



**UNIVERSIDADE FEDERAL DA GRANDE DOURADOS  
FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA  
TRABALHO DE CONCLUSÃO DE CURSO  
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

**PBL-SCRUM - UMA APLICAÇÃO MÓVEL PARA APOIO AO ENSINO  
DE SCRUM VIA APRENDIZAGEM BASEADA EM PROBLEMAS**

**CAIO FERNANDES LIMA**

**DOURADOS-MS  
2021**



**UNIVERSIDADE FEDERAL DA GRANDE DOURADOS  
FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA  
TRABALHO DE CONCLUSÃO DE CURSO  
CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**PBL-SCRUM - UMA APLICAÇÃO MÓVEL PARA APOIO AO ENSINO  
DE SCRUM VIA APRENDIZAGEM BASEADA EM PROBLEMAS**

**CAIO FERNANDES LIMA**  
**Bacharelado em Engenharia de Computação**

Trabalho de conclusão de curso apresentado ao Curso de Engenharia de Computação da Faculdade de Ciências Exatas e Tecnologia da Universidade Federal da Grande Dourados, como pré-requisito à obtenção do título de Bacharel em Engenharia de Computação.

Área de concentração: Sistemas de Informação.

**Orientador: Prof<sup>a</sup>. Dr<sup>a</sup>. Silvana Morita Melo**

**DOURADOS-MS  
2021**



MINISTÉRIO DA EDUCAÇÃO  
FUNDAÇÃO UNIVERSIDADE FEDERAL DA GRANDE DOURADOS

### AVALIAÇÃO DE PROJETO DE TCC OU TCC

**TÍTULO DO TRABALHO:** PBL-SCRUM - Uma Aplicação Móvel para Apoio ao Ensino de SCRUM via Aprendizagem Baseada em Problemas

**NOME DO(S) ESTUDANTE(S):** Caio Fernandes Lima

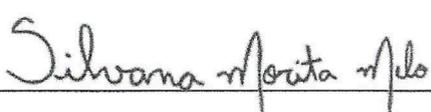
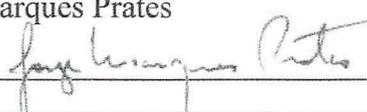
**ORIENTADOR(A):** Silvana Morita Melo

**CO-ORIENTADOR(A):** \_\_\_\_\_

**DATA DA APRESENTAÇÃO:** 10/05/2021

**HORÁRIO:** 14:00 horas

#### AVALIAÇÃO DA COMISSÃO EXAMINADORA:

<input type="checkbox"/> Trabalho de Conclusão de Curso I	<input checked="" type="checkbox"/> Trabalho de Conclusão de Curso II
Presidente: Silvana Morita Melo Assinatura: _____	
Membro 1: Jorge Marques Prates Assinatura: _____	
Membro 2: Anderson Bessa da Costa Assinatura: _____	
Resultado:	<input checked="" type="checkbox"/> Aprovado <input type="checkbox"/> Aprovado condicionado a modificações (especificar no campo observações) <input type="checkbox"/> Reprovado
<b>Observações<sup>1</sup>:</b>	
Eu, Caio Fernandes Lima, Orientador Silvana Morita Melo, e Coordenador do Curso Marcos Paulo Moro, declaramos para os devidos fins que estamos de acordo com a defesa do Trabalho de Conclusão de Curso, seja no formato remoto (via Google Meet), e que a mesma será gravada.	



# PBL-SCRUM - Uma Aplicação Móvel para Apoio ao Ensino de SCRUM via Aprendizagem Baseada em Problemas

Caio F. Lima<sup>1</sup>, Silvana M. Melo<sup>1</sup>

<sup>1</sup>Faculdade de Ciências Exatas e Tecnológicas (FACET)

Universidade Federal da Grande Dourados (UFGD)

Caixa Postal 364 – 79.804-97015.064 – Dourados – MS – Brasil

[caio.lima424@academico.ufgd.edu.br](mailto:caio.lima424@academico.ufgd.edu.br), [silvanamelo@ufgd.edu.br](mailto:silvanamelo@ufgd.edu.br)

**Abstract.** *The Scrum framework has been widely used in the software industry, among other advantages for allowing frequent deliveries of functional software with a return on the client's investment and reducing project risks. However, recent research has shown that computer graduates leave universities without adequate knowledge of development and management cycles, such as Scrum. Some proposed alternatives to address this problem are the use of student-centered teaching approaches and tools to support teaching that implement these approaches. In this sense, this article describes the current state of development of a mobile tool to support the teaching of Scrum via Problem Based Learning. The main objective of the research is to provide a tool that efficiently automates the teaching of agile software development methodologies through an active teaching approach, helping to cover this topic within computer courses.*

**Resumo.** *O framework Scrum tem sido amplamente utilizado na indústria de software, dentre outras vantagens por permitir entregas frequentes de software funcional com retorno ao investimento do cliente e redução dos riscos do projeto. Entretanto, pesquisas recentes têm demonstrado que egressos dos cursos de computação deixam as universidades sem o conhecimento adequado de ciclos de desenvolvimento e gerenciamento, como o Scrum. Algumas alternativas propostas para enfrentar esse problema são a utilização de abordagens de ensino centradas no aluno e ferramentas de apoio ao ensino que implementem essas abordagens. Nesse sentido, este artigo descreve o atual estado de desenvolvimento de uma ferramenta mobile para apoio ao ensino de Scrum via Aprendizagem Baseada em Problemas. O Principal objetivo da pesquisa é fornecer uma ferramenta que automatize de maneira eficiente o ensino de metodologias ágeis de desenvolvimento de software por meio de uma abordagem ativa de ensino auxiliando a cobertura desse tópico dentro dos cursos de computação.*

## 1. Introdução

Nos dias de hoje, os softwares desempenham papel fundamental dentro das companhias, estando intimamente ligado às suas operações de negócios. Em um ambiente global de mudanças rápidas, o desenvolvimento rápido de software tornou-se essencial às companhias em ordem destas obterem vantagens competitivas e acesso a oportunidades de mercado [Sommerville 2011].

Dessa necessidade de processos que permitissem o desenvolvimento rápido de software surgiram os métodos ágeis de desenvolvimento. Dentre eles pode-se destacar o

Scrum, que por possuir uma abordagem iterativa e incremental que permite entregas frequentes de software funcional e retorno ao investimento do cliente, tem sido apontado como um dos métodos mais utilizados no gerenciamento de projetos de software pela indústria [Fonseca and Campos 2008, Sabbagh 2014].

Embora as metodologia ágeis, como o Scrum, façam parte da ementa das disciplinas de Engenharia de Software, empresas de software afirmam que os egressos de cursos da área da computação possuem conhecimentos técnicos e teóricos nessa área, mas não habilidades práticas suficientes para sua aplicação, faltando, por exemplo, habilidades de comunicação e trabalho em equipe e experiência ou vivência de ciclos de desenvolvimento de software [Castro et al. 2017, Mills et al. 2003]. Uma das principais causas é a utilização de abordagens de ensino centradas no professor nas disciplinas de Engenharia de Software. A abordagem tradicional ainda continua sendo a mais utilizada pelos professores. Neste sentido, o uso de ferramentas de apoio ao ensino e abordagens de ensino-aprendizagem centradas nos alunos, como a *Problem Based Learning* (PBL), têm sido discutidas para enfrentar essa situação [Bessa et al. 2012].

Neste contexto, este trabalho tem como objetivo oferecer uma ferramenta de apoio à abordagem de aprendizagem baseada em problemas (PBL) para o ensino do Scrum. Contribuindo, não somente para a melhora no aprendizado dessa metodologia de desenvolvimento ágil em disciplinas de Engenharia de Software, mas também para ampliar a formação do aluno dos cursos da área da computação. Além de auxiliar o professor na cobertura deste tópico.

Para o desenvolvimento do trabalho foram realizadas revisões de literatura a fim de obter uma visão da área de ensino de metodologias ágeis. Algumas ferramentas foram encontradas e são descritas em detalhes na seção de trabalhos relacionados. Buscando alinhar-se às metodologias ativas de ensino, que têm sido amplamente citadas como vantajosas ao ensino de Engenharia de Software, a abordagem PBL foi o foco deste trabalho. As tecnologias utilizadas na construção da ferramenta são apresentadas e um exemplo prático é discutido a fim de demonstrar a aplicação prática da ferramenta proposta.

O restante do artigo está organizado como segue. Na Seção 2 são abordados os principais conceitos relacionados ao trabalho, que envolvem: métodos ágeis, Scrum, PBL e ferramentas de ensino. Na Seção 3 são listados alguns trabalhos relacionados. Na Seção 4 é descrita a ferramenta proposta, sua arquitetura, os principais módulos que a compõem, bem como as tecnologias utilizadas para seu desenvolvimento. Na Seção 5 é apresentado um exemplo de aplicação. Na Seção 6 discutem-se os resultados obtidos. Na Seção 7 são listadas alternativas à trabalhos futuros. Por fim, na Seção 8 são discutidas as principais conclusões obtidas com a condução desta pesquisa.

## **2. Métodos Ágeis**

Em um ambiente global de rápidas mudanças tecnológicas e comerciais, desenvolver software rapidamente e com qualidade tornou-se essencial às organizações em ordem destas obterem vantagens competitivas [Sommerville 2011].

Enquanto os métodos tradicionais de desenvolvimento seguem o princípio de que é necessário requisitos bem compreendidos e pouco prováveis de serem alterados durante o desenvolvimento do sistema, porém, este cenário é pouco provável de encontrar-se numa empresa inserida num ambiente de mudanças rápidas em que o requisitos do sistema serão clareados a medida que entrega-se o software, que os usuários obtêm experiência e de acordo com *feedback* dos clientes [Sommerville 2011].

Por volta da década de 90, os métodos tradicionais baseados em extensas especificações de requisitos e documentação abrangente já não se adequavam ao tipo de desenvolvimento rápido de software, desse modo, desenvolvedores e consultores passaram a experimentar novas práticas de desenvolvimento e novas formas de trabalho e organização. O resultado dessa experimentação foi o surgimento dos métodos ágeis, os quais são baseadas em abordagem incremental, com entrega frequente de software e colaboração próxima entre times auto-organizáveis e *stakeholders* de negócio [Agile Alliance 2015].

## 2.1. Scrum

O Scrum é um framework ágil para o gerenciamento de projetos de software, o qual utiliza uma abordagem iterativa e incremental que possibilita mudanças no decorrer do projeto e entregas de valor frequente, reduzindo os riscos do projeto, sendo, atualmente, uma das abordagens mais utilizadas no gerenciamento de projetos de software [Fonseca and Campos 2008, Sabbagh 2014, Sutherland, J. e Schwaber, K. 2018].

O processo do Scrum [Sutherland, J. e Schwaber, K. 2018] é ilustrado na Figura 1. No início de um projeto, a força de trabalho envolvida, chamada de Time Scrum, é dividida em papéis que os ajudam a entender funções e deveres. O Scrum descreve 3 papéis: i) *Scrum Master*: membro da equipe que atua como facilitador e garante que os princípios do Scrum sejam entendidos por todos; ii) *Product Owner*: membro da equipe que gerencia uma lista de itens, chamada *Product Backlog*, a qual representa o que ele acha que deve ser produzido ao longo do projeto; iii) *Time de Desenvolvimento*: uma equipe multidisciplinar e auto-organizável que é responsável pela entrega de incrementos.

O Scrum promove o desenvolvimento em *Sprints*, que são *timebox* nas quais um incremento funcional e entregável é desenvolvido. No início da *Sprint* ocorre a *Sprint Planning*, uma reunião na qual o *Product Owner* negocia com o Time de Desenvolvimento o objetivo que a *Sprint* deve realizar e os itens de *Product Backlog* que, se completados na *Sprint*, atingirão o objetivo da *Sprint*; os itens escolhidos são movidos do *Product Backlog* para o *Sprint Backlog*. A principal função do *Sprint Backlog* é tornar visível o trabalho em andamento, por isso, é comum utilizar-se um quadro *Kanban* [Sabbagh 2014].

Além da *Sprint Planning*, outros eventos ocorrem dentro da *Sprint*. O *Daily Scrum* é uma reunião diária curta de no máximo 15 minutos, cujo objetivo é garantir visibilidade sobre o trabalho em andamento; A *Sprint Review* é realizada no final da *Sprint* para inspecionar o incremento e adaptar o *Product Backlog* se necessário; A *Sprint Retrospective* também ocorre ao fim da *Sprint* e permite ao time *Scrum*

inspecionar a si próprio para criar um plano de melhorias a serem aplicadas na próxima *Sprint* [Sutherland and Schwaber 2018].

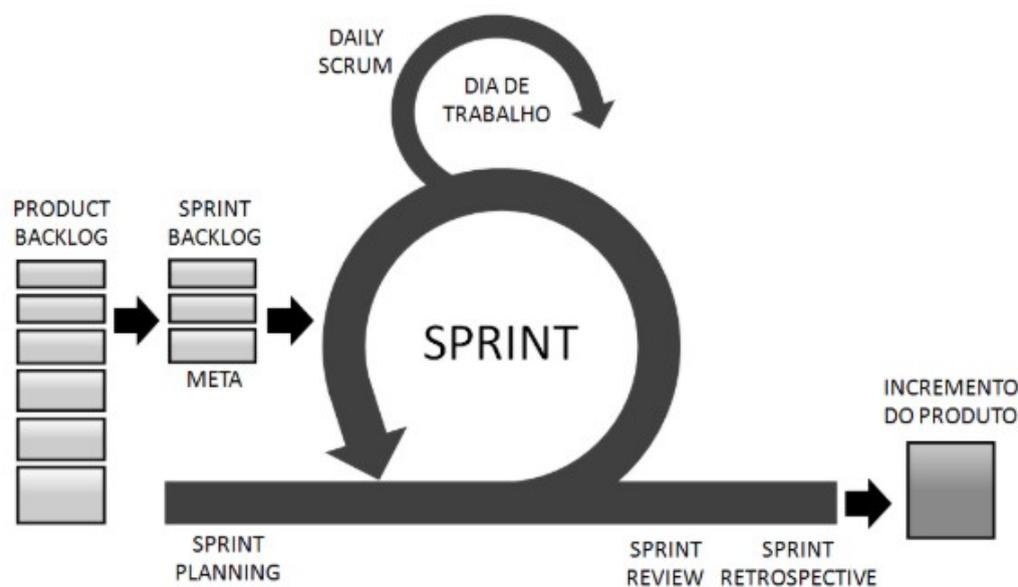


Figura 1. O ciclo Scrum adaptado de [Sabbagh 2014].

## 2.2. Ensino de Métodos Ágeis

Em geral, as metodologia ágeis e os frameworks de desenvolvimento, como o Scrum, são ensinados durante as aulas de Engenharia de Software (ES), porém, alguns estudos atuais apontam que o método tradicional de ensino, preponderantemente adotado no ensino de ES, carece em atingir tal objetivo [Castro et al. 2017]. As empresas relatam que os graduados não possuem experiência de ciclos de desenvolvimento e carecem de habilidades de comunicação e trabalho em equipe [Meireles and Bonifácio 2015, Mills et al. 2003].

Para enfrentar tal situação, abordagens de ensino-aprendizagem centradas no aluno, como o PBL, e ferramentas de apoio ao ensino têm sido estudadas e discutidas na literatura [Ribeiro and Ribeiro 2020, Bessa et al. 2012, Macedo et al. 2019, Gramigna, 1994].

Segundo Ribeiro and Ribeiro. (2020), o PBL (*Problem-Based Learning*) ou aprendizagem baseada em problemas é uma metodologia de ensino e aprendizagem que utiliza problemas – coerentes para com a futura atuação dos alunos como profissionais e cidadãos – para iniciar, enfocar e motivar a aprendizagem dos conhecimentos conceituais, procedimentais e atitudinais objetivados.

Na PBL os problemas são reais e admitem múltiplas soluções, deste modo, os estudantes separados em grupos e facilitados por um tutor debatem entre si os

conhecimentos que acreditam ser necessários para resolver o problema; individualmente eles buscam esse conhecimento e então reúnem-se para compartilhá-lo e discutir soluções para o problema; o processo pode repetir-se caso o problema não seja resolvido, ou seja, os estudantes voltam a buscar novos conhecimentos e fazer novas discussões. Sendo assim, o PBL é capaz de criar um ambiente de aprendizagem dinâmico e estimulante ao aproximar a experiência acadêmica com a prática do ambiente de trabalho, trazendo como isso os benefícios de promover habilidades tais como proatividade, iniciativa, capacidade de autoaprendizagem, de comunicação e de trabalho em equipe, requisitos essenciais à moderna profissão de engenharia.

As características apresentadas acima reforçam a PBL como um abordagem apta a melhorar o ensino de métodos ágeis dentro da engenharia de software; foi portanto, como o objetivo de trazer essas vantagens para o contexto do ensino de Scrum que a abordagem foi implementada na ferramenta proposta.

### **2.2.2 Ensino apoiado por ferramentas**

Além da metodologia de ensino PBL, a utilização de ferramentas de apoio ao ensino tem sido destacada na literatura como forma de melhorar o aprendizado [Bessa et al. 2012]. Ferramentas interativas, como jogos ou simuladores, utilizadas dentro do ambiente de ensino são capazes de despertar maior interesse e motivação no aluno [Gramigna 1994].

Além do carácter motivacional, as ferramentas contribuem para o ensino por apresentar o conteúdo de forma mais amigável e prazerosa por meio de interfaces interativas, cenários estimulantes e dinâmicas desafiadoras e recompensadoras. Outra vantagem do uso de ferramentas é poder simular um ambiente empresarial, no qual os estudantes possam testar e aplicar seus conhecimentos técnicos e de gerência, dentro de um escopo onde recursos de tempo e orçamento são controlados, podendo ser avaliada deficiências na formação dos alunos [Savi and Ulbricht 2008, Porto et al. 1999].

A ferramenta EngSoft, por exemplo, simula um ambiente empresarial real de fábrica de software, no qual os usuários podem atuar como clientes (demandando software), funcionários da fábrica (desenvolvendo software), ou consultores (facilitadores no processo de desenvolvimento) [Bessa et al. 2012].

Neste contexto, este trabalho visa combinar essas abordagens (PBL e ensino apoiado por ferramentas) e fornecer uma ferramenta que automatize o processo de ensino-aprendizagem e possa auxiliar os professores na utilização de abordagens ativas de ensino para metodologias ágeis, com foco no Scrum.

## **3. Trabalhos Relacionados**

Foi feita uma revisão não sistemática de maneira *ad-hoc* no *google scholar* no período de novembro de 2020 a fevereiro de 2021 buscando pelos termos ferramenta de ensino e métodos ágeis. Puderam ser encontrados na literatura alguns trabalhos propondo ferramentas para apoio ao ensino de metodologias ágeis em disciplinas de Engenharia de Software. Esses trabalhos incluem jogos e simuladores utilizados dentro do ambiente

educacional. Na tabela 1 são listadas as principais ferramentas identificadas, considerando a similaridade com o problema abordado neste trabalho.

**Tabela 1. Ferramentas de apoio ao ensino de Scrum**

Jogo/simulador	Referência	Plataforma	Descrição
Scrumie	[Marinho et al. 2020]	Web	O Scrumie é um jogo no qual um jogador deve seguir um fluxo que o permite gerenciar projetos de software, aplicando seus conhecimentos teóricos sobre Scrum.
Virtual Scrum	[Rodriguez et al. 2015]	Desktop	A Virtual Scrum é um software educacional que fornece um ambiente virtual 3d que simula uma sala real de um time Scrum, provendo, através de elementos virtuais, acesso fácil a quadros brancos que exibem o Product e Sprint Backlog, e às métricas de desempenho dos estudantes.
eduScrum	[Borges et al. 2014]	Web	A eduScrum é uma ferramenta desenvolvida para apoiar a metodologia eduScrum. É uma solução virtual do <i>Kanban</i> , mas permite também a gerência de usuários e projetos.
SCRUMIA	[Von Wangenheim et al. 2013]	Não Automatizada	A Scrumia lida com o planejamento e execução de um <i>sprint</i> de um hipotético projeto aplicando <i>Scrum</i> .

A PBL-Scrum permite ao usuário interagir com os artefatos do Scrum, criar times e gerenciar papéis Scrum, e alertar e interagir com os eventos e reuniões Scrum.

O foco da Scrumie está em ampliar os conhecimentos teóricos dos estudantes, enquanto nossa abordagem enfoca tanto conhecimentos teóricos quanto a prática de operação e gerência do Scrum.

A Virtual Scrum aproxima-se à proposta deste trabalho, pois implementa o Product Backlog como um lista de User Stories, o Sprint Backlog como um quadro Kanban, e fornece métricas de desempenho através de um gráfico Burndown, assim como fazemos. A principal diferença fica por conta da forma de apresentação do conteúdo, a Virtual Scrum simula um ambiente 3D, e a plataforma de desenvolvimento, a Virtual Scrum é uma aplicação desktop, enquanto a PBL-Scrum é uma aplicação mobile, possibilitando acesso mais amplo para os alunos.

A ferramenta *Scrumia* apresenta um jogo que utiliza materiais diversos como papel e caneta e está interessada em fortalecer o entendimento dos conceitos Scrum, mas não em acompanhar um projeto real ou acadêmico como propomos nesse trabalho.

A *eduScrum* oferece um quadro *Kanban* para acompanhar o progresso do trabalho, assim como nós fornecemos; porém, a eduScrum confere uma gerência de usuário rígida, delegando o papel de *Scrum Master* ao professor e não permitindo que

alguém atue no papel de *Product Owner*. Já na ferramenta proposta, há a possibilidade de delegar a função de *Product Owner* a alguém e o professor pode assumir qualquer função dentro do projeto.

A próxima seção apresenta a ferramenta proposta neste trabalho, descrevendo as principais tecnologias utilizadas, bem como o porquê da opção por elas; são descritos também os principais módulos e funcionalidades e como eles interagem.

#### **4. PBL-Scrum**

A ferramenta proposta oferece uma interface que possibilita a criação de um projeto que representa um espaço de trabalho no qual os estudantes separados em grupo, atuando dentro dos papéis Scrum, e auxiliados por um professor, o qual pode assumir um papel no time Scrum, possam cooperar e debater a solução de problemas-projetos reais preconizados pela abordagem PBL.

Dentre as principais funcionalidades da ferramenta estão a definição do time *Scrum* e seu autogerenciamento de acordo com os papéis *Scrum* (*Product Owner*, Time de Desenvolvimento e *Scrum Master*), interação e modificação dos artefatos do *Scrum* (*Product Backlog* e *Sprint Backlog*), e recebimento de alertas sobre eventos (*Sprint* e *Sprint Planning*) e reuniões (*Daily Scrum*, *Sprint Review* e *Sprint Retrospective*) que são descritas em detalhes na Seção 4.3.

##### **4.1 Definição da Ferramenta**

A interface de usuário foi construída utilizando o *framework react-native*, o qual é um framework *javascript* que permite criar apps nativos para *Android* e *IOS* [React Native 2020]. Para armazenamento das informações do usuário e dados de projetos foi utilizado o *Firestore Realtime Database* do *Google*, que permite que os dados sejam sincronizados em tempo real com todo cliente conectado, ou seja, permite que os clientes automaticamente recebam atualizações com os dados mais recentes [Firestore Database 2020].

O aplicativo foi desenvolvido utilizando a plataforma Expo, que facilita o desenvolvimento de aplicações *React Native* por permitir fácil acesso às *APIs* nativas do dispositivo não necessitando instalar a *SDK* do *Android* ou o *XCode* para *Mac*, isso porque a Expo possui um aplicativo móvel instalável pelas lojas do *Android/iOS* que contém todo código nativo necessário pelo *React Native* para iniciar sua aplicação [Expo, Rocketseat].

A plataforma mobile foi escolhida porque os dispositivos móveis estão em contato mais próximo e frequente com o usuário, facilitando, assim, notificações e mensagens que permitam manter o usuário atualizado. Além disso, muitos apps (aplicativos), que são os softwares dos dispositivos mobile, conseguem "agregar em uma única ferramenta recursos visuais e auditivos capazes de estimular o estudo com auxílio de interfaces atraentes e extremamente amigáveis e intuitivas" [de Oliveira and de Menezes Alencar 2017].

## 4.2 Arquitetura da Ferramenta

### 4.2.1 Módulos

A ferramenta é composta de quatro módulos principais que são responsáveis por sua operação, são eles:

#### 1. Módulo de Cadastro, Login e Controle de Acesso:

“Os aplicativos precisam reconhecer a identidade do usuário. Ter essa informação permite que um aplicativo salve os dados do usuário na nuvem com segurança e forneça uma experiência personalizada para cada usuário” [Firebase].

O *SDK do Firebase Authentication* fornece maneiras fáceis de autenticar um usuário. A maneira aqui utilizada é a Autenticação baseada em e-mail e senha; para este tipo de autenticação, o *SDK do Firebase Authentication* fornece métodos para criar e gerenciar usuários a partir do email e senha fornecidos pelo usuário.

O módulo de Cadastro, Login e Controle de Acesso é responsável por fornecer o método de Autenticação baseada em e-mail e senha do *SDK do Firebase Authentication* para a aplicação.

#### 2. Módulo de exibição e criação de projetos:

A primeira função deste módulo é mostrar ao usuário os projetos que ele tem. Essa primeira função do módulo atua de forma a recuperar os nomes dos projetos que o usuário tem armazenado no banco de dados (no caso, o banco de dados utilizado é o *firebase realtime database*) e então exibir cada nome de projeto como um ícone clicável que leva à área de operação do projeto.

A segunda função deste módulo é permitir a criação de novos projetos. O usuário é levado por uma série de telas que lhe pedem para informar o nome do novo projeto, dar uma descrição para o novo projeto e montar um time Scrum. De posse dessas informações, o módulo de criação de projetos é responsável por gravar essas informações no banco de dados.

#### 3. Módulo de envio de notificações:

A *expo-notifications* oferece uma *API* que permite apresentar notificações no lado do cliente. A *Expo* também oferece um servidor pelo qual é possível enviar as notificações para os usuários.

O Processo funciona da seguinte maneira:

- No lado do cliente, é necessária a permissão do usuário para enviar a ele *push notifications*.

- Garantida a permissão, a *Expo* gera um *Token* chamado *ExpoPushToken* que identifica o aparelho do usuário.
- Quando o usuário deseja programar o envio de uma notificação, uma requisição do tipo POST é realizada num servidor node.js, criado para o âmbito desta ferramenta. Esse servidor é responsável por programar o disparo de uma requisição para o servidor da *Expo*, informando os *ExpoPushTokens* dos usuários que deverão receber a notificação.
- O servidor da *Expo* se encarrega de mandar as notificações para os *ExpoPushTokens* (os usuários) que deverão receber a notificação.
- Por fim, a notificação é exibida nos dispositivos dos usuários.

#### 4. Módulo de operação do projeto:

Este módulo é responsável por exibir uma página que mostra um esquema interativo do ciclo Scrum; a maioria das funcionalidades operacionais disponibilizadas por esta ferramenta estão acessíveis através da interação do usuário com esta página.

### 4.3 Funcionalidades

O usuário da ferramenta deve criar uma conta ou logar numa já existente para então ser direcionado para a área de projetos, onde ele pode criar um novo projeto ou acessar um já existente; acessando ou criando um projeto, o usuário é levado para uma interface que disponibiliza um ciclo Scrum interativo, onde o usuário pode experimentar interagir e gerenciar o *Product Backlog* e o *Sprint Backlog*, criar alertas de reuniões e ter acesso à métricas de desempenho. A figura 2 ilustra o fluxo de execução da ferramenta.

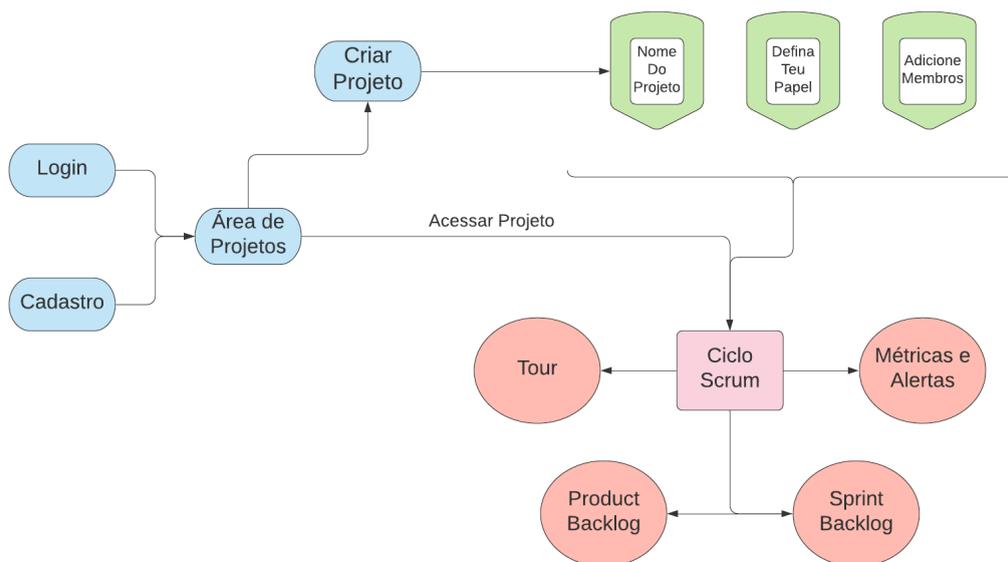


Figura 2. Fluxograma de execução da ferramenta.

#### 4.3.1 Cadastro

Os usuários podem efetuar cadastro informando um nome, um email e uma senha. Após a inserção dessas informações o módulo de Cadastro, Login e Controle de Acesso certifica-se de que o email informado é um email válido e de que a senha informada tem seis ou mais caracteres, apenas satisfeitas essas condições, o usuário poderá efetuar cadastro com sucesso.

#### 4.3.2 Login

Os usuários podem efetuar login informando o email e senha previamente cadastrados; o Módulo Cadastro, Login e Controle de Acesso certifica-se de verificar se existe algum usuário cadastrado com aquele email e se a aquela senha está correta. Em caso afirmativo, o usuário efetua login com sucesso e é redirecionado para a página de projetos. Em caso negativo, mensagens de erro são exibidas orientando o usuário sobre como proceder; algumas mensagens de erro e seu significado são: “E-mail Inválido”, neste caso, o usuário forneceu um email que não existe; “Usuário não encontrado”, neste caso, o usuário forneceu um email que não possui conta; “Senha Errada”, neste caso, o usuário forneceu uma senha errada.

#### 4.3.3 Criação de um Projeto

Os usuários logados podem efetuar a criação de um novo projeto, para isso, eles devem informar um nome e uma descrição para o projeto, e convidar colaboradores para serem membros do time Scrum.

#### 4.3.4 Criação de um time Scrum

Durante a criação de um projeto, ao usuário é requisitado que ele monte um time Scrum, para isso ele deve fornecer os emails dos colegas que farão parte daquele projeto.

#### 4.3.5 Atribuição de papéis

Durante a criação de um time Scrum, ocorre a atribuição de papéis; campos específicos requerem que o usuário, que está criando o projeto, indique quem será o Product Owner, quem será o Scrum Master, e quem fará parte do Time de Desenvolvimento.

#### 4.3.6 Tour pelos elementos do Ciclo Scrum

O usuário é levado por um tour pelos elementos do Ciclo Scrum, neste tour, uma breve explicação sobre cada elemento é exibida para o usuário. Este tour é especialmente adequado e instrutivo para usuários inexperientes com o Scrum.

#### 4.3.7 Criação de *User Stories* para o *Product Backlog*

As *User Stories* criadas para o *Product Backlog* tem um título e a seguinte estrutura que deve ser completada pelo usuário. i) *As a*: descreve quem você é; ii) *I want to*: descreve o que você quer; e iii) *so that*: descreve para que você quer.

Elas também têm um campo numérico, chamado prioridade, que é usado para ordená-las em ordem crescente e uma cor de fundo para facilitar diferenciar umas das outras.

#### 4.3.8 Implementação do *Sprint Backlog* como um quadro *Kanban*

O *Sprint Backlog* reúne os itens que se completados durante a *Sprint* geram um incremento de software funcional e entregável. Um quadro *Kanban* facilita visualizar o que deve ser feito e acompanhar a evolução das tarefas.

Neste caso, o quadro *Kanban* foi implementado utilizando-se quatro colunas. A primeira coluna é chamada *Backlog*; nela estão as *User Stories* que o time Scrum moveu do *Product Backlog* para o *Sprint Backlog* e que devem ser completadas no decorrer da *Sprint*.

As outras três colunas são chamadas *To Do*, *Doing*, e *Done*; nelas estão cartões responsáveis por quebrar a *User Stories* em tarefas menores. A coluna *To Do* contém os cartões que ainda não estão sendo feitos; a coluna *Doing* contém os cartões atualmente em desenvolvimento; e a coluna *Done* contém os cartões que já foram finalizados. Os cartões são movidos de uma coluna para outra conforme o avanço no desenvolvimento.

Os cartões criados possuem um título e uma descrição que representam uma tarefa a ser executada; eles podem ser designados a determinados membros do Time de Desenvolvimento, e podem também ser atribuídos a uma *User Story* específica. Atribuir um cartão a uma *User Story* específica significa dizer que aquele cartão é uma tarefa menor, que se completado, contribui para que a *User Story* seja satisfeita.

#### 4.3.9 Disponibilização de Métricas de Desempenho

Um gráfico *Burndown* dos cartões do *Sprint Backlog* é disponibilizado ao usuário. Basicamente, na eixo Y fica a quantidade de cartões que o usuário não completou (cartões nas colunas *To Do* e *Doing*), no eixo X ficam os dias; com o passar do tempo e conforme o usuário move os cartões para a coluna *Done*, o gráfico exibirá uma curva descendente mostrando a evolução na completude dos cartões.

#### 4.3.10 Criação de Alertas de Reuniões

O usuário pode criar alertas que enviam notificações aos usuários do grupo. O usuário pode determinar o horário da *Daily Scrum*, e neste caso, todos os dias de segunda a sexta, uma notificação será enviada no horário definido. A outra reunião que o usuário pode programar é a *Sprint Retrospective*, neste caso, ele deve escolher o dia e horário específicos que a notificação deverá ser enviada.

## 5. Exemplo de aplicação

A fim de exemplificar a aplicação da ferramenta por meio de um exemplo prático, considerando a aprendizagem baseada em problemas, descreve-se aqui um exemplo de problema que pode ser utilizado de maneira didática para a utilização da ferramenta em ambiente acadêmico; esse problema foi extraído de [Barbosa et al. 2016] e aborda o desenvolvimento de um sistema de gerenciamento de empréstimos de livros de uma biblioteca.

Nesse contexto, um sistema de empréstimo de livros de biblioteca é um sistema que auxilia a gerência de empréstimos de livros feitos na biblioteca. Para um usuário-estudante, o sistema permite reservar livros pela internet; para um usuário atendente da biblioteca, o sistema permite consultar a localização de livros para pegá-los rapidamente e consultar os dados dos alunos para entregar-lhes os livros; e para o administrador do sistema, permite visualizar os livros que não foram entregues e contatar os estudantes que os emprestaram.

O *Product Backlog* e o *Sprint Backlog*, os quais descrevem o sistema de gerenciamento de empréstimos de livros e são fornecidos pelo exemplo de aplicação, serão demonstrados dentro da ferramenta nas imagens a seguir. Assim, espera-se demonstrar a aptidão da ferramenta em auxiliar o desenvolvimento de um problema-projeto, bem como a possibilidade de interagir com os elementos do Scrum.

O *Product Backlog* do exemplo é formado por itens compostos de uma *User Story* e um campo indicando a prioridade do item. Dois itens do *Product Backlog* demonstrando a perspectiva de dois diferentes usuários deste sistema podem ser vistos logo abaixo.

- Item 1
  - Descrição: Como aluno, eu quero reservar um livro pela internet para que não tenha que ir até a biblioteca.
  - Prioridade: Alta.
- Item 2
  - Descrição: Como atendente, eu quero consultar a localização de um livro para que eu possa pegá-lo rapidamente.
  - Prioridade: Média.

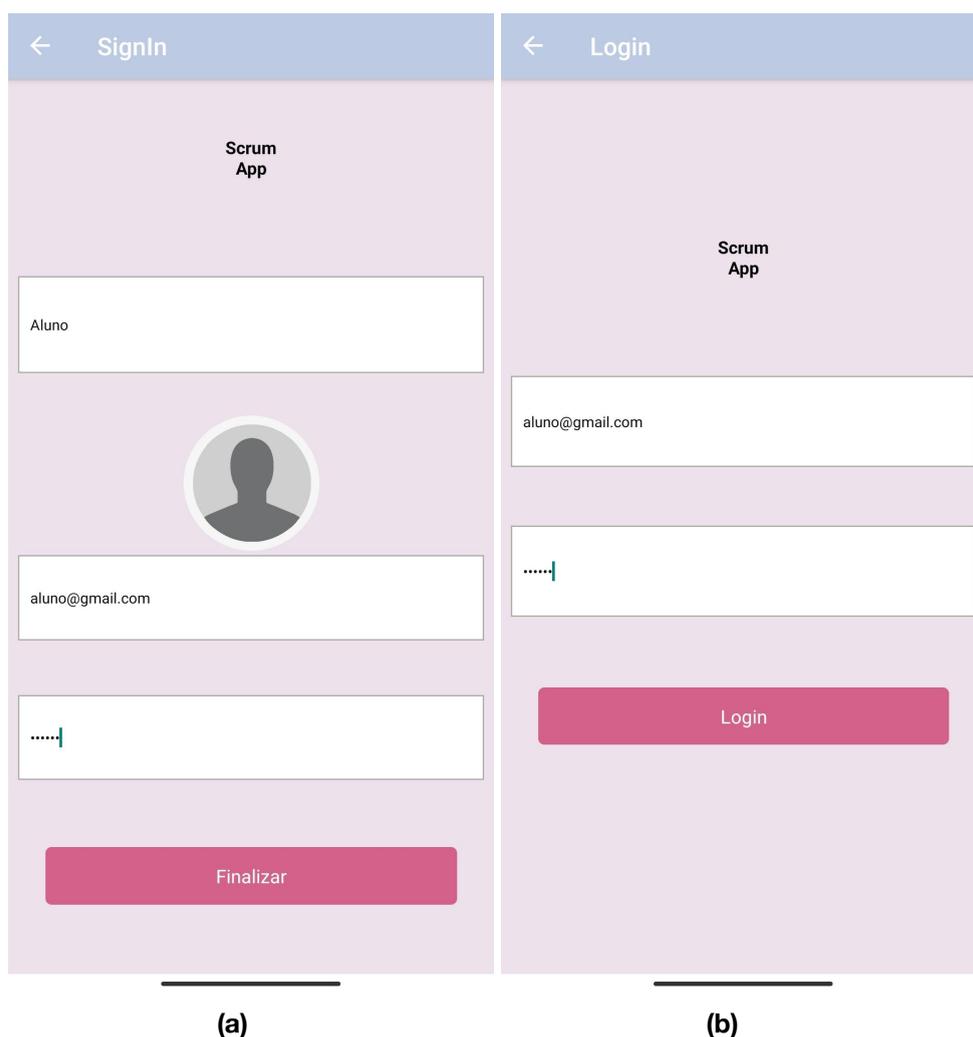
O *Sprint Backlog* do exemplo mostra uma tabela que quebra as *User Stories* do *Product Backlog* em tarefas menores. Abaixo, é mostrado um dos itens do *Sprint Backlog* do exemplo.

- User Story: Como aluno, eu quero reservar um livro pela internet para que não tenha que ir até a biblioteca.
  - tarefa 1: Projetar tela de pesquisa de livro;
  - tarefa 2: Codificar módulo de reserva;
  - tarefa 3: Criar módulo para envio de e-mail;
  - tarefa 4: Criar comprovante de reserva.

A ferramenta oferece um espaço compartilhado, que representa um projeto, no qual usuários *logados* e reunidos em grupo possam trabalhar como time *Scrum*,

experimentando interagir com os artefatos do Scrum, atuar dentro dos papéis Scrum e gerenciar reuniões e eventos Scrum.

As figuras 3(a) e 3(b) ilustram a interface de **Cadastro e Login** respectivamente. Na interface de Cadastro, o usuário pode criar uma conta informando nome, email e senha; mensagens de erro e orientação são mostradas ao usuário caso ele tenha informado um email inválido ou uma senha com menos de seis caracteres. Através da interface de Login, o usuário pode acessar através de email e senha uma conta previamente cadastrada.



**Figura 3: Tela de Cadastro e Login : (a) Tela de Cadastro; (b) Tela de Login.**

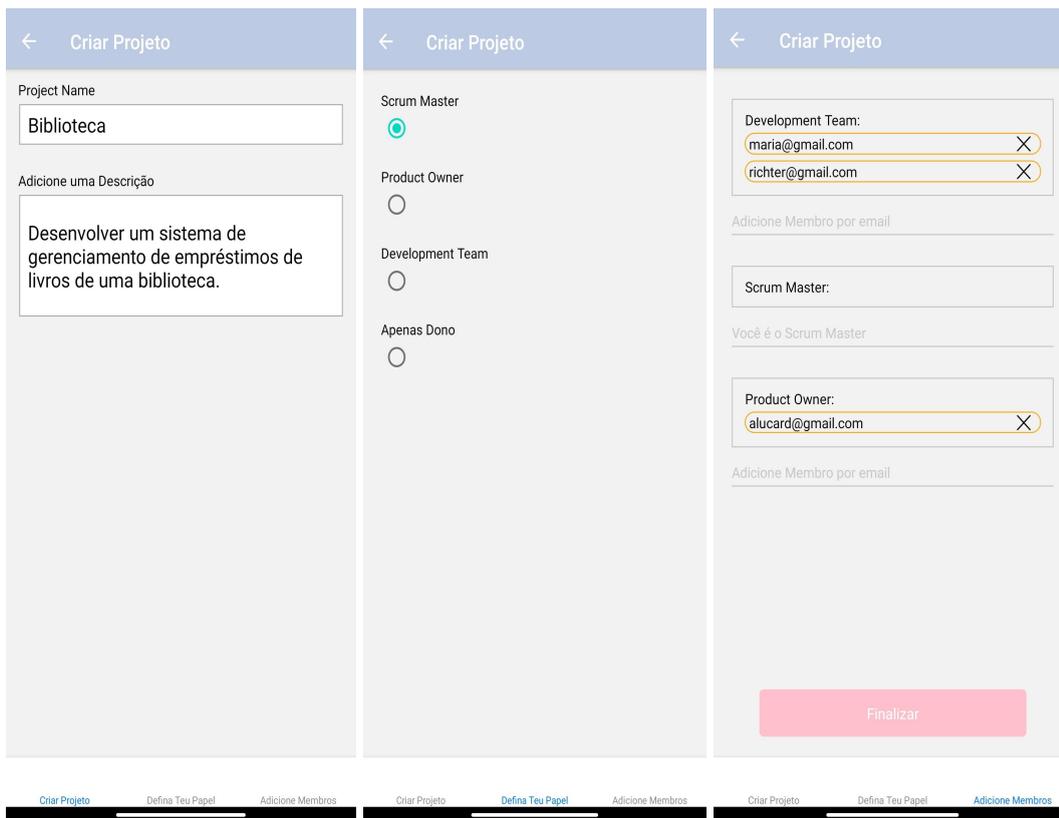
Efetuada login ou criada uma conta, o usuário é direcionado para a Área de Projetos; nesta área encontram-se tanto os projetos que o usuário criou, bem como, os projetos que foram compartilhados com este usuário. A figura 4 ilustra um exemplo de Área de Projetos; neste exemplo, o usuário possui um projeto próprio chamado Biblioteca e um projeto compartilhado com ele chamado Barbearia.



**Figura 4: Área de Projetos**

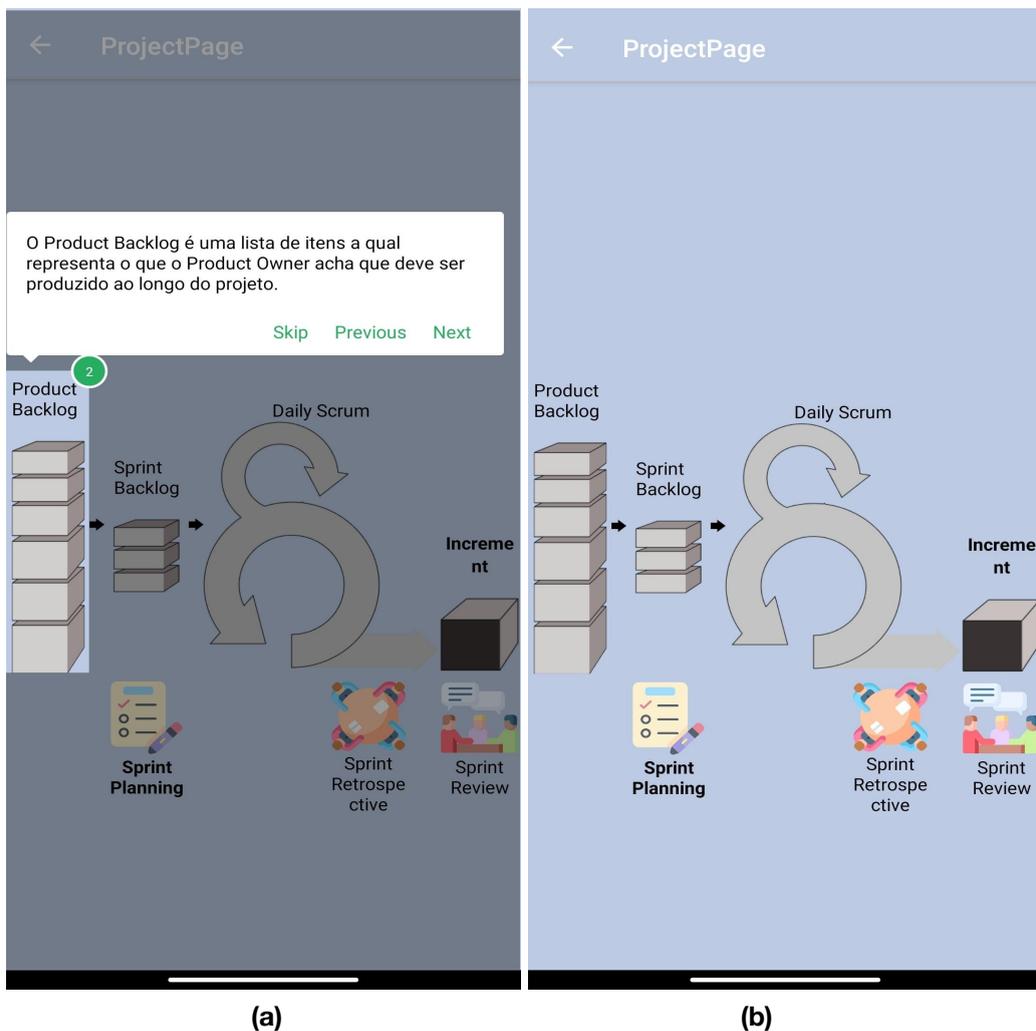
Clicando no botão flutuante (botão circular com sinal de adição) no canto inferior direito da Área de Projetos (figura 4), o usuário é levado através de três telas (figuras 5(a), 5(b), e 5(c)) que permitem-no efetuar a **Criação de novos Projetos**.

A primeira tela (figura 5(a)), chamada criar Projeto, permite ao usuário definir um nome e uma descrição a um novo projeto; a segunda tela (figura 5(b)), chamada Defina seu Papel, pede ao usuário que escolha qual papel ele desempenhará dentro do projeto; a terceira tela (figura 5(c)), chamada Adicione Membros, permite ao usuário convidar pessoas para atuarem no projeto dentro de algum papel que Scrum prescreve.



(a) (b) (c)  
**Figura 5: Interfaces de criação de projetos: (a) criar Projeto; (b) Defina teu Papel; (c) Adicione membros.**

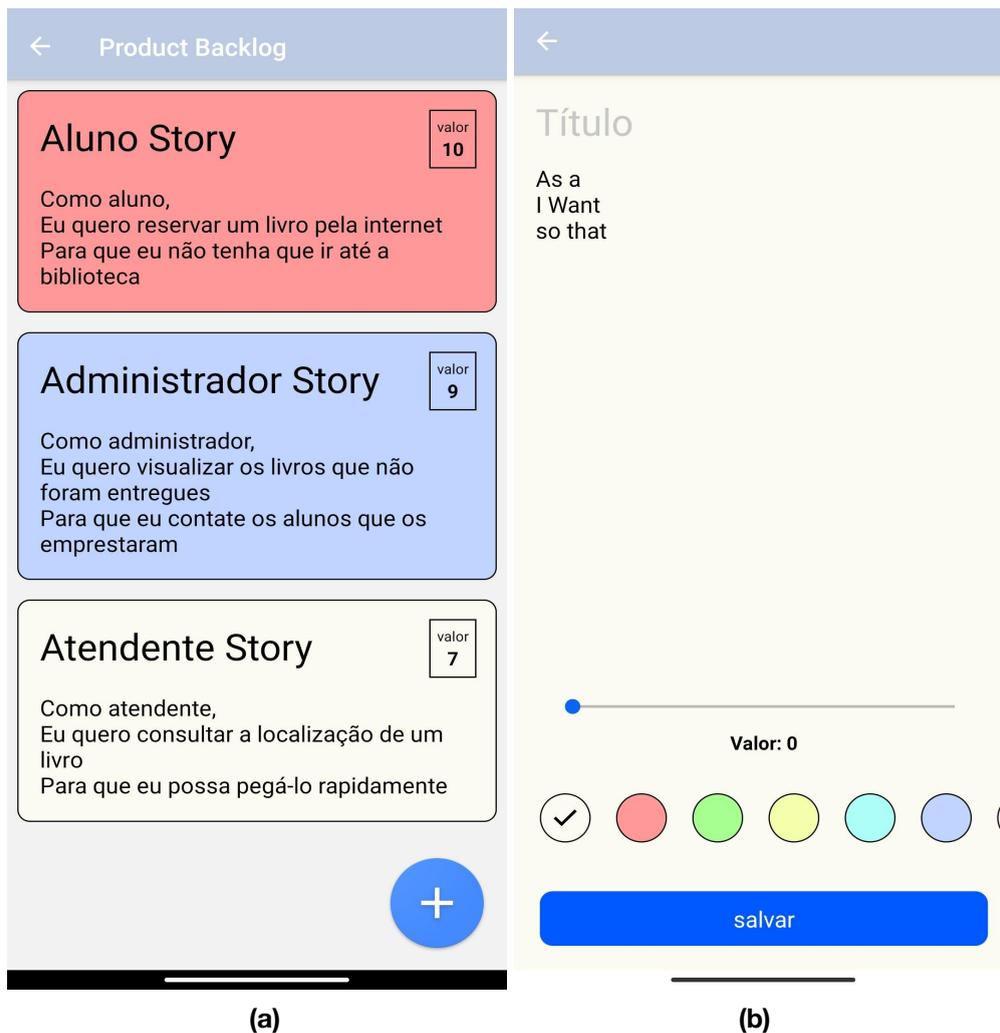
Criado um projeto, o usuário é direcionado a **Interface Principal do Projeto** (figura 6(a)). Esta interface apresenta um esquema interativo do ciclo Scrum pelo qual o usuário tem acesso a maioria das funcionalidades disponibilizadas pela ferramenta. A primeira funcionalidade desta interface é prover um tour pelo Ciclo Scrum, evidenciando cada elemento do Scrum e exibindo uma breve explicação sobre eles.



**Figura 6: Interface Principal do Projeto : (a) Tour pelos elementos do Scrum (b) Ciclo Scrum interativo.**

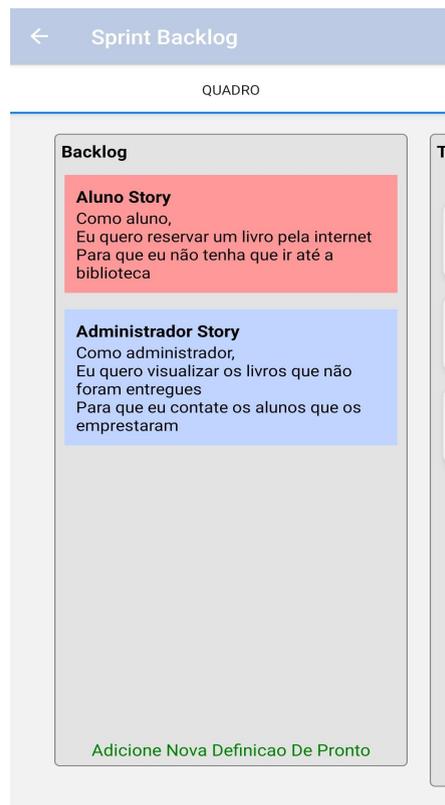
Através da Interface Principal do Projeto, clicando no ícone do *Product Backlog*, o usuário pode criar *User Stories* e notas como itens para *Product Backlog*. A figura 7(a) exibe um exemplo de *Product Backlog*. Note que as *User Stories* adicionadas são justamente as *User Stories* extraídas do exemplo de aplicação e que o campo prioridade, que no exemplo é um valor categórico, foi transformado no campo prioridade do tipo numérico.

Outra característica do *Product Backlog* da ferramenta é ordenar as *User Stories* em ordem decrescente pelo valor prioridade.



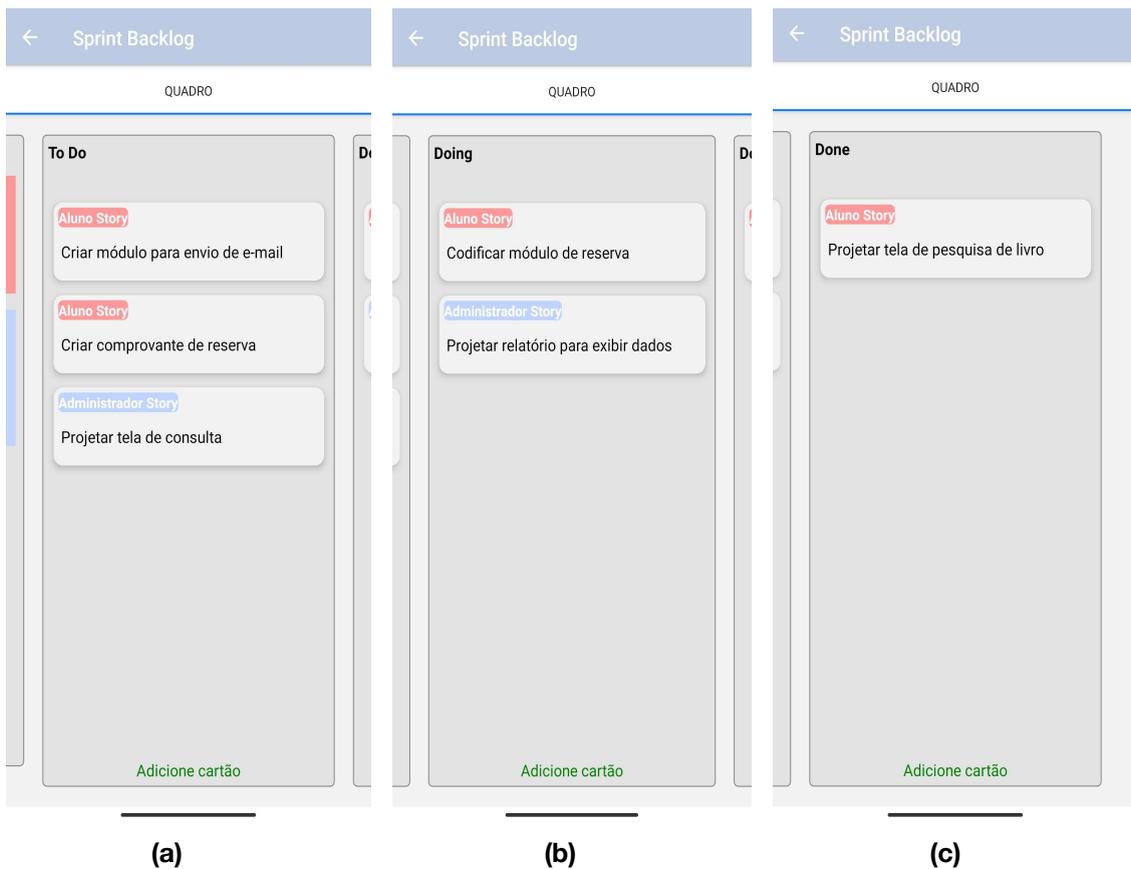
**Figura 7: Product Backlog : (a) Lista de User Stories adicionadas (b) Formulário de criação de novas User Stories.**

Através da Interface Principal do Projeto, clicando no ícone do *Sprint Backlog*, o usuário tem acesso a um quadro *Kanban* com colunas (*Backlog*, *To Do*, *Doing*, *Done*) para dar maior visibilidade sobre a atual situação dos itens da *Sprint* em desenvolvimento. A coluna *Backlog* (figura 8) permite ao usuário adicionar *User Stories* que ele criou previamente e que estão armazenadas no *Product Backlog*.



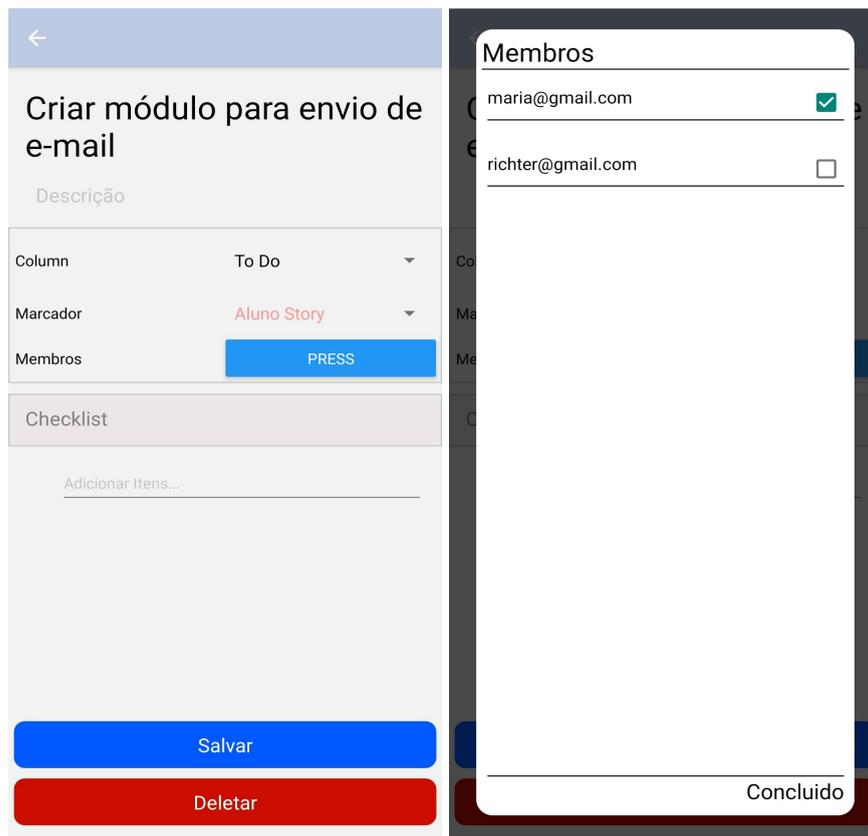
**Figura 8: Coluna *Backlog***

As demais colunas (*To Do*, *Doing*, *Done*) reúnem cartões que são atribuídos a *User Stories* como forma de quebrá-las em tarefas menores (figura 9). Note que os cartões refletem justamente as tarefas extraídas do *Sprint Backlog* do exemplo de aplicação e que a *tag* no canto superior esquerdo nos diz a qual *User Story* este cartão está designado.



**Figura 9: Sprint Backlog : (a) Coluna To Do; (b) Coluna Doing; (c) Coluna Done.**

Os cartões adicionados ao quadro *kanban* mostram as seguintes características quando clicados: eles têm um título, têm uma descrição, podem ser movidos de colunas (figura 10(a)), podem ser atribuídos a uma *User Story*, podem ser designados para membros da equipe (figura 10(b)), e eles possuem uma área de *checklist*.

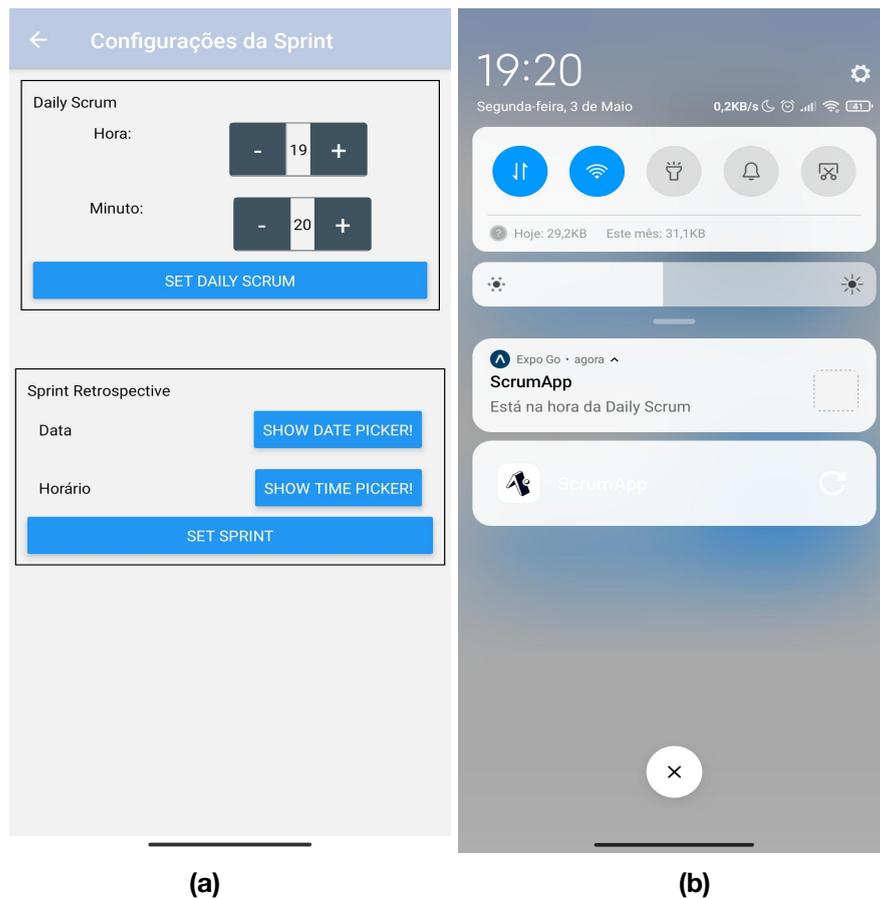


(a)

(b)

**Figura 10: Cartões do Kanban : (a) características dos cartões; (b) Designar membros para o cartão.**

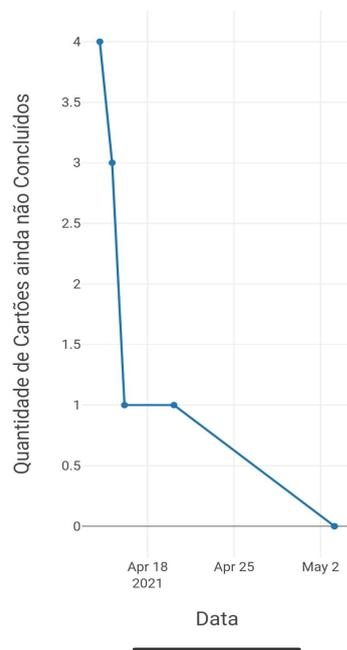
O usuário também tem acesso a uma interface (figura 11(a)) que lhe permite programar notificações e alarmes. São duas reuniões, as quais o usuário pode programar o horário em que uma notificação será enviada como alerta para o time, a *Daily Scrum* e a *Sprint Retrospective*. No caso da *Daily Scrum*, todos os dias de segunda a sexta, uma notificação é enviada para o time no horário programado. No caso da *Sprint Retrospective*, uma notificação é enviada no dia e hora específicos informados pelo usuário. A figura 11(b) mostra uma notificação recebida pelo usuário.



**Figura 11: Programar Notificações (a) Programar *Daily Scrum* e *Sprint Retrospective* (b) Notificação recebida.**

Sob o ícone da *Sprint Retrospective*, o usuário tem acesso a um gráfico *Burndown* da evolução da completude dos cartões do *Sprint Backlog* (figura 12); a ideia é fornecer métricas de desempenho para que o time possa durante a *Sprint Retrospective* auto inspecionar-se.

Gráfico burndown dos cartões do Sprint Backlog

**Figura 12: Gráfico *Burndown* dos cartões do *Sprint Backlog***

## 6. Resultados e Discussão

Para atender o princípio PBL de trabalho em equipe, a ferramenta disponibiliza um espaço compartilhado, o qual representa um projeto. Os professores podem ter papéis dentro dos projetos dos alunos e atuarem como facilitadores (o que é muito similar à função do *Scrum Master* dentro do *Scrum*) como preconiza o PBL. Os alunos têm à disposição um quadro *Kanban* que os permite auto-gerenciarem-se e auto-planejarem-se.

Visando ensinar Scrum adequadamente, a ferramenta pretende dar uma experiência prática, ou seja, os alunos poderão gerenciar um *Product Backlog*, isto é, entender do que ele é composto e quais características possui, e também negociar itens do *Product Backlog* que devem ser construídos, movê-los para o *Sprint Backlog*, e visualizar o progresso do trabalho no quadro *Kanban*. Ademais, questões de prazo como a *Sprint*, e datas de reuniões podem ser melhor compreendidas com o uso da ferramenta.

A fim de avaliar a ferramenta, buscou-se desenvolver uma análise da aplicabilidade por meio de um exemplo de aplicação para verificar a validade da ferramenta e a capacidade de disponibilizar artefatos que contribuam com o ambiente de prática de ensino. Para realizar essa análise, desejou-se encontrar um exemplo de *Sprint* documentada cujos *Product Backlog* e *Sprint Backlog* pudessem ser demonstrados dentro da ferramenta. Encontrou-se um exemplo de aplicação que abordava o

desenvolvimento de um sistema de gerenciamento de empréstimos de livros de uma biblioteca (descrito na seção anterior) que disponibilizava um *Product Backlog* composto por *User Stories* e um *Sprint Backlog* que quebrava as *User Stories* em tarefas menores.

Como mostrado ao longo do passo a passo da execução da ferramenta nas imagens da seção anterior, a ferramenta conseguiu com sucesso se ajustar ao exemplo de aplicação encontrado.

Por fim, pode-se listar o que acredita-se que sejam algumas das principais contribuições da ferramenta quando adotada para apoio ao ensino de Scrum dentro de uma disciplina de engenharia de software:

- I. Criar um ambiente para trabalho em grupo e compartilhamento de informações;
- II. Criar visibilidade sobre as tarefas em execução;
- III. Permitir manipulação e interação com os artefatos do Scrum;
- IV. Permitir que os usuários atuem dentro das atribuições dos papéis Scrum.

## 7. Trabalhos Futuros

Planeja-se que trabalhos futuros incluam as seguintes atividades:

- O desenvolvimento de funcionalidades adicionais, como por exemplo, uma dinâmica que automatizasse a *Sprint Planning*, ou seja, que permitisse ao Time Scrum negociar através da ferramenta os itens do *Product Backlog* que deveriam ser completados durante a *Sprint*.
- A implementação da ferramenta como ferramenta de apoio ao ensino de Scrum dentro de uma disciplina de Engenharia de Software ou correspondentes, a ser ministrada no curso de Engenharia de Computação e/ou Sistemas de Informação da Universidade Federal da Grande Dourados (UFGD).
- A aplicação de estudos empíricos para avaliar a eficácia e efetividade da ferramenta, sob o ponto de vista de alunos e professores que utilizaram a ferramenta para o auxílio ao processo de ensino-aprendizagem, bem como a análise, sumarização e divulgação dos resultados em artigos científicos a serem publicados em conferências da área.

## 8. Conclusão

Este artigo apresentou o desenvolvimento de uma ferramenta *mobile* para apoio ao ensino de Scrum via Aprendizagem Baseada em Problemas. Atualmente, a ferramenta conta com funcionalidades de cadastro e login, gerência de projetos, gerência de Time Scrum, manipulação do *Product Backlog* e do *Sprint Backlog*, criação de alertas de Reuniões, e Disponibilização de Métricas de Desempenho por um gráfico *Burndown*.

Avaliou-se a aplicabilidade da ferramenta por meio de um exemplo-problema que abordava o desenvolvimento de um sistema de gerenciamento de empréstimos de livros de uma biblioteca o qual demonstrou a aptidão da ferramenta em auxiliar a gerência, o desenvolvimento, e a resolução de problemas-projeto, e a competência da

ferramenta em disponibilizar artefatos que contribuam com o ambiente de prática de ensino.

As próximas etapas da pesquisa envolvem a avaliação empírica da ferramenta, bem como a divulgação dos resultados encontrados em eventos científicos da área.

Considerando o estado atual da área de pesquisa, em especial do ensino de Scrum, e observando que ferramentas de apoio ao ensino e abordagens de ensino centradas no aluno têm sido amplamente estudadas como soluções, espera-se que a ferramenta proposta possa contribuir para aprimorar o ensino de métodos ágeis em especial a metodologia Scrum em cursos de computação no país.

## Referências

Agile Alliance (2015). What is agile software development? <https://www.agilealliance.org/agile101/>. Accessed: 2020-10-07.

Barbosa, R. A. et al. (2016). Análise de 'backlog' de histórias de usuário por meio de agrupamento de textos e similaridade semântica.

Bessa, B., Cunha, M., and Furtado, F. (2012). Engsoft: Ferramenta para simulação de ambientes reais para auxiliar o aprendizado baseado em problemas (pbl) no ensino de engenharia de software. In Anais do XX Workshop sobre Educação em Informática. Curitiba-PR.

Borges, K. S., Schmitt, M. A. R., and Nakle, S. M. (2014). Eduscrum projetos de aprendizagem colaborativa baseados em scrum. *RENOTE-Revista Novas Tecnologias na Educação*, 12(1)

Boulic, R. and Renault, O. (1991) "3D Hierarchies for Animation", In: *New Trends in Animation and Visualization*, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons Ltd., England.

Castro, R. M., Siqueira, S. W. M., de Almeida, D. N., and Nascimento, F. C. (2017). Agility scrum-um jogo para ensino da metodologia scrum. In Anais do XXV Workshop sobre Educação em Computação. SBC

Dyer, S., Martin, J. and Zulauf, J. (1995) "Motion Capture White Paper", [http://reality.sgi.com/employees/jam\\_sb/mocap/MoCapWP\\_v2.0.html](http://reality.sgi.com/employees/jam_sb/mocap/MoCapWP_v2.0.html), December.

Expo. Introduction to expo. <https://docs.expo.io/>. Accessed: 2021-04-04

Firebase Database (2020). Firebase realtime database. <https://firebase.google.com/docs/database/>. Accessed: 15/12/2020

Firebase Authentication. Firebase authentication. <https://firebase.google.com/docs/auth/>. Accessed: 2021-04-04

Fonseca, I. and Campos, A. (2008). Por que scrum? *Engenharia de Software Magazine*, DevMedia, pages 30–35.

Gramigna, M. R. M. (1994). *Jogos de Empresa*. MAKRON BOOKS.

- Holton, M. and Alexander, S. (1995) "Soft Cellular Modeling: A Technique for the Simulation of Non-rigid Materials", *Computer Graphics: Developments in Virtual Environments*, R. A. Earnshaw and J. A. Vince, England, Academic Press Ltd., p. 449-460.
- Knuth, D. E. (1984), *The TeXbook*, Addison Wesley, 15<sup>th</sup> edition.
- Macedo, C. H. B., Sotero, S., de Souza Rabelo, D. S., and Valentim, R. (2019). Utilização do pbl para o acompanhamento na aprendizagem de scrum. In *Anais do Workshop de Informática na Escola*, volume 25, page 1079.
- Marinho, L. L., dos Santos, S. R. C., Andrade, L., Cons, B. C., Schots, M., and Werneck, V. M. (2020). Scrumie: Scrum teaching agent oriented game. *Revista de Informática Teórica e Aplicada*, 27(2):140–152.
- Meireles, M. C. and Bonifácio, B. (2015). Uso de métodos ágeis e aprendizagem baseada em problema no ensino de engenharia de software: Um relato de experiência. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 180
- Mills, J. E., Treagust, D. F., et al. (2003). Engineering education—is problem-based or project-based learning the answer. *Australasian journal of engineering education*, 3(2):2–16.
- Oliveira, A. R. F. and de Menezes Alencar, M. S. (2017). O uso de aplicativos de saúde para dispositivos móveis como fontes de informação e educação em saúde. *RDBCI: Revista Digital de Biblioteconomia e Ciência da Informação*, 15(1):234–245.
- PORTO, A. J. V. ; CARVALHO, Anna Cristina Barbosa Dias de . Jogos de empresas. In: *ENEGEP - Encontro Nacional de Engenharia de Produção, 1999, Rio de Janeiro. Anais do XIX EBEGEP*. Rio de Janeiro: Editora da ABEPRO, 1999.
- React Native (2020). React native. <https://reactnative.dev/docs/getting-started/>. Accessed: 15/12/2020.
- Ribeiro, L. and Ribeiro, C. (2020). Aprendizagem baseada em problemas (pbl) na educação em engenharia.
- Rocketseat. Expo: o que é, para que serve e quando utilizar? <https://blog.rocketseat.com.br/expo-react-native/>. Accessed: 2020-04-04.
- Rodriguez, G., Soria, A., and Campo, M. (2015). Virtual scrum: A teaching aid to introduce undergraduate software engineering students to scrum. *Computer Applications in Engineering Education*, 23(1):147–156.
- Sabbagh, R. (2014). *Scrum: Gestão ágil para projetos de sucesso*. Editora Casa do Código
- Savi, Rafael, and Vania Ribas Ulbricht. "Jogos digitais educacionais: benefícios e desafios." *RENTE-Revista Novas Tecnologias na Educação* 6.1 (2008).
- Smith, A. and Jones, B. (1999). On the complexity of computing. In *Advances in Computer Science*, pages 555–566. Publishing Press.
- Sommerville, I. (2011). *Engenharia de software*. PEARSON BRASIL

Sutherland, J. e Schwaber, K. (2018).]. Scrum Guide; . Disponível em:  
<https://scrumguides.org>.

Von Wangenheim, C. G., Savi, R., and Borgatto, A. F. (2013). Scrumia—an educational game for teaching scrum in computing courses. *Journal of Systems and Software*, 86(10):2675–2687.