

Coding Dojo como Prática de Aprendizagem Colaborativa para Apoiar o Ensino Introdutório de Programação: Um Estudo de Caso

Géssica M. da S. Alves^{1,2}, Ayla Dantas Rebouças², Pasqueline D. Scaico²

¹ Universidade Federal de Campina Grande (UFCG)

² Universidade Federal da Paraíba (UFPB)

gessica@copin.ufcg.edu.br, {ayla, pasqueline}@dce.ufpb.br

Abstract. *Learning programming is a challenging task for many novice students in Computer Science programs. This article proposes the use of Coding Dojos while teaching introductory programming aiming to evaluate if this practice influences aspects related to motivation, collaboration and learning. Data obtained are a result of surveys applied with novices of two introductory programming courses. The results point to the following general positive conclusions: programming Dojos might encourage students to learn in a more participative and motivated way; and create a collaborative environment that works in favor of learning. On the other hand, some negative aspects emerged: in some classes coding with a colleague is a big challenge.*

Resumo. *Aprender programação é uma tarefa desafiadora para muitos alunos ingressantes dos cursos da área de Computação. Este artigo propõe o uso de Coding Dojos no ensino introdutório de programação e avalia se a prática influencia positivamente nos aspectos de motivação, colaboração e percepção de aprendizagem. Os dados obtidos são fruto de questionários aplicados em turmas de programação do primeiro ano de cursos da área de tecnologia. Os resultados apontam as seguintes conclusões gerais positivas: a promoção de Dojos de programação pode incentivar a participação e motivação de alunos e criar um ambiente colaborativo que favorece o aprendizado. Por outro lado, os resultados obtidos também identificaram alguns aspectos negativos: em algumas turmas trabalhar em par é um grande desafio.*

1. Introdução

As disciplinas que envolvem o ensino de Lógica de Programação têm se mostrado um grande obstáculo para alguns alunos da área de Computação. Muitos têm apresentado dificuldades em entender conceitos, conteúdos, desenvolver a capacidade de abstração e lógica para resolução de problemas. Conforme mencionado por Silva et al. (2009), tais disciplinas apresentam altos índices de evasão e reprovação, o que gera um ciclo negativo, que afeta a motivação¹ dos alunos.

Um dos motivos para evasão e desinteresse na área são problemas na formação básica dos alunos que ingressam nas universidades [Paula et al. 2009]. Além disso, deficiências na base matemática e dificuldades de interpretação de problemas fazem

¹ Neste artigo, os termos motivação e interesse são utilizados de maneira similar.

com que muitos alunos não apresentem as competências necessárias para a resolução de problemas, o que prejudica sua aprendizagem de programação.

As atividades propostas pelos professores para que os estudantes pratiquem programação, muitas vezes, são realizadas através de estratégias de tentativa e erro, que causam frustração na busca pela solução [Ettles; Luxton-Reilly; Denny 2018]. Esse processo chamado de tentativa e erro se mantém até que a solução correta seja alcançada [Almeida et al. 2017]. Entretanto, as tentativas frustradas podem se tornar frequentes, fazendo com que o aluno sinta desmotivação para continuar praticando a atividade, um fator que pode levá-lo a desistir de entregar as atividades ou entregá-las incompletas [Oliveira 2018].

Impactos na motivação dos iniciantes são observados, dentre outras razões, porque muitos não possuem conhecimentos prévios na área, o que influencia o nível de complexidade que alguns julgam estar associado ao processo de aprender a codificar [Berssnette 2016].

A motivação dos alunos é fator indispensável para o seu sucesso em qualquer processo de aprendizagem. De acordo com Guimarães e Boruchovitch (2004), quando estão motivados, eles apresentam entusiasmo para realizar atividades e costumam orgulhar-se do próprio desempenho.

Visando motivar estudantes, alguns educadores buscam experimentar diferentes metodologias, técnicas e ferramentas para apoiar o ensino-aprendizagem de programação, como é o caso da aprendizagem ativa. A aprendizagem ativa visa fazer com que os estudantes se tornem ativamente envolvidos no seu próprio aprendizado, buscando ampliar suas próprias descobertas. Sendo assim, a técnica conhecida como *Coding Dojo* (CD) foi utilizada no estudo descrito neste artigo para auxiliar a implementação de práticas de aprendizagem ativa.

Recentemente, o CD surgiu como uma prática de programação colaborativa e motivadora, com potencial para proporcionar um ambiente de aprendizagem não-competitivo, e que é realizada, frequentemente, em pares [Estácio et al. 2015a]. Experiências com esta prática ocorrem na literatura, principalmente, com profissionais no mercado de trabalho ou alunos de programação mais avançados [Luz et al. 2013] [Estácio et al. 2016].

Todavia, nesta pesquisa, as pesquisadoras estavam instigadas a saber: Será que a técnica de CD pode contribuir para a motivação e percepção de aprendizagem de iniciantes em programação se aplicada em sala de aula? Será que é capaz de estimular que os estudantes aprendam a trabalhar colaborativamente?

Diante disso, o presente trabalho teve por objetivo avaliar se a aplicação de *Coding Dojos* poderia influenciar aspectos relacionados à motivação, colaboração e percepção da aprendizagem em disciplinas introdutórias de programação. A fase inicial desta pesquisa consistiu em um levantamento bibliográfico com o objetivo de identificar artigos relacionados ao uso de CD no ensino introdutório de programação. Em seguida, o método científico foi discutido. Optou-se por conduzir um estudo de caso, com caráter exploratório e descritivo. Os dados foram obtidos através de questionários online. O ambiente de investigação consistiu em duas turmas da Universidade Federal da Paraíba (Campus IV/ Rio Tinto – PB), uma de Introdução à Programação e outra de Linguagem de Programação (disciplina do segundo período). Por meio do estudo, pôde-se analisar a

motivação dos estudantes, as formas com que colaboraram e como percebiam seu aprendizado ao longo das sessões de CD, as quais focavam na resolução dos desafios de programação.

Os resultados obtidos, apesar de preliminares e limitados ao contexto do caso que foi estudado, têm valia, uma vez que podem auxiliar educadores que pretendem adotar novas práticas para apoiar o processo de ensino-aprendizagem de programação, além de apresentar uma visão geral sobre aspectos relacionados à motivação, colaboração e aprendizagem durante o uso de *Coding Dojos*.

O trabalho está organizado nas seguintes seções: o referencial teórico encontra-se na Seção 2; os trabalhos relacionados ao presente estudo estão descritos na Seção 3; na Seção 4 é apresentada a metodologia adotada, bem como o instrumento de coleta de dados; a caracterização das sessões de Dojo está detalhada na Seção 5; na Seção 6 são apresentados os resultados do estudo. Por fim, na Seção 7, as pesquisadoras apresentam algumas conclusões preliminares e possibilidades para trabalhos futuros.

2. Referencial Teórico

2.1. Aprendizagem Ativa

A aprendizagem ativa, em inglês *Active Learning*, é definida por Prince (2004) como qualquer método instrucional que envolva os alunos no processo de aprendizagem. Nessa aprendizagem os alunos são estimulados a fazer atividades significativas por meio das suas próprias descobertas, fazendo reflexões sobre o que estão fazendo durante todo processo.

Prince (2004) ainda apresenta e define três abordagens de aprendizagem ativa: Aprendizagem Colaborativa, Aprendizagem Cooperativa e Aprendizagem Baseada em Problemas. Aprendizagem Colaborativa, ou do inglês, Collaborative Learning, pode referir-se a qualquer método instrucional no qual os estudantes trabalham juntos em pequenos grupos em direção a um propósito comum. A aprendizagem cooperativa é um tipo de abordagem colaborativa no qual ocorre o trabalho em grupo de forma estruturada; valoriza-se a aplicação de incentivos que possam ser construtivos (em detrimento ao encorajamento pela competição, por exemplo) e no qual a avaliação tende a ser individualizada. Aprendizagem baseada em problemas é um método instrucional onde problemas relevantes são introduzidos no início do ciclo de instrução, usados para fornecer a contextualização e motivação para aprendizagem que se segue pelos estudantes.

Aprendizagem Colaborativa é uma metodologia ativa que permite ao aluno tornar-se participante ativo do processo, onde a construção do conhecimento é mediante a discussão e interação com colegas e educadores [Harasim 1989]. Assim, em ambientes colaborativos é essencial para aprendizagem a participação de outras pessoas que desejem de maneira entrelaçada compartilhar seus conhecimentos para realização de uma tarefa ou atividade. No contexto de programação, as práticas de Programação em Par e Dojo são representantes de metodologias colaborativas, pois permitem a dinâmica de duas ou mais pessoas trabalhando em conjunto no mesmo algoritmo ou código. A colaboração é utilizada como recurso indispensável para aprendizagem de novos conceitos e a motivação dos programadores [Estácio et al. 2015a].

2.2. Motivação e Engajamento

A motivação é definida por Bzuneck (2000, p. 9) como “aquilo que move uma pessoa ou que a põe em ação [...]”. Para outros, a motivação é um conjunto de aspectos que se referem a estar feliz ou não realizando uma tarefa; sentir-se bem no ambiente; entrosar-se com pessoas mantendo-se em paz e harmonia [Klava 2010].

É importante diferenciar motivação e engajamento. Engajamento remete a envolvimento, dedicação; pode ser entendido como a interação de um indivíduo com um dado contexto [Fredricks et al. 2004]. Segundo Fredricks e McColskey (2012, p. 765) “o indivíduo está engajado em algo (isto é, tarefa, atividade e relacionamento), e seu engajamento não pode ser separado de seu ambiente”. Além disso, o engajamento tende a ser pensado em termos de ação, ou manifestações comportamentais, emocionais e cognitivas da motivação. Logo, o engajamento é um componente presente em pessoas que estão motivadas. Assim, é um componente essencial para medir a motivação das pessoas, uma vez que materializa a conexão do meio com o indivíduo [Scaico 2018].

2.3. Coding Dojo (CD)

O termo Dojo está relacionado ao lugar onde se pratica artes marciais ou um local de meditação na cultura japonesa [Delgado et al. 2012] [Luz e Neto 2012]. Portanto, em inglês, *Coding Dojo* nada mais é que um “local de treinamento de código”, ou “local de treinamento de programação” [Bonfim 2014].

Dojo de Programação é uma sessão onde um grupo de participantes se reúne para praticar programação [Heinonen et al. 2013]. Dentre os princípios básicos de um Dojo está a criação de um ambiente seguro, colaborativo, inclusivo e não competitivo, onde as pessoas possam aprender continuamente [Sato et al. 2008].

Os Coding Dojos apresentam um ou vários desafios de programação nos quais os participantes são incentivados a participar e compartilhar suas habilidades [Sato et al. 2008]. As pessoas envolvidas na resolução dos desafios geralmente possuem diferentes níveis de habilidades e conhecimentos. Este desequilíbrio permite que os participantes colaborem e troquem experiências desenvolvendo assim novas competências [Rodrigues et al. 2017].

Existem diferentes formatos de Coding Dojo [Sato et al. 2008]. Os formatos mais conhecidos são: *Kata*, *Randori* e *Kake*. Em todos os formatos a programação em par é o principal mecanismo que permite participação no grupo, ou seja, como o nome já sugere, consiste em dois desenvolvedores trabalhando colaborativamente na mesma atividade.

A primeira variante de Dojo é chamada de *Kata*. Neste formato, a dinâmica é iniciada com um mentor apresentando a solução desde o início às duplas, que são fixas. Nelas, um participante atua como piloto (escrevendo o código) e outro como co-piloto (sendo ele “observador”, mas podendo ajudar e apoiar o piloto). Durante a apresentação da solução, é descrito todo o passo a passo aos membros envolvidos, podendo qualquer um interromper o apresentador, pedir explicações, sugerir alternativas ou tirar dúvidas. O objetivo final é que todos possam reproduzir a solução alcançando os mesmos resultados [Rodrigues et al. 2017].

Na sessão *Randori*, o grupo de participantes trabalham juntos seguindo a dinâmica: (i) um participante atua como piloto, (ii) outro atua como o co-piloto e (iii) os

demais participantes atuam como o público, ficando a maior parte do tempo em silêncio prestando atenção no par que está mexendo no código que vai sendo projetado para todos enquanto é construído. No entanto, todo público é capaz de participar das discussões, mas sugere-se que isso ocorra só após os testes de unidade passarem. As posições (piloto e co-piloto) dos participantes são alternadas em rodadas [Oliveira et al. 2016] [Estácio et al. 2015a].

Sato et al. (2008) recomendam que cada rodada deve durar de 5 a 7 minutos. No final de uma rodada, o piloto se junta ao público, o co-piloto torna-se o novo piloto e um outro participante da plateia se torna o novo co-piloto. Cada participante age pelo menos uma vez como piloto e uma vez como co-piloto. Não existe uma organização exata para a formação das duplas, os próprios participantes são estimulados a se voluntariar e contribuir com o(s) desafio(s) no decorrer da sessão.

Por fim, o Kake é conhecido por ser semelhante ao Randori, porém com algumas diferenças. Neste formato, múltiplos pares em computadores diferentes buscam resolver simultaneamente o mesmo problema. Desta maneira, a ociosidade dos participantes por longos períodos de tempo (como pode ocorrer no Randori) é descartada, dado que todos eles estão desenvolvendo em um computador na aplicação do Kake. A dinâmica de piloto e co-piloto permanece durante toda sessão. Porém, com a seguinte distinção em cada rodada: (i) cada piloto em um par se torna um co-piloto em outro par e (ii) cada co-piloto em um par se torna um piloto com um outro par. A rotatividade também dura alguns minutos e os participantes são instigados a trocar por um par diferente a cada rodada [Rodrigues et al. 2017] [Schoeffel 2016].

O local de realização das sessões exige apenas um ou vários computadores e um meio de divulgação do código, sendo usualmente um projetor. As sessões geralmente apresentam o mesmo processo, conforme descrito por Sato et al. (2008). Primeiramente é realizada a escolha/apresentação do problema (5 a 10 minutos), seguida da discussão sobre o problema (10 a 20 minutos) e sessão para a sua resolução (1 a 2 horas). Finaliza-se com a retrospectiva (10 a 20 minutos). O tempo da sessão de resolução do problema pode ser adaptado de acordo com a necessidade do contexto [Souza-Concilio et al. 2015].

É importante destacar que o ambiente de um Dojo é planejado para incentivar seus frequentadores a participarem ativamente da programação [Carmo 2012]. Além de engajá-los a aprender e refletir sobre várias formas de soluções com os outros participantes, visa-se ajudar os participantes a melhorar suas próprias práticas de desenvolvimento [Heinonen et. al. 2013]. Nos Dojos, o recurso de retrospectiva é normalmente realizado ao final do evento, devido a ser uma forma eficaz e extremamente importante para discutir sobre o problema e refletir sobre o que foi aprendido, verificando ainda se realmente a sessão atendeu seus objetivos.

3. Trabalhos Relacionados

Considerando trabalhos semelhantes ao do presente artigo, o trabalho de Estácio et al. (2015a) foi considerado um dos mais relacionados. Os autores buscaram identificar a influência de novas práticas colaborativas com alunos iniciantes de programação, sendo utilizado *Coding Dojo Randori* como uma dessas práticas. Eles avaliaram ainda aspectos relacionados à motivação e aprendizagem dos estudantes. No estudo da motivação os autores utilizaram uma escala já desenvolvida [Wangenheim et al. 2013] e

adaptaram com foco apenas nos aspectos de atenção, relevância e confiança, e aprendizagem a longo prazo. Como resultado, eles comprovaram que CD gera melhores resultados nas habilidades de programação, tratando-se de programadores com pouca ou nenhuma experiência prática com desenvolvimento de software. No presente artigo é explorada uma variação do formato Kake de CD, como será detalhado posteriormente.

Rodrigues et al. (2017) relatam suas experiências com o uso da técnica Dojo visando superar algumas dificuldades dos alunos dos cursos de Engenharia de Software e Ciência da Computação da UNIPAMPA. No trabalho são apresentadas resumidamente quais foram as percepções do que funcionou e do que não funcionou no estudo. Ao final, a importância dessas novas práticas é demonstrada, como também o quanto os alunos se sentiram engajados para solucionar problemas.

No Brasil, Sato et al. (2018) expõem em seu trabalho a iniciativa do grupo DojoSP, que utiliza a prática colaborativa utilizada nesta pesquisa para criar um ambiente de aprendizado de métodos ágeis em que se busca estimular o uso de técnicas como desenvolvimento orientado a testes, refatoração, programação em par, entre outros. A diferença do presente trabalho é o foco dado ao uso da técnica com iniciantes.

4. Metodologia

Avaliou-se adequado como método de pesquisa o estudo de caso. Segundo Yin (2010), os estudos de caso têm por objetivo explorar, descrever e explicar um evento ou fornecer uma compreensão mais profunda de um fenômeno.

A pesquisa desenvolvida tem caráter exploratório e descritivo, uma vez que permite uma maior familiaridade entre o pesquisador e o fenômeno que deve ser investigado, além de descrever como pode se dar o uso do Dojo no ensino introdutório de programação em termos da motivação dos alunos, colaboração e percepção da aprendizagem [Theodorson 1970].

No intuito de observar aspectos relativos à motivação, colaboração e percepção da aprendizagem de alunos participando de sessões de Coding Dojo, a coleta de dados foi realizada com estudantes matriculados em duas disciplinas introdutórias de programação. Ambas são ofertadas nos cursos Licenciatura em Ciência da Computação e Bacharelado em Sistemas de Informação. IP é ofertada no primeiro semestre e LP no segundo. As duas turmas foram conduzidas pela mesma professora. A coleta de dados ocorreu uma vez, após todas as sessões, que aconteceram entre agosto e setembro de 2018.

Quanto à estratégia para coleta de dados, foi utilizado um questionário online composto por dez questões abertas e fechadas (<https://goo.gl/Nx21wV>). Os dados coletados são apresentados e discutidos na Seção 6.

5. Caracterização das sessões de *Coding Dojo*

As sessões nas turmas de Introdução à Programação e Linguagem de Programação utilizaram, respectivamente, as linguagens de programação Python e Java e apenas na segunda havia um foco na Programação Orientada a Objetos. Em ambas, as sessões seguiram a mesma estrutura, sendo feitas em sala de aula, e algumas vezes, em laboratório, com duração de 40 a 50 minutos.

A dinâmica era a seguinte: inicialmente o desafio de programação era apresentado pela professora. Logo em seguida, os alunos eram estimulados a se juntar em pares (um participante atuando como piloto e o outro como co-piloto). Quando a sessão ocorria no laboratório, apenas um computador era utilizado pela dupla. Quando ocorria em sala de aula, uma folha de papel era compartilhada pela dupla. Decidiu-se utilizar uma variação do formato Kake.

No formato adotado permitia-se aos alunos pedir explicações ou tirar dúvidas a qualquer momento com a professora. Esta decisão foi tomada, uma vez que se percebeu que, por se tratar de estudantes aprendendo a programar, muitos estavam tendo o primeiro contato com programação ou com as linguagens de programação adotadas. A escolha de uma variação do formato Kake fundamentou-se em função do seguinte: foi visto no trabalho de Rodrigues et al. (2017) que em turmas maiores de programação torna-se mais produtivo trabalhar com o Kake devido ao método proporcionar que todos trabalhem simultaneamente, diferentemente do formato Randori, que em turmas maiores faz com que os estudantes possam passar muito tempo sem programar. Com Randori, a sessão passa a requerer uma duração maior para que todos participem ao menos uma vez como piloto e co-piloto.

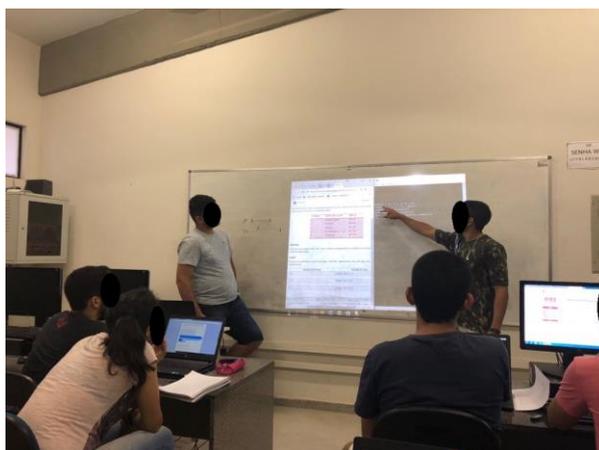


Figura 1. Ilustração de uma sessão Dojo de Programação com participantes do estudo

Na variação do Kake adotada, as posições (piloto e co-piloto) dos participantes eram alternadas em rodadas, com a diferença que o participante que atuava como piloto, passava a ser co-piloto no seu par, enquanto o que atuava como co-piloto tornava-se piloto. As trocas com os outros pares na sessão ocorriam apenas após a resolução e discussão do desafio posto, conforme apresentado na Figura 1, com a apresentação da solução de uma das duplas e discussões sobre soluções diferentes. Os alunos eram estimulados a buscar sempre trocar de dupla em diferentes sessões, interagindo assim com vários colegas ao longo da disciplina.

Diferentemente do que foi recomendado por Sato et al. (2008), de que cada rodada deveria durar de 5 a 7 minutos, na variação adotada, a rodada teve a duração de 10 minutos, tempo que se mostrou suficiente para que eles exercessem melhor o papel de piloto e co-piloto e praticassem com mais calma a solução do desafio.

6. Resultados e Discussão

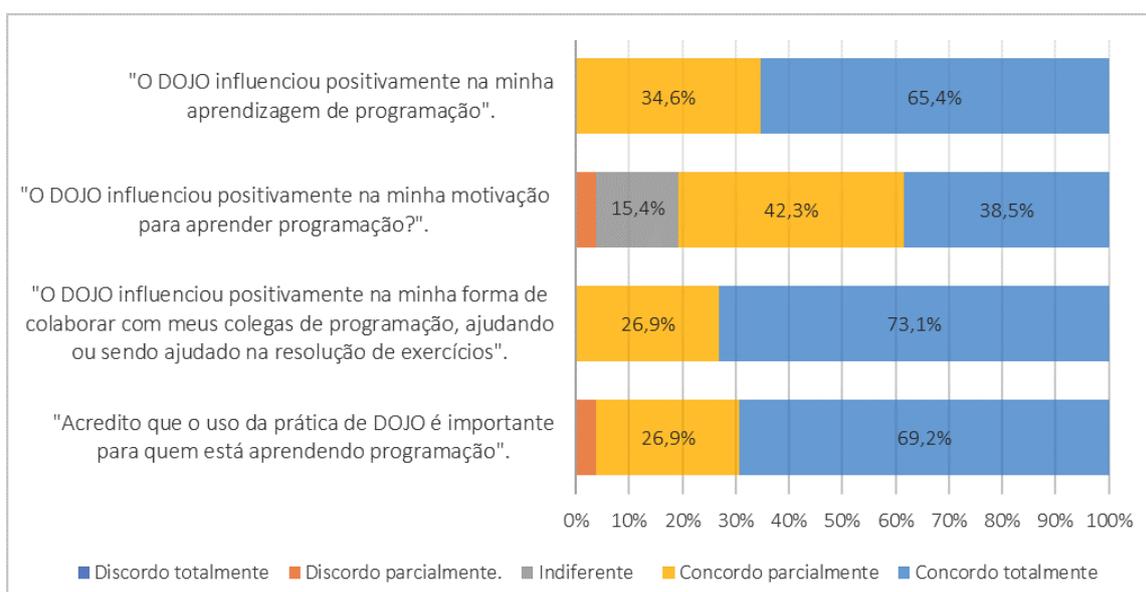
No total foram realizadas seis sessões de CD na disciplina de IP durante as aulas, e cinco em LP. A seguir serão apresentados os resultados obtidos por turma.

6.1. Dojo no ensino de programação de Introdução à Programação

A aplicação do questionário ocorreu no mês de setembro de 2018, com um total de 26 alunos da disciplina de IP respondentes (53,3% dos alunos matriculados), sendo preservada a identidade de todos.

Os dados correspondentes às respostas à questão 1 (contido no link mencionado na página 7) estão ilustrados pelo Gráfico 1. Nesta questão estavam sendo considerados os aspectos gerais de motivação, colaboração e percepção de aprendizagem dos alunos por meio de afirmativas e o grau de concordância dos alunos. De maneira geral foram observados resultados positivos principalmente nas frases em que se destaca a influência do Dojo nos aspectos de colaboração e aprendizagem. Nas afirmativas desta natureza, observou-se que a maioria dos alunos concordavam totalmente ou parcialmente. Além disso, houve alta concordância sobre o fato de a técnica ser importante para quem está aprendendo programação (69,2% concordaram totalmente e 26,9% concordaram parcialmente).

Gráfico 1. Aspectos gerais de motivação, colaboração e percepção de aprendizagem



Com relação à frase "O Dojo influenciou positivamente na minha motivação para aprender programação" observou-se uma menor concordância por parte dos alunos. Mesmo assim, 80,8% concordam totalmente ou parcialmente com a frase.

A questão sobre o que os alunos consideravam bom quando estavam participando de uma sessão Dojo obteve os principais aspectos: a resolução de problemas de modo interativo com os colegas; a discussão sobre os exercícios e a forma como a técnica os ajuda e faz aprenderem com os seus colegas. Alguns discursos que ilustram essas opiniões são apresentados a seguir: "A interação com minha dupla, uma ajuda mútua, além do momento de debate sobre a resolução do problema" (P01); "A

colaboração dos meus colegas que tive na hora de conseguir a solução dos exercícios” (P02); *“A possibilidade de um ajudar o outro e juntos resolver e aprender o problema”* (P03); *“A disponibilidade em ajudar o que não foi compreendido, e ter discussões para o melhor aprendizado”* (P04) e *“Os revezamentos e as trocas de informação com o parceiro”* (P05). Com base nestas respostas, observa-se que o aspecto que foi mais estimulado com as sessões foi a colaboração. A maioria considerou a interação, a colaboração e a ajuda como primordiais para seu aprendizado em programação.

A maioria dos alunos não percebeu aspectos negativos ao participar dos Dojos, porém existiram algumas menções quanto ao tempo que poderia ser maior para resolução dos exercícios, como pode ser visto nas seguintes falas: *“O tempo às vezes limita o que você pode contribuir (...)”* (P06). Outros participantes relataram sobre conflitos com colegas por não saberem ou não gostarem de trabalhar em dupla: *“(...) a relação de interação por vezes tem certos conflitos de opiniões”* (P07); *“Às vezes ocorria conflitos na forma de solucionar uma determinada questão”* (P08) e *“Algumas duplas não entendiam o significado de ser co-piloto”* (P05). Por fim, apenas um aluno afirmou: *“Pessoalmente, não gosto de trabalhar em grupo”* (P09).

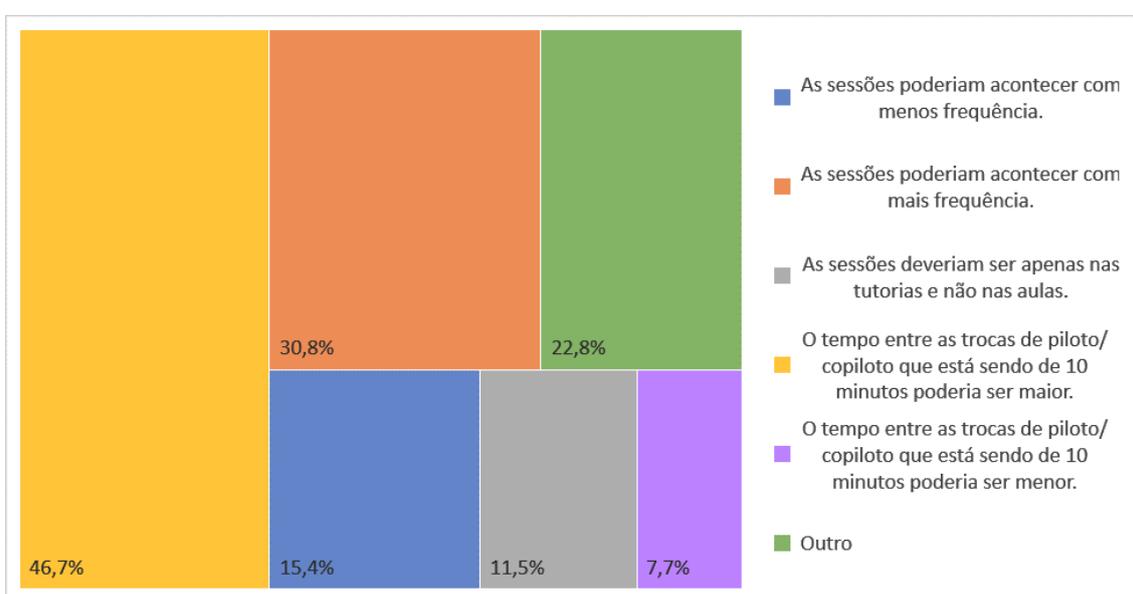
A experiência de trocar de posição de piloto e co-piloto agradou a maioria dos participantes do Dojo, uma vez que 96,2% afirmaram que gostaram sim de permitir essa troca, enquanto apenas um disse não gostar. As justificativas obtidas como resposta para esta questão ressaltaram a importância do trabalho em grupo; do aprendizado obtido e de a técnica ajudar aos que se sentem inseguros, conforme se percebe através das seguintes respostas: *“Foi bom para aprender a trabalhar em grupo e para a construção do meu conhecimento, uma vez que muitas vezes eu acabava aprendendo algo novo, pois as pessoas com quem eu fazia dupla tinham maneiras diferentes da minha de fazer as coisas”* (P10); *“O bom é que todos conseguem mexer no programa e você consegue ver outro modo de ver alguns códigos”* (P11); *“Pois me fez pensar em soluções que eu não teria pensado só”* (P12) e *“Pois assim os que têm menos segurança, conseguem ganhar a mesma e melhorar na programação”* (P13).

A estratégia dos alunos buscarem uma dupla nova a cada sessão também foi explorada no questionário, tendo em vista que a estratégia proporciona que todos colaborem com diferentes colegas. Obteve-se o seguinte: 84,6% dos alunos (22) responderam que gostaram, afirmando positivamente com respostas sobre a troca de experiências e a importância do contato com outros colegas. Algumas respostas a seguir são fornecidas: *“Troca de experiência é sempre importante e quando a dupla muda, consequentemente "concatenamos" ideias diferentes. Isso contribui ainda mais para a aprendizagem”* (P14); *“Foi muito bom, porque me trouxe experiências novas e contato com mentes(ideias) diferentes”* (P02); *“Pois me fez conhecer várias formas de pensar”* (P12) e *“A dinâmica de dupla sempre é boa para o "fortalecimento" da turma”* (P15). Por outro lado, 15,4% dos alunos (quatro) não gostaram de ter uma nova dupla a cada sessão. Como justificativa para essa questão um aluno mencionou a questão de afinidade: *“Porque é bom trabalhar com quem temos mais afinidade”* (P16). Outro ficou indiferente ao fato de ter uma nova dupla, afirmando: *“Acho indiferente”* (P17). Os outros dois preferiram não fornecer justificativa.

Visando implantar melhorias em sessões futuras de Dojo, foi perguntado aos alunos sobre o que eles consideravam que poderia mudar, oferecendo-lhes algumas opções de seleção e a opção "Outros" (vide Questão 8 do questionário, que é do tipo

múltipla escolha e, por isso, no Gráfico 2 os percentuais ultrapassam 100%). Quase metade (46,7%) respondeu que o tempo de 10 minutos para as trocas de piloto/co-piloto poderia ser maior. Um ponto positivo, é que 30,8% dos alunos demonstraram aceitação com a prática, solicitando que as sessões ocorressem com mais frequência. Seis alunos (22,8%) adicionaram outras respostas nesta questão, sendo que três delas se referiram ao uso da prática de Dojo individualmente, uma outra resposta sugeriu o uso de desafios mais difíceis onde todos os alunos se juntassem em uma única máquina para resolvê-lo, um outro aluno mencionou que a troca entre os pares deveria ser determinada entre eles caso sentissem necessidade, e por fim, um aluno elogiou que a forma aplicada do Dojo na sala de aula estava bom, e que não precisava de ajustes nas futuras sessões.

Gráfico 2. Melhorias sugeridas pelos participantes para sessões futuras de Dojo



Muitos dos alunos que participaram das sessões de Dojo na turma foram por mais de quatro vezes (46,2%) ou três a quatro vezes (46,2%). Apenas dois (7,7%) alunos responderam que participaram aproximadamente de uma ou duas sessões.

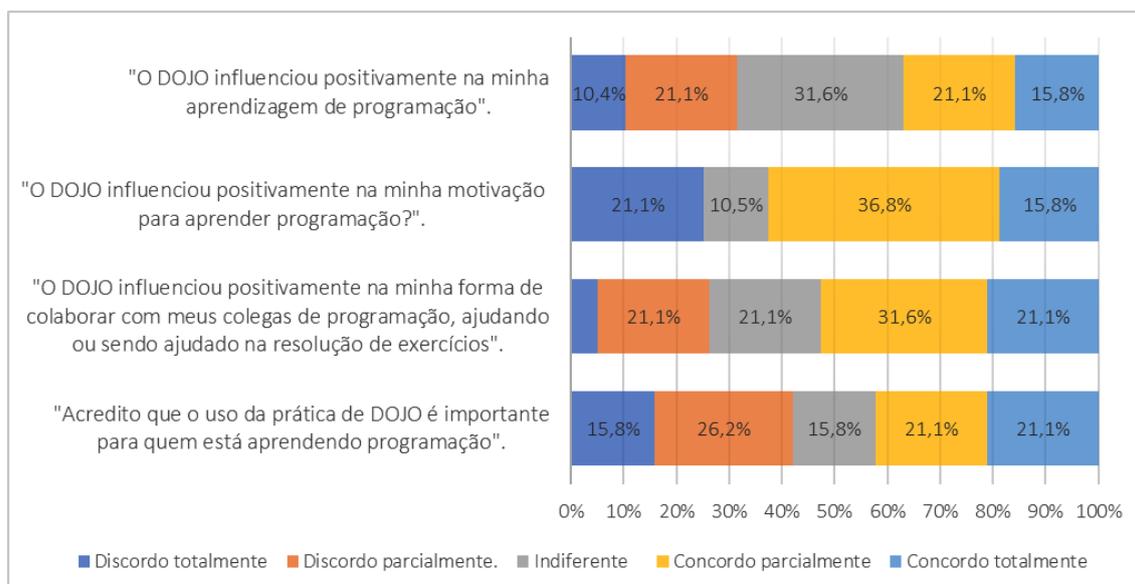
6.2 Dojo no ensino de programação de Linguagem de Programação

Na turma de Linguagem de Programação, o mesmo questionário foi aplicado e obteve a resposta de 19 alunos (42,2% dos 45 matriculados).

No Gráfico 3 estão as respostas dos alunos referentes à questão 1. Todas as afirmativas receberam respostas bem divididas quanto aos aspectos estudados, variando todas as respostas na escala de discordo totalmente a concordo totalmente. Quando perguntados sobre se acreditavam que a sessão Dojo influenciou positivamente na sua aprendizagem de programação, muitos (31,6%) marcaram como indiferente e 10,5% responderam que discordam totalmente. Apesar disso, 36,9% concordaram totalmente ou parcialmente com a frase. Com relação à influência do Dojo em sua motivação e colaboração, os alunos em sua maioria concordaram parcialmente (36,8% e 31,6%) e totalmente (15,8% e 21,1%) que o Dojo influenciou positivamente nesses aspectos.

Na frase "Acredito que o uso da prática de Dojo é importante para quem está aprendendo programação", as respostas foram diversas, onde a maior parte (42,2%) dos alunos concordaram parcialmente ou totalmente quanto a esta afirmação. Por outro lado, 42% discordaram parcialmente ou totalmente. Tais resultados mostram que a turma dos alunos do segundo período ficou bem dividida quanto a essa questão.

Gráfico 3. Aspectos gerais de motivação, colaboração e percepção de aprendizagem



Quanto ao que os alunos de LP consideraram bom na prática *Coding Dojo*, a maioria destacou a troca de conhecimento, as diferentes formas de resolver os problemas, a colaboração, a interação e o tempo estabelecido só para tirar dúvidas dos exercícios. Algumas das respostas dos alunos nessa questão foram: “*A colaboração entre os alunos; a prática do entendimento do código de outro programador*” (P01); “*Me ajudou a identificar quais dúvidas eu tinha em comum com minha dupla, fazendo com que nós tomássemos alguma decisão para resolver o problema*” (P02); “*As diferentes formas de resolver os problemas propostos*” (P03) e “*A interação com outro programador*” (P04).

Ao levantarem os aspectos negativos do uso do Dojo, os alunos destacaram a dificuldade de programar em par ou o fato de não gostarem de trabalhar dessa forma por experiências prévias infelizes. Algumas respostas que ilustram estes aspectos são as seguintes: “*Programar em dupla é um pouco complicado*” (P05); “*A falta de interação em alguns casos, onde às vezes um esquece de comunicar a outro o que está fazendo no código, o que atrapalha no entendimento coletivo do código*” (P06) e “*Por alguns alunos estarem atrasados no assunto, muitas vezes o DOJO não consegue trabalhar como deveria*” (P07).

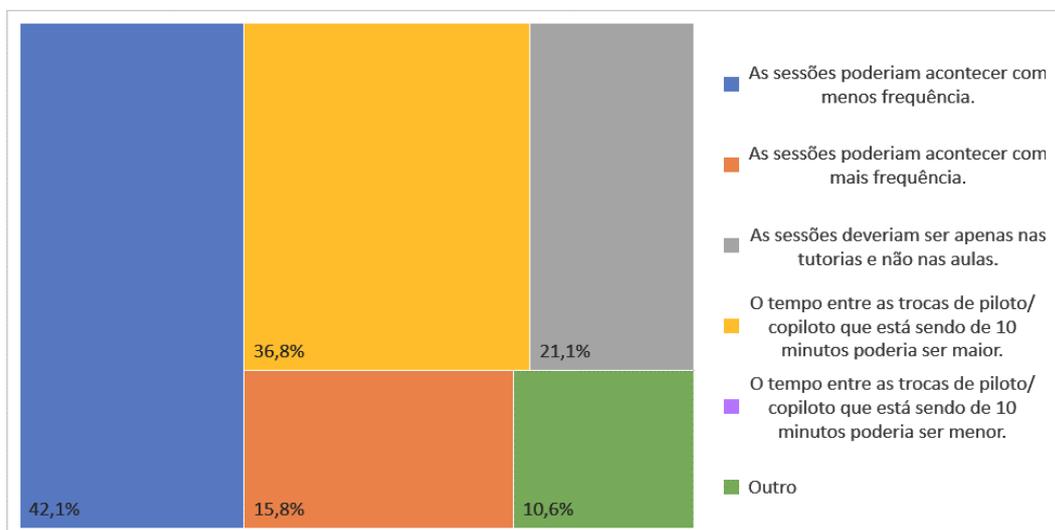
Diferentemente da turma de Introdução à Programação, os alunos de LP marcaram que não gostaram (52,6%) da experiência de trocar de posição de piloto e copiloto durante as sessões de Dojo, e 47,4% afirmaram que gostaram. Dessa maneira, eles justificaram que não gostaram da troca pelo fato de em alguns momentos quebrar o raciocínio quando estavam resolvendo o exercício, como ressaltado nas respostas: “*Achei que cortou meu pensamento em alguns momentos, então prefiro que ao invés de*

trocar de posição a cada tempo determinado, cada grupo tivesse um "piloto" e um "co-piloto" fixos e mudasse conforme as dúvidas” (P02) e *“Às vezes pensamos melhor sem ter que se preocupar com isso de tá trocando de piloto”* (P08). Outros alunos justificaram positivamente afirmando que é bom pela troca de conhecimento, por permitir a visualização de diferentes maneiras de resolução e por reforçar o trabalho em equipe.

A maioria (63,2%) dos alunos responderam que não gostaram da estratégia de buscar ter sempre uma dupla nova a cada sessão de Dojo, destacando que a turma apresenta problemas com afinidade e trabalhar colaborativamente. Algumas das respostas foram *“Por a turma não ser muito colaborativa”, “Acho que seria melhor duplas fixas. Pois as vezes não temos afinidade com outros colegas e de certa forma atrapalha. Com duplas fixas seria um "atalho", já que saberíamos as facilidades/dificuldades do colega”* (P09) e *“Pois de certa forma não temos um entrosamento de ideias”* (P01). Por outro lado, 36,8% dos alunos consideraram que é bom para ambos a experiência em virtude de se ajudarem na resolução, por poderem ir tirando dúvidas e por se prepararem para o trabalho em equipe, como destacado nestas respostas: *“Troca de conhecimentos, é bom para mim que posso saber menos que o parceiro e ele me ajudar. Como posso saber mais e poder ajudá-lo”* (P04); *“Pois isso ajuda a nos preparar a como trabalhar com equipes de pessoas com ideias e pensamentos diferentes”* (P02); e *“Pois eu costumava ir tirando eventuais dúvidas ao decorrer da atividade”* (P10).

De acordo com o Gráfico 4, 42,1% dos alunos de LP gostariam que as sessões ocorressem com menos frequência, 36,8% consideraram que o tempo das trocas deveria ser maior, 21,1% achavam que as sessões de Dojo deveriam ser apenas na tutoria, e 15,8% consideraram que as sessões poderiam acontecer com mais frequência. Essa mesma questão ainda recebeu duas respostas como novas propostas de melhorias para as sessões de Dojo. Uma delas foi para que fossem feitas com duplas fixas e nas próximas aulas após o conteúdo ter sido ministrado na sala de aula; a outra resposta levou em consideração o tempo de troca do piloto e co-piloto, em que citou que *“As trocas de piloto e copiloto devem ser mais dinâmicas e não determinadas por tempo”* (P02).

Gráfico 4. Melhorias sugeridas pelos participantes para sessões futuras de Dojo



A maioria (47,4%) dos alunos que responderam o presente questionário participaram mais de 4 vezes das sessões, enquanto 31,6% dos alunos realizaram os exercícios com a prática 3 ou 4 vezes, e os outros (21,1%) participaram 1 ou 2.

Como se pôde observar, a técnica Dojo na disciplina de Linguagem de Programação foi menos aceita que na turma de IP, algo que pode ter sido um reflexo de parte da turma ter expressado ter dificuldade ou não gostar de fazer trabalho em equipe.

7. Conclusões e trabalhos futuros

Analisando-se os resultados deste estudo, foi visto que para as disciplinas de Introdução à Programação e Linguagem de Programação a técnica colaborativa Coding Dojo contribuiu, de maneira geral, positivamente para a motivação, colaboração e percepção de aprendizagem dos discentes que participaram do estudo.

Nas análises realizadas, observou-se uma aceitação da técnica e percepção de sua importância para a aprendizagem introdutória de programação por um maior percentual da turma de IP do que na turma de LP. Como relatado pela maioria dos alunos de IP no questionário, e por parte dos alunos de LP, o Dojo permitiu melhorias na interação, troca de conhecimentos e colaboração com os colegas. Por outro lado, percebeu-se pelas respostas que alguns alunos demonstraram ainda a dificuldade ou resistência em trabalhar em dupla.

No entanto, acredita-se que é importante explorar em sala de aula o uso de estratégias metodológicas que estimulem a colaboração entre os estudantes, como a que foi aplicada no presente estudo, visando contribuir para a formação de profissionais que possam trabalhar bem em equipe. Apesar da resistência por parte de alguns alunos, é importante destacar que graças à técnica, muitos alunos com dificuldades puderam ser ajudados e houve uma importante troca de experiências entre os discentes, conforme ressaltado por alguns discursos analisados. Aos educadores, vale ainda destacar que sistematizar práticas de aprendizagem ativa, como sessões de CD, é algo desafiador pois em algumas turmas experiências prévias negativas ou problemas de relacionamento entre os alunos podem impedir bons resultados do uso da técnica.

Mesmo assim, percebeu-se que a prática de CD foi considerada positiva para o ensino introdutório de programação, principalmente pelos alunos no primeiro período do curso, e permitiu que os alunos pudessem tirar dúvidas entre si e compartilhassem diferentes conhecimentos sobre a linguagem em estudo e sobre formas de resolver problemas. Como trabalhos futuros, propõe-se: (i) analisar de forma mais abrangente a técnica realizando avaliações de aprendizagem; (ii) explorar a utilização desta prática nos seus diferentes formatos, a fim de identificar qual melhor se encaixaria no contexto da sala de aula com turmas relativamente grandes (35-50 alunos); e (iii) criar um protocolo de observação para compreender em diferentes ambientes as nuances do uso da técnica de Dojo.

Agradecimentos

As autoras gostariam de agradecer aos alunos por terem participado sem receios neste relevante estudo, permitindo que este se tivesse concretizado. Ficamos extremamente gratas pelo apoio, dedicação e cuidado em todos os momentos da pesquisa.

Referências

- Almeida, T. O.; Netto, J. F. M.; Silva, R. G. S.; Custódio, T. P. (2017) “Laboratório Remoto de Robótica como Elemento Motivador para a Aprendizagem de Programação”. In Anais do XXVIII Simpósio Brasileiro de Informática na Educação.
- Berssanette, J. H. (2016) “Ensino de Programação de Computadores: Uma Proposta de Abordagem Prática Baseada em Ausubel.” Dissertação de Mestrado. Ponta Grossa, 144 f: il.
- Bonfim, M. (2015) “O que é o Coding Dojo”. Acessado em: 27 de junho de 2018. Disponível em: <<http://www.devmedia.com.br/o-que-e-o-coding-dojo/30517>>.
- Bzuneck, J. A. (2000) “As Crenças de Auto-Eficácia dos Professores.” In: F.F. Sisto, G. de Oliveira, & L. D. T. Fini (Orgs.). Leituras de psicologia para formação de professores. Petrópolis, Rio de Janeiro: Vozes.
- Carmo, D.H.; Braganholo, V. (2012) “Um Estudo Sobre o Uso Didático de Dojos de Programação”. In Anais do XX Workshop sobre Educação em Computação, Curitiba, Paraná.
- Delgado, C.; Toledo, R.; Braganholo, V. (2012) “Uso de Dojos no Ensino Superior de Computação”. In Anais do XX Workshop sobre Educação em Computação, Curitiba, Paraná.
- Estácio et al. (2015a) “Evaluating collaborative practices in acquiring programming skills: Findings of a controlled experiment”. In SBES’15.
- Estácio et al. (2015b) “Evaluating the use of pair programming and coding dojo in teaching mockups development: An empirical study”. In Proceedings of the Annual Hawaii International Conference on System Sciences, Vol. 2015-March. IEEE, Kauai, 5084–5093. DOI:<http://dx.doi.org/10.1109/HICSS.2015.602>
- Estácio et al. (2016) “On the randori training dynamics”. In 9th International Workshop on Cooperative and Human Aspects of Software Engineering. ACM, Austin, 44–47. DOI:<http://dx.doi.org/10.1145/2897586.2897603>.
- Ettles, A; Luxton-Reilly, A., and Denny, P. “Common Logic Errors Made by Novice Programmers,” in Proceedings of the 20th Australasian Computing Education Conference, 2018, pp. 83–89.
- Fredricks, A. J.; McColskey, W. (2012) “The Measurement of Student Engagement: A Comparative Analysis of Various Methods and Student Self-report Instruments.” S.L. Christenson et al. (eds.), Handbook of Research on Student Engagement, 763 DOI 10.1007/978-1-4614-2018-7_37, © Springer Science+Business Media, LLC.
- Guimarães, S. E.; Boruchovitch, E. (2004) O Estilo Motivacional do Professor e a Motivação Intrínseca dos Estudantes: uma Perspectiva da Teoria da Autodeterminação. Psicologia Reflexão e Crítica, Porto Alegre, v. 17, n. 2, p. 143-150.
- Harasim, L. (1989) On-line Education: A New Domain. In: Mason, R; Kaye, A. (Ed.). Mindweave: communication, computers and distance instruction. Oxford: Pergamon.
- Heinonen, K. et. al (2013) “Learning Agile Software Engineering Practices Using Coding Dojo.” In Proceedings of the 2013 ACM SIGITE Annual Conference on Information

Technology Education. ACM, New York, 97–102.
DOI:<http://dx.doi.org/10.1145/2512276.2512306>

- Klava, V. (2010) “Motivação Empresarial - O Desafio do Século XXI”. Disponível em: <http://www.administradores.com.br/informe-se/artigos/motivacao-empresarial-o-desafio-do-seculo-xxi/48844/>.
- Luz, R.B.; Neto, A. (2012) “Usando Dojos de Programação para o Ensino de Desenvolvimento Dirigido por Testes.” Anais do 23.º Simpósio Brasileiro de Informática na Educação (SBIE), Rio de Janeiro.
- Luz, R.B.; Neto, A.G; Noronha, R. (2013) “Teaching TDD, the Coding Dojo Style”. International Conference Advanced Learning Technologies (ICALT), (Beijing, China),pp.371-375, IEEE, 2013.
- Oliveira, T. A. N. de (2018) “O uso de Pair Programming para Apoiar o Ensino Introdutório de Programação: Um Estudo Qualitativo”. Universidade Federal da Paraíba - Campus IV.
- Oliveira, R. et al. (2016) “Identifying Code Smells with Collaborative Practices: A Controlled Experiment”. In 2016 X Brazilian Symposium on Components, Architectures and Reuse Software 978-1-5090-5086-4/16 \$31.00 © 2016 IEEE DOI 10.1109/SBCARS.2016.18 61 2016.
- Prince, M. (2004) “Does Active Learning Work? A Review of the Research”. In Journal of Engineering Education, 93(3), 2223-231.
- Paula, L. Q., Piva Jr., D., Freitas, R. L. (2009) “A Importância da Leitura e da Abstração do Problema no processo de formação do raciocínio lógico-abstrato em alunos de Computação.” Anais do XVII Workshop sobre Educação em Informática. São Leopoldo – RS, Brasil.
- Rodrigues, da R. L. P., Franz P. L., Cheiran P. F. J., Silva, da S. P. J., Bordin S. A. (2017) “Coding Dojo as a transforming practice in collaborative learning of programming: an experience report.” In Proceedings of SBES’17, Fortaleza, CE, Brazil, September 20–22, 2017, 10 pages.
- Sato, D. T.; Corbucci, H.; Bravo, M. V. (2008) “Coding dojo: An environment for learning and sharing agile practices”. In Proceedings of the Agile 2008, AGILE ’08, pages 459–464, Washington, DC, USA. IEEE Computer Society.
- Scaico, P. D. (2018) “Um Estudo Sobre O Desenvolvimento De Interesse Pela Aprendizagem De Programação”. Tese de Doutorado. Universidade Federal de Pernambuco - CIN - Centro de Informática.
- Souza-Concílio, I. de A.; Corrêa, A. G. D.; Pacheco, B. de A. (2015) “Coding Dojo Como Metodologia De Ensino: A Visão Dos Educadores”. Direitos autorais 2015 Revista Educação & Tecnologia.
- Silva, I. F. A.; Silva, I. M. M.; Santos, M. S. (2009) “Análise de problemas e soluções aplicadas ao ensino de disciplinas introdutórias de programação”. Acesso em: 13 de julho de 2018.
- Yin, K. R. (2015) “Estudo de Caso: planejamento e métodos”. Tradução de Cristhian Matheus Herrera. 5o ed. Editora Bookman.