

DMÓVEL: DESENVOLVIMENTO DE COMPETÊNCIAS EM PRODUÇÃO DE SOFTWARE, GERÊNCIA DE PROJETOS E TRABALHO REMOTO

Agostinho L. S. Castro – agcastro@ufpa.br

Ronaldo F. Zampolo – zampolo@ufpa.br

Alan M. Oliveira – alan.olvm@gmail.com

Edinaldo L. Santos – edinaldoluisdossantos@gmail.com

Pedro V. C. Nascimento – pedrovictorcosta7@gmail.com

Universidade Federal do Pará, Instituto de Tecnologia

Faculdade de Engenharia da Computação e Telecomunicações

Campus Belém – Av. Augusto Corrêa, 1

66075-110 – Belém – Pará

Resumo: *As escolas de engenharia devem preparar seus alunos para os desafios da atualidade que, em geral, demandam soluções complexas fruto de trabalho colaborativo e interdisciplinar. O cenário pandêmico da COVID-19 expôs ainda mais a necessidade de desenvolver habilidades de comunicação, e gestão do conhecimento e de equipes, cujos elementos podem estar distribuídos geograficamente. Este artigo apresenta a experiência de execução de um projeto de desenvolvimento de aplicativo e sua adaptação ao trabalho remoto. A equipe é formada por alunos de graduação, professores e profissionais das áreas de engenharia e ciência da computação, psicologia, e saúde, envolvendo ainda a coordenadoria de acessibilidade da instituição executora. O aplicativo usa MongoDB, Flutter e Laravel. O gerenciamento das atividades adota Scrum, sistemas Git, e Trello. Em face do necessário isolamento social, a equipe ajustou-se a uma rotina de encontros diários virtuais, e de reuniões quinzenais por videoconferência para avaliação/planejamento. A gestão de documentos e acompanhamento de estudos à distância foram de relativa fácil execução. Contudo, problemas de infraestrutura em ambiente residencial mostraram-se de difícil solução e impactaram sobremaneira no avanço do projeto. Conclui-se que projetos práticos, associados ao emprego de ferramentas de comunicação e estratégias de gerenciamento, são alternativas satisfatórias para aproximar o engenheiro em formação a cenários típicos do exercício profissional do século XXI. Atenção especial, contudo, deve ser dada às condições de infraestrutura de comunicação e computação para que a experiência seja positiva e eficaz em caso de projetos executados por equipes com colaboradores geograficamente dispersos.*

Palavras-chave: *Extensão universitária. Aprendizagem por projeto. Scrum. Desenvolvimento de aplicativo. Formação em engenharia.*

1 INTRODUÇÃO

Em um momento onde as relações de interdependência entre nações do mundo se evidenciam cada vez mais, a formação de profissionais preparados para lidar com a realidade do século XXI apresenta-se como um grande desafio, em particular para as escolas de engenharia. Em tempos de crises, em geral, encontram-se discursos indicando que esses cenários podem gerar oportunidades para inovações. No tocante ao recente cenário pandêmico da COVID-19, com diversos problemas associados não só à saúde mas também a todos os setores da sociedade, o mesmo pode ser associado a esse evento indutor de soluções. Notadamente, a conjuntura de calamidade mundial tem suscitado a união de esforços para o combate ao vírus e a doença. A rapidez com que a humanidade irá transpor essa situação depende, em grande parte, da habilidade com que o conhecimento, espalhado e diverso, seja reunido e usado com a eficiência e rapidez. Trata-se, portanto, de interação cooperativa onde diversos atores provêm “pedaços de conhecimento” de acordo com as respectivas especialidades para materializar soluções. Tão importante quanto a parte técnica são as questões relacionadas à gestão de pessoas e do próprio conhecimento. Reunir pessoas, concatenar ideias e coordenar a execução das atividades é uma tarefa desafiadora.

O cenário descrito acentuou a necessidade de cooperação, porém o mundo já estava gerando demandas onde, inevitavelmente, precisavam-se de métodos de trabalho que conseguissem harmonizar os recursos disponíveis, sejam técnicos, humanos e financeiros. O universo empresarial foi o primeiro a perceber essa importância e implantar o trabalho colaborativo entre setores. Como resultado, criaram-se métodos de gestão de pessoal, infraestrutura e de reaproveitamento de ferramental existente. Na área acadêmica, este entendimento tem sido ratificado, mediante a crescente produção de conhecimento científico de forma cooperativa envolvendo diversas áreas de conhecimento, o que tem provocado, cada vez mais, ações interdisciplinares e interinstitucionais.

Com base neste contexto, este artigo pretende registrar a experiência dos autores no desenvolvimento de um projeto que reúne diferentes especialidades na produção de uma ferramenta móvel para fomento à autonomia de pessoas com deficiência (PcDs). Tal ferramenta permitirá o mapeamento de deficiências em uma dada região e o pronto acesso a informações sobre a rede de serviços de apoio ao público-alvo nas áreas de saúde, educação, lazer, cultura e espaços públicos. Na execução do projeto, há o envolvimento de participantes de várias áreas de conhecimento que englobam diferentes institutos, faculdades, núcleos de pesquisa, uma coordenadoria institucional, e uma instituição filantrópica com atividades nas áreas de assistência social e saúde. O projeto iniciou com boa parte de suas ações, principalmente as de gerenciamento da equipe de desenvolvimento, sendo desenvolvidas em grupo nos espaços de laboratório da UFPA. Contudo, a necessidade do isolamento social e a suspensão de atividades presenciais da universidade, fizeram com que a equipe tivesse que se adaptar rápido e completamente à modalidade de trabalho remoto. Dessa forma, este texto apresenta um recorte da experiência de como está sendo gerido projeto, quais ferramentas estão sendo usadas na gestão e no desenvolvimento técnico, que rotinas de trabalho foram implementadas e como a experiência tem sido avaliada.

O restante deste texto está organizado como se segue. Na Seção 2, será abordada a cultura adotada para gestão do projeto. Na Seção 3, os *frameworks* utilizados no desenvolvimento da solução pretendida serão apresentados. A Seção 4 mostra o estado atual do trabalho, bem como os passos futuros. Nas Conclusões é realizada uma análise da experiência reportada, ao que se seguem breves agradecimentos e a lista de referências consultadas.

2 ESTRATÉGIAS E FERRAMENTAS PARA GESTÃO DO PROJETO

O aprendizado do aluno participante do projeto vai além das ferramentas e conhecimentos técnicos estritamente necessários para desenvolvimento do aplicativo em si. A equipe trabalha dentro de uma cultura colaborativa que adota as diretrizes de condução de projeto preconizadas pelo *framework* de gerenciamento Scrum, em associação com o aplicativo Trello e o sistema de controle de versão Git. A seguir, abordamos o essencial de cada um desses elementos, destacando como são usados na prática diária de condução do projeto.

2.1 Scrum

O Scrum é um *framework* de gerenciamento criado por Jeff Sutherland e Ken Schwaber (2020), alinhado com os princípios exarados no Manifesto Ágil (LAYTON; MORROW, 2018). Alguns autores são reticentes ao classificar o Scrum como uma metodologia, justificando que o Scrum assemelha-se mais a um conjunto de diretrizes gerais, permitindo adaptabilidade a diferentes contextos. O Scrum se dá mediante três elementos: artefatos, eventos, e papéis.

Os artefatos são listas de tarefas ou protótipos obtidos ao final de ciclos de desenvolvimento, a saber:

- *Backlog* do produto (*product backlog*): lista de todas as tarefas necessárias para a execução do projeto. Tal lista é dinâmica, podendo acomodar mudanças de requisitos em tempo de execução do projeto, e é ordenada por prioridade.
- *Backlog* do sprint (*sprint backlog*): lista de tarefas selecionadas de acordo com sua importância para serem realizadas no próximo período de desenvolvimento.
- Incrementos do produto: resultados ao final de um ciclo de desenvolvimento. Em geral, são versões incompletas, mas utilizáveis, do produto final almejado pelo projeto. Também chamadas de protótipos.

Os eventos típicos do Scrum são:

- *Sprint*: é o período ou a janela de desenvolvimento. Normalmente, os *sprints* variam de uma semana a dois meses. A ideia básica é que os *sprints* sejam curtos para que a equipe do projeto seja realimentada o mais cedo possível e de maneira permanente com avaliações feitas nos protótipos produzidos. Inicialmente, no projeto Dmóvel, os *sprints* eram de uma semana e, atualmente, são de duas semanas. A mudança foi uma adaptação em razão das características e disponibilidade de tempo dos membros da equipe do projeto.
- Scrum diário (*daily scrum*): é um encontro breve (15 minutos de duração, no máximo) que ocorre em cada dia de um *sprint*. Nesses encontros, cada um dos participantes deve responder às perguntas: quais tarefas do projeto fiz no dia anterior? Que tarefas pretendo fazer hoje? Antevjo algum problema, ou limitação técnica, que dificulte ou impeça atingir as metas do *sprint*? O objetivo do scrum diário é manter uma comunicação interna de boa qualidade, onde todos sabem o que todos estão fazendo e podem atuar em conjunto para suplantarem as eventuais dificuldades.
- Revisão do *sprint* (*sprint review*): ao final de cada *sprint*, a equipe deve apresentar quais atividades foram realizadas e, sobretudo, demonstrar o protótipo produzido. Essa

reunião tem por objetivo obter *feedback* sobre as funcionalidades do protótipo, avaliando se o produto está sendo desenvolvido na direção certa.

- Retrospectiva do *sprint* (*sprint retrospective*): pode-se dizer que a retrospectiva é a reunião de avaliação do *sprint*. Busca-se verificar o que funcionou ou não, o quê e como melhorar o desempenho da equipe. Por meio da retrospectiva do *sprint*, a equipe obtém elementos para se auto aprimorar.
- Planejamento do *sprint* (*sprint planning*): encerrado um *sprint*, o próximo é planejado, ou seja, é definida a meta e selecionadas as tarefas do próximo *sprint*. O planejamento sofre influência tanto da reunião de retrospectiva quanto da reunião de revisão do *sprint*.

Originalmente, os papéis em um Scrum são os seguintes:

- *Product owner*: é o elemento que têm a visão do produto e que é o responsável pelo *backlog* do produto. Ele é o derradeiro responsável por incluir/retirar tarefas no/do *backlog* do produto, estabelecer prioridades de tarefas e funcionalidades do produto, e indicar a conclusão de uma tarefa.
- *Scrum master*: é o responsável pela realização dos eventos Scrum, sinergia interna da equipe, e manutenção das boas condições de trabalho do time de desenvolvimento.
- Time de desenvolvimento: são os que efetivamente irão “meter a mão na massa”, realizar as tarefas selecionadas para o *backlog* do *sprint*. Segundo os preceitos do Scrum, as equipes mais eficientes devem ser auto organizáveis, autônomas, e competentes. Significa dizer que a equipe de desenvolvimento tem poder para decidir quando e como irá realizar uma tarefa do *sprint backlog*, e reúne membros cuja soma de habilidades lhes permitem atingir as metas do *sprint*.

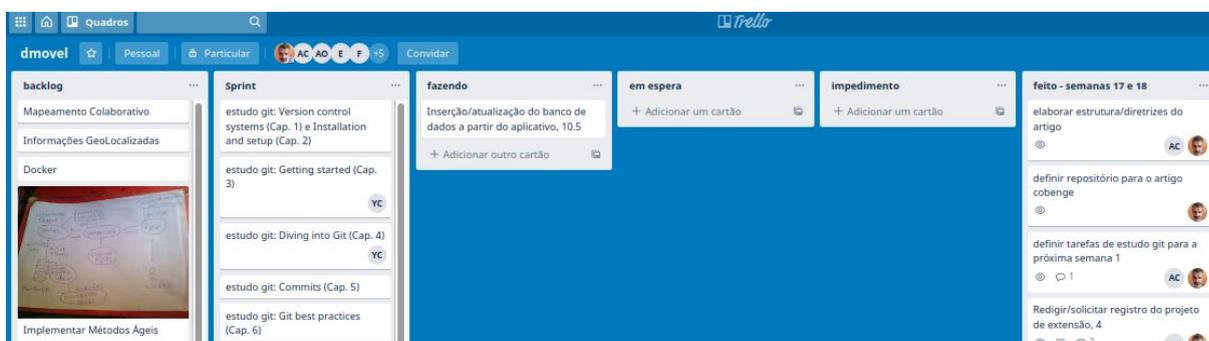
É comum o uso de quadros Kanban para visualizar o fluxo de execução de tarefas. Tais quadros são dispostos em colunas (variações são possíveis): *product backlog*, *sprint backlog*, *fazendo*, e *feito*. As tarefas inicialmente se encontram no *product backlog*, sendo paulatinamente movidas para a coluna *sprint backlog* como resultado das reuniões de planejamento de *sprint*. Durante o *sprint*, os membros da equipe de desenvolvimento deslocam as tarefas da coluna *sprint backlog* para a coluna *fazendo*, e após a conclusão para a coluna *feito*. Algumas observações sobre o manejo de tarefas no Scrum: a) uma vez definido o *sprint backlog*, o mesmo não deve ser alterado durante o *sprint*; b) é o time de desenvolvimento que decide quando e como uma tarefa do *sprint backlog* deve ser executada; e c) deve-se evitar a execução simultânea de tarefas. Tais observações têm por objetivo favorecer o foco no trabalho, respeitando o ritmo próprio do time de desenvolvimento na realização das atividades planejadas.

2.2 Trello

Para que qualquer membro da equipe possa acompanhar o trabalho sem que precise estar fisicamente na sala do projeto, optamos por manter uma réplica *online* do quadro físico de visualização do fluxo tarefas. Dentre as várias soluções disponíveis, escolhemos o aplicativo Trello (trello.com) por funcionar em plataformas *mobile* e *web*, e ter uma versão gratuita com conjunto de funcionalidades suficiente para as necessidades da equipe. As colunas do nosso quadro de atividades diferem um pouco das relacionadas na Seção 2.1. Baseado em

experiências progressas, a equipe definiu que as colunas seriam em número de 6: *backlog* (correspondendo a *product backlog*), *sprint* (correspondendo a *sprint backlog*), *fazendo*, *em espera*, *impedimento*, e *feito* (Figura 1). As atividades posicionadas na coluna *em espera* e *impedimento* são aquelas que em determinado momento de suas execuções precisam ser suspensas em decorrência do surgimento de dependências de terceiros, interrupção de acesso à rede, pane em equipamento, etc.

Figura 1 – Screenshot do quadro de atividades do projeto Dmóvel no Trello



Fonte: Autores

2.3 Git/GitLab

O projeto Dmóvel tem por meta a implementação de um aplicativo, requerendo o uso de um sistema para controle de suas diferentes versões. A equipe optou pelo sistema de controle de versão distribuído Git (GIT, 2020) em razão de seu desempenho, qualidade, flexibilidade de adaptação a diferentes fluxos de trabalho, e facilidade de aprendizado. Em resumo, o Git permite manter registro de comentários, datas e autorias de alterações de código, bem como recuperar versões anteriores em caso de necessidade.

O GitLab (GITLAB, 2020), por sua vez, é um aplicativo *online* baseado em Git que oferece recursos variados para desenvolvimento colaborativo de código: gerenciamento de repositórios com controle de acesso, ferramentas para definição e acompanhamento de *issues* (tarefas a serem feitas, funcionalidades a alterar/acrescentar/retirar, correção de *bugs*, etc.), página *wiki* para documentação do projeto, e suporte à requisição de fusão de código.

2.4 Discussão

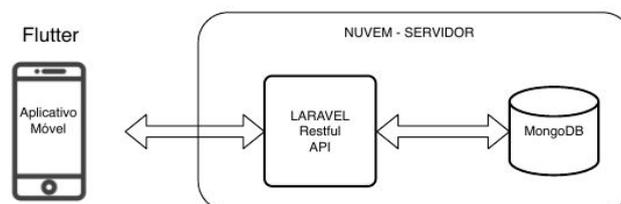
O contexto de trabalho do projeto Dmóvel favorece o aprendizado de competências imprescindíveis ao futuro engenheiro, dentre as quais destacamos o exercício de trabalho colaborativo, capacidade de priorização de atividades, manutenção de foco na realização de tarefas, e estímulo à iniciativa e responsabilidade individual. No caso específico do desenvolvimento de *software*, a vivência com Scrum e Git/GitLab confere ao discente uma vantagem profissional estratégica, uma vez que competências relacionadas à gestão projetos, desenvolvimento ágil, e suas ferramentas têm sido cada vez mais requisitadas, tanto pelo mercado quando pela academia. Embora cada colaborador tenha um foco de atuação específico no projeto, a dinâmica de desenvolvimento colaborativo descrita permite a todos o conhecimento geral do projeto. Todos visualizam o que todos fazem e podem contribuir e

opinar sobre problemas que não necessariamente estão sob sua responsabilidade. Isto permite a construção de um espírito uno de equipe, onde os resultados e problemas são de todos os colaboradores. Em tempo de pré pandemia, *daily scrums*, revisões, retrospectivas e planejamento de *sprint* eram realizados de forma presencial na universidade. Após a indicação de isolamento social, a gestão do trabalho está sendo realizada usando diversos mecanismos de teletrabalho. Os *daily scrums* acontecem regularmente via sistema de mensagem eletrônica em horários fixos e o acompanhamento das atividades se dá facilmente pelo Tello. Os outros eventos do Scrum, a saber, reuniões para revisões, retrospectivas e planejamento de *sprint*, vêm ocorrendo através de sistemas de videoconferência. Entretanto, há perdas no andamento do projeto. Estas estão principal e diretamente associadas à qualidade da infraestrutura de processamento e comunicação disponíveis no ambiente doméstico de cada colaborador, que por vezes dificulta a interação entre membros da equipe e retarda avanços no desenvolvimento do aplicativo. De maneira geral, a adaptação à nova rotina foi realizada sem muito esforço, em pouco mais de uma semana. Contudo, soluções para as questões da referida infraestrutura não se podem improvisar.

3 FRAMEWORKS DE DESENVOLVIMENTO

Em termos técnicos, o desenvolvimento do aplicativo segue metodologias de engenharia de software e estratégias de desenvolvimento ágil. A Figura 2 apresenta uma visão geral da ferramenta em desenvolvimento.

Figura 2 - Visão geral da software DMóvel



Fonte: Autores

A construção *backend* do aplicativo será realizada usando-se o *framework* Laravel. O Laravel (OTWELL, 2020) é um *framework* PHP, livre e *open-source* para o desenvolvimento de sistemas web, com o objetivo de criar aplicações seguras e com bom desempenho de forma estruturada e rápida. As principais características do Laravel são: a utilização da arquitetura MVC (model, view, controller); emprego do gerenciador de pacotes Composer; documentação oficial completa e intuitiva; compatibilidade com vários bancos de dados; e comunidade de desenvolvimento grande e ativa. Ele foi implementado no projeto para construção do serviço RESTful (back-end) da aplicação, que é a linha de comunicação entre os dados e a interface do usuário (Flutter).

Para a construção da aplicação móvel (*frontend*) será utilizado o *framework* Flutter, que é um pacote de desenvolvimento que possui recursos para a criação de aplicativos de modo rápido. Este *framework* é um SDK (*Software Development Kit*) de interface do usuário do Google para criar aplicativos compilados nativamente para dispositivos móveis, web e desktop a partir de um único código base (Google, 2020). O objetivo do Flutter é obter um desenvolvimento rápido, interfaces de usuário expressivas e bonitas e um desempenho nativo. Suas principais características são: a utilização da linguagem de programação *script* Dart;

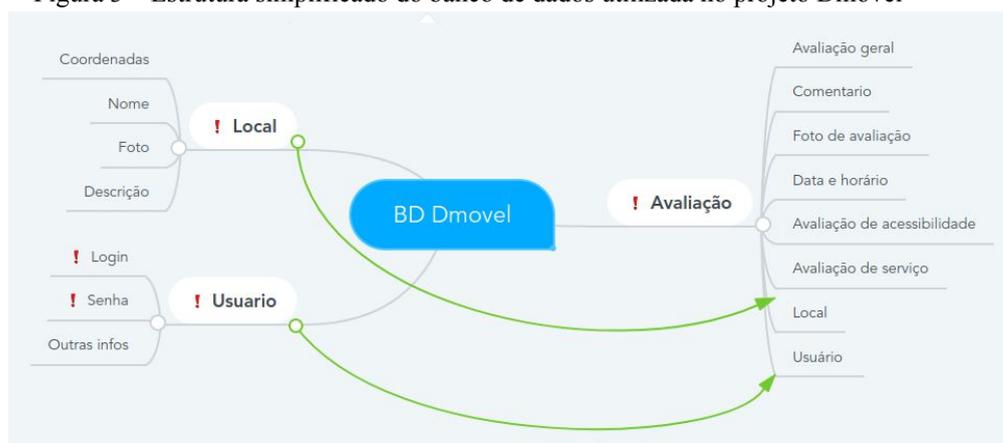
escrita de um único código base para gerar versão para qualquer plataforma; documentação extensa; vasto repositório de plugins, pacotes e bibliotecas; e 100% de integração com o Sistema Operacional Android. Ele foi escolhido para ser implementado no projeto para construção do aplicativo móvel (*front-end*) por suas características que facilitam e diminuem o tempo gasto em seu desenvolvimento e ter suporte à integração de recursos de acessibilidade que é um dos focos do projeto.

O *manuseio* dos dados será realizado usando um software de banco de dados chamado de MongoDB, o qual é um banco de dados baseado em documentos, de propósito geral, código aberto, gratuito e de alta performance (MONGODB, 2020). Suas principais características são: armazenamento dos dados no formato *JSON* (*JavaScript Object Notation*); modelo de armazenamento não relacional de dados (*NoSQL*); documentação oficial extensa; e desenvolvimento voltado para grande quantidade de dados (*Big Data*). Pelo projeto ter o intuito de armazenar a maior quantidade de dados possíveis ao longo do tempo de sua utilização, o MongoDB foi escolhido pela sua alta performance no tratamento de grande quantidades grandes de dados.

4 ATUAL ESTADO DO DESENVOLVIMENTO E PRÓXIMOS PASSOS

Atualmente, a equipe tem se dedicado principalmente ao *backend* do aplicativo, ou seja, à infraestrutura de *software* do lado do servidor necessária para viabilizar o sistema. Um ambiente de teste foi montado na sala do projeto contendo um servidor web local, acessível por uma rede sem fio. Neste servidor local, foi instalado um banco de dados (MONGODB) com informações fictícias de usuários, locais, e avaliações, cuja estrutura simplificada é apresentada na Figura 3.

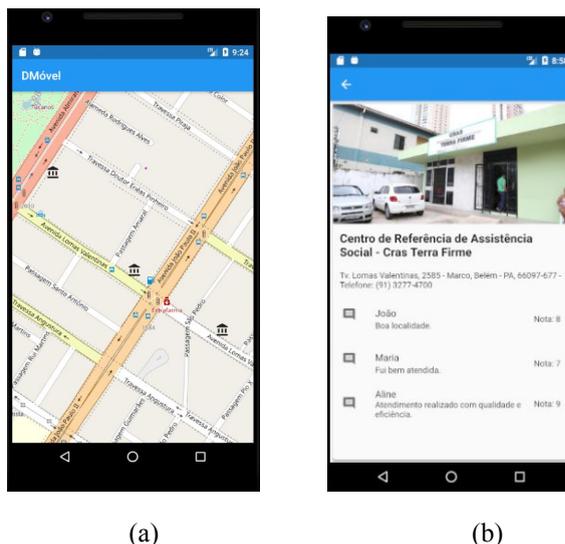
Figura 3 – Estrutura simplificado do banco de dados utilizada no projeto Dmóvel



Fonte: Autores

Uma interface simples para consulta remota ao banco de dados com informações fictícias foi criada e pode ser instalada em celulares Android. Essa interface é essencialmente um mapa (OPENSTREETMAPS, 2020) em que os locais cadastrados aparecem indicados (Figura 4 (a)). Ao ser selecionado um local, o aplicativo apresenta uma nova tela contendo informações básicas e avaliações associadas (Figura 4 (b)). Do aplicativo é possível também fazer consulta ao banco de dados usando filtros, selecionando informações de interesse, como por exemplo as avaliações de um determinado local realizadas por usuário e/ou datas específicas.

Figura 4 – Interface do aplicativo Dmóvel. (a) Mapa para consulta de locais cadastrados; (b) Tela com informações e avaliações associadas



Fonte: Autores

Os próximos passos no desenvolvimento do aplicativo compreendem a adição da funcionalidade de inserção de informações no banco de dados a partir do dispositivo móvel, tais como cadastro de novos locais, envio de novas avaliações com texto e fotos, a criação de sistema de cadastro e autenticação de usuários, e elaboração de protótipo de interface de usuário visando avaliação com público-alvo. Pretende-se, então, iniciar um projeto piloto com discentes vinculados à CoAccess, ainda em âmbito interno da UFPA, que atuarão como equipe de apoio para o teste de funcionalidades e aprimoramento da interface do usuário.

5 CONCLUSÕES

Este artigo apresentou o registro de um projeto de extensão em andamento para a criação de um aplicativo voltado para pessoas com deficiência e sua adaptação para o contexto de trabalho remoto em razão do isolamento social provocado pela pandemia da COVID-19.

Verificou-se que são três os principais elementos que estruturam as atividades do projeto conferindo-lhes ritmo e foco: cultura de trabalho, *frameworks* de desenvolvimento, e infraestrutura computacional e de comunicação. A cultura de trabalho envolve a adoção do Scrum para o gerenciamento de atividades, o uso de quadros no Trello em lugar do *kanban* físico, e o emprego de sistemas Git para controle de versão de *software*. Os *frameworks* reduzem a curva de aprendizado e proporcionam qualidade à produção. Contudo, a infraestrutura computacional e de comunicação é o elemento-chave através do qual efetivamente o trabalho remoto se dá, sendo essencial haver qualidade mínima que viabilize a interação fácil entre colaboradores.

A experiência relatada demonstra que projetos de cunho prático, ou seja, que pretendam gerar uma solução em resposta a um problema ou necessidade concreta são um dos instrumentos que as escolas de engenharia podem lançar mão para proporcionar aos seus

alunos experiências mais ricas no sentido da interdisciplinaridade e do trabalho colaborativo, em particular, à distância.

Dadas as características geográficas e socioeconômicas brasileiras, possíveis investigações nessa mesma linha temática podem incluir estratégias e soluções técnicas para mitigar os referidos problemas de infraestrutura de maneira que projetos continuem a ser executados a despeito dos entraves de comunicação.

Agradecimentos

Os autores gostariam de registrar seu agradecimento ao apoio ao projeto dado pela Coordenadoria de Acessibilidade (CoAcess) da Universidade Federal do Pará, e ao Projeto D+Eficiência pelo apoio financeiro parcial.

REFERÊNCIAS

GIT Community. **Git --fast-version-control**. Disponível em: <https://git-scm.com/>. Acesso em: 25 abr. 2020.

GITLAB. **GitLab Docs: User Docs**. Disponível em: <https://docs.gitlab.com/ee/user/index.html>. Acesso em: 25 abr. 2020.

GOOGLE. **Flutter**. Disponível em: <https://flutter.dev/>. Acesso em 25 abr. 2020.

LAYTON, Mark C.; MORROW, David. **Scrum for dummies**. 2ª edição, New Jersey: John Wiley & Sons, 2018.

MONGODB. Disponível em: <https://www.mongodb.com>. Acesso em: 25 abr. 2020.

OPENSTREETMAP Foundation. **OpenStreetMap**. Disponível em: <https://www.openstreetmap.org/>. Acesso em: 25 abr. 2020.

OTWELL, Taylor. **Laravel**. Disponível em: <https://laravel.com/>. Acesso em: 25 abr. 2020.

SCHWABER, Ken; SUTHERLAND, Jeff. **The Scrum Guide - The definitive guide to Scrum: the rules of the game**. Disponível em: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>. Acesso em: 25 abr. 2020.

DMÓVEL: LEARNING SKILLS IN SOFTWARE DEVELOPMENT, PROJECT MANAGEMENT, AND REMOTE WORK

Abstract: *Engineering students should be trained to face the challenges of the XXI century, which mostly require complex solutions as a result of collaborative and interdisciplinary work. The current coronavirus pandemic scenario has reinforced the need to develop communication and team management skills, especially if coworkers are geographically apart. This work depicts the experience of the team in executing a project aimed to develop a mobile application, whose activities were adapted to be accomplished remotely. The team is interdisciplinary, composed by undergrad students, professors and professionals from the areas of computer engineering and science, psychology, and health, in conjunction with the accessibility department of the host institution. The intended application is based on MongoDB, Flutter, and Laravel. We adopt Scrum, Git systems, and Trello for activity management. After a period of adaptation, our team got used to daily meetings by instant messaging, and to fortnightly video conferences for evaluation and planning. The team found somewhat easy to manage text production, and to follow studies remotely. However, home infrastructure proved to be an issue with considerable impacts upon the evolution of the project. The reported experience indicates that practical projects, associated with digital communication tools and management strategies are suitable to bring real context regarding remote work to engineering educational environments. On the other hand, for effectiveness, special attention should be paid to computer and communication infrastructure at collaborator's home side.*

Keywords: *Undergrad projects. Project-based learning. Scrum. Application development. Engineering education.*