaborativa a partir da participação dos alunos em fóruns de discussão e *blogs*, utiliza elementos de *gamification* como medalhas e permite ao aluno o acompanhamento do seu progresso e ao professor o acompanhamento individual de cada aluno da sua turma.

O curso de Python utilizado nesta pesquisa e disponibilizado na ferramenta Codecademy está estruturado da seguinte forma:

- Sintaxe Python;
- Strings e Exibição no Console;

⁴⁰ <https://www.codecademy.com/pt>

- Fluxo de Controle e Condições;
- Funções, Listas e Dicionários;
- Laços;
- Tópicos Avançados em Python;
- Introdução às Classes; e
- Entrada e Saída de Arquivos.

Para a realização do experimento com os alunos, o assunto visto pela turma foi limitado aos seguintes itens: Sintaxe Python, Strings e Exibição no Console, Fluxo de controle e Condições.

Como visto, a ferramenta Codecademy auxilia no ensino da Linguagem de Programação Python, utilizando conceitos de *Gamification* para maior engajamento dos alunos e utiliza fóruns para que os alunos possam interagir, aplicando os conceitos de aprendizagem colaborativa. O fórum é bastante utilizado como forma dos alunos trocarem conhecimento, porém, muitas das informações se perdem devido ao grande fluxo dessas informações.

Visando aumentar a interação entre os alunos e tornar mais ativo o seu papel no controle da sua aprendizagem, a codificação em pares em tempo real seria um requisito a mais para essa ferramenta, o qual ela não possui. Logo, têm-se uma lacuna que será preenchida por meio do protótipo desenvolvido, CodeinPY, criado com o intuito de observar a interação entre os pares e aumentar a motivação pela aprendizagem dos conceitos de Programação de Computadores, incentivando aos alunos aprenderem concomitantemente.

5.4 Procedimentos

Inicialmente, a pesquisadora buscou manter contato com a professora participante e os coordenadores do curso por meio de reuniões nas quais foram explanados os objetivos geral e específicos da pesquisa, quando apresentaram interesse em participar do estudo.

O Instituto Federal do Piauí foi escolhido como ambiente para aplicação da pesquisa devido a oferecer as condições necessárias e adequadas aos interesses sugeridos pela investigação, pois, seria necessário uma turma inexperiente em relação aos conhecimentos sobre Algoritmos e Programação de Computadores.

Após a concordância dos coordenadores e da professora da disciplina em auxiliar na

pesquisa, um cronograma foi definido com as atividades e os períodos estabelecidos conforme o Quadro 5.1, iniciando-se em Maio de 2017.

Quadro 5.1 - Cronograma das etapas da Pesquisa

Período	Atividade
02/05/2017 a 08/05/2017 - 1 semana	Formação da Professora
15/05/2017 a 17/05/2017 - 3 dias	Cadastros dos alunos na ferramenta
18/05/2017 a 19/05/2017 - 2 dias	Preparação dos ambientes Codecademy e CodeinPY
25/05/2017 a 08/06/2017 - 2 semanas	Observação em sala de aula
13/06/2017 a 19/07/2017 - 1 mês e 1 semana	Observação em laboratório ferramentas Codecademy e CodeinPY
20/07/2017 - 1 dia	Aplicação de avaliação e questionário
21/07/2017 a 23/07/2017 - 3 dias	Extração das respostas do questionário
24/07/2017 a 08/08/2017 - 2 semanas	Análise dos registros de atividades em laboratório
10/08/2017 a 31/08/2017 - 3 semanas	Análise das respostas do questionário

Fonte: Elaborado pela Autora (2017)

A professora da disciplina Algoritmos e Programação foi selecionada para participar da pesquisa já que no período ministrava aulas em uma turma apta à realização dos experimentos, uma vez que havia a necessidade dos alunos serem inexperientes em relação ao conhecimento de Algoritmos e Programação de Computadores, como já citado.

A professora recebeu orientação de forma individual sobre a metodologia de ensino *Flipped Classroom* e como será aplicada em sala de aula. Foi visto o uso da ferramenta Codecademy e como será realizado o acompanhamento dos alunos por meio da ferramenta e a forma como o material é disponibilizado. Para a aprendizagem colaborativa foi apresentada a ferramenta CodeinPY e a sua utilização.

5.5 Coleta de Dados

Esta seção apresenta como foi realizada a coleta dos dados nesta pesquisa. Na Seção 5.5.1 expõe-se como aconteceram as observações e a separação das etapas para realização dos experimentos. Já a

Seção 5.5.2 descreve como foram realizados os experimentos e apresenta a descrição das observações em laboratório do uso das ferramentas Codecademy e CodeinPY. Por último, na Seção 5.5.3 descreve-se a avaliação e os questionários aplicados com os alunos.

5.5.1 Observações e Organização dos Experimentos

As observações em sala de aula ocorreram duas vezes na semana, na disciplina de Algoritmos e Programação. Durante esses dias, as observações das aulas tinham duração de 100 minutos.

Nos primeiros dias da observação, foi realizado um levantamento de informações se a turma já possuía conhecimento em Algoritmos ou em alguma Linguagem de Programação e foi constatado que a turma não possuía esse conhecimento prévio.

Com isso, as aulas iniciaram com uma explanação inicial sobre o que são Algoritmos, e sobre o que são Linguagens de Programação para que os alunos conseguissem se familiarizar com os conceitos. Nesse primeiro momento, todos os 40 alunos assistiram às aulas seguindo apenas a metodologia tradicional de ensino. Essa observação durou duas semanas e serviu para um processo introdutório do conteúdo aos alunos.

Ao final desse período foram realizadas algumas reuniões entre a professora e a pesquisadora, nas quais se discutia como seriam as observações em laboratório e como seria realizada a coleta e análise dos dados coletados.

Após essas duas semanas, foram explanadas aos alunos a motivação da pesquisa e como seriam realizadas as suas etapas. Também foi explicado que haveria uma divisão da turma e que todas as aulas seriam realizadas em laboratório e em ambiente controlado. Os experimentos foram realizados seguindo as etapas apresentadas a seguir:

- **Etapa 1** - Separação da turma em duas partes, onde foram utilizados 2 laboratórios, sendo alocados 20 alunos em cada. Foi levada em consideração a ordem do número de chamada, os 20 primeiros em um laboratório e o restante em outro;
- **Etapa 2** - Após a separação da turma, foi definido que uma parte teria as aulas com a metodologia tradicional, de forma expositiva, onde o professor expõe o conteúdo da disciplina e ao final atribui atividades para casa e na outra parte foi aplicada a metodologia *Flipped Classroom*, onde os alunos tinham acesso ao conteúdo da aula por meio da ferramenta Codecademy, e as resoluções e discussões das atividades seriam realizadas em

sala de aula, utilizando-se o protótipo CodeinPY;

- **Etapa 3** - Preparação da aula expositiva com a professora para que fosse equivalente à ordem do material disponibilizado na Codecademy, de modo que os alunos vissem o mesmo conteúdo na mesma sequência;
- **Etapa 4** - Treinamento dos alunos que iriam utilizar a ferramenta Codecademy. Foram criados usuários e senhas para cada aluno e realizados testes em laboratório;
- **Etapa 5** - Início do experimento com os alunos em laboratório, de forma simultânea, onde a professora participante ministrava aula expositiva em um laboratório para parte da turma e a pesquisadora em outro laboratório observava os alunos utilizando a ferramenta Codecademy;
- **Etapa 6** - Treinamento dos alunos na ferramenta CodeinPY, a qual foi demonstrada para todos os alunos de ambos laboratórios e realizados testes de interação entre eles;
- **Etapa 7** - Observação em laboratório dos alunos que viram o conteúdo da disciplina pela metodologia *Flipped Classroom* utilizando a ferramenta Codecademy e das eventuais dificuldades que foram surgindo;
- **Etapa 8** - Observação da interação dos alunos na utilização do protótipo CodeinPY como meio para resolução das atividades em laboratório, aplicação da aprendizagem colaborativa (Programação em Pares);
- **Etapa 9** - Ensaio do experimento "A", explicação do passo a passo de como irá funcionar o experimento a fim de simular o evento;
- **Etapa 10** - Realização do experimento "A" valendo 2 pontos para avaliação do protótipo CodeinPY, investigação se a mesma foi efetiva em relação a interação entre os laboratórios e se os alunos conseguiram aprender algo juntos;
- **Etapa 11** - Realização do experimento "B" valendo 8 pontos com todos os alunos da turma para avaliar a absorção do conteúdo por ambas metodologias. Esse experimento foi realizado em dia diferente ao experimento "A";
- **Etapa 12** - Coleta dos dados por meio da aplicação de questionários com todos os alunos da turma: um sobre a utilização e efetividade do protótipo CodeinPY e outro apenas com os alunos que viram o conteúdo pela metodologia *Flipped Classroom* para avaliar a ferramenta Codecademy;
- **Etapa 13** - Coleta dos dados enviados pela professora participante com o resultado da

junção das notas do experimento "A" e "B" que formaram a primeira nota da disciplina;

- **Etapa 14** - Aplicação de questionário com a professora participante a fim de avaliar a experiência da utilização das ferramentas propostas como estratégias de ensino/aprendizagem dos conceitos de Programação de Computadores; e
- **Etapa 15** - Análise da coleta de todos os dados colhidos pela investigação a fim de se verificar se foram atingidos ou não os objetivos propostos por esse estudo. Os experimentos citados nas etapas 10 e 11 serão detalhados mais adiante.

As observações em laboratório buscaram analisar a utilização das ferramentas Codecademy e CodeinPY. Todas as aulas onde os alunos viam o conteúdo na Codecademy foram realizadas em laboratório, em ambiente controlado e observado pela pesquisadora.

Conforme os alunos iam vendo o conteúdo na Codecademy, antes de darem início a realização das atividades em sala e resolução de problemas, foi demonstrada outra ferramenta, a CodeinPY, desenvolvida pela pesquisadora para que os alunos pudessem escrever e executar seus códigos de forma colaborativa em tempo real, onde poderiam trocar informações e tentar resolver em conjunto as atividades aplicadas.

Essa mesma ferramenta foi utilizada em ambos os laboratórios, tanto com os alunos que utilizaram a metodologia tradicional de ensino quanto com os que utilizaram o *Flipped Classroom*.

O que se pôde perceber durante o período de observações foram as dúvidas que iam surgindo durante a utilização da ferramenta Codecademy. Muitas delas se davam por problemas de interpretação de texto, e após breve explicação, os alunos conseguiam prosseguir adiante no conteúdo de forma autônoma. Percebeu-se também um maior estímulo em resolver as atividades propostas pela própria ferramenta, já que o aluno só consegue evoluir no assunto após cumprir o desafio determinado pela ferramenta, como se fosse um jogo, no qual o aluno tem que passar por fases onde há níveis de dificuldade.

A ferramenta Codecademy permite ao professor avaliar por meio da observação da ferramenta a evolução de cada aluno de forma individualizada, já que cada aluno tem seu próprio ritmo de aprendizado, possibilitando a averiguação da dificuldade que cada um possui de modo a criar formas de se trabalhar essas dificuldades de maneira personalizada.

Na observação do protótipo CodeinPY o que se percebeu foi a real interação entre os alunos buscando responder as atividades juntos e compartilhando dúvidas por meio do bate-papo. Apenas com treinamento simples de utilização desse protótipo foi possível a utilização dele sem surgimento

de maiores dúvidas, o que confirma ser uma ferramenta simples e de fácil utilização.

Tendo em vista aumentar a participação dos alunos durante a resolução das atividades, eles foram estimulados a responder e expor suas respostas aos outros alunos, onde os outros alunos buscam soluções semelhantes e apresentam uma forma de resolução alternativa a apresentada utilizando o protótipo CodeinPY.

A seção a seguir apresentará de forma mais detalhada a descrição de cada um dos experimentos realizados e quais objetivos alcançados.

5.5.2 Descrição dos Experimentos

Primeiramente, um dos pontos a serem observados durante a utilização das ferramentas é verificar como os alunos avaliariam o uso da ferramenta Codecademy, se sentiram dificuldade ou se de alguma forma não conseguiram entender os assuntos disponibilizados na ferramenta. Outro ponto foi avaliar se o protótipo CodeinPY possibilitou a esses alunos interagirem e aprenderem de forma colaborativa, ou seja, se utilizando o protótipo conseguiriam aprender algo novo juntos.

Para isso, durante o período de observação, foram realizados dois experimentos distintos "A" e "B", detalhados a seguir:

Experimento "A" - Neste experimento, os 40 alunos que já estavam separados em dois laboratórios distintos, foram reunidos em grupos contendo 4 alunos cada, sendo que desses quatro alunos, 2 estavam no Laboratório 01 e os outros 2 estavam no Laboratório 02.

A definição desses grupos se deu de forma aleatória e após a definição de todos os grupos, totalizou-se a criação de 10 grupos distintos.

No Laboratório 01 estavam os 20 alunos que viram todo o conteúdo pela metodologia *Flipped Classroom* utilizando a ferramenta Codecademy e no laboratório 02 os outros 20 alunos que assistiram aula pela metodologia tradicional de ensino.

Dois alunos de cada laboratório formavam um mesmo grupo, ou seja, os grupos eram formados por dois alunos que viram o assunto pela metodologia tradicional e os outros dois pela metodologia *Flipped Classroom*. A ideia é estarem em laboratórios distintos e utilizarem o protótipo proposto, o CodeinPY, para responderem juntos a um "desafio": construção de um algoritmo baseado nos assuntos aprendidos em sala de aula.

Cada laboratório ficou com 10 grupos distintos, sendo assim, 10 algoritmos diferentes foram utilizados. Ficou determinado que o Grupo 01 do Laboratório 01 trabalharia em conjunto com o

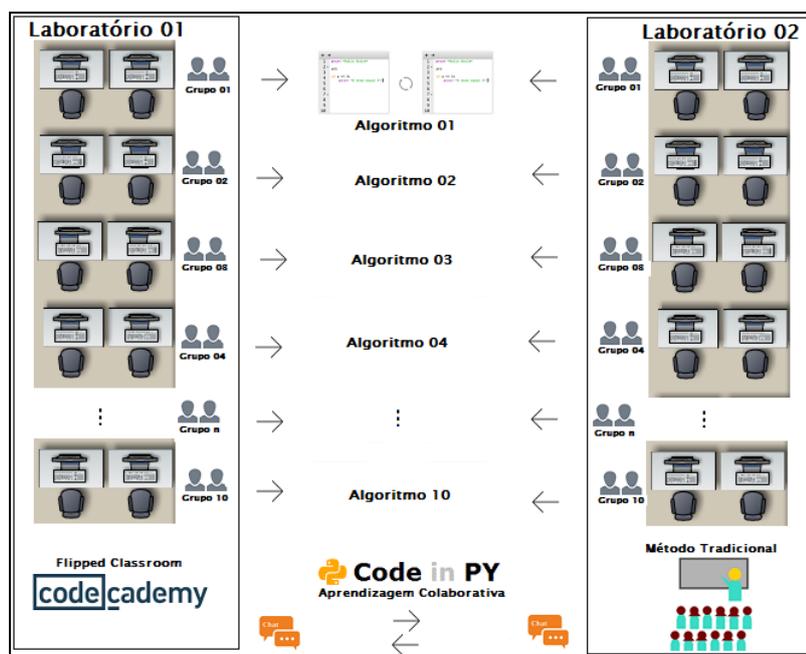
Grupo 01 do laboratório 02, e assim sucessivamente até o grupo 10. Logo, os grupos de mesma numeração, mesmo em laboratórios/locais diferentes, tentariam responder juntos ao desafio, codificando ao mesmo tempo utilizando o protótipo CodeinPY.

O experimento funcionava da seguinte forma: cada grupo criava uma sala virtual na CodeinPY e os 4 alunos participantes que compartilhavam a mesma sala se comunicavam e codificavam juntos utilizando a ferramenta. O compartilhamento da sala se dava pela utilização de todos os participantes do grupo da mesma URL. Logo, todos os alunos que utilizavam o mesmo endereço se encontravam na mesma sala virtual.

Todos os 40 alunos foram treinados para utilização do protótipo CodeinPY e o experimento consistiu em os participantes de um mesmo grupo codificarem de forma simultânea e colaborativa utilizando o protótipo proposto para responderem a um Algoritmo "desafio" trocando informações e chegando a uma resposta única. A pesquisadora observou a interação dos alunos na utilização do protótipo e verificou se cada grupo conseguiu responder ao algoritmo, colaborando entre si e aprendendo algo juntos. Esse experimento aplicado foi dado como qualitativo na nota dos alunos e valeu 2 pontos num total de 10 da nota total individual de cada aluno.

O experimento "A" é detalhado conforme a Figura 5.1 e as figuras 5.2 e 5.3 apresentam os alunos utilizando o protótipo e respondendo as atividades de forma colaborativa:

Figura 5.1 - Experimento "A"



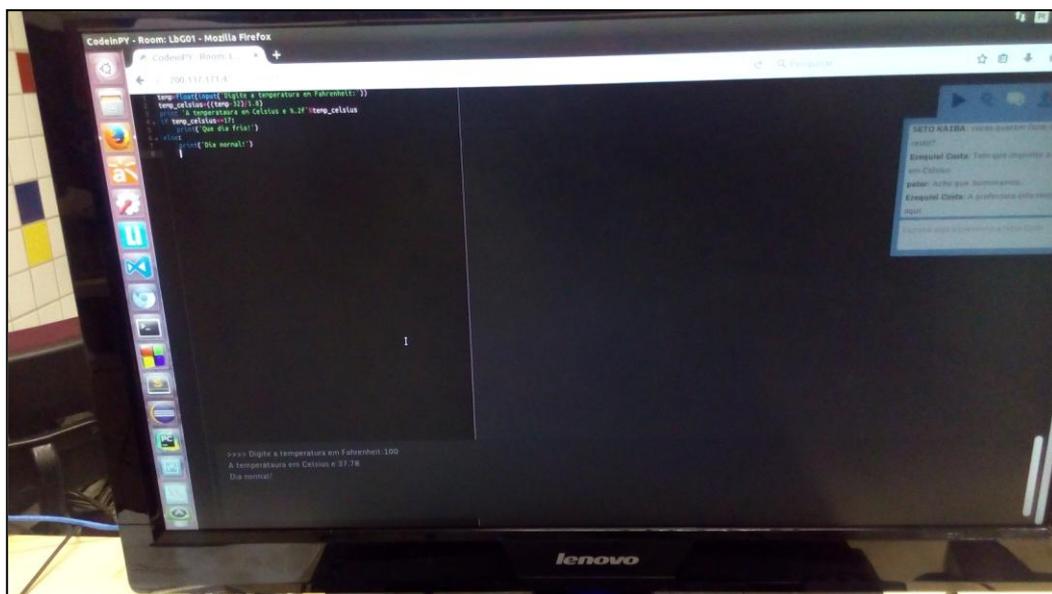
Fonte: Elaborada pela Autora (2017)

Figura 5.2 - Experimento "A" - Utilização do Protótipo CodeinPY



Fonte: Elaborada pela Autora (2017)

Figura 5.3 - Experimento "A" - Tela de Grupo resolvendo Algoritmo no Protótipo CodeinPY



Fonte: Elaborada pela Autora (2017)

A função do protótipo CodeinPY nesse experimento é proporcionar a esses alunos um ambiente colaborativo no qual possam interagir e tentar responder juntos ao desafio proposto, buscando assim aprenderem juntos resolvendo a mesma atividade.

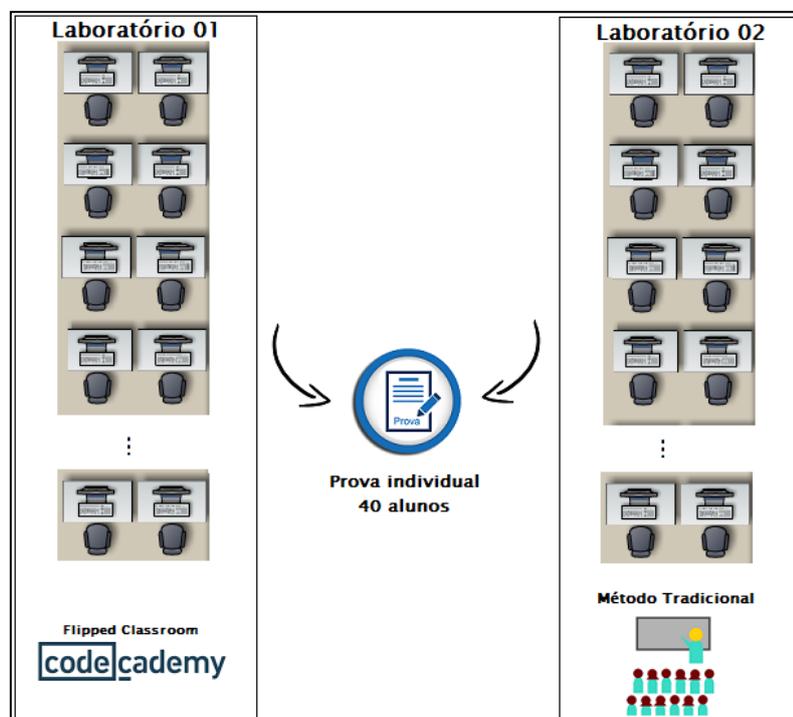
O objetivo a ser atingido por esse experimento foi verificar se a utilização do protótipo foi

efetiva como composição das estratégias para ensino/aprendizagem dos conceitos de Programação de Computadores aplicando o conceito da Aprendizagem Colaborativa, ou seja, se como ferramenta, possibilitou aos alunos aprenderem juntos de forma colaborativa trabalhando em equipe baseando-se na Programação em Pares.

Experimento "B" - Na semana seguinte ao término do experimento "A" foram finalizados os assuntos de Python vistos em laboratório durante as observações, tanto via metodologia tradicional de ensino quanto via metodologia *Flipped Classroom*. O conteúdo visto pelos alunos de ambos os laboratórios seguiu a mesma sequência (presente na ferramenta Codecademy) para que não houvesse distinção do conteúdo apresentado, mas apenas a diferença na metodologia de ensino aplicada.

Os 40 alunos da turma foram submetidos a uma mesma avaliação valendo 8 pontos sendo a prova formada por quatro questões valendo dois pontos cada e avaliaria o aprendizado de todo o conteúdo visto até então. Para responder a essa avaliação os alunos não precisaram utilizar nenhuma das ferramentas apresentadas. A prova foi individual e aplicada na forma escrita. A Figura 5.4 demonstra como foi realizada a aplicação dessa avaliação:

Figura 5.4 - Experimento "B"



Fonte: Elaborada pela Autora (2017)

O objetivo desse experimento foi avaliar o aprendizado de todos os alunos independentemente da metodologia de ensino aplicada, comprovando a efetividade da metodologia de ensino *Flipped Classroom* para o ensino dos conceitos de Programação de Computadores mediado por ambientes virtuais de ensino.

A coleta dos dados se deu por meio das técnicas de observação presencial em laboratório e aplicação de formulário eletrônico. Os alunos sabiam que estavam sendo observados e não se seguiu um padrão pré-definido para as observações.

Após a realização dos experimentos, dois questionários foram aplicados a fim de consultar os alunos sobre o impacto da utilização das ferramentas propostas no processo de ensino-aprendizagem, verificando se o uso dessas inovações tecnológicas os auxiliaram na compreensão dos conceitos de Programação de Computadores.

As perguntas dos questionários tiveram o intuito de avaliar as ferramentas analisando sua efetividade em relação ao ensino/aprendizagem de Programação de Computadores suscitando possíveis estratégias para que os professores possam aproveitar-se dessas inovações tecnológicas como seus instrumentos pedagógicos.

Sendo que o questionário sobre a ferramenta Codecademy foi respondido apenas pelos alunos que viram o conteúdo pela metodologia *Flipped Classroom* (Laboratório 01), enquanto que o questionário sobre o CodeinPY foi respondido por todos, pois, todos os alunos de ambos laboratório utilizaram o protótipo para responderem os Algoritmos no experimento "A", portanto todos podem avaliar a sua utilização.

Nos quadros 5.2 e 5.3 estão relacionadas as perguntas presentes nos dois formulários aplicados.

Quadro 5.2 - Questionário sobre Codecademy

Pergunta	Opções	Pergunta	Opções
Qual sua idade?	Campo aberto	Qual seu sexo?	Masculino/Feminino
Sobre a metodologia "Flipped Classroom". Como você avalia a experiência de ter acesso previamente ao conteúdo da aula através da ferramenta Codecademy?	1 - Péssimo; 2 - Ruim; 3 - Regular; 4 - Bom; 5 - Ótimo.	Como você avalia a utilização do Codecademy como ferramenta de aprendizagem em programação?	1 - Péssimo; 2 - Ruim; 3 - Regular; 4 - Bom; 5 - Ótimo.
Como você avalia sua experiência na utilização da ferramenta Codecademy?	1 - Péssimo; 2 - Ruim; 3 - Regular; 4 - Bom; 5 - Ótimo.	Os conhecimentos e prática experimentados utilizando a ferramenta Codecademy lhe ajudaram satisfatoriamente como uma boa base para continuação dos estudos de forma autônoma?	Sim; Não; Em parte.
O conteúdo visto na ferramenta Codecademy foi útil para seu aprendizado, ou seja, conseguiu aprender utilizando a ferramenta?	Sim; Não.	Encontrou dificuldades na utilização da ferramenta Codecademy?	Sim; Não; Em parte.
Na sala de aula, conseguiu sanar suas dúvidas sobre o conteúdo visto previamente na Codecademy?	Sim; Não.	Teve dificuldade em resolver as atividades propostas pelo professor apenas acessando o conteúdo pela Codecademy?	Sim; Não; Em parte.
O conteúdo visto na ferramenta Codecademy foi suficiente para resolver as atividades passadas em sala de aula?	Sim; Não; Em parte.	Deseja incluir alguma observação sobre a ferramenta Codecademy?	Campo aberto

Fonte: Elaborado pela Autora (2017)

Objetivo: Investigar se os alunos realmente aprenderam os conceitos de Programação de Computadores utilizando a ferramenta Codecademy seguindo a metodologia *Flipped Classroom*, buscando responder a um dos questionamentos desta pesquisa:

- Qual a efetividade da adoção de ambientes virtuais de aprendizagem como apoio à metodologia *Flipped Classroom* na composição das estratégias para ensino-aprendizagem dos conceitos de Programação de Computadores?

Quadro 5.3 - Questionário sobre CodeinPY

Pergunta	Opções	Pergunta	Opções
Qual sua idade?	Campo aberto	Qual seu sexo?	Masculino/Feminino
Sobre aprendizagem colaborativa. Como você avalia a sua experiência na utilização da ferramenta CodeinPY?	1 - Péssimo; 2 - Ruim; 3 - Regular; 4 - Bom; 5 - Ótimo.	Como avalia a experiência de codificar em tempo real de forma colaborativa?	1 - Péssimo; 2 - Ruim; 3 - Regular; 4 - Bom; 5 - Ótimo.
Como você avalia a utilização do CodeinPY como ferramenta de aprendizagem em programação?	1 - Péssimo; 2 - Ruim; 3 - Regular; 4 - Bom; 5 - Ótimo.	Conseguiu resolver as atividades de aula através da ferramenta CodeinPY?	Sim; Não.
A ferramenta CodeinPY foi útil para trocar conhecimento/dúvidas com outros alunos?	Sim; Não.	Conseguiu aprender ou ensinar algo para seu colega utilizando a ferramenta colaborativa CodeinPY?	Sim; Não.
Sentiu dificuldade em resolver atividades em colaboração com outros alunos através da ferramenta CodeinPY?	Sim; Não; Em parte.	Conseguiu ajudar/ser ajudado a resolver alguma atividade através da ferramenta CodeinPY?	Sim; Não.
Deseja incluir alguma observação sobre a ferramenta CodeinPY?	Campo aberto		

Fonte: Elaborado pela Autora (2017)

Objetivo: Investigar se os alunos conseguiram aprender juntos de forma colaborativa por meio do protótipo CodeinPY avaliando a interação entre os pares e sua aprendizagem, buscando responder a outro questionamento da pesquisa:

- Quais estratégias e inovações tecnológicas podem ser utilizadas para resolver problemas envolvendo a aprendizagem de Programação de Computadores?

5.6 Análise de Dados

Os dois questionários aplicados foram criados na plataforma Google Forms ⁴¹ e a captação dos dados e as informações apuradas serão apresentadas em forma de gráficos. Dois alunos deixaram de preencher aos questionários, logo, no questionário sobre Codecademy há apenas 18 respostas e sobre a CodeinPY apenas 38 respostas. Foram selecionadas três perguntas de cada formulário como mostra o Quadro 5.4 e os demais gráficos demonstram os dados coletados apresentados nas figuras 5.5 a 5.10.

⁴¹ <https://www.google.com/forms/about/>

Quadro 5.4 - Corte do Questionário sobre CodeinPY

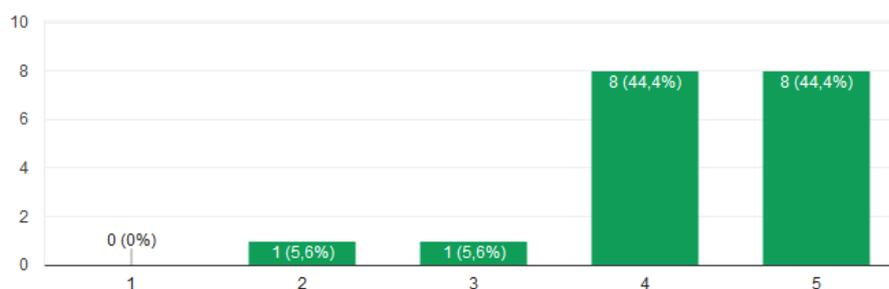
Perguntas Codecademy	Perguntas CodeinPY
Sobre a metodologia "Flipped Classroom". Como você avalia a experiência de ter acesso previamente ao conteúdo da aula através da ferramenta Codecademy?	Como você avalia a utilização do CodeinPY como ferramenta de aprendizagem em programação?
O conteúdo visto na ferramenta Codecademy foi útil para seu aprendizado, ou seja, conseguiu aprender utilizando a ferramenta?	Conseguiu aprender ou ensinar algo para seu colega utilizando a ferramenta colaborativa CodeinPY?
Deseja incluir alguma observação sobre a ferramenta Codecademy?	Deseja incluir alguma observação sobre a ferramenta CodeinPY?

Fonte: Elaborado pela Autora (2017)

Figura 5.5 - Avaliação da Experiência na Utilização da Ferramenta Codecademy

Sobre a metodologia "Flipped Classroom". Como você avalia a experiência de ter acesso previamente ao conteúdo da aula através da ferramenta Codecademy?

18 respostas



Fonte: Elaborada pela Autora (2017)

O que se pôde verificar na Figura 5.5 é uma avaliação positiva da ferramenta e metodologia, pois, 44,4% (8) dos alunos avaliaram a experiência como ótima, 44,4% (8) como boa, 5,6% (1) como regular e 5,6% (1) avaliou como péssimo, totalizando 100% (18) dos alunos participantes da aplicação deste questionário que viram o conteúdo pela metodologia *Flipped Classroom*.

Figura 5.6 - Avaliação da Efetividade da Ferramenta Codecademy para o Ensino dos Alunos

O conteúdo visto na ferramenta Codecademy foi útil para seu aprendizado, ou seja, conseguiu aprender utilizando a ferramenta?

18 respostas



Fonte: Elaborada pela Autora (2017)

Analisando o gráfico da Figura 5.6, verifica-se que a ferramenta foi efetiva em relação ao ensino dos conceitos de Programação de Computadores, pois 17 dos 18 alunos que responderam ao questionário informaram que a Codecademy foi útil para seu aprendizado. Apenas 1 aluno informou não ter conseguido aprender o conteúdo utilizando a ferramenta. Ou seja, 94,4% (17) avaliaram positivamente, enquanto 5,6% (1) avaliou negativamente.

Figura 5.7 - Observações sobre a Ferramenta Codecademy

Deseja incluir alguma observação sobre a ferramenta Codecademy?

11 respostas

Não
NÃO
não!
Eu realmente gostei muito!Eu estava preocupada se não ia conseguir aprender. Mas a ferramenta me ajudou a entender o assunto!Obrigado pela experiência!
não

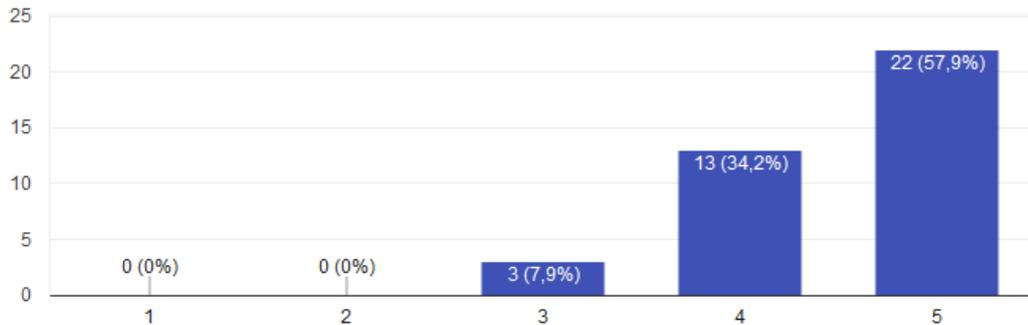
Fonte: Elaborada pela Autora (2017)

Essa questão não era obrigatória e permitia aos alunos deixarem sugestões ou críticas sobre a ferramenta Codecademy. 11 alunos deixaram sua avaliação e a Figura 5.7 expôs algumas das respostas apresentadas. As figuras 5.8 a 5.10 apresentam a coleta dos dados sobre o protótipo CodeinPY.

Figura 5.8 - Avaliação do Protótipo CodeinPY como Ferramenta de Aprendizagem

Como você avalia a utilização do CodeinPY como ferramenta de aprendizagem em programação?

38 respostas



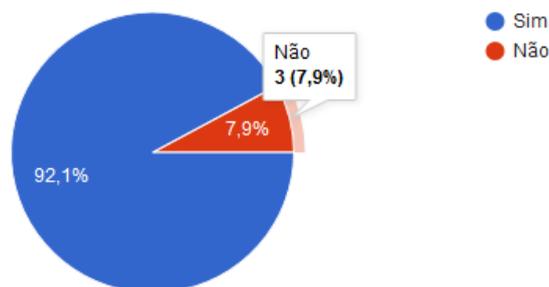
Fonte: Elaborada pela Autora (2017)

Analisando o gráfico da avaliação do protótipo CodeinPY na Figura 5.8, observa-se que a maioria, 57,9% (22) dos alunos o avaliaram como ótimo, 34,2% (13) como bom e apenas 7,9% (3) avaliaram a ferramenta como regular. O que é um resultado positivo, pois, apesar de ser um protótipo, foi muito bem avaliado, não recebendo conceitos negativos como ruim ou péssimo, o que o evidencia como uma boa opção de ferramenta para ensino/aprendizagem dos conceitos de Programação de Computadores.

Figura 5.9 - Avaliação da Efetividade do Protótipo CodeinPY em Relação ao Ensino/Aprendizagem

Conseguiu aprender ou ensinar algo para seu colega utilizando a ferramenta colaborativa CodeinPY?

38 respostas



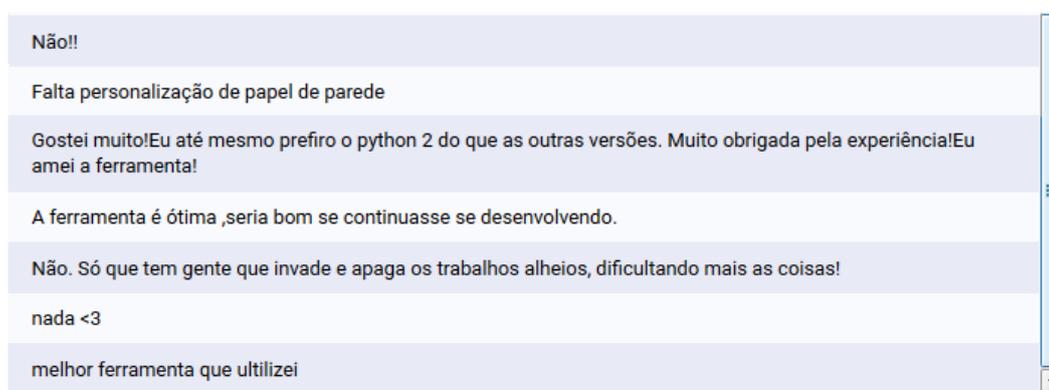
Fonte: Elaborada pela Autora (2017)

Em relação à efetividade do aprendizado dos alunos e à colaboração entre as partes utilizando o protótipo CodeinPY, foi constatado que 92,1% (35) dos alunos afirmaram ter aprendido ou ensinado algo por meio do protótipo, enquanto 7,9% (3) afirmam o contrário. Isto evidencia outro ponto positivo para a ferramenta, já que quase 100% (38) dos alunos afirmaram conseguir aprender o conteúdo juntos de forma colaborativa.

Figura 5.10 - Observações sobre o Protótipo CodeinPY

Deseja incluir alguma observação sobre a ferramenta CodeInPy?

21 respostas



Fonte: Elaborada pela Autora (2017)

Sobre as possíveis sugestões ou críticas em relação à utilização do protótipo, pôde-se perceber que os alunos gostaram da experiência em utilizá-lo e expuseram suas avaliações, em sua maioria positiva. Na Figura 5.10 foram selecionadas algumas dessas opiniões e se pode verificar o entusiasmo de alguns alunos em relação à ferramenta. Observa-se também recomendações para dar continuidade ao desenvolvimento da mesma e sugestões para melhoria do protótipo.

Na análise das respostas dos questionários avaliativos das ferramentas Codecademy e CodeinPY, para investigar a efetividade de ambas em relação ao ensino/aprendizagem dos conceitos de Programação de Computadores, pôde-se perceber durante as observações em laboratório e apuração dos dados coletados, um acréscimo na motivação e satisfação dos alunos no que se refere ao seu aprendizado.

No que diz respeito à análise dos experimentos realizados em laboratório, o experimento "A" avaliou se os alunos conseguiram se comunicar por meio do protótipo CodeinPY e resolverem o algoritmo proposto. A análise dos dados se deu pela observação das interações entre os alunos e a verificação se cada grupo conseguiu responder ao desafio juntos. O objetivo foi verificar se iriam

conseguir por meio do protótipo CodeinPY resolver ao Algoritmo desafio de forma colaborativa e foi visto que todos os grupos formados conseguiram resolver a atividade proposta.

A análise dos dados do Experimento "B" se deu por meio da coleta das médias das notas individuais de cada aluno na prova aplicada, e a partir daí, foi gerada a média geral dos alunos que viram todo o conteúdo pela metodologia *Flipped Classroom* e dos alunos que viram pela metodologia tradicional de ensino. O objetivo é verificar se houve uma diferenciação ou equivalência entre as notas apresentadas pelos alunos em cada metodologia de ensino aplicada. O Quadro 5.5 mostra as médias gerais dos alunos por metodologia de ensino, enquanto o Quadro 5.6 mostra a quantidade de alunos por intervalo de notas.

Quadro 5.5 - Média Geral das Notas dos Alunos

Média Geral	
<i>Flipped Classroom</i>	Tradicional
8,15	7,80

Fonte: Elaborado pela Autora (2017)

Onde:

- Nota = pontos experimento "A" + pontos experimento "B"
 - Nota máxima = 02 + 08 = 10 pontos
 - Média geral = $\frac{\sum \text{notas}}{\text{total de alunos por metodologia}}$

Quadro 5.6 - Quantidade de Alunos por Intervalo de Notas

Quantidade de Notas					
<i>Flipped Classroom</i>			Tradicional		
Notas = 10	10 > Notas >=7	Notas < 7	Notas = 10	10 > Notas >=7	Notas < 7
6	10	4	4	10	6

Fonte: Elaborado pela Autora (2017)

Observa-se no Quadro 5.5 que a média geral dos alunos que viram o conteúdo pela metodologia *Flipped Classroom* foi superior à média geral dos que viram pela metodologia

tradicional de ensino, já o Quadro 5.6 apresenta um quantitativo das notas em relação a determinados intervalos e verificou-se que houve uma diferença entre os intervalos inferior e superior com relação às duas metodologias, o que mostra um aumento em relação ao número de notas dez e um decréscimo em notas menores que sete para os alunos que utilizaram a metodologia *Flipped Classroom*, o que representa mais um ponto positivo e contribuição desta pesquisa.

Para que houvesse uma validação estatística dos dados, foi realizada a aplicação das fórmulas do desvio padrão e variância com a finalidade de verificar a homogeneidade das notas em relação à média verificada em cada metodologia. Os resultados obtidos são apresentados no Quadro 5.7, onde:.

- Fórmula para a variância amostral: $s^2 = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n - 1}$
- Fórmula para desvio padrão amostral: $s = \sqrt{s^2}$

Quadro 5.7 - Desvio Padrão e Variância

Metodologia	Notas	Variância	Desvio Padrão
<i>Flipped Classroom</i>	9.5;10;7;4;9;9;9.5;10;10;9;8;10; 10;4;4;6;8.5;8;7.5;10	4.45	2.110
Tradicional	10;9.5;4.5;10;6.5;3.5;7;9.5;9.5;1 0;4;10;7.5;9.5;9.5;9;8;5.5;8;5;	4.98	2.233

Fonte: Elaborado pela Autora (2017)

Analisando o Quadro 5.7 observou-se que a variância e o desvio padrão das notas dos alunos da metodologia *Flipped Classroom* foi menor do que os apresentados pelos alunos da metodologia tradicional, ou seja, as notas se mostraram mais uniformes em relação à média geral apresentada por cada metodologia, não apresentando grandes discrepâncias entre elas e a média.

Além disso, também foi aplicado o método T de *Student*, bastante utilizado para testes de hipótese para avaliar se há diferença em relação à significância entre as médias de duas amostras.

Para tal, esse método foi aplicado utilizando-se o programa R⁴² que é *Software* livre e muito utilizado para fins estatísticos e análises de dados.

A princípio, para a realização deste teste, deve-se definir quais as suposições a serem feitas acerca dos dados, logo, determina-se as hipóteses h_0 e h_1 , onde h_0 seria a hipótese nula, ou seja, a

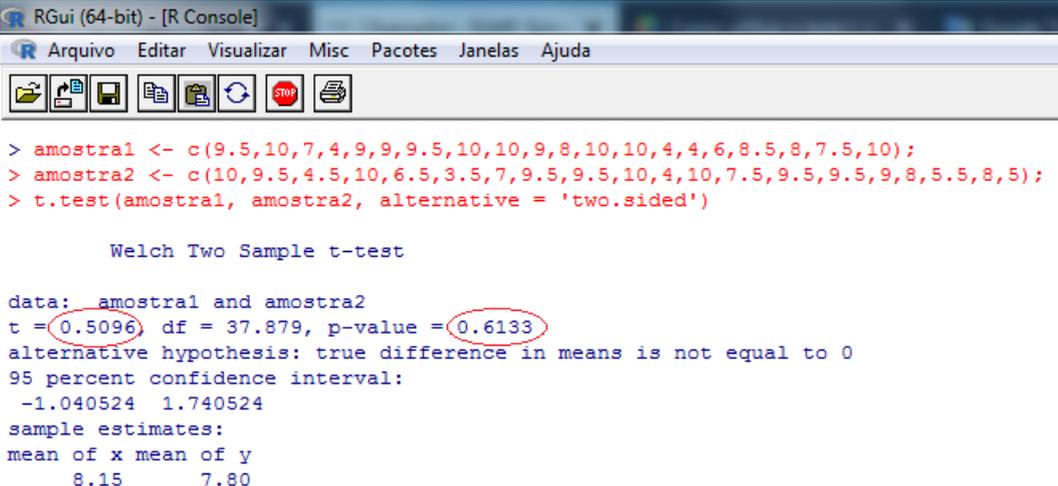
42 <https://www.r-project.org/>

média sendo a mesma para as duas amostras e h_1 a hipótese alternativa, as médias seriam diferentes e determinar o intervalo de confiança. Nesse estudo, foi aplicado no *Software R* com o teste T de *Student* para o intervalo de confiança de 95% como mostra a Figura 5.11.

Portanto, as hipóteses são:

- **h_0** - Não existe diferença entre as médias dos alunos utilizando a metodologia *Flipped Classroom* em relação a metodologia tradicional de ensino;
- **h_1** - Existe diferença entre as médias dos alunos utilizando a metodologia *Flipped Classroom* em relação a metodologia tradicional de ensino.

Figura 5.11 - Aplicação do Método T de *Student* no Programa R



```

RGui (64-bit) - [R Console]
Arquivo Editar Visualizar Misc Pacotes Janelas Ajuda

> amostra1 <- c(9.5,10,7,4,9,9,9.5,10,10,9,8,10,10,4,4,6,8.5,8,7.5,10);
> amostra2 <- c(10,9.5,4.5,10,6.5,3.5,7,9.5,9.5,10,4,10,7.5,9.5,9.5,9,8,5.5,8,5);
> t.test(amostra1, amostra2, alternative = 'two.sided')

Welch Two Sample t-test

data: amostra1 and amostra2
t = 0.5096, df = 37.879, p-value = 0.6133
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.040524  1.740524
sample estimates:
mean of x mean of y
 8.15      7.80

```

Fonte: Elaborado pela Autora (2017)

Um dos resultados desse teste é o p-valor (significância estatística) e um p-valor superior a 0,05 significa que há uma diferença não significativa, logo, o p-valor encontrado no teste foi de 0,6133 maior que 0,05. Nesse caso, aceita-se a hipótese h_0 , ou seja, apesar da média aritmética das notas dos alunos utilizando a metodologia *Flipped Classroom* ter sido maior que a dos alunos utilizando a metodologia tradicional, não há uma diferença significativa entre as médias de cada metodologia.

O ideal para se ter resultados mais concretos, seria a realização desse mesmo estudo em mais turmas para que se obtivesse dados mais consistentes, mesmo assim, o que se evidencia

através dessa análise é que a utilização da sala de aula invertida foi tão efetiva quanto a metodologia tradicional há bastante tempo utilizada.

O que também se evidenciou foi a metodologia *Flipped Classroom* em ação conjunta com as ferramentas propostas, Codecademy e CodeinPY apresentando resultados positivos como estratégia de aprendizagem, resultando em um desempenho equivalente ou até mesmo superior ao da metodologia tradicional de ensino.

Em relação à professora participante, foi aplicado um questionário a fim de avaliar o seu ponto de vista sobre a utilização das metodologias ativas *Flipped Classroom*, *Gamification* e aprendizagem colaborativa aplicadas como estratégias para o ensino/aprendizagem dos conceitos de Programação de Computadores mediado por ambientes virtuais de ensino.

Buscou-se verificar se as ferramentas apresentadas foram úteis para o processo de aprendizagem dos alunos e se a sua utilização impactou positivamente nos alunos. As perguntas realizadas são apresentadas no Quadro 5.8.

Quadro 5.8 - Questionário para Avaliação Docente

Pergunta	Opções	Pergunta	Opções
1. Como você avalia a experiência da utilização em conjunto do Flipped Classroom+Gamification+Aprendizagem colaborativa como estratégias para o ensino/aprendizagem de Programação de Computadores?	1 - Péssimo; 2 - Ruim; 3 - Regular; 4 - Bom; 5 - Ótimo.	2. Utilizaria em outras turmas/disciplinas as estratégias apresentadas?	Sim; Não; Em parte.
3. Após a junção da turma, percebeu alguma diferença entre os alunos que tiveram aula pela metodologia tradicional em relação aos que viram pela sala de aula invertida?	Sim; Não; Em parte.	4. Caso a resposta seja sim na pergunta anterior, o que percebeu?	Campo aberto.
5. Conversou com os alunos sobre a experiência? Acredita que os alunos avaliaram positivamente ou negativamente as ferramentas e estratégias utilizadas?	Avaliaram Positivamente; Avaliaram Negativamente; Não conversei sobre o assunto.	6. Após os experimentos realizados em laboratório, percebeu interesse dos alunos em continuarem utilizando as ferramentas apresentadas?	Sim; Não; Em parte.
7. Percebeu os alunos mais ativos/participativos em relação ao seu auto aprendizado?	Sim; Não; Em parte.	8. Acredita que os alunos que viram o conteúdo pela metodologia Flipped Classroom utilizando as ferramentas Codecademy e CodeinPY realmente aprenderam o conteúdo?	Sim; Não; Em parte.
9. Como avalia a ferramenta Codecademy para acompanhamento do aprendizado dos alunos?	1 - Péssimo; 2 - Ruim; 3 - Regular; 4 - Bom; 5 - Ótimo.	10. Como avalia o protótipo CodeinPY como ferramenta para Aprendizagem Colaborativa dos conceitos de Programação de Computadores?	1 - Péssimo; 2 - Ruim; 3 - Regular; 4 - Bom; 5 - Ótimo.
11. Continuaria a utilizar as ferramentas Codecademy/CodeinPY em próximas experiências?	Sim; Não; Em parte.	12. Gostaria de informar sugestões/críticas ao protótipo CodeinPY ou a ferramenta Codecademy?	Campo aberto.

Fonte: Elaborado pela Autora (2017)

O que se pôde verificar pelas respostas do questionário foi uma avaliação positiva da professora em relação à experiência realizada. Sobre a percepção de alguma diferença entre o comportamento dos alunos que viram as aulas pela metodologia *Flipped Classroom* dos que viram pela metodologia tradicional de ensino, a mesma informou que os alunos se divertiram mais com a sala de aula invertida o que a torna um bom mecanismo para despertar o interesse dos alunos.

A professora declarou ainda que continuaria a utilizar as estratégias aplicadas neste estudo

em outras turmas e que percebeu interesse dos alunos em continuar utilizando as ferramentas apresentadas, o que representa uma maior motivação dos alunos em relação ao seu aprendizado. As respostas do questionário encontram-se no Apêndice G.

5.7 Considerações Finais do Capítulo

Este capítulo relatou os experimentos realizados para avaliar a utilização da ferramenta Codecademy e do protótipo desenvolvido CodeinPY como estratégias para ensino/aprendizagem dos conceitos de Programação de Computadores, analisando sua efetividade em relação ao aprendizado dos alunos.

Foi descrito como foram realizados os experimentos, os seus participantes e procedimentos, cronograma das etapas executadas, além das ferramentas empregadas.

Foram detalhadas as etapas para a coleta dos dados e os métodos aplicados para tal, como a aplicação de questionários, observação sistemática e avaliação por notas. Foi exposta a análise dos dados obtidos por meio de gráficos e quadros.

A análise dos questionários de avaliação das ferramentas propostas pela pesquisa verificou sua efetividade em relação ao ensino/aprendizado de Programação de Computadores, além da motivação dos alunos no que se refere a sua própria aprendizagem.

Analisou-se os experimentos realizados examinando as médias gerais das notas dos alunos em cada metodologia, investigando qual o desempenho de cada uma delas, a fim de mensurar se as metodologias ativas *Flipped Classroom* + Aprendizagem Colaborativa + *Gamification* mediado por ambientes virtuais de aprendizagem foi tão efetivo quanto a metodologia tradicional de ensino. Como contribuição positiva deste trabalho, também pode ser destacada a apresentação de novas formas de ensinar/aprender Programação de Computadores por meio da aplicação em conjunto das metodologias ativas e ferramentas propostas pela pesquisa.

O próximo capítulo relata as principais contribuições desta pesquisa, as limitações encontradas durante o seu processo de desenvolvimento e sugestões para possíveis trabalhos futuros buscando inovações tecnológicas para o ensino/aprendizagem dos conceitos de Programação de Computadores.

6 CONSIDERAÇÕES FINAIS

Este trabalho promoveu a união dos conceitos de *Flipped Classroom*, Aprendizagem Colaborativa e *Gamification* aplicados em um ambiente colaborativo para ensino/aprendizagem de Programação de Computadores. Este capítulo tem por finalidade apresentar as considerações finais sobre o referido trabalho, destacando suas principais contribuições, as limitações encontradas no decorrer da pesquisa, além de sugestões para trabalhos futuros.

6.1 Principais Contribuições

Uma das principais contribuições deste trabalho foi a busca por ferramentas/ambientes virtuais de ensino que poderiam ser utilizadas como apoio à metodologia *Flipped Classroom*, as quais possuísem como diferencial a aplicação dos conceitos de *Gamification* e Aprendizagem Colaborativa.

A metodologia *Flipped Classroom* não define ferramentas para a aplicação dos seus conceitos, logo, o levantamento desses ambientes virtuais que permitam a inversão da sala de aula e uma aprendizagem ativa dos alunos é de relevante contribuição tanto para o professor que poderá utilizá-los para melhoria de suas práticas docentes quanto para os alunos que poderão ter acesso ao conteúdo da aula previamente a qualquer momento, se tornando responsável pelo seu próprio aprendizado.

Outra importante contribuição desta pesquisa foi o desenvolvimento de um protótipo para aprendizagem colaborativa com a finalidade de aumentar o envolvimento dos alunos no processo de aprendizagem, possibilitando uma melhor experiência na interação e comunicação entre eles, permitindo que escrevam seus códigos de forma compartilhada em tempo real e aprendam o conteúdo juntos, colaborando entre si a qualquer hora e em qualquer lugar, apoiando a metodologia *Flipped Classroom*. O seu código-fonte⁴³ encontra-se disponível e está acessível para que todos possam baixá-lo e personalizá-lo conforme o interesse. Este trabalho não somente levantou ferramentas para apoio a metodologias *Flipped Classroom*, mas também realizou a análise da efetividade da utilização delas em relação ao aprendizado dos alunos dos conceitos de Programação de Computadores. Para tanto foram aplicados questionários e realizados experimentos para

43 <https://github.com/gisapslee/codeinpy>

implementar esta investigação, o que foi retratado no capítulo anterior.

O que se pôde verificar foi uma aceitação dos alunos em relação às metodologias aplicadas e às ferramentas utilizadas. Os alunos se mostraram mais motivados em relação ao seu aprendizado, pois, durante as observações em laboratório, sempre quando surgiam perguntas ou dúvidas, os próprios alunos buscavam colaborar e responder as dúvidas de outros alunos, ou seja, se mostraram mais ativos em relação ao seu papel no processo de aprendizagem.

Em relação à primeira nota da disciplina, constatou-se que a média geral dos alunos que tiveram aula pelas metodologias *Flipped Classroom* + *Gamification* + aprendizagem colaborativa foi levemente superior à média dos alunos que assistiram pela metodologia tradicional de ensino, o que revela que as metodologias aplicadas em conjunto foram efetivas no que trata o ensino/aprendizagem de Programação de Computadores.

Por fim, conclui-se que o objetivo estabelecido no princípio do desenvolvimento do trabalho - analisar a efetividade da adoção da metodologia *Flipped Classroom* em conjunto com outras metodologias ativas no processo de ensino-aprendizagem de Programação de Computadores mediado por ambientes virtuais de aprendizagem - foi cumprido.

Para isso, os objetivos específicos definidos para alcançar esse objetivo geral também foram alcançados:

- a) Pesquisar e comparar ferramentas (ambientes virtuais de aprendizagem) que possam ser utilizadas como apoio à metodologia da "Sala de aula invertida" - **Ferramenta Codecademy**;
- b) Desenvolver um protótipo de ferramenta para aplicação da aprendizagem colaborativa - **Protótipo CodeinPy**; e
- c) Avaliar as ferramentas por meio de experimentos e identificar as limitações e dificuldades encontradas - **Observação em laboratório, aplicação de questionários, experimentos com as ferramentas**.

A seção a seguir apresenta as limitações encontradas no desenvolvimento deste trabalho.

6.2 Limitações

Durante o desenvolvimento deste trabalho algumas dificuldades surgiram. Uma delas foi encontrar uma turma com o perfil necessário para realização desta pesquisa: alunos com total inexperiência nos conceitos de Algoritmos e Linguagens de Programação. Outro ponto foi a necessidade de uma

turma controle que houvesse visto o mesmo conteúdo disponibilizado na ferramenta Codecademy via metodologia tradicional de ensino a fim de se comparar o desempenho dos alunos em relação a cada metodologia aplicada.

Essas dificuldades foram superadas optando-se pela divisão da turma selecionada em duas para que se pudesse ter o parâmetro do desempenho dos alunos que viram o conteúdo pela metodologia *Flipped Classroom* e dos alunos que viram o conteúdo pela metodologia tradicional de ensino. Apesar dos problemas citados, não chegaram a causar uma limitação à realização da pesquisa.

Uma das principais limitações deste trabalho encontra-se no ambiente utilizado para a realização dos testes do protótipo, já que sem acesso à Internet não é possível utilizar as ferramentas apresentadas. Outro ponto foi que não foram realizadas simulações prévias em relação à quantidade de acessos simultâneos. Logo, essa avaliação ocorreu durante a execução do trabalho e utilização do protótipo.

É importante ressaltar que os testes realizados não podem ser generalizados a uma população distinta da aqui apresentada, já que não há como prever o que ocorreria se uma quantidade maior de acessos simultâneos ocorresse.

Durante as observações, por algumas vezes o protótipo parou de funcionar, porém, não foi possível detectar a real causa do problema: se por excesso de acessos simultâneos ao protótipo ou por problemas no servidor de teste. A solução temporária foi reiniciar o processo da aplicação sempre quando o problema ocorria. Porém, isso resolve a situação temporariamente, já que a real causa merece ser investigada e o problema totalmente sanado.

Portanto, apesar da existência de limitações, os testes puderam ser executados utilizando-se o protótipo desenvolvido, sendo possível realizar uma avaliação específica conforme o planejado.

Na seção seguinte, apresentam-se as sugestões para trabalhos futuros.

6.3 Trabalhos Futuros

Como proposta para trabalhos futuros, recomenda-se dar continuidade ao desenvolvimento do protótipo CodeinPY aperfeiçoando-o e adicionando novas funcionalidades, levando em consideração as sugestões coletadas por meio dos questionários aplicados. Uma delas seria simular a utilização do protótipo em uma maior quantidade de acessos simultâneos e encontrar uma solução

definitiva para o problema das "quedas" do protótipo.

Outras funcionalidades, como a inclusão de transmissão por *streaming* de vídeo ou aceitação de pessoas específicas para acesso a determinadas salas virtuais, podem vir a potencializar a comunicação entre os usuários e evitar que usuários intrusos alterem o código sem permissão, respectivamente, além de outras possibilidades.

Salvar os códigos escritos durante a aula e armazená-los para serem utilizados posteriormente se faz interessante para que os alunos retomem sua codificação de onde terminaram para concluírem o seu raciocínio. Outras funcionalidades, por exemplo, presentes na plataforma Slack⁴⁴ como compartilhamento de arquivos, imagens, documentos, entre outros, incluídas no protótipo aumentaria o leque de possibilidades para uma maior interação entre os alunos e professor presentes na sala virtual criada.

Outra recomendação é investigar a utilização dos conceitos propostos, *Flipped Classroom*, Aprendizagem Colaborativa e *Gamification*, integrados a outras estratégias de ensino como PBL ou *Peer Instruction*, por exemplo, e avaliar a efetividade destas em conjunto com as metodologias apresentadas neste estudo.

Realizar esse mesmo estudo com outras turmas seria recomendável para a realização de uma pesquisa quantitativa buscando implementar uma análise estatística dos dados coletados a fim de se obter uma comparação efetiva entre as metodologias tradicionais e as apresentadas neste estudo.

Por fim, sugere-se pesquisar novas maneiras de ensinar os conceitos de Programação de Computadores apresentando inovações tecnológicas que auxiliem o professor a melhor compreender e orientar seus alunos em busca de aumentar o envolvimento destes no processo de aprendizagem.

44 <https://slack.com/> - plataforma de comunicação que permite compartilhamento de documentos entre outras mídias.

REFERÊNCIAS

ARBELAITZ, Olatz; MARTÍN, José I.; MUGUERZA; Javier. **Analysis of Introducing Active Learning Methodologies in a Basic Computer Architecture Course.** IEEE Transactions on Education, Vol. 58, p.110-116, 2015.

AUDSLEY, Samantha; FERNANDO, Kalyani; MAXSON, Bronwen. **An Examination of Coursera as an Information Environment: Does Coursera Fulfill its Mission to Provide Open Education to All?.** The Serials Librarian, Vol. 65, p.136–166, 2013.

BERGMANN, Jonathan; SAMS, Aaron. **Sala de Aula Invertida: Uma metodologia ativa de aprendizagem.** 1. ed, Rio de Janeiro: LTC, 2016.

CÁMARA, Luis Miguel Serrano;VELASCO, Maximiliano Paredes; ALCOVER, Carlos-María; ITURBIDE, J. Ángel Velazquez. **An evaluation of students' motivation in computer-supported collaborative learning of programming concepts.** Journal Computers in Human Behavior, Vol.31, p.499-508, 2014.

COLOMBO, Cristiano da S.;STAHL, Nilson S. P; DUNCAN, Camila P. F. R. ; SCHROETTER, Sandra M. (2014): **O modelo sala de aula invertido aplicado em uma disciplina do curso técnico em Informática.** Disponível em: <<http://ocs.ifes.edu.br/index.php/sepinfo/ENRETESI/paper/viewFile/1442/426>>. Acesso: 03 Set. 2016.

ELGAMAL, A. F.; ABAS, H.A.;BALADOH, E-S.M. **An interactive e-learning system for improving web programming skills.** Journal Education and Information Technologies, Vol.18, n. 1, p.29-46, 2013.

ESTÁCIO, Bernardo José da Silva; PRIKLADNICKI, Rafael. **Distributed Pair Programming: A Systematic Literature Review.** Information and Software Technology, Vol. 63, p.1-10, 2015.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa.** 4. ed. - São Paulo : Atlas, 2002.

ÍBAÑES, María-Blanca; DI-SERIO, Ángela; DELGADO-KLOOS, Carlos. **Gamification for Engaging Computer Science Students in Learning Activities: A Case Study.** IEEE Transactions on Learning Technologies, Vol.7, n. 3, p.291-301, 2014.

JONES, Lucas Herbert. (2016): **Aprendizado Invertido em Compiladores**. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/147664/000999687.pdf?sequence=1>>. Acesso: 05 Set. 2016.

JR, Dilermando Piva; CORTELAZZO, Angelo Luis. (2015): **Sala de aula invertida, ambientes de aprendizagem e educação online: a junção de três métodos para potencialização do ensino de algoritmos**. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/6291/4400>>. Acesso: 05 Set. 2016.

KIM, Ada S; KO, Andrew J. **A Pedagogical Analysis of Online Coding Tutorials**. In: ACM SIGCSE Technical Symposium on Computer Science Education, 2017, New York, p. 321 a 326.

LAW, Kris M. Y; LEE, Victor C. S.; YU, Y. T. **Learning motivation in e-learning facilitated computer programming courses**. Journal Computers & Education, Vol.55, n. 1, p.218-228, 2010.

LEE, Michael J.;KO. Andrew J. **Comparing the Effectiveness of Online Learning Approaches on CS1 Learning Outcomes**. In: ACM ICER '15, 2015, Nebraska, p. 237-246.

MENEZES, Nilo Ney Coutinho. **Introdução a programação com Python: Algoritmos e lógica de programação para iniciantes**. 2. ed, São Paulo: Novatec, 2014.

MENSING, Karen; MAK, Janice, BIRD, Megan; BILLINGS, Jeff. **Computational, Model Thinking and Computer Coding for U.S. Common Core Standards with 6 to 12 Year Old Students**. In: IEEE International Conference on Emerging eLearning Technologies and Applications ICETA 2013 11th, 2013, Slovakia, p. 17-22.

MOURA, João Cesar Mota; TEIXEIRA, Katiúscia Costa Barros. **Active Methodologies for Teaching Linear Algebra in an Engineering Course**. IEEE LATIN AMERICA TRANSACTIONS, Vol. 14, n. 2, p.837-843, 2016

OLIVEIRA, Allison Lopes de; GONÇALVES, Willian Antonio; HOED, Raphael Magalhães. (2014): **Arduino: Uma Proposta para o ensino introdutório de Programação**. Disponível em: <<http://www.abenge.org.br/cobenge-2014/Artigos/129086.pdf>>. Acesso: 01 Set. 2016.

OLSSON, Marie; MOZELIUS, Peter; COLLIN, Jonas. **Visualisation and Gamification of e-Learning and Programming Education.** *Electronic Journal of e-Learning*, Vol.13, n. 6, p. 441 - 454, 2015.

PANWONG, Patcharaporn; KEMAVUTHANON, Kemachart. **Problem-Based Learning Framework for Junior Software Developer: Empirical Study for Computer Programming Students.** *Wireless Personal Communications*, Vol.76, n. 3, p.603-613, 2014.

PEREIRA, Caio Ribeiro. **Aplicações web real-time com Node.js.** 1. ed. - São Paulo: Casa do Código, 2014.

RODRIGUES, Sandra. (2015): **Da Flipped Classroom à Flipped Learning.** Disponível em: <<http://www.hoper.com.br/single-post/2015/03/23/DA-FLIPPED-CLASSROOM-%C3%80-FLIPPED-LEARNING>>. Acesso: 05 Set. 2016.

SILVA, Edna L.;MENEZES, Estera M. (2005): **Metodologia da Pesquisa e Elaboração de Dissertação.** Disponível em: https://projetos.inf.ufsc.br/arquivos/Metodologia_de_pesquisa_e_elaboracao_de_teses_e_dissertacoes_4ed.pdf>. Acesso: 10 Set. 2016.

SOHRABI, Babak; IRAJ, Hamideh. **Implementing flipped classroom using digital media: A comparison of two demographically different groups perceptions.** *Computers in human behavior*, Vol. 60, p.514 -524, 2016.

TORRES, Patrícia Lupion; IRALA, Esrom Adriano F. (2014): **Aprendizagem Colaborativa: Teria e Prática.** Disponível em: <http://www.agrinho.com.br/site/wp-content/uploads/2014/09/2_03_Aprendizagem-colaborativa.pdf>. Acesso: 06 Jan. 2018.

VALENTE, José Armando. (2014): **Aprendizagem Ativa no Ensino Superior: a proposta da sala de aula invertida.** Depto. de Multimeios, Nied e GGTE - Unicamp & Ced - PucSP. São Paulo. Disponível em: <http://www.pucsp.br/sites/default/files/img/aci/27-8_aguardar_proec_textopara280814.pdf> Acesso: 05 Set. 2016.

WAZLAWICK, Raul Sidnei. **Metodologia de Pesquisa para Ciência da Computação.** 2. ed. Rio de Janeiro: Elsevier, 2014.

ZACHARIS, Nick Z. **Measuring the Effects of Virtual Pair Programming in an Introductory Programming Java Course.** IEEE TRANSACTIONS ON EDUCATION, Vol. 54, n.1, p.168-170,2011.

ZELLE, John M. Prefácio. In: ZELLE, John M. **Python Programming: An introduction to Computer Science.** U.S.A: Franklin, Beedle & Associates, Incorporated, 2004.

ZHONG,Baichang;WANG,Qiyun;CHEN, Jie. **The impact of social factors on pair programming in a primary school.** Computers in Human Behavior, Vol. 64, p.423-431, 2016.

APÊNDICE A - Quadro com Nota dos Alunos Avaliados

Quadro A1 - Notas dos alunos por metodologia aplicada

Nº na chamada	Nota	Metodologia	Nº na chamada	Nota	Metodologia
1	9,50	<i>Flipped Classroom</i>	21	10,00	Tradicional
2	10,00	<i>Flipped Classroom</i>	22	9,50	Tradicional
3	7,00	<i>Flipped Classroom</i>	23	4,50	Tradicional
4	4,00	<i>Flipped Classroom</i>	24	10,00	Tradicional
5	9,00	<i>Flipped Classroom</i>	25	6,50	Tradicional
6	9,00	<i>Flipped Classroom</i>	26	3,50	Tradicional
7	9,50	<i>Flipped Classroom</i>	27	7,00	Tradicional
8	10,00	<i>Flipped Classroom</i>	28	9,50	Tradicional
9	10,00	<i>Flipped Classroom</i>	29	9,50	Tradicional
10	9,00	<i>Flipped Classroom</i>	30	10,00	Tradicional
11	8,00	<i>Flipped Classroom</i>	31	4,00	Tradicional
12	10,00	<i>Flipped Classroom</i>	32	10,00	Tradicional
13	10,00	<i>Flipped Classroom</i>	33	7,50	Tradicional
14	4,00	<i>Flipped Classroom</i>	34	9,50	Tradicional
15	4,00	<i>Flipped Classroom</i>	35	9,50	Tradicional
16	6,00	<i>Flipped Classroom</i>	36	9,00	Tradicional
17	8,50	<i>Flipped Classroom</i>	37	8,00	Tradicional
18	8,00	<i>Flipped Classroom</i>	38	5,50	Tradicional
19	7,50	<i>Flipped Classroom</i>	39	8,00	Tradicional
20	10,00	<i>Flipped Classroom</i>	40	5,00	Tradicional
Média	8,15		Média	7,80	

Fonte: Elaborado pela Autora (2017)

APÊNDICE B - Avaliação Aplicada - Experimento "B"

Disciplina: Algoritmos e Programação

Nome: _____ No. _____

PROVA No. 1

1. Qual o valor final das variáveis A e B, após a execução das instruções abaixo? (2.0 pts)

A = 2

B = A * 5

A = B + 2

B = A - 5

Valor A: _____

Valor B: _____

2. Mostre o que será impresso pelo seguinte algoritmo: (2.0 pts)

B=28

A= (B*2)

C=A-B

print(A)

print(C)

print("C")

Resposta: _____

3. Faça um algoritmo no Python que leia o nome e a idade de uma pessoa: (2.0 pts)

- Imprima o nome da pessoa

- se a pessoa tem 18 anos, imprima "A pessoa tem 18 anos!!!";

- se a pessoa tem mais de 18 anos, imprima "A pessoa tem mais de 18 anos!!!";

- se a pessoa tem menos de 18 anos, imprima "A pessoa tem menos de 18 anos!!!";

4. A loja Mamão com Açúcar está vendendo seus produtos em 5 (cinco) prestações sem juros. Faça um algoritmo que receba um valor de uma compra e mostre o valor das prestações. (2.0 pts)

APÊNDICE C - Algoritmos Utilizados - Experimento "A"

Grupo 01: Escreva um algoritmo que leia uma temperatura em Fahrenheit e imprima essa temperatura em Celsius ($^{\circ}\text{C} = (^{\circ}\text{F}-32)/1.8$). Caso a nova temperatura em Celsius seja menor ou igual a 17°C exiba "Que dia frio!" caso contrário exiba "Dia normal!".

Grupo 02: Escreva um algoritmo que peça ao usuário o ano atual e o seu ano de nascimento e calcule a sua idade. Caso a idade seja menor que 18 exiba "Menor de idade" caso contrário "Maior de idade".

Grupo 03: Escreva um algoritmo que leia um número e exiba o dia correspondente da semana. (1- Domingo, 2- Segunda, 3- Terça, 4 - Quarta, 5- Quinta, 6- Sexta), se digitar outro valor imprimir "valor inválido".

Grupo 04: Escreva um algoritmo que leia a idade de um carro e exiba se o carro é velho ou novo. Se a idade for menor ou igual a 3 anos imprima "Seu carro é novo" caso contrário "Seu carro é velho".

Grupo 05: Escreva um algoritmo que leia uma palavra, veja a quantidade de caracteres dessa palavra e verifique se a quantidade é par ou ímpar, se for PAR exiba "Quantidade de caracteres Par" caso contrário "Quantidade de caracteres Ímpar".

Grupo 06: Escreva um algoritmo que leia a quantidade de horas trabalhadas e o nível do professor, calcule e exiba o seu salário com base nos valores da tabela a seguir:

- Caso nível do professor igual a "1" (salário = quantidade de horas * 12);
- Caso nível do professor igual a "2" (salário = quantidade de horas * 15);
- Caso nível do professor igual a "3" (salário = quantidade de horas * 25);

Grupo 07: Escreva um algoritmo que pergunte a velocidade do carro de um usuário. Caso ultrapasse 80 km/h, exiba uma mensagem dizendo que o usuário foi multado. Nesse caso, exiba o valor da multa, cobrando R\$ 5 por km acima de 80 km/h.

Grupo 08: Escreva um algoritmo que leia a idade de uma pessoa e imprima a sua classe eleitoral conforme tabela a seguir:

Idade < 16 anos	Não eleitor
Idade \geq 18 anos e < 65 anos	Voto obrigatório
Idade > 65 anos	Eleitor facultativo

Grupo 09: Faça um algoritmo que leia os valores A, B, C e imprima na tela se a soma de $A + B$ é menor C.

Grupo 10: Faça um algoritmo que leia dois valores inteiros A e B se os valores forem iguais deverá se somar os dois, caso contrário multiplique A por B. Atribua esses resultados a uma variável C e exiba o seu valor.

APÊNDICE D - Figura com Resposta de Simulação - Experimento "A"

AVALIAÇÃO TESTE - GRUPO 05

2ip

Para responder a questão a seguir utilize o endereço: <http://200.137.171.4:1337/LbG05>

1 - Escreva um algoritmo que leia uma palavra e verifique se ela inicia com uma vogal e imprima "Inicia com uma vogal" senão imprima "Não é uma vogal".

```

letra = raw_input('Escreva a palavra:')
if letra[0].upper() == "A" or letra[0].upper() == "E" or letra[0].upper() == "I" or letra[0].upper() == "O" or letra[0].upper()

```

1- Você conseguiu falar com seu colega de grupo da outra sala?

Sim

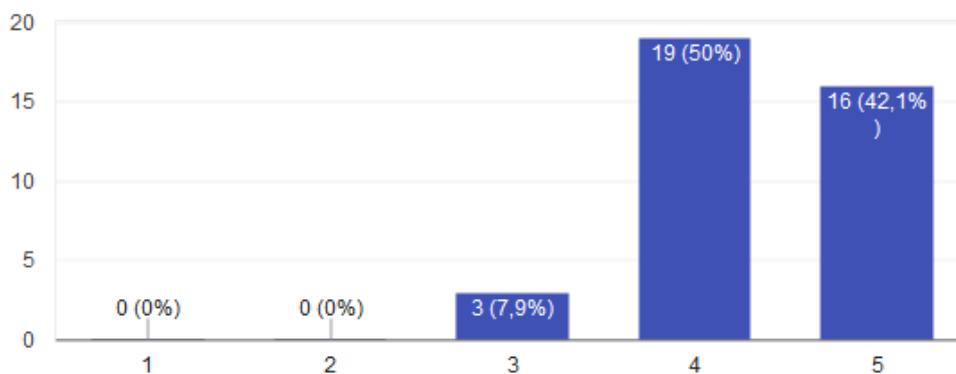
2- A comunicação com o seu colega através do chat do Moodle ajudou na resolução do problema? Concretiza!

APÊNDICE E - Gráficos com Dados Coletados - Questionário CodeinPY

As figuras a seguir apresentam os gráficos com a coleta dos dados dos questionários aplicados com os alunos para avaliar a experiência de utilização do protótipo CodeinPY.

3. Sobre aprendizagem colaborativa. Como você avalia a sua experiência na utilização da ferramenta CodeinPY?

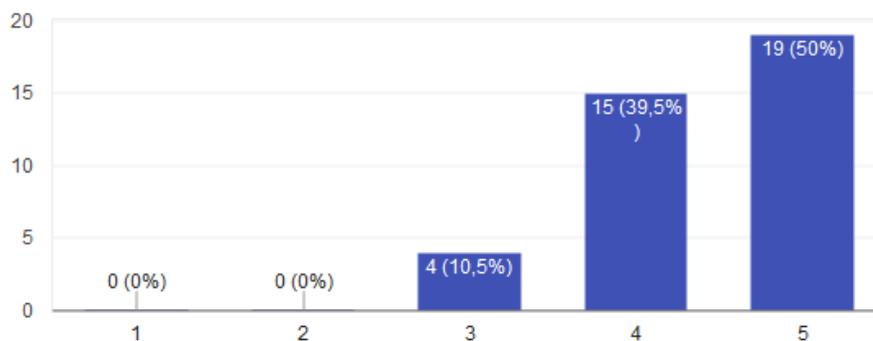
38 respostas



Fonte: Elaborado pela Autora (2017)

4. Como avalia a experiência de codificar em tempo real de forma colaborativa?

38 respostas



Fonte: Elaborado pela Autora (2017)

6. Conseguiu resolver as atividades de aula através da ferramenta CodeinPY?

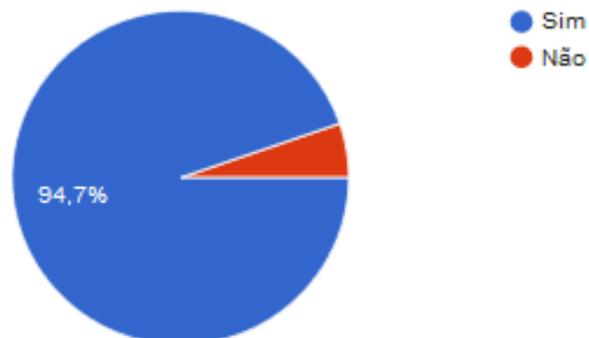
38 respostas



Fonte: Elaborado pela Autora (2017)

7. A ferramenta CodeinPy foi útil para trocar conhecimento/dúvidas com outros alunos?

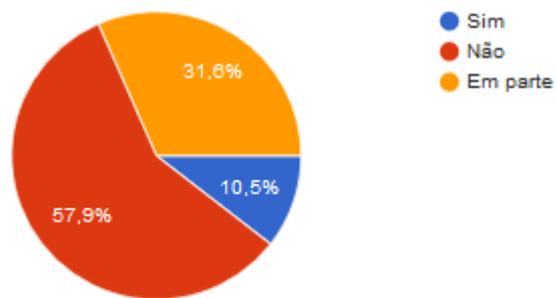
38 respostas



Fonte: Elaborado pela Autora (2017)

9. Sentiu dificuldade em resolver atividades em colaboração com outros alunos através da ferramenta CodeinPY?

38 respostas



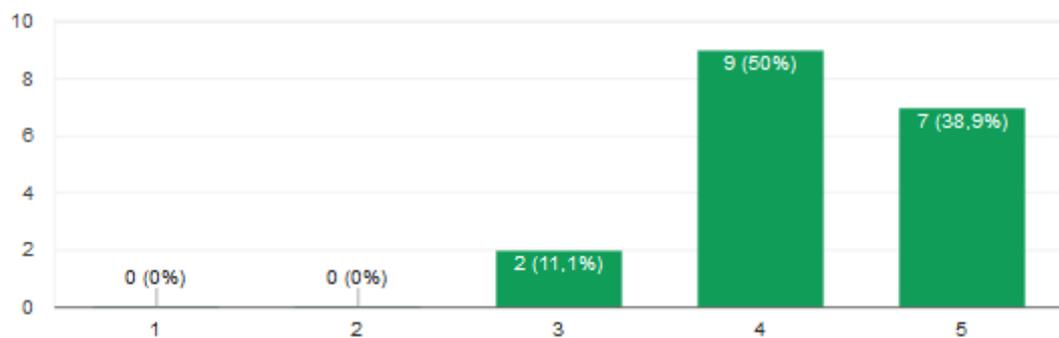
Fonte: Elaborado pela Autora (2017)

APÊNDICE F - Gráficos com Dados Coletados - Questionário Codecademy

As figuras a seguir apresentam os gráficos com a coleta dos dados dos questionários aplicados com os alunos para avaliar a experiência de utilização da ferramenta Codecademy.

4. Como você avalia a utilização do Codecademy como ferramenta de aprendizagem em programação?

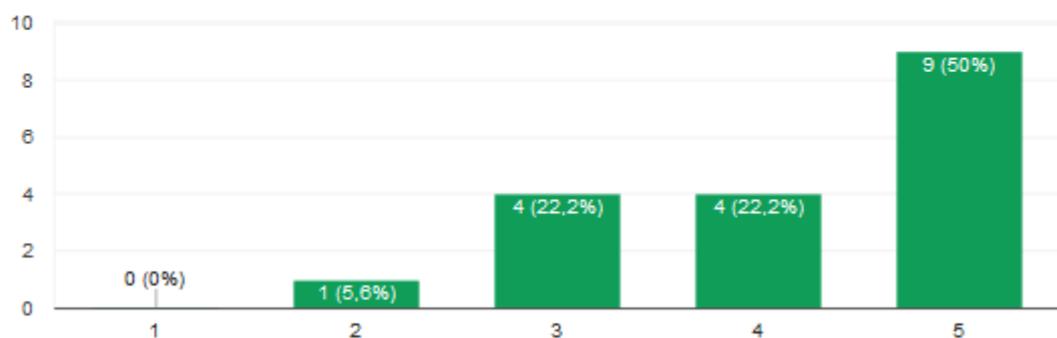
18 respostas



Fonte: Elaborado pela Autora (2017)

5. Como você avalia sua experiência na utilização da ferramenta Codecademy?

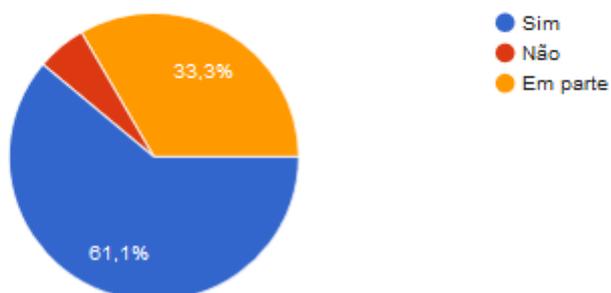
18 respostas



Fonte: Elaborado pela Autora (2017)

6. Os conhecimentos e prática experimentados utilizando a ferramenta Codecademy lhe ajudaram satisfatoriamente como uma boa base para continuação dos estudos de forma autônoma?

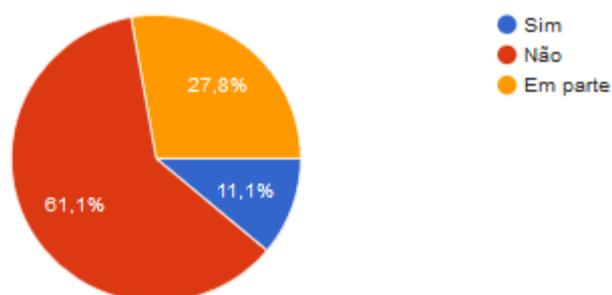
18 respostas



Fonte: Elaborado pela Autora (2017)

8. Encontrou dificuldades na utilização da ferramenta Codecademy?

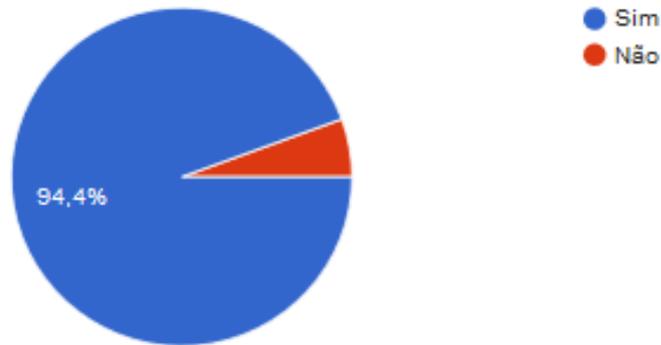
18 respostas



Fonte: Elaborado pela Autora (2017)

9. Na sala de aula, conseguiu sanar suas dúvidas sobre o conteúdo visto previamente na Codecademy?

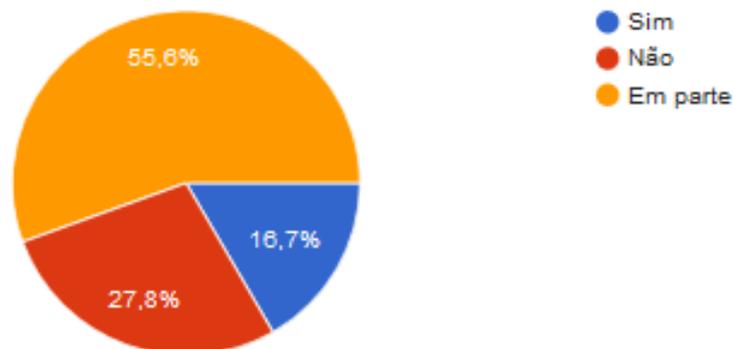
18 respostas



Fonte: Elaborado pela Autora (2017)

10. Teve dificuldade em resolver as atividades propostas pelo professor apenas acessando o conteúdo pela Codecademy?

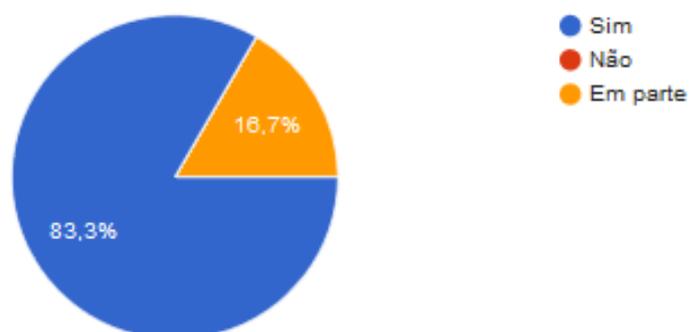
18 respostas



Fonte: Elaborado pela Autora (2017)

11. O conteúdo visto na ferramenta Codecademy foi suficiente para resolver as atividades passadas em sala de aula?

18 respostas



Fonte: Elaborado pela Autora (2017)

APÊNDICE G - Respostas do Questionário da Avaliação Docente

1. Como você avalia a experiência da utilização em conjunto do Flipped Classroom+Gamification+Aprendizagem colaborativa como estratégias para o ensino/aprendizagem de Programação de Computadores? *

1 - Péssimo; 2 - Ruim; 3 - Regular; 4 - Bom; 5 - Ótimo.

	1	2	3	4	5	
Péssimo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Ótimo

2. Utilizaria em outras turmas/disciplinas as estratégias apresentadas? *

- Sim
 Não
 Em parte

3. Após a junção da turma, percebeu alguma diferença entre os alunos que tiveram aula pelo método tradicional em relação aos que viram pela sala de aula invertida? *

- Sim
 Não
 Em parte

4. Caso a resposta seja sim na pergunta anterior, o que percebeu?

Os alunos se divertem mais com a sala invertida, é um bom mecanismo para despertar o interesse dos alunos

5. Conversou com os alunos sobre a experiência? Acredita que os alunos avaliaram positivamente ou negativamente as ferramentas e estratégias utilizadas? *

- Avaliaram Positivamente
 Avaliaram Negativamente
 Não conversei sobre o assunto

6. Após os experimentos realizados em laboratório, percebeu interesse dos alunos em continuarem utilizando as ferramentas apresentadas? *

- Sim
 Não
 Em parte

7. Percebeu os alunos mais ativos/participativos em relação ao seu auto aprendizado? *

- Sim
- Não
- Em parte

8. Acredita que os alunos que viram o conteúdo pela metodologia Flipped Classroom utilizando as ferramentas Codecademy e CodeinPY realmente aprenderam o conteúdo? *

- Sim
- Não
- Em parte

9. Como avalia a ferramenta Codecademy para acompanhamento do aprendizado dos alunos? *

1 - Péssima; 2 - Ruim; 3 - Regular; 4 - Boa; 5 - Ótima.

	1	2	3	4	5	
Péssimo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Ótimo

10. Como avalia o protótipo CodeinPY como ferramenta para Aprendizagem Colaborativa dos conceitos de Programação de Computadores? *

1 - Péssima; 2 - Ruim; 3 - Regular; 4 - Boa; 5 - Ótima.

	1	2	3	4	5	
Péssimo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Ótimo

11. Continuará a utilizar as ferramentas Codecademy/CodeinPY em próximas experiências? *

- Sim
- Não
- Em parte

12. Gostaria de informar sugestões/críticas ao protótipo CodeinPY ou a ferramenta Codecademy?