Blocos de Lego no Ensino da Metodologia Ágil Scrum em Cursos de Tecnologia da Informação através de um Jogo Iterativo

REVISTA ELETRÔNICA MULTIDISCIPLINAR
UNIFACEAR

Douglas Augusto Barcelos Bica[1]; Carlos Alexandre Gouvea da Silva[1]
[1] *Centro Universitário de Araucária (UNIFACEAR)*

RESUMO

O século 21 trouxe grandes avanços tecnológicos no cotidiano das pessoas. Além disso, ativos relacionados a tecnologias e sistemas de informação se tornaram imprescindíveis para a estratégia comercial e competitiva de qualquer empresa ou organização. Esse avanço cresce ainda mais com a constante entrega de aplicativos e sistemas computacionais para diferentes segmentos de mercado. O processo de entrega desses sistemas envolve desde a fase de concepção das ideias até a fase de implementação para o público interessado. Contudo, o desenvolvimento desses sistemas pode apresentar desafios aos programadores e gestores de projetos de Tecnologia da Informação (TI). Atualmente, as empresas de TI utilizam diversas abordagens na construção dos sistemas, em especial às relacionadas a desenvolvimento ágil, como por exemplo o Scrum ou Extreme Programming. Embora muitos profissionais de TI tenham conhecimento sobre os conceitos de desenvolvimento ágil, muitos não possuem práticas acadêmicas que aprimorem o conhecimento mais prático dessas abordagens quando vão para o mercado de trabalho. Desta forma, o objetivo deste artigo é apresentar os resultados da realização de uma atividade dinâmica utilizando blocos de Lego no ensino de Scrum com estudantes do curso Análise e Desenvolvimento de Sistemas e Sistemas de Informação em 2019 da UNIFACEAR. Ao final os alunos foram perguntados quanto a efetividade da dinâmica.

Palavras chave: Educação na Informática, Desenvolvimento Ágil, Scrum.

ABSTRACT

The 21st century brought great technological advances in people's daily lives. In addition, assets related to information technology and systems have become essential for the commercial and competitive strategy of any company or organization. This advancement grows even more with the constant delivery of applications and computer systems to different market segments. The delivery process of these systems involves from the design phase of the ideas to the implementation phase for the interested public. However, the development of these systems can present challenges to programmers and managers of Information Technology (IT) projects. Today, IT companies use a variety of approaches to building systems, especially those related to agile development, such as Scrum or Extreme Programming. Although many IT professionals are knowledgeable about agile development concepts, many do not have academic practices that enhance the more practical knowledge of these approaches when they go into the job market. In this way, the objective of this article is to present the results of the accomplishment of a dynamic activity using Lego blocks in the teaching of Scrum with students of the Analysis and Development of Systems and Information Systems course in 2019 of UNIFACEAR. In the end the students were asked about the effectiveness of the dynamics.

Key Words: Education on Informatic, Agile Developement, Scrum.

1. INTRODUÇÃO

O desenvolvimento de software a partir dos anos 60 começou a modificar os diferentes padrões de relacionamento entre o ser humano e as tecnologias computacionais inerentes a época. Essa interação acarretou no aumento pela procura e necessidade de ferramentas computacionais em diversos segmentos, gerando assim ao longo dos anos o aparecimento de processos de desenvolvimento de software convencionais como o cascata, o evolutivo, espiral e entre outros (SATO, 2007).

No mercado profissional de desenvolvimento de software uma das grandes necessidades é a formação de uma equipe de desenvolvimento que utilize diferentes técnicas e tecnologias, assim como as metodologias ágeis de desenvolvimento, na construção de sistemas computacionais. Dentre essas metodologias, o Scrum se destaca como uma das metodologias mais utilizadas e aplicadas nas empresas de software. Uma das grandes dificuldades em se obter profissionais com qualificação neste tipo de metodologia, estão nas instituições acadêmicas, que muitas das vezes não oferecem formação adequada quanto ao tipo e uso dessas metodologias ágeis.

Em cursos de formação voltados à Informática ou à Computação, a grande variedade de ferramentas e técnicas existentes permitem uma interação e imersão mais efetiva entre alunos, professores e as tecnologias da área. O desenvolvimento ágil é um conceito utilizado no mercado de trabalho, com objetivo de maximizar os resultados e na obtenção de melhores níveis de qualidade de produtos de software. Métodos ágeis, como o Scrum, vem sendo utilizados como um recurso didático para a prática e desenvolvimento do aprendizado acadêmico de alunos na realização de projetos e estudos relacionados (SILVA *et. al.*, 2016).

Para suprir e contornar esses desafios, professores e instituições tem utilizados diversas formas e ferramentas para o ensino de diversas disciplinas em sala de aulas. Em especial, o Scrum devido a sua grande utilização por profissionais em grupos de desenvolvimento, tem ganhado diversas maneiras de ser ensinado nas instituições de ensino, como por exemplo, através de jogos (GODOY e BARBOSA, 2010) (WANGENHEIM, 2013) (MAY *et. al.*, 2016), projetos reais de desenvolvimento de software (SCHARFF e VERMA, 2010) (SILVA *et. al.*, 2016), simuladores (LEE, 2016) (RODRIGUEZ *et. al.*, 2015), vídeo aulas, entre outros. Embora essas abordagens apresentem alternativas interessantes, o uso de outros recursos alternativos são estudados para o ensino de Scrum dentro de sala de aula.

O objetivo desse artigo é apresentar os resultados da aplicação de uma dinâmica acadêmica realizada com dois grupos de alunos do curso de Análise Desenvolvimento de Sistemas e Sistemas de Informação da UNIFACEAR. Os estudantes

participaram de um *workshop* referente a metodologia ágil Scrum e depois realizaram a construção de uma cidade utilizando peças de Lego. A construção da cidade foi realizada e envolvia todos os aspectos do Scrum. Ao final da dinâmica os alunos foram questionados quanto a efetividade da dinâmica no aspecto de ensino aprendizagem.

2. DESENVOLVIMENTO

2.1. Scrum

O método de desenvolvimento ágil Scrum é utilizado amplamente por equipes de desenvolvimento em todo o mundo com objetivo de maximizar as tarefas e a produtividade em um projeto de software com foco na qualidade de produtos de sistemas. O Scrum foi criado em 1993 por Ken Schwaber e Jeff Sutherland e constitui um conjunto de práticas e técnicas utilizadas por anos, sendo inicialmente apresentada em 1995 na OOPSLA (*Object-Oriented Programming, Systems, Languages Applications*) e aprimorada ao longo dos anos (LIMA, 2011). O OOPSLA é uma conferência de pesquisa realizada anualmente pela *Association for Computing Machinery* (ACM) e tem como principal objetivo a difusão de trabalhos que tratem de qualquer aspecto o desenvolvimento de software como: requisitos, modelagem, prototipagem, *design*, implementação, geração, análise, verificação, validação, teste, manutenção, reutilização, substituição e linguagens de programação (SILVA *et. al.*, 2016).

O nome Scrum surgiu da comparação entre desenvolvedores e os jogadores de Rugby. Scrum é a denominação da rápida reunião que ocorre quando os jogadores de Rugby iniciam uma jogada de realização do jogo (CARVALHO e MELO, 2012). A Figura 1 apresenta o modelo representativo do Scrum por etapas.

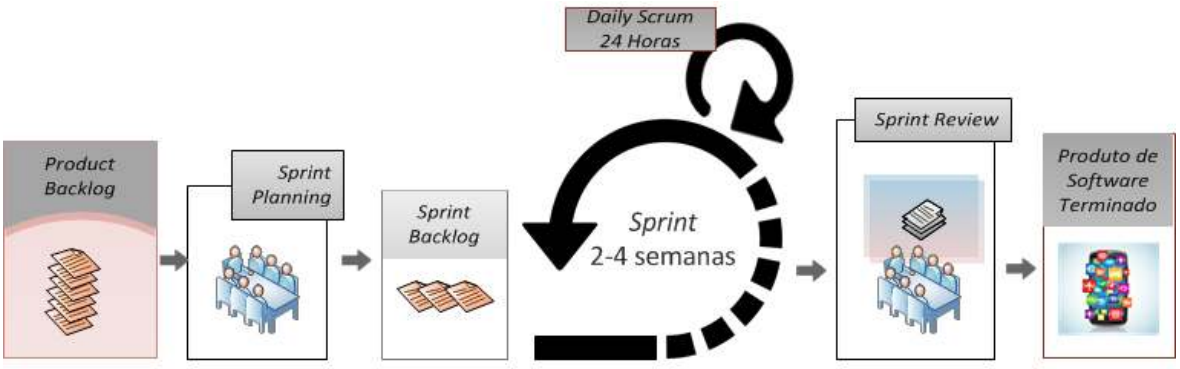


FIGURA 1: SEQUÊNCIA DE ATIVIDADES DO SCRUM
FONTE: OS AUTORES (2019)

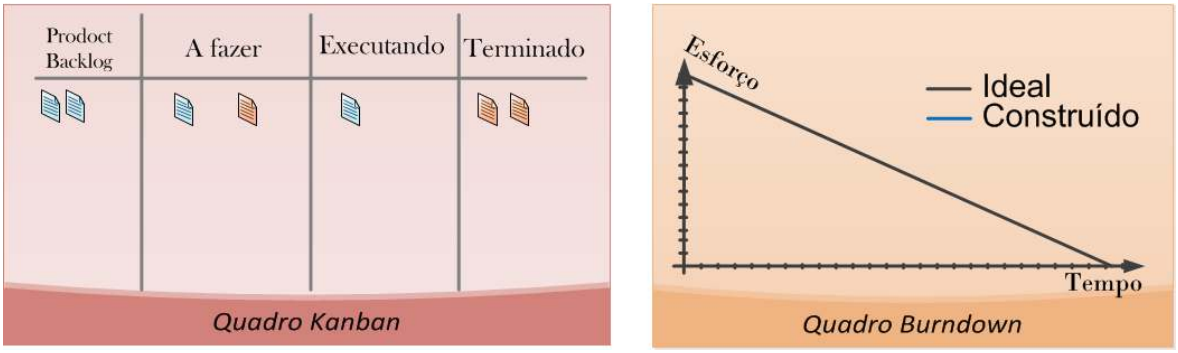
No Scrum, três funções de envolvidos são definidas: primeiro o *Product Owner* (PO), responsável pela aquisição de recursos para desenvolvimento do software e também é conhecedor das necessidades do cliente principal do software; segundo a Equipe de Desenvolvimento ou equipe Scrum, que é responsável por desenvolver o software desejado pelo PO; e por fim o *Scrum Master* (SM), responsável por orientar e ajudar todos os envolvidos, assim como eliminação de dificuldades e impedimentos (SILVA *et. al.*, 2016).

Primeiramente o PO gera uma lista chamada de *Product Backlog* das necessidades do software que necessita ser desenvolvido. Essa lista é criada e mantida pelo PO e sempre que necessário ela pode ser atualizada com a inserção ou retirada de necessidades. Dentro da metodologia, essas necessidade são chamadas de Histórias de Usuários (US, *user stories*), ou no geral também é conhecido como requisito de software. Cada US possui uma prioridade baseada na necessidade do PO, sendo primordial a execução das atividades com maior prioridade primeiro (SILVA *et. al.*, 2016).

Após definido o *Product Backlog* e suas prioridades é realizado uma reunião chamada *Sprint Planning*. Nessa reunião o PO indica para a equipe de desenvolvimento quais as atividades mais importantes que ele julga que devem ser inicialmente desenvolvidas. Nesta etapa, a equipe de desenvolvimento associa a cada US um número chamado de "esforço". Esse número representa o quanto a equipe entende ser difícil ou complexo o desenvolvimento daquela US e geralmente é mensurada a partir de uma sequência de Fibonacci com números 1, 2, 3, 5 e 8, contudo outras sequencias de Fibonacci também são aceitas (TAMRAKAR e JØRGENSEN, 2012). Essa estimativa é importante, pois ajuda a equipe mensurar exatamente o nível e quantidade de trabalho necessário para executar uma determinada US. Na prática, uma US que a equipe entende que não conseguirá executar não poderá ser aceita. Esse mesmo esforço mensurado pela equipe servirá como índice de desempenho da equipe durante o desenvolvimento de todo o software através da atualização do quadro de *Burndown* futuramente (SILVA *et. al.*, 2016).

A lista de US aceitas pela equipe é chamada de *Sprint Backlog* e reflete um acordo entre o PO e equipe. Após definido o *Sprint Backlog*, o mesmo não poderá ser alterado ou modificado. A execução das US do *Sprint Backlog* são executados dentro de um ciclo chamado de *Sprint.* Um *Sprint* deve ocorrer entre 2 a 4 semanas. Durante o *Sprint* a equipe realiza reuniões diárias com o SM para verificar o andamento das atividades. Essas reuniões diárias são chamadas de *Daily Scrum* e ocorrem em períodos curtos de 15 minutos no qual é atualizado o *Kanban* e o *Burndown* (SILVA et. al., 2019). O *Kanban* é um quadro que mostra o estado atual das atividades e o *Burndown*

apresenta o desempenho esperado da equipe, como mostrado respectivamente nas Figuras 2A e 2B.



(A) (B)

FIGURA 2: MODELO DE QUADROS UTILIZADOS NO SCRUM: (A) KANBAN E (B) BURNDOWN. FONTE: OS AUTORES (2019)

Finalizado o *Sprint* é realizado uma reunião de verificação e validação das US realizadas. Essa reunião é chamada de *Sprint Review* e conta com a presença do PO, de um ou mais representantes da equipe de desenvolvimento e o SM. Nesta reunião a equipe apresenta o que foi desenvolvido. O PO avalia as US que foram entregues e define como aceito, rejeitado ou necessita ajustes. A US aceitas são incorporados ao sistema final de forma que o projeto de software possa ser desenvolvido de forma incremental e iterativa devido a presença constante do PO. Depois disso, a equipe se reúne para uma autoavaliação do que foi feito corretamente e o que pode ser melhorado para futuros *Sprints.* Em seguida, é realizado um novo *Sprint Planning* no qual o PO e a equipe definem outras US pendentes para serem realizadas, seguido da execução de um novo *Sprint.* Esse processo ocorre várias vezes até que todas as US do *Product Backlog* sejam finalizados.

2.2. Ensino de Scrum

O processo de aprendizagem de alunos pode ser afetado por um conjunto de elementos, incluindo professores e alunos não motivados, tipo de ferramentas disponíveis usadas e métodos aplicados nas aulas. O uso de atividades acadêmicas práticas é uma alternativa interessante no processo de aprendizagem baseado no Scrum. Nos ambientes universitários, o aluno aprende a desenvolver habilidades e atitudes usando atividades práticas (SILVA, SANTOS e PELACINI, 2018). Nesta seção, mostramos uma breve visão geral de alguns trabalhos que usaram diferentes métodos para ensinar Scrum em sala de aula.

Lee (2016) sugeriu usar um software chamado Scrum-X que ajuda os alunos a reforçar sua compreensão dos conceitos discutidos na metodologia Scrum. Scrum-X é uma plataforma de simulação desenvolvida para melhorar os conceitos da metodologia Scrum através de exercícios experimentais. O impacto da aprendizagem nos alunos foi realizado com um projeto piloto com 15 alunos. No final do experimento, um questionário foi realizado para avaliar a eficácia do aprendizado do Scrum. Os resultados apontam para um potencial positivo na contribuição para o ensino, em que os autores reconhecem o impacto positivo na motivação para aprender e, ao mesmo tempo, proporcionam um engajamento dos alunos.

Kropp *et. al*., (2014) sugeriram uma atividade baseada no jogo Scrum Lego em que os alunos têm que construir uma cidade. A construção da cidade foi realizada em 4 *Sprints* de 25 minutos cada. Os autores observaram que as equipes começaram com um primeiro *Sprint* fraco, mas ao longo do experimento eles atingiram o objetivo estimado. Os resultados mostraram que os alunos gostaram da abordagem prática e apreciaram a intensa comunicação e colaboração dos membros da equipe. Por outro, Steghofer *et. al*., (2017) apresentaram resultados sobre mais de 450 alunos participantes que participaram de *workshops* Scrum construindo uma cidade Lego em múltiplos *Sprints* curtos.

Baseado também em blocos de Lego, Paasivaara *et. al*., (2014) simulou o Scrum como um jogo com alunos de uma Universidade na Finlândia. Durante o jogo, as equipes de alunos aprendem os papéis, etapas, eventos e conceitos do Scrum na prática, simulando vários *Sprints* de desenvolvimento, incluindo o planejamento e a criação de um produto de blocos de Lego. Os resultados mostram que os alunos aprendem rapidamente e estavam satisfeitos com o jogo.

2.3. Planejamento da Dinâmica

A realização da dinâmica acadêmica foi realizada com duas turmas distintas. A primeira foi realizada em março de 2019 com uma turma de 14 alunos do sétimo período do curso de Sistemas de Informação. A segunda dinâmica foi realizada no mês de junho com 27 alunos do primeiro ou sexto período dos cursos de Tecnologia em Análise e Desenvolvimento de Sistemas e Bacharelado em Sistemas de Informação. Nestas duas dinâmicas os alunos foram distribuídos em grupos de trabalho de 4 a 6 alunos cada um.

Inicialmente os estudantes foram apresentados ao mesmo *Product Backlog,* formado por 18 elementos (US) que deveriam ser criados de forma a criar uma cidade completa utilizando peças de Lego. Ao todo foi disponibilizado duas mil peças de

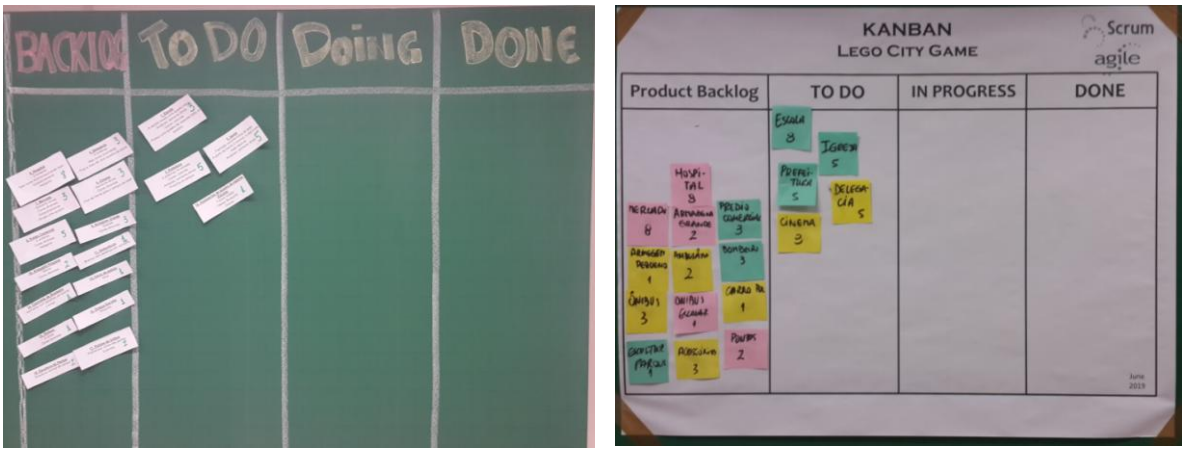
diferentes tamanho e cores. A lista com os elementos propostos e suas respectivas prioridades são apresentados na Tabela 1.

TABELA 1 – PRODUCT BACKLOG COM AS USER STORIES A SEREM DESENSOLVIDAS

ITEM	PRIORIDADE	ITEM	PRIORIDADE
Escola	3	Armazém pequeno	1
Igreja	3	Ambulância	2
Prefeitura	3	Caminhão bombeiros	2
Delegacia	3	Carros de policia	2
Hospital	3	Ônibus	1
Cinema	3	Ônibus escolar	2
Mercado	2	Esculturas parque	1
Prédio comercial	2	Pontos de ônibus	1
Armazém grande	2	Acessórios quadra	1

FONTE: OS AUTORES (2019)

Na primeira turma foi utilizado um quadro negro para criação e demonstração do *Kanban*, como mostrado na Figura 3A. Contudo, na segunda turma foi criado um *Kanban* impresso no qual ficou mais fácil a manipulação da US do *Product Backlog,* como mostrado na Figura 3B. Além disso, foi utilizado recursos de *post-it* coloridos e autoadesivos, o que geralmente é feito em ambientes de criação e desenvolvimento de software que utilizam o Scrum.

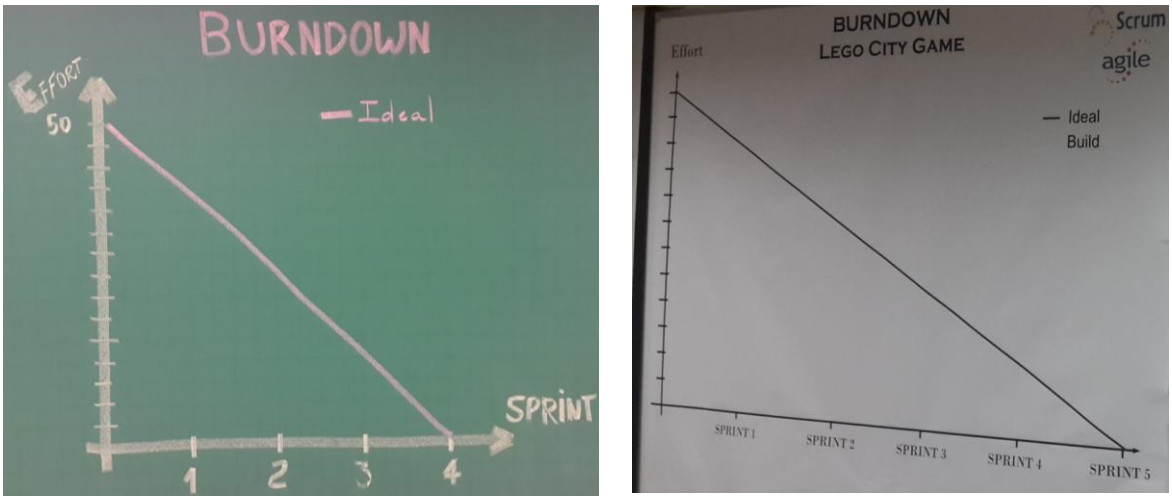


(A) (B)

FIGURA 3: KANBANS UTILIZADOS NAS DINÂMICAS: (A) TURMA 1 (B) TURMA 2.
FONTE: OS AUTORES (2019)

Após apresentado o *Product Backlog* os alunos realizaram o *Planning Pocker* para estimação do esforço necessário para construção de cada elemento da cidade. O

esforço foi estimado a partir da complexidade de cada elemento, onde deveriam considerar diferentes características como: tamanho, cores, quantidade máxima de peças, forma, modelo e demais requisitos definidos em cada elemento. Após a definição do esforço necessário para desenvolver cada elemento da cidade, foi disponibilizado o *Burndown* que seria utilizado pelos alunos para acompanhar o desempenho dos projetos, como mostrados nas Figuras 4A e 4B.



(A) (B)

FIGURA 4: BURNDOWS UTILIZADOS NAS DINÂMICAS: (A) TURMA 1 (B) TURMA 2.
FONTE: OS AUTORES (2019)

No *Sprint Planning* os alunos definiram quais elementos cada time poderia desenvolver no primeiro *Sprint.* Após definido quais elementos deveriam ser desenvolvidos, os alunos tiveram um tempo cronometrado para construção de cada elemento utilizando as peças de Lego. Ao final do *Sprint* o PO realizou o *Sprint Review* com as equipes de forma a avaliar o que foi desenvolvido. Em seguida, um novo *Sprint Planning* foi planejado para o segundo *Sprint.* Esse processo se repetiu até que todos os elementos fossem desenvolvido e assim finalizasse a dinâmica.

2.4. Avaliação da Dinâmica

Ao final das dinâmicas, os alunos foram convidados a responder um questionário online com o objetivo de avaliar a efetividade da dinâmica. As perguntas e questionamentos realizados são indicados abaixo:

- Em uma escala, você CONCORDA que a dinâmica Lego City permitiu um entendimento maior do SCRUM quando comparado se fosse apenas uma aula teórica? Escalas: 1 - Discordo totalmente até 5 - Concordo totalmente.

- Das alternativas abaixo, assinale a MELHOR opção para um estudante que nunca ouviu falar de SCRUM aprender/entender mais rapidamente a metodologia. Foram apresentados 4 alternativas - Assistindo uma aula teórica - Lendo um artigo técnico sobre Scrum - Realizando uma atividade prática acadêmica ou profissional - ou Assistindo um vídeo.

3. RESULTADOS

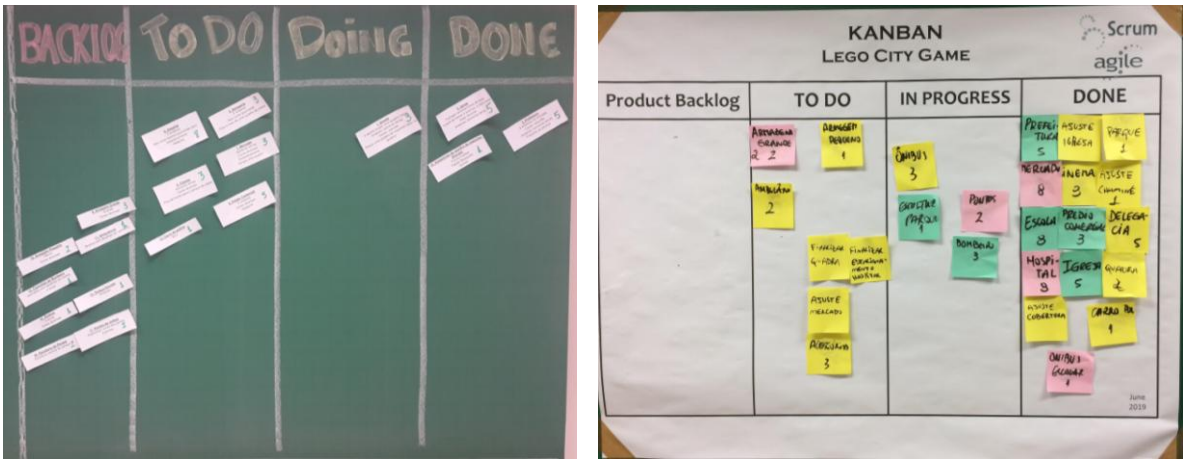
A linha do tempo de execução das diversas etapas da dinâmica com as duas turmas são apresentadas na Tabela 2. Como é possível identificar os alunos tiveram pouco tempo para executar as atividades. Esses tempos curtos tiveram como principal objetivo apresentar aos alunos um cenário simulando um ambiente real com pouco tempo para execução de atividades e com alta cobrança de gestores para finalização do *Product Backlog*. Ainda, pode-se verificar que o tempo de execução de *Sprints* é diminuído ao fato de que os alunos se empenharam ao máximo na execução das atividades, o que consequentemente fez com que algumas atividades fossem finalizadas rapidamente.

TABELA 2 – TEMPO DAS ATIVIDADEDES EXECUTADAS NAS DINÂMICAS

ATIVIDADE REALIZADA	TEMPO EM MINUTOS
Apresentação dos objetivos da dinâmica aos alunos	10
Apresentação do *Product Backlog* e PO aos alunos	15
Definição do esforço das atividades	15
Spring Planning 1	5
Sprint 1	15
Sprint Retrospective 1	5
Spring Planning 2	5
Sprint 2	15
Sprint Retrospective 2	5
Spring Planning 3	3
Sprint 3	8
Sprint Retrospective 3	2
Spring Planning 4	3
Sprint 4	5
Sprint Retrospective 4	2
Encerramento da dinâmica	10

FONTE: OS AUTORES (2019)

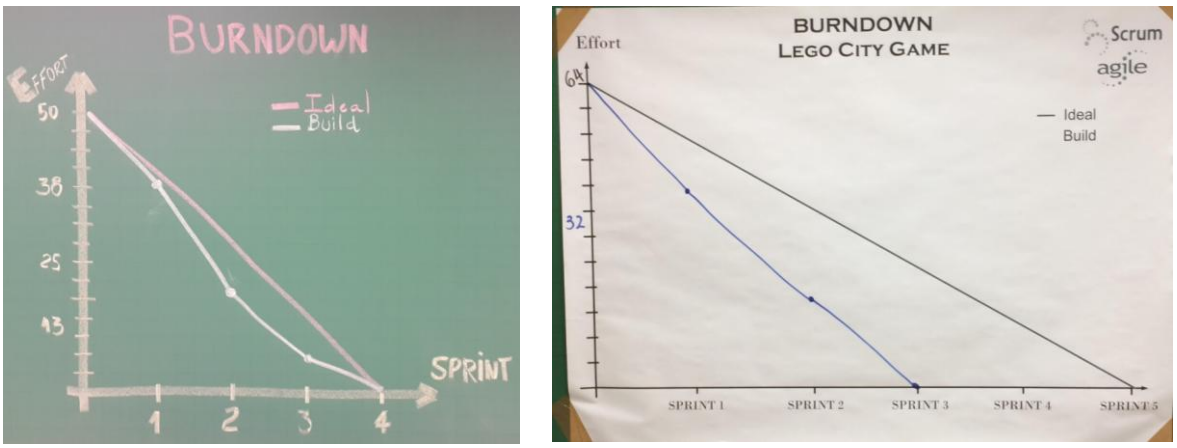
Os quadros de *Kanban* e *Burndown* apresentaram grande vantagem no acompanhamento das atividades realizadas nas dinâmicas. A Figura 5 apresenta a atualização do Kanban, no qual os alunos puderam praticar uma atividade amplamente realizada em ambientes reais de desenvolvimento de software.



(A) (B)

FIGURA 5: KANBAN ATUALIZADOS NAS DINÂMICAS: (A) TURMA 1 (B) TURMA 2.
FONTE: OS AUTORES (2019)

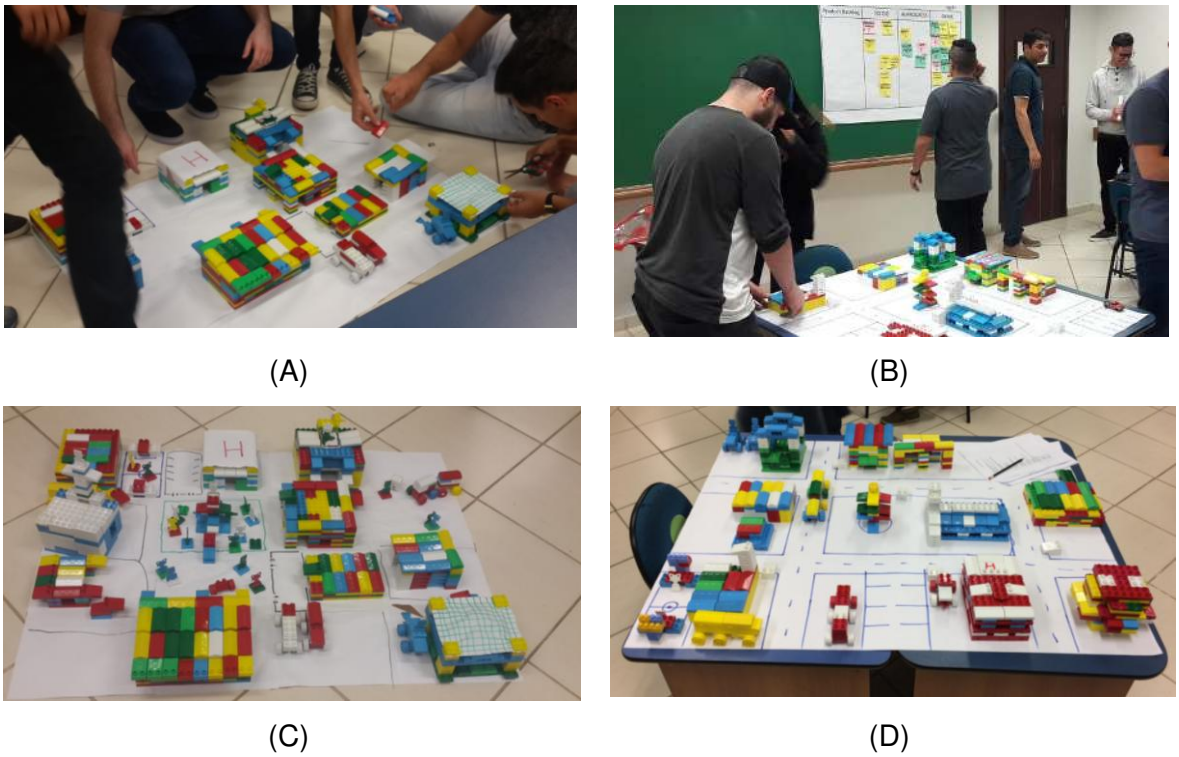
Na Figura 6 o *Burndown* com o desempenho dos alunos é mostrado, onde é possível ver um desempenho adequado quando comparado com a linha do ideal. Uma melhoria significativa entre as dinâmicas foi em relação a segunda turma, que realizou o desenvolvimento de todo o projeto com apenas 3 *Sprints*. Podemos relacionar essa melhoria do fato de que a quantidade de alunos era maior na segunda turma.



(A) (B)

FIGURA 6: BURNDOWN ATUALIZADOS NAS DINÂMICAS: (A) TURMA 1 (B) TURMA 2.
FONTE: OS AUTORES (2019)

A execução das atividades é mostrada nas Figuras 7. A execução das atividades resultaram na criação de duas cidades utilizando peças de Lego a partir da execução de execuções iterativas e incrementais. Essa abordagem permitiu que os alunos presenciassem e participassem de uma dinâmica acadêmica com o objetivo de aprender de uma maneira lúdica as diversas etapas e atividades do Scrum. A dinâmica fez com que os estudantes experimentassem situações similares ao que ocorrerá em ambientes profissionais reais.



(A) (B)

(C) (D)

FIGURA 7: RESULTADOS FINAIS DAS DINÂMICAS: (A-C) TURMA 1 (B-D) TURMA 2.
FONTE: OS AUTORES (2019)

Os alunos foram questionados quanto a efetividade das dinâmicas no aprendizado do Scrum em ambiente acadêmico. Na primeira pergunta eles foram questionados quanto a efetividade das dinâmicas se comparado a aulas tradicionais em sala de aula. Os resultados mostraram que 78,7% dos alunos concordam que a dinâmica em uma escala de 1 à 5 (1 - menos efetiva até 5 - muito efetiva) é muito efetiva no aprendizado do Scrum. Outros 14,9% consideram uma escala 4 e outros 6,4% consideram uma escala de 3. Nenhum dos alunos indicaram escalas de 1 ou 2.

A Figura 8 apresenta o resultado quanto as diferentes opções que um estudante pode aprender ou entender o Scrum. Para 91,5% dos alunos, realizar uma atividade prática acadêmica ou profissional é a opção mais rápida em se aprender o Scrum. Para

4,3% dos alunos esse aprendizado pode ser mais efetivo assistindo vídeos sobre o assunto, assim como 4,3% que responderam que assistir uma aula teórica pode ser mais efetiva. Não houve respostas para ler um artigo técnico sobre o Scrum.

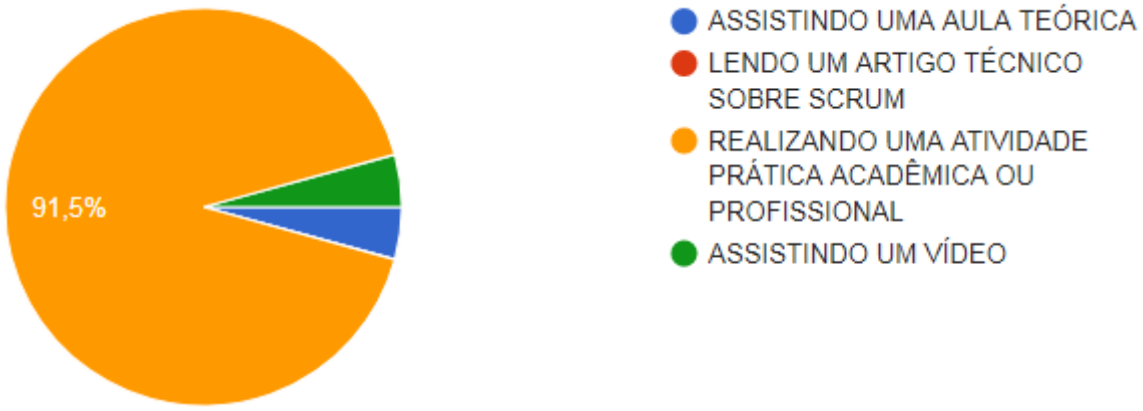


FIGURA 8: RESULTADOS DA SEGUNDA QUESTÃO.
FONTE: OS AUTORES (2019)

4. CONCLUSÃO

O presente artigo teve como principal objetivo apresentar os resultados da aplicação de uma dinâmica acadêmica com alunos de dois cursos de desenvolvimento de softwares. Durante as dinâmicas os alunos realizaram o desenvolvimento de uma cidade utilizando peças de Lego a partir das etapas e atividades da metodologia ágil Scrum. A finalidade foi apresentar uma maneira lúdica e de fácil aprendizado o funcionamento do Scrum em ambiente acadêmico. Os resultados das criações foram satisfatórios e alcançaram a estimativa inicial. Para a maioria dos alunos, as dinâmicas foram produtivas e são mais efetivas no ensino quando comparadas a métodos tradicionais.

5. REFERÊNCIAS

SATO, D. T. Uso eficaz de métricas em métodos ágeis de desenvolvimento de software. **Instituto de Matemática e Estatística**, Universidade de São Paulo, São Paulo, v. 139, 2007.

SILVA, C. A. G.; SANTOS, E. L.; ANGELO, L. M.; OLIVEIRA, M. A. C. S.; MORAES, R. V. *A utilização do SCRUM como recurso educacional no processo de aprendizagem em Engenharia de Software.* **International Journal on Alive Engineering Education**, v. 3, n. 2, p. 87-102, 2016.

GODOY, A; BARBOSA, E. F. Game-Scrum: *An approach to agile game development*. **Proceedings of SBGames,** p. 292-295, 2010.

WANGENHEIM, C. G. V.; SAVI, R.; BORGATTO, A. F. *SCRUMIA—An educational game for teaching SCRUM in computing courses*. **Journal of Systems and Software**, v. 86, n. 10, p. 2675-2687, 2013.

MAY, J.; YORK, J.; LENDING, D. *Teaching tip: Play ball: Bringing scrum into the classroom.* **Journal of Information Systems Education**, v. 27, n. 2, p. 87, 2016.

SCHARFF, C.; VERMA, R. Scrum to support mobile application development projects in a just-in-time learning context. In: **Proceedings of the 2010 ICSE workshop on cooperative and human aspects of software engineering**. ACM, 2010. p. 25-31.

LEE, W. L. SCRUM-X: *An interactive and experiential learning platform for teaching scrum*. 2016.

RODRIGUEZ, G.; SORIA, Á.; CAMPO, M. Virtual Scrum: *A teaching aid to introduce undergraduate software engineering students to scrum*. **Computer Applications in Engineering Education**, v. 23, n. 1, p. 147-156, 2015.

LIMA, W. V.. Scrum no Brasil. 2011. **Dissertação de Especialização em Engenharia de Projeto de Software**, Universidade do Sul de Santa Catarina, 27 de Junho de 2011, Florianópolis, SC.

TAMRAKAR, Ritesh; JØRGENSEN, Magne. *Does the use of Fibonacci numbers in Planning Poker affect effort estimates?*. In: **16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012)**. 2012.

CARVALHO, B. V.; MELLO, C. H. P.. Aplicação do método ágil Scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica. **Gestão & Produção,** v. 19, n. 3, p. 557-573, 2012.

SILVA, C. A. G.; SANTOS, E. L.; PELACINI, D. A. F. *Evaluation of Academic Experience in Learning Education over Simulators Softwares*. **International Journal on Alive Engineering Education**, v. 5, n. 2, p. 23-40, 2018.

KROPP, M.; MEIER, A.; MATEESCU, M.; ZAHN, C. *Teaching and learning agile collaboration*. In: **2014 IEEE 27th conference on software engineering education and training**. IEEE, 2014. p. 139-148.

STEGHÖFER, J.; BURDEN, H.; ALAHYARI, H.; HANEBERG, D. *No silver brick: Opportunities and limitations of teaching scrum with lego workshops*. **Journal of Systems and Software**, v. 131, p. 230-247, 2017.

PAASIVAARA, M.; HEIKKILÄ, V.; LASSENIUS, C.; TOIVOLA, T. *Teaching students scrum using LEGO blocks*. In: **Companion Proceedings of the 36th International Conference on Software Engineering**. ACM, 2014. p. 382-391.