

# Learning by Doing em Fábrica de Software: Relato de uma Experiência no Mestrado Profissional em Computação Aplicada

Tainara Lucateli Bernardi, Alexandre Lazaretti Zanatta,  
Jucélia Giacomelli Beux, Daiana Biduski, Éricles Andrei Bellei

Programa de Pós-Graduação em Computação Aplicada (PPGCA)  
Universidade de Passo Fundo – Caixa Postal 611 – 99.001-970 – Passo Fundo – RS

{tainarab, zanatta, 68428, 130011, 168729}@upf.br

***Abstract.** Practical teaching approaches, such as Learning by Doing, are fundamentals in the academic training offered by professional master's programs, which seek to apply the obtained knowledge to a field of specific professional action. This paper reports the experience, within a Software Factory, executed in a discipline of the professional masters in Applied Computing, promoting academics integration with real work processes and problems. Formulated reflections and indications show increase of synergy among groups, creativity development, ability to plan among others. The experience was valid as an application in teaching practice and student training.*

***Resumo.** Abordagens práticas de ensino, como Learning by Doing, são fundamentais na formação acadêmica proposta por programas de mestrado profissional, os quais buscam a aplicação orientada para um campo de atuação profissional específico. O presente artigo relata a experiência, no âmbito de uma Fábrica de Software, realizada em uma disciplina do mestrado profissional em Computação Aplicada, promovendo a integração dos acadêmicos com processos de trabalho e problemas reais. As reflexões e indícios levantados apontam para o aumento da sinergia entre os grupos, o desenvolvimento da criatividade, capacidade de planejar entre outros. Foi válida e proveitosa a experiência como aplicação na prática docente e na formação discente.*

## 1. Introdução

A transformação na forma de desenvolver software faz com que várias áreas sejam conectadas e integradas para atender às demandas do mercado. Nesse contexto, a busca por software que atende às necessidades e desejos dos clientes aumentam, transformando as maneiras de desenvolvimento e entrega. A demanda e complexidade no cenário de desenvolvimento de software exigem maior produtividade, associada a um elevado grau de qualidade. Para tanto, é importante observar e enriquecer os processos de aprendizagem utilizados para difundir esse novo contexto de desenvolvimento. Tais questões podem ser aperfeiçoadas em um ambiente educacional com a integração de abordagens práticas de aprendizagem e a aplicação de determinadas técnicas e modelos de funcionamento de Fábricas de Software [Ahmad *et al.* 2014].

Nesse contexto, o conceito da metodologia *Learning by Doing* (LBD) exemplifica uma forma de aprendizagem experiencial, em que os estudantes ganham experiência prática na resolução de problemas do mundo real. Em sua teoria, Dewey (2011) argumenta que para a educação ser mais efetiva, os indivíduos devem receber oportunidades de aprendizagem que lhes permitam vincular o conteúdo presente a experiências e conhecimentos anteriores. A educação também deve preparar os alunos para assumirem uma parte plena e ativa na formação de sua sociedade futura.

A metodologia LBD baseia-se na abordagem construcionista e promove um ambiente de aprendizagem orientado para o aluno. Ela promove o desenvolvimento de habilidades e a aprendizagem de informações por meio da implementação prática de conceitos ou conteúdos [Shank *et al.* 2013]. Nessa metodologia, mentores, que podem ser os professores, auxiliam os alunos no curso de seu trabalho ao fornecer sugestões e não soluções prontas. Ao contrário do modo de ensino convencional de sala de aula, a dependência entre estudante e professor é baixa, o que permite a orientação e o autocrescimento do aluno de maneira prática e consistente [Karri and Kode 2011].

Diante do exposto, este estudo tem como objetivo apresentar a experiência dos discentes do curso de mestrado profissional em computação aplicada da Universidade de Passo Fundo. Aplicaram-se os conceitos de fábrica de software associados à metodologia *Learning by Doing* com o desafio de implementar um software de agendamento online para um departamento dentro da mesma universidade.

O texto está organizado da seguinte maneira: na seção 2 são apresentadas algumas definições conceituais e estudos relacionados. A seção 3 descreve o projeto e detalha a organização do trabalho. Na seção 4 são expostas discussões sobre o esforço e as escolhas feitas. Na seção 5 são apresentadas algumas dificuldades e lições aprendidas com a execução do projeto. A seção 6 encerra com algumas considerações sobre a experiência obtida, limitações e trabalhos futuros.

## **2. Fundamentação Teórica e Trabalhos Relacionados**

Uma fábrica de software é um ambiente definido a partir de processos para realizar as atividades relacionadas ao ciclo de vida de software, abrangendo determinadas práticas de manufatura industrial, com processos, uso de normas e modelos de qualidade, cumprimento de cronogramas e entregas. Os conceitos e as propostas de fábrica de software estão relacionados às linhas de produtos de software, às linguagens para domínios específicos, ao uso de ferramentas computacionais específicas e ao uso de processos [Borsoi 2008].

[Fernandes and Teixeira 2007] afirmam que uma fábrica de software é um processo estruturado, controlado e melhorado de forma contínua, orientado para o atendimento de demandas de natureza e escopo distintas, visando a geração de produtos de software que atendam às necessidades dos usuários da forma mais produtiva e econômica possível. No ambiente acadêmico, pode haver a inserção dos conceitos de fábrica de software com o objetivo de integração entre teoria e prática, onde o discente tem oportunidade de vivenciar problemas reais. [Ahmad *et al.* 2014] explicam que as fábricas de software podem servir como suporte de testes para ideias de engenharia de software e uma fonte para pesquisa básica de desenvolvimento de software científico. Explicam também que essa prática pode ser um veículo educacional para universidades,

onde os artefatos produzidos na fábrica servem para melhorar a aprendizagem e fornecer materiais didáticos em estreita colaboração com a indústria.

O estudo de [Karri and Kode 2011] apresenta um programa de treinamento desenvolvido pelos autores a fim de melhorar a qualidade da educação em Tecnologia da Informação nas faculdades indianas de engenharia no ano de 2008. A metodologia adotada para esse processo foi a LBD, com o objetivo de fornecer aos membros da faculdade conhecimento sobre habilidades e ferramentas necessárias para integrar a tecnologia no ensino e na aprendizagem. A abordagem LBD foi customizada de acordo com o programa elaborado. Em seus resultados, observaram que alunos que passaram pelo treinamento customizado aprenderam de forma mais rápida, demonstraram melhorias em seus níveis de habilidade e expressaram maior satisfação com a experiência de aprendizagem do que aqueles que expostos ao aprendizado sem personalização. Dessa forma, a metodologia LBD mostrou-se satisfatória, não substituindo o método utilizado pela faculdade, mas sim complementando-o de forma refinada.

[Mansker *et al.* 2016] investigaram como a incorporação da experiência prática com aprendizado de serviço – *service learning* – poderia influenciar a aprendizagem e o interesse por serviço comunitário nos alunos integrantes de uma turma de gerenciamento de reuniões e eventos do programa Hospitality and Restaurant Management em uma universidade nos Estados Unidos. Os resultados indicaram que a aprendizagem de serviços juntamente com a atividade prática aumenta a compreensão dos participantes sobre os conceitos abordados no curso, além de facilitar a compreensão de tempo e esforço para planejar um evento, aumentando o desempenho acadêmico dos participantes.

As implicações para os alunos incluem a melhoria da realização acadêmica resultante de um aumento do nível de interesse no curso e maior nível de aprendizagem, levando a uma experiência acadêmica mais positiva. Os estudantes acreditam que a abordagem aumenta sua capacidade de aplicar conceitos aprendidos a novas situações, sendo capaz de direcionar sua escolha profissional para um mercado onde eles serão inseridos já possuindo conhecimentos prévios, baseados na experiência de campo.

A LBD pode ser utilizada em diversas áreas do conhecimento, permitindo que os alunos se envolvam no processo e busquem soluções próprias para tarefas práticas. A experiência por meio dessa aprendizagem acontece de maneira diferente para cada indivíduo que, guiado por suas experiências passadas, conhecimento pessoal e características individuais, pode interagir com a informação de diferentes maneiras e assim, alcançar resultados distintos, tornando o ambiente de sala de aula uma verdadeira representação do mundo real.

O trabalho de [Ahmad *et al.* 2014] relata as impressões e lições aprendidas com a criação de um laboratório de fábrica de software em uma universidade da Finlândia, concebida com o objetivo de facilitar a aprendizagem dos estudantes e atender às necessidades da indústria. No estudo, é examinado como os fatores no ambiente de aprendizado da fábrica de software podem afetar os alunos ao longo de um curso relacionado ao assunto, tendo em vista a avaliação do projeto sob a perspectiva dos estudantes. Dentre as técnicas empregadas nesse processo, destaca-se o uso da aprendizagem colaborativa por meio da LBD, que juntamente com outras técnicas, proporcionou resultados significativos na aprendizagem dos estudantes, em suas realizações acadêmicas e no ganho de habilidades profissionais.

### 3. O Projeto Desenvolvido

O projeto atual foi desenvolvido na disciplina de Fábrica de Software com carga horária de 30 horas, divididas em 15 encontros semanais, ofertada uma vez ao ano em um Mestrado Profissional em Computação Aplicada. O objetivo principal da disciplina envolveu a construção do conhecimento e o entendimento dos conceitos e atividades realizadas no processo de trabalho de fábricas de software. A adoção da metodologia LBD visou transformar a abordagem da aula em uma experiência de aprendizagem ativa, em que os alunos participam ativamente na busca pelo conhecimento, deixando de serem apenas ouvintes de conteúdo para serem partícipes do processo. Nesse modelo, os alunos, com autonomia, se engajam na realização de projetos que os proporcionam aprendizados com compreensão conceitual, experiência e desenvolvimento de suas habilidades [Revans 2011].

No início dos trabalhos do projeto, em março de 2017, os participantes tomaram diversas decisões, tais como: os papéis dentro do projeto e da estrutura da fábrica de software, o processo a ser seguido, as ferramentas a ser utilizadas, as formas de comunicação, entre outras. Para organizar os trabalhos, o primeiro passo foi definir um plano de projeto com todas as informações necessárias para que o trabalho fosse conduzido da melhor forma possível. Esse planejamento continha uma visão geral do projeto: objetivos, escopo, papéis e responsabilidades, riscos identificados, um plano de comunicação entre as equipes de trabalho e o cliente e, finalmente, o cronograma com as entregas. O prazo final do projeto foi junho de 2017.

O processo avaliativo da disciplina teve três momentos. O primeiro, ao final da disciplina, os discentes realizaram uma descrição (relatório) com suas percepções sobre a disciplina em si e a metodologia utilizada. O segundo, no último encontro, fez-se uma discussão em classe com todos participantes, tendo o professor da disciplina como moderador. O cliente foi chamado para avaliar – conforme as suas experiências – o software entregue. Vale lembrar que uma avaliação institucional foi realizada. Esses dados foram recolhidos e analisados utilizando algumas práticas da Análise de Conteúdo [Bardin 2005].

A fim de organizar e distribuir as atividades, fez-se a análise do perfil de cada integrante do projeto com o intuito de identificar as habilidades e possibilitar a distribuição de tarefas condizentes aos interesses particulares. Os discentes foram organizados em equipes de trabalho, em que cada um assumiu algum papel dentro do contexto de fábrica e, assim, formaram-se três equipes, com as finalidades de análise de requisitos, desenvolvimento, testes e qualidade de software. Para a distribuição de papéis, levou-se em consideração o conhecimento pessoal, as experiências anteriores e as características individuais de cada aluno, seguindo o que preconiza a LBD. Complementando os papéis, um aluno assumiu a responsabilidade de gerente de projeto, com o objetivo de planejar, organizar o cronograma, acompanhar o andamento dos trabalhos e gerenciar o plano de comunicação. Cabe ressaltar que nenhum dos integrantes havia trabalhado anteriormente com outro membro do projeto.

Participaram dez alunos com graduação na área da tecnologia da informação e faixa etária entre 20 e 40 anos, com experiência entre 1 e 5 anos em projetos de desenvolvimento de software. Embora sendo um projeto acadêmico de pós-graduação, buscou-se uma experiência próxima dos desafios de um contexto real de fábrica de software, com a demanda de um software a ser desenvolvido para um cliente real e com

prazos determinados para as entregas de cada tarefa. Após a compreensão geral do escopo do software a ser desenvolvido, os alunos dividiram-se em equipes de trabalho para iniciar as atividades do projeto.

As atividades foram desenvolvidas conforme a estrutura de trabalho utilizada na fábrica de software envolvida no projeto (Figura 1), a qual utiliza o modelo de processo de desenvolvimento iterativo e incremental, acrescido de algumas práticas ágeis. Na figura, podem ser observadas as atividades definidas no processo de desenvolvimento utilizado pela fábrica de software. As equipes do projeto foram definidas e organizadas conforme as áreas de processos abordadas na engenharia de software, as quais são identificadas no fluxo apresentado. Inicialmente, as atividades referem-se ao processo de engenharia de requisitos, na continuidade ao desenvolvimento (projeto e implementação) e, posteriormente, às atividades de verificação e validação.

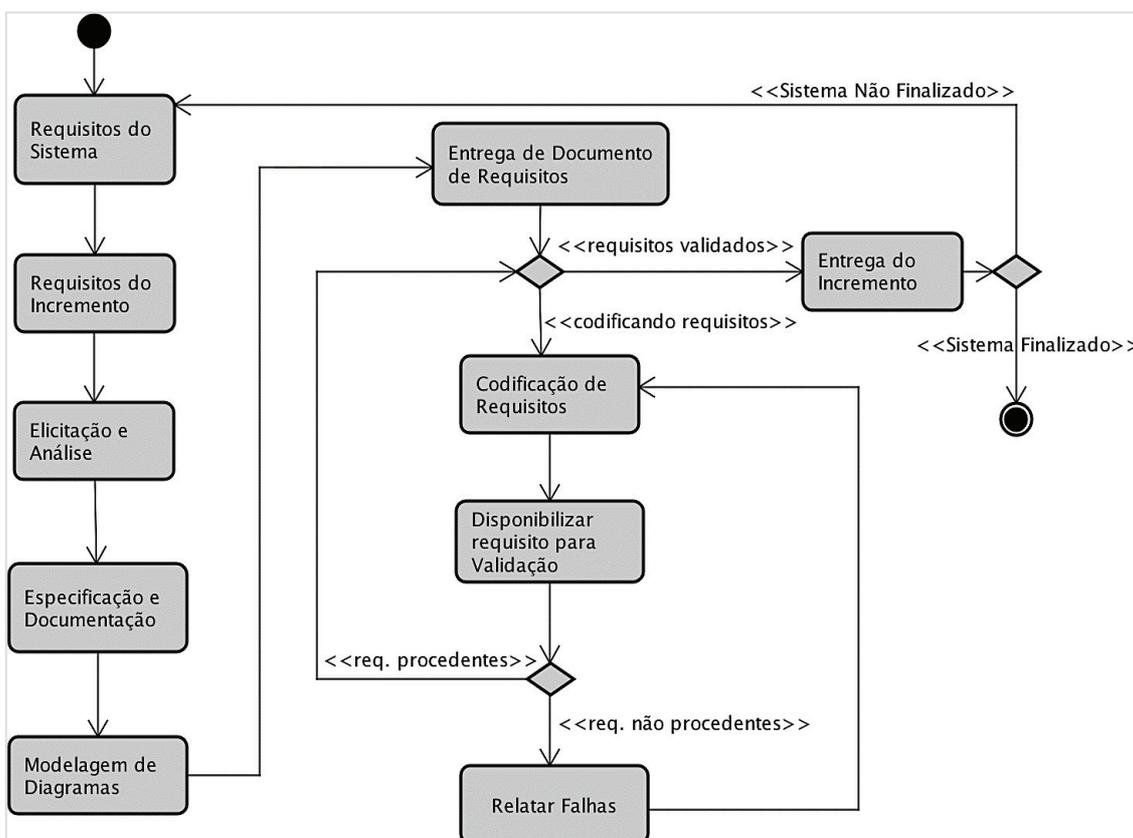


Figura 1. Fluxo de atividades da Fábrica de Software

A equipe responsável pelo processo de análise de requisitos iniciou seu trabalho fazendo a elicitação e análise dos requisitos do software. O processo de especificação e documentação dos requisitos foi construído gradativamente conforme o andamento dos ciclos de trabalho (*sprints*) definidos no projeto. Inicialmente, um formato robusto de documentação foi utilizado, com descrições detalhadas acompanhadas de alguns diagramas da UML (casos de uso, atividades e diagrama de classes). Além disso, *mockups* com a representação das informações de cada funcionalidade foram utilizados para documentar os requisitos funcionais do software. Em um segundo momento, a equipe substituiu o modelo de documentação, passando a utilizar o formato de *User Stories*,

juntamente com diagramas de UML e *mockups*. Desse modo, a equipe aprofundou seus conhecimentos em duas técnicas diferentes de especificação e documentação de requisitos, podendo visualizar as diferenças existentes entre elas. Todos o processo organizacional, como a definição das atividades, a escolha das técnicas utilizadas e a distribuição dos papéis, foram autogerenciadas pelos integrantes da própria equipe, possibilitando que cada integrante realizasse atividades condizentes com suas habilidades e disponibilidade de tempo. Em consequência, todos demonstraram empenho e dedicação para que as atividades fossem realizadas da melhor forma possível.

A equipe de desenvolvimento iniciou o projeto com a estruturação do ambiente de desenvolvimento, bem como a definição da gestão de configuração. Neste projeto, o banco de dados utilizado foi Oracle, o *backend* foi desenvolvido em Java com o framework Spring<sup>1</sup> e o *frontend* com AngularJS<sup>2</sup>. A cada ciclo de desenvolvimento, a equipe recebia os requisitos da equipe de análise, codificava e aplicava os testes de desenvolvimento. Depois disso, encaminhava a funcionalidade para uma nova equipe no projeto. A gestão de configuração foi organizada utilizando a ferramenta GitLab<sup>3</sup>, a qual contribui na qualidade do trabalho colaborativo realizado pela equipe. Toda a organização e gestão das atividades realizadas na equipe de desenvolvimento foram gerenciadas pelos próprios integrantes da equipe. Os mesmos consideraram os conhecimentos técnicos e as habilidades de cada membro para a definição dos papéis e responsabilidades na equipe. Esta análise e organização contribuiu significativamente para o bom andamento das atividades do projeto.

O processo de teste e qualidade de software ficou sob responsabilidade de uma terceira equipe, a qual foi analisou, documentou, implementou e executou os testes de cada incremento do software. Para tanto, foram efetuados diferentes tipos de teste que contribuíram com a qualidade do produto de software final. Os problemas identificados eram reportados ao gerente do projeto que, imediatamente, via ferramenta de gerenciamento de atividades, repassava aos desenvolvedores as demandas de ajustes.

O tempo reservado para as aulas semanais da disciplina foi utilizado para reuniões das equipes de trabalho e alinhamento das atividades gerais do projeto. Acompanhados pelo professor, os membros decidiram sobre o andamento de suas tarefas, verificação de dependências, entregas de artefatos e outras necessidades. As equipes tiveram autonomia, como preconiza a LBD, para fazer a divisão de tarefas e seu autogerenciamento, acompanhadas pelo gerente de projeto. O período disponível para o projeto foi de quatro meses e, a partir desse prazo, foram estipuladas datas para entregas parciais e a entrega final do software. As reuniões posteriores com o cliente foram feitas pelo gerente de projeto e pela equipe de análise de requisitos. Durante as reuniões, foram apresentados protótipos da interface gráfica do software para detalhamento das funcionalidades ao cliente, que verificavam a conformidade à sua demanda e opinavam sobre necessidades de ajustes. Os protótipos da interface do software, assim como os *mockups* utilizados na documentação de requisitos, foram elaborados no software Balsamiq<sup>4</sup>.

---

<sup>1</sup> <https://spring.io>

<sup>2</sup> <https://angularjs.org>

<sup>3</sup> <https://about.gitlab.com>

<sup>4</sup> <https://balsamiq.com>

Assim como acontece em um ambiente de fábrica de software [Fernandes 2009], os requisitos do software desenvolvido na disciplina foram divididos por ordem de importância e urgência, separados em duas modalidades: requisitos do usuário e requisitos do administrador do software. Os requisitos essenciais para o funcionamento do software foram movidos para o início da lista de prioridades, enquanto as sugestões de melhorias e os ajustes necessários foram analisados e organizados para um segundo momento. Para os requisitos com baixa complexidade, reservou-se uma semana para o desenvolvimento e teste, enquanto outros, como requisitos fundamentais e requisitos com alta complexidade, tinham 20 dias para o desenvolvimento e entrega parcial.

No contexto, muitas das atividades foram desenvolvidas em ambiente distribuído, possibilitando aos discentes uma nova experiência, pois nos dias atuais, muitas empresas e fábricas de software adotam metodologias de trabalho que independem da presença física dos colaboradores no ambiente corporativo [Fernandes 2009]. Desse modo, para facilitar a organização, colaboração e comunicação entre os participantes, foram utilizadas algumas ferramentas de trabalho, as quais fazem parte do portfólio de ferramentas da fábrica de software da instituição. O repositório e o controle de versionamento de artefatos da documentação do projeto e dos códigos fonte foram gerenciados com o GitLab. A organização do fluxo de realização das tarefas aconteceu na ferramenta Trello<sup>5</sup>. Além disso, a comunicação no formato de listas de discussão, fórum e chats entre os participantes, aconteceu na ferramenta Slack<sup>6</sup> e nos grupos criados no WhatsApp<sup>7</sup>.

#### 4. Discussão

Além da consolidação dos conteúdos da disciplina, os alunos que participaram desse projeto aproveitaram a vivência para conhecer os processos de trabalho de uma fábrica de software e conviver com a resolução de problemas reais. A forma de trabalho utilizada na disciplina materializa o objetivo do mestrado profissional de estreitar as relações entre as universidades e o setor produtivo, visando a capacitação de profissionais nas diversas áreas do conhecimento, mediante o estudo de técnicas, processos ou temáticas que atendem a alguma demanda do mercado de trabalho [MEC 2009]. Essa modalidade de pós-graduação *stricto sensu* apresenta forte relação didática com as abordagens práticas de ensino, como *Learning by Doing*, pois ambos buscam a aplicação orientada para um campo de atuação profissional, em que a aprendizagem experiencial é utilizada para que os estudantes adquiram experiências práticas na resolução de problemas do mundo real.

A partir da experiência, várias reflexões e discussões foram realizadas com os alunos com o objetivo de analisar o aprendizado construído por meio da vivência real do ambiente de trabalho. O trabalho distribuído e a colaboração foram as características mais preocupantes do projeto, considerando o agravante de que nenhum dos participantes, nem mesmo o gerente de projetos, tinham experiência em trabalhar com esse perfil de equipe. Nesse cenário, as ferramentas de trabalho auxiliaram o processo de comunicação e de gerenciamento das atividades a desenvolvidas em cada equipe dentro do contexto de fábrica de software. A comunicação com o cliente, predominantemente feita pelo gerente de projetos e pela equipe de análise de requisitos, colocou construtivamente os alunos no

---

<sup>5</sup> <https://trello.com/home>

<sup>6</sup> <https://slack.com>

<sup>7</sup> <https://web.whatsapp.com>

ambiente real de trabalho, onde puderam perceber que a comunicação e o processo de negociação são importantes e decisivos para o sucesso de um projeto.

O *Learning by Doing* traz uma nova perspectiva de aprendizagem pelo fato de promover o autodesenvolvimento, o qual foi enfaticamente relatado pelos discentes participantes da disciplina. Alguns dos alunos não tinham experiência profissional e nunca haviam trabalhado em um ambiente real de produção de software. Os mesmos relataram que os desafios inicialmente pareciam assustadores, mas que a vivência das atividades ampliou significativamente o conhecimento e o desenvolvimento particular. Um fator a ser considerado da experiência vivenciada refere-se aos diferentes perfis de profissionais integrantes do projeto. Ressalta-se que ninguém conhecia a forma de trabalho utilizada por cada um de seus colegas, o que poderia ter dificultado o andamento do trabalho. Porém, os relatos demonstram que isso agregou muito crescimento pessoal e profissional no que se refere ao trabalho colaborativo e distribuído.

## 5. Desafios e Lições Aprendidas

Durante o andamento do projeto, a equipe enfrentou alguns desafios relacionados a cronograma, planejamento de atividades, comprometimento, domínio de tecnologias, heterogeneidade da equipe e dinâmica de trabalho. Estes desafios levaram a algumas lições aprendidas consideradas importantes para a execução de novos projetos.

Em relação ao cronograma e o planejamento, o cumprimento dos prazos foi um dos maiores desafios enfrentados no projeto. Inicialmente, fez-se o planejamento considerando quatro horas semanais de trabalho para cada integrante do projeto. Porém, nem todos os discentes conseguiram disponibilizar o tempo planejado para a realização das atividades. Por outro lado, alguns relataram maior dedicação do que o tempo planejado, dedicando aproximadamente seis horas semanais. Foi necessário ajustar o cronograma e adiar as entregas ao cliente, pois a tarefa de um integrante acabava comprometendo o cronograma de realização das atividades subsequentes. Houve incremento na demanda de horas de trabalho, ocasionado pela dificuldade de integração do software desenvolvido com a estrutura da Fábrica, sendo esse o principal problema de cunho técnico do projeto.

Oficialmente, a disciplina finalizou em junho de 2017. Todavia, como o projeto não foi concluído, os alunos se comprometeram em terminar o projeto após o encerramento da disciplina, mesmo estando em período de férias. Isso implicou trabalho extra de motivação para a sua conclusão. Após 15 dias do encerramento da disciplina, foi possível disponibilizar uma versão funcional do software para a análise do usuário. A versão final do software foi entregue ao cliente 30 dias após a data planejada e estipulada durante a estruturação da disciplina. Deste modo, foi possível confirmar a existência de readequações no planejamento das atividades da disciplina.

Particularmente nesse projeto, pelo fato de trabalhar com discentes de uma disciplina de mestrado, o comprometimento estava fortemente relacionado com o interesse particular de cada aluno. Foi difícil granjear o comprometimento de todos os integrantes numa mesma proporção, considerando que cada indivíduo tem uma percepção individual e objetivos diferentes sobre o comprometimento na realização das tarefas. Logo no início do projeto, um dos integrantes da equipe de testes e qualidade de software solicitou desligamento do programa de mestrado. Esse fato impactou diretamente no andamento do projeto, pois houveram dificuldades no repasse das atividades que até então

estavam sob sua responsabilidade. Dessa forma, foi necessário readequar os integrantes da equipe e aperfeiçoar a comunicação para que não se tornasse uma vulnerabilidade no projeto. Ao se aproximar da conclusão das atividades, novas dificuldades foram enfrentadas nessa mesma equipe de trabalho, todas elas decorrentes do comprometimento e conhecimento técnico dos integrantes. As demais equipes não apresentaram problemas. Nessa perspectiva, acredita-se que o interesse individual de cada integrante foi fator preponderante para o sucesso na realização das atividades.

A falta de domínio das tecnologias escolhidas e a pouca experiência em desenvolvimento de software, por parte de alguns componentes da equipe, foram fatores desafiadores. Parte do trabalho aconteceu remotamente e os integrantes não estavam habituados a trabalhar juntos e em um projeto colaborativo com a equipe distribuída. Nesse fator, a comunicação da equipe foi um dos elementos mais proeminentes para compartilhamento de soluções. Soma-se o fato de que a ocasião foi a primeira experiência com um projeto de trabalho diferenciado em uma disciplina do mestrado e, por isso, suscitava dúvidas e inseguranças quanto ao resultado final.

Dentre as várias lições aprendidas, destaca-se a importância de estabelecer canais de comunicação eficientes entre todos os integrantes e utilizar ferramentas para gerenciar as atividades de cada equipe do projeto, como destacam [Fuks and Pimentel 2011]. Essa organização facilitou o trabalho dos participantes e levou ao sucesso das atividades. Outro destaque deve-se ao trabalho com grupo heterogêneo de pessoas, as quais não se conheciam e apresentavam perfis profissional diferentes. Os participantes, além de desenvolver um software com prazo estabelecido para a sua conclusão, precisaram resolver conflitos de opiniões e dinâmica de trabalho diferentes do comumente utilizado.

## **6. Considerações Finais**

A partir da experiência relatada, é conciso afirmar que as pessoas precisam receber oportunidades de aprendizagem que possibilitem o vínculo dos conteúdos da atualidade com as experiências e conhecimento já vivenciados. Dessa forma, os envolvidos estarão se preparando para assumir e atuar ativamente em projetos futuros. Considerando os indícios levantados que apontaram para o aumento da sinergia entre os grupos, o desenvolvimento da criatividade, capacidade de planejamento, entre outras melhorias, conclui-se que foi válida a experiência na formação discente. Além disso, foi considerada válida a experiência como aplicação na prática docente e na formação de discente.

Cabe ressaltar que a experiência confirmou as expectativas de sucesso e de ganhos didáticos ao integrar os conhecimentos teóricos sobre fábrica de software e a vivência prática em um ambiente de trabalho profissional promovido pela abordagem LBD. Isso mostra que uma das formas mais assertivas de aprendizado é a prática. Acredita-se que essa abordagem de ensino poderá ser aplicada em novas experiências e contribuir significativamente como técnica de ensino em ambientes de mestrados profissionais.

Por outro lado, mais pesquisas envolvendo a abordagem LBD precisam ser executadas, investigando todo o conjunto de características, possibilidades e aplicações. Dessa forma, pode-se confirmar que os alunos realmente adquirem e fazem uso do conhecimento e das habilidades transmitidas por meio do ensino, corroborando que a metodologia LBD é uma excelente maneira de alcançar os resultados almejados [Karri and Kode 2011].

Durante a realização do projeto, foram percebidas algumas limitações. A experiência relatada refere-se a uma primeira aplicação da abordagem LBD no trabalho com fábrica de software no mestrado profissional da instituição. Consequentemente, os resultados atingidos limitam-se a um projeto e a um modelo de processo de trabalho específico. Além do mais, até o momento ainda não foi possível validar o produto de software com o cliente, pois encontra-se em processo de implantação. Da mesma forma, a validação do conhecimento construído na disciplina foi baseada apenas nos indícios observados e no relato dos participantes do projeto. Como trabalhos futuros, espera-se realizar novo projeto com o mesmo modelo utilizado e validar o conhecimento adquirido pelos participantes.

## Agradecimentos

Agradecemos a todos alunos que participaram dessa atividade, ao gestor da Fábrica de software e à Divisão de Tecnologia da Informação da Universidade de Passo Fundo pela oportunidade na realização do presente trabalho.

## Referências

- Ahmad, M. O., Liukkunen, K. and Markkula, J. (abr 2014). Student perceptions and attitudes towards the software factory as a learning environment. In *2014 Global Engineering Education Conference (EDUCON)*. IEEE.
- Bardin. L. (2005), *Análise de conteúdo*, Lisboa: Edições 70.
- Borsoi, B. T. (1 jul 2008). *Arquitetura de processo aplicada na integração de fábricas de software*. Biblioteca Digital de Teses e Dissertações da Universidade de São Paulo.
- Dewey, J. (2012), *Democracy and education*, Milton Keynes: Simon & Brown.
- Fernandes, A. A., Teixeira, D. S. (2009), *Fábrica de Software*, São Paulo: Atlas.
- Fuks, H., Pimentel, M. (2011), *Sistemas colaborativos*, Elsevier Brasil.
- Karri, S. K. R. e Kode, S. (2011). Effectiveness of *Learning by Doing* Methodology in Training Programs. In *2011 11th International Conference on Advanced Learning Technologies*. IEEE, p. 227-231.
- Mansker, V., Fulks, R., Peters, B., Curtner, R. e Ogbeide, G.-C. (2016). *Learning by Doing: A Case Study of Hospitality Students' Learning Experience via Service Learning/Hands-on Experience*. *Tourism Travel and Research Association: Advancing Tourism Research Globally*.
- MEC, Ministério da Educação (2009). "Portaria Normativa nº 17, de 28 de dezembro de 2009". <http://bit.ly/2sI53we>.
- Revans, R. W. (2011), *ABC of action learning*, New York: Gower.
- Shank, R. C., Berman, T. R. e Macpherson, K. A. (2013). *Learning by Doing*. In: Reigeluth, C. M. *Instructional-design Theories and Models: A New Paradigm of Instructional Theory*. II ed. Evanston: Taylor & Francis. p. 161–181.