



**VI Jornada Ibero-Americana de Pesquisas
em Políticas Educacionais e
Experiências Interdisciplinares na Educação**

13, 14 e 15
junho de 2022

ISSN: 2525-9571

Vol. 6 | Nº. 1 | Ano 2022

Vanessa Desiderio
IFRN

vanessa.desiderio@ifrn.edu.br

Hortevan Marrocos Frutuoso
UFRN

hortevan.mf@gmail.com

Eixo TEMÁTICO: Educação, ciência,
tecnologia e informação

**PROPOSTA DE MODELO DE
SOFTWARE PARA APOIO ÀS
ATIVIDADES DE PRÁTICA
PROFISSIONAL DOS ALUNOS DO
ENSINO MÉDIO**

**PROPUESTA DE MODELO DE
SOFTWARE PARA APOYAR LAS
ACTIVIDADES DE PRÁCTICA
PROFESIONAL DE ESTUDIANTES DE
SECUNDARIA**



RESUMO

A pandemia COVID-19 forçou algumas mudanças como o teletrabalho e ensino remoto. O ensino remoto seja síncrono ou assíncrono assim como o teletrabalho se tornam cada vez mais presentes. Na academia as fábricas de software já existiam antes da pandemia como estratégia para a prática profissional dos alunos, mas a literatura carece de mais estratégias sobre formas de mensurar o aprendizado do aluno nesse tipo de prática profissional. Pensando nisso e analisando a experiência de outros colegas, criou-se uma proposta de modelo de gerenciamento de equipes de trabalho que desenvolvem softwares em instituições de ensino. O artigo teve como objetivo apresentar os principais requisitos para o protótipo, para tanto utilizou-se a pesquisa bibliográfica e análise qualitativa de outras experiências similares para construção do protótipo. Ainda sobre a perceptiva metodológica, trata-se de uma forma narrativa de relato de experiência. O presente relato descreve a experiência de construção do protótipo, cujos resultados e diagramas apontam para a efetividade. Infelizmente não houve tempo hábil para versões de teste em ambiente real, mas através desse trabalho foi possível criar as orientações e diagramas principais. Espera-se com o compartilhamento da experiência a troca de saberes e o avanço nesse segmento.

Palavras-chave: Prática profissional. Fábrica de Software. Ensino Médio. Remoto.

RESUMEN

La pandemia del COVID-19 obligó a que algunos cambios como el teletrabajo y la enseñanza a distancia. La enseñanza a distancia ya sea síncrona o asíncrona así como el teletrabajo estén cada vez más presentes. En la academia ya existían fábricas de software antes de la pandemia como estrategia para la práctica profesional de los estudiantes, pero la literatura carece de más estrategias sobre formas de medir el aprendizaje de los estudiantes en este tipo de práctica profesional. Con esto en mente y analizando la experiencia de otros colegas, se elaboró una propuesta de modelo de gestión para equipos de trabajo que desarrollan software en instituciones educativas. El artículo tuvo como objetivo presentar los principales requisitos para el prototipo, para lo cual se utilizó la investigación bibliográfica y el análisis cualitativo de otras experiencias similares para la construcción del prototipo. Aún en lo perceptivo metodológico, es una forma narrativa de relato de experiencia. Este informe describe la experiencia de construcción del prototipo, cuyos resultados y diagramas apuntan a la efectividad. Desafortunadamente, no hubo suficiente tiempo para versiones de prueba en un entorno real, pero a través de este trabajo fue posible crear las principales pautas y diagramas. Con el intercambio de experiencias, se espera el intercambio de conocimientos y el avance en este segmento.

Palabras Clave: Practica profesional. Fábrica de software. Escuela secundaria. Remoto.



1. INTRODUÇÃO

A atividade de prática profissional é um componente curricular obrigatório nos cursos técnicos integrados com o ensino médio nos institutos federais brasileiros, já nos cursos superiores esse componente é desejável, mas nem sempre obrigatório. Durante a pandemia COVID-19 as aulas remotas substituíram as aulas presenciais e os laboratórios de informática (espaço físico onde ocorre a maior parte das aulas de prática profissional) ficaram fechados. Para os cursos técnicos da área de informática a prática de programação é importante e geralmente é executada através de atividades em grupos, onde cada componente (colaborador) tem uma importante contribuição no desenvolvimento do todo.

Para Mota *et. Al.* (2014) a prática profissional favorece aluno e professor, pois além da correta interdisciplinaridade entre as atividades que regem a área de programação para ter sucesso no processo pedagógico é preciso ofertar ambientes para que o aluno possa, no desenvolvimento de suas atividades profissionais, manusear e aplicar os conceitos adquiridos em sala de aula no desenvolvimento de soluções. A questão é que alguns alunos são mais hábeis que outros no desenvolvimento das atividades práticas e durante a pandemia com a suspensão dos laboratórios, a avaliação do professor se torna ainda mais subjetiva.

Pensando nessa problemática, criou-se um protótipo para apoiar professores, profissionais e alunos na execução da prática profissional na área de desenvolvimento de software. Ainda sem nome, o protótipo se baseia nas experiências de Fábrica de Software, que são projetos de extensão e/ou núcleo de práticas vivenciados em alguns institutos federais. Embora a solução seja uma resposta à situação pandêmica, é possível utilizar pós pandemia, na verdade esse é o objetivo, aplicar a solução, testar versões e averiguar a necessidade de aperfeiçoamento da solução. Além disso, a ideia é expandir a experiência para quaisquer profissionais ou estudantes interessados pela temática e não restringir ao ambiente acadêmico.

1.1 Relato de Experiências semelhantes que inspiraram o projeto

As fábricas de software se apresentam na literatura no contexto comercial e acadêmico, para fins desse trabalho, o enfoque será o acadêmico, a experiência de Mota *et. Al.* (2014) e Santos *et. Al.* (2021) surgem como uma resposta a necessidade de estágio curricular do ensino



superior em Sistemas de Informação e Engenharia de Software respectivamente, cujo objetivo é acompanhar o aluno dentro de projetos relacionados com a área de Sistemas de Informação, em um ambiente controlado; permite a certificação de que os conceitos apresentados em sala sejam aplicados pelos alunos no momento do desenvolvimento de soluções. Permite também a identificação de eventuais desvios na formação do profissional, mantendo um ajuste entre o projeto pedagógico do curso e a formação do egresso.

Com a crescente expansão das empresas juniores e incubadoras de empresas nas instituições de ensino superior, as fábricas de software acadêmicas são cada vez mais comuns, isso porque elas têm grande potencial de transformação em *spin-offs* acadêmicas (empresas criadas dentro de instituições de ensino de forma a comercializar as pesquisas e conhecimentos ali criados através de empresas incubadas, por exemplo) associando pesquisa acadêmica com experiência de mercado. Os egressos do ensino superior são o público mais comum para essas ações, as experiências com egressos do ensino médio se mostram menos exitosas devido a pouca maturidade para empreender profissionalmente.

Concorrente a isso, o presente trabalho tem como foco o ensino médio integrado com o técnico, cada vez mais disputado pelos projetos societários e mais recentemente evidenciado pela mídia por causa exigência das competências da BNCC (BRASIL, 2018) e diretrizes diversas tais como a PORTARIA Nº 1.432/2018 (BRASIL, 2019) que regulam o novo ensino médio, exigindo cada vez mais do professor a criação de ambientes de prática voltadas para o mercado.

A experiência da Fábrica de Software relatada por Santos e Neto (2018) é realizada com alunos do ensino médio do curso técnico de informática para internet, baseia-se em um modelo iterativo-incremental distribuídos em fases, com a inserção de atividades inspiradas em metodologias ágeis, como *Scrum*, e ferramentas de modelos de negócios, como *business model canvas*. Ao todo, quatro tipos de estruturas principais formam o processo: fases, atividades, tarefas e artefatos. Cada fase é dividida em atividades, as quais são divididas em tarefas, que, por sua vez, geram artefatos. Também foram definidos guias para os artefatos gerados nas tarefas. Esses guias podem ser exemplos ou *templates* e alguns possuem indicações de ferramentas e de tutoriais. O processo proposto foi dividido em 4 fases: Iniciação, na qual busca-



se responder à pergunta “*O que será feito?*”; Elaboração, na qual a pergunta é “*Como será feito?*”; Construção, que corresponde ao desenvolvimento real do projeto descrito e elaborado anteriormente; e Transição, para últimos ajustes e implantação.

O protótipo desenvolvido e relatado nesse trabalho foi feito com base na experiência vivenciada no campus Natal Zona Norte, do Instituto Federal de Educação Ciência e Tecnologia (IFRN) em 2018-2020, observando o projeto FASES, apresentado por Santos e Neto (2018) e na experiência com incubação de empresas no campus João Câmara (2014-2022) e Natal Zona Norte (2018-2020), estimulando a criação de um núcleo de prática no curso de informática do campus João Câmara, semelhante ao que é feito no campus Natal Zona Norte. Trata-se, portanto de um relato de experiência com apresentação de protótipo de Fábrica de Software “virtual” para gerenciamento da prática profissional, visto que as experiências relatadas por Mota (2014) e Santos e Neto (2018) utilizam-se de um espaço físico ou ambiente controlado e a proposta em construção é uma adaptação para a situação pandêmica. A experiência de *home office* por cinco meses de Santos et. Al (2021) também colaborou para a construção do projeto.

A atividade de desenvolvimento de software naturalmente pode ser realizada de qualquer lugar do globo terrestre. Através da Internet, profissionais podem interagir e desenvolver softwares bastante complexos, através de ferramentas de colaboração. Porém verificamos nas experiências vivenciadas que o problema se dá justamente na gestão de uma equipe de desenvolvimento geograficamente dispersa. Desse modo, apresentamos uma metodologia de gerenciamento de equipes a distância. Foram analisados diversos modelos de desenvolvimento em busca de um modelo que melhor se adaptasse ao desenvolvimento a distância, conservando conceitos como reuso de componentes e trabalho colaborativo. Apresentamos com base na experiência vivenciada um modelo de software que permite integrar os diversos membros de uma equipe de desenvolvimento, bem como fornece as ferramentas certas que permitem aos gestores de uma fábrica escola de software, através de indicadores, gerenciar os projetos de softwares desenvolvidos à distância.

2. PROBLEMATIZAÇÃO E PROPOSTA DE SOLUÇÃO

Antes da pandemia, o cenário de caos no trânsito, somado ao crescimento desordenado



das cidades, problemas de segurança pública, entre outros já vislumbravam o teletrabalho como alternativa para muitos profissionais. Nas instituições de ensino, apenas os cursos de Educação à Distância (EaD) utilizavam o trabalho e ensino remoto; com a pandemia, tudo mudou, houve necessidade de formação e adaptação para alunos e trabalhadores da educação concerne ao modo remoto. A priori as aulas teóricas foram supridas, mas as aulas práticas permaneceram suspensas. A prática profissional oferece ao aluno a experiência de trabalho com supervisão profissional e orientação docente. Algumas práticas são realizadas em laboratórios, outras em ambientes de empresas parceiras por meio de estágios. Além disso, alguns alunos optam pela pesquisa e/ou extensão que promovem a experiência de criar algo inovador ou o desenvolvimento de softwares para fins de atender uma necessidade da sociedade através de disciplinas ou projetos com orientação docente.

No cenário pandêmico, de isolamento social, o trabalho/estudo remoto se constitui uma saída para manutenção das atividades de prática profissional, apesar de saber que em algumas situações o trabalho/estudo remoto não permitiu melhor rendimento por vários motivos (falta de empatia dos familiares, desorganização dos horários de estudo/trabalho, distrações diversas, falta de equipamentos ou equipamentos sem a devida manutenção; falta de internet de qualidade ou de licença de software, por exemplo). Vencidos esses obstáculos, surgiram outros: para desenvolvimento de software é interessante a colaboração de forma integrada, dificilmente um aluno do curso técnico integrado com ensino médio consegue desenvolver um projeto de prática profissional sem apoio dos colegas, principalmente porque a prática profissional é obrigatória no último ano, junto com outras disciplinas e preparação para o ingresso na graduação.

Mangan (2006) concorda que “o software é o resultado do esforço de diversos indivíduos que contribuem com capacidades e perspectivas diferentes”. Por esta definição percebe-se a necessidade de uma forte integração entre os membros de uma equipe de desenvolvimento. O autor ainda explica que, apesar de cada colaborador receber uma atividade específica, essa interfere nas demais. Dessa forma, o fato de os desenvolvedores estarem distantes entre si torna mais difícil promover esta interação entre os membros da equipe.

Durante a construção do projeto surgiram alguns questionamentos: como gerar confiança entre as partes? Como criar um gerenciamento eficaz no processo de desenvolvimento? Peter



Drucker afirmou que “o que não se pode medir não se pode gerenciar” Drucker (*apud* GRAEML, 2000). Por gerenciar, entende-se analisar, observar os pontos fortes e fracos, corrigir o que for necessário de modo a melhorar a qualidade do serviço. Para que uma equipe qualquer possa oferecer o melhor de si, o gestor/líder/orientador deve ter condições de metrificar corretamente o trabalho realizado e apontar os ajustes necessários.

Deve-se ainda escolher uma estratégia que facilite o gerenciamento de tarefas e o reuso de código, pois um dos problemas experimentados na prática docente e na gestão de incubadoras foi o fato de que se perdem excelentes projetos simplesmente porque não houve tempo hábil para amadurecer ou aperfeiçoá-lo no período em que ocorreu a prática profissional e a turma (equipe) que pega o projeto no período seguinte não consegue retomar por falta de documentação ou desorganização em relação ao uso do código, havendo muitas vezes retrabalho e perda de dados.

A estratégia correta precisava responder a seguinte questão: De que forma é possível gerir uma equipe de desenvolvimento de software geograficamente dispersa e conectada remotamente? Para responder a esta pergunta é necessário saber, dentre os processos de software conhecidos, qual ou quais deles se adéquam ao trabalho remoto.

2.1. Processo de desenvolvimento de Software

Sommerville (2011) define processo de software como “um conjunto de atividades e resultados associados que produz um produto de software”. Ele discute em sua obra três modelos básicos de processos de software: o modelo em cascata, desenvolvimento iterativo-evolucionário e engenharia de software baseada em componentes. Para atender ao objetivo deste trabalho foram utilizados os modelos iterativo-evolucionário e engenharia baseada em componentes. Utilizou-se também o modelo ágil *Scrum*. Larman (2007) define o modelo evolucionário como um processo de software onde o desenvolvimento é dividido em pequenos pedaços chamados de iteração. A cada iteração agrega-se mais uma parte ao projeto principal. Ela é tratada como um pequeno sistema, com suas próprias etapas de levantamento, análise de requisitos, projeto, desenvolvimento e teste. A principal vantagem deste modelo é o fato de que o sistema é feito em pequenos pedaços que são aos poucos integrados ao projeto principal.



Como o projeto é feito aos poucos, torna-se mais fácil garantir que o produto final esteja de acordo com os requisitos solicitados pelo cliente. Uma vez que, a cada iteração, temos um subsistema que será apresentado ao cliente e este avaliará se os requisitos funcionais daquele “pedaço” foram atingidos. Caso haja um desvio nesses requisitos, este “pedaço” poderá ser corrigido mais rapidamente, sem que para isso todo o sistema seja alterado.

O modelo iterativo tem sido amplamente utilizado na Programação Orientada a Objetos (POO). Observa-se que a programação POO se presta muito bem ao uso deste modelo. Uma vez que o programa é construído através de classes e pacotes. Fowler (*apud* LARMAN, 2007) afirma que o uso do modelo iterativo melhora as chances de um projeto de software ser bem-sucedido. Aliando-se esta técnica ao desenvolvimento baseado em componentes conseguiu-se otimizar o processo de desenvolvimento de software. Deve-se tomar cuidado para que cada módulo desenvolvido possua uma alta coesão e um acoplamento baixo o suficiente para permitir sua reutilização em outro projeto. Esta prática torna viável a criação de um banco de componentes. Assim, o trabalho desenvolvido por um profissional poderá ser reaproveitado por outro profissional em um projeto diferente.

2.2. Orientações para desenvolvimento do modelo proposto

Se a instituição de ensino possui uma incubadora de empresas é comum ter um banco de ideias com resultado de concurso ou laboratório de ideias, trata-se de um momento anterior à prática profissional, para reunir soluções de problemas ou necessidades demandas pela sociedade, elas servirão de controle de entrada, pode ser realizado também através de edital aberto à sociedade que reportará demandas específicas que poderão ser atendidas e uma triagem realizada por grupo de professores.

Após verificar quais ideias são potenciais de desenvolvimento, uma outra triagem deve ser discutida, a divisão de equipes de alunos, observando habilidades de *front-end* e *back-end*. Feito isso, em uma primeira reunião deve ser gerada uma lista contendo os principais requisitos do projeto a ser desenvolvido. Neste momento não deve haver preocupações com os detalhes de cada requisito. Apenas devem ser anotados e listados os requisitos que sejam importantes para o software. Esses requisitos deverão ser agrupados por prioridade. A lista então gerada é

chamada *product backlog*.

O ciclo de desenvolvimento começa com uma nova reunião chamada planejamento de versão (*Release Planning*), onde a partir da *product backlog*, um grupo de requisitos de maior prioridade é separado e passa a formar uma nova lista chamada *release backlog*. Estes requisitos são novamente analisados. Agora de forma mais detalhada, gerando uma lista de atividades a serem realizadas. São estimados o tempo e os recursos necessários para a execução de cada atividade, bem como a ordem em que deverão ser executadas. As atividades são agrupadas gerando pequenas listas de atividades chamadas de *sprint de trabalho*. Cada *sprint* contém um subconjunto de atividades que implementam um subgrupo de requisitos do *release backlog*. Durante um intervalo de tempo de duas a quatro semanas, a *sprint* é desenvolvida.

O sistema deve ainda conter uma biblioteca de componentes reutilizáveis bem documentados que permitam a qualquer programador fácil reuso de componentes ou mesmo obter auxílio rápido a dúvidas simples, como descobrir a assinatura de um método em um determinado componente. É necessário que todo código seja gravado dentro do sistema através de um servidor de código integrado à ferramenta. É importante também que exista uma ferramenta wiki que permita armazenar o conhecimento aprendido sobre o projeto. Todos devem ser estimulados a documentar o código e a compartilhar conhecimento. Seria bom, mas não imprescindível, que todas essas funções (ao menos as mais importantes) existissem em uma única ferramenta, que através da internet, integrasse o trabalho de todos os membros da equipe, com uso de senha e registro de logs. Agregando-se os requisitos acima especificados, identificaram-se os seguintes atores (figura 1):

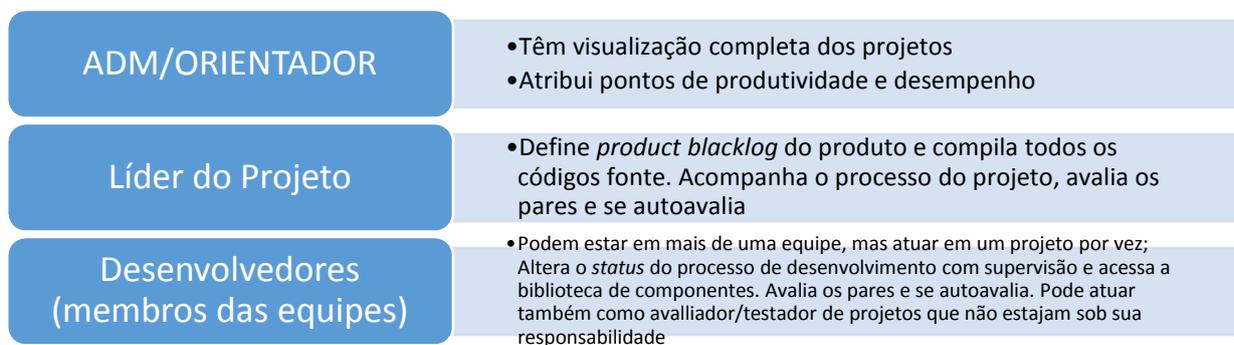


Figura 1 – Atores do Sistema

Durante a construção da proposta foi feito o levantamento dos casos de uso com base na articulação entre os atores, no entanto, não é possível colocar no presente trabalho por falta de espaço, mas como se trata de uma proposta em construção, pode ser compartilhado posteriormente caso o leitor tenha interesse. A título de exemplo, será apresentado uma descrição de caso de uso (figura 2).

U001 - Ver Usuários Online	<p>Descrição Exibe os usuários que estão conectados no momento.</p> <p>Descrição Detalhada Exibir lista de usuários do sistema cujo <i>status</i> seja “Online”</p> <p>Pré-Condições Usuário deve estar conectado e autenticado</p> <p>Fluxo Principal 1) Listar todos os usuários com <i>status</i> “Online”</p>
----------------------------------	---

Figura 2 – Descrição de caso de uso dos usuários do sistema

Após o diagrama dos casos de uso, foi elaborado o diagrama de classes com 5 pacotes: Cadastro, Acesso, Comunicação, Projeto e Produtividade. O passo seguinte foi a criação do Diagrama de Entidade de Relacionamentos (DER) seguido dos diagramas de sequências. Os Diagramas de Atividades revelaram o fluxo de comandos de um conjunto de atividades dentro de um sistema. Ao passo que o diagrama de estados exibiu como o estado de uma classe muda conforme a execução das atividades dentro de um sistema.

3. CONSIDERAÇÕES FINAIS

O software aqui proposto foi desenhado por meio dos vários diagramas para ser capaz de integrar as principais etapas necessárias à comunicação entre os atores durante as etapas de análise, desenvolvimentos e teste de software, em um modelo ágil – baseado no *scrum* – aplicado a distância. Não foi possível, neste trabalho, avaliar na prática o funcionamento do software. Contudo, se utilizado de forma correta, e com o devido acompanhamento por parte dos administradores/orientadores (professores ou coordenadores de projeto de extensão, por exemplo) e líderes de equipe (alunos engajados), uma fábrica escola de softwares ou até uma empresa júnior de desenvolvimento pode conceder diversos benefícios para os colaboradores e melhorar seu desempenho/produtividade.



Acompanhar os erros e acertos no processo ensino aprendizagem da prática profissional é a principal característica desse modelo. Não foi analisado aqui o efeito social e/ou psicológico de colaboradores que trabalham em outro local fora a instituição de ensino. Assim administrar dificuldades como a motivação, otimização do tempo ou até mesmo a possibilidade de “retrocesso acadêmico/profissional” por parte dos alunos não faz parte do escopo desta proposta. Foram levadas em conta apenas as dificuldades de ordem técnica e remota aplicados ao desenvolvimento de software, a fim de se construir um software que ajudasse a contorná-las. Inclusive recomenda-se não levar o termo “à distância” a um extremismo cibernético. É importante que existam – se não frequentemente – ao menos esporadicamente, reuniões síncronas ou na medida do possível presenciais. Por melhor que o um software possa ser, dispor de ferramentas de interação e comunicação entre usuários, ele não substitui o contato humano. Para trabalhos futuros, sugere-se avaliar o impacto social/psicológico em profissionais da área de TI trabalhando a distância.

5. REFERÊNCIAS

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília, 2018.

BRASIL. Ministério da Educação. Gabinete do Ministro. **PORTARIA Nº 1.432, DE 28 DE DEZEMBRO DE 2018**. Estabelece os referenciais para elaboração dos itinerários formativos conforme preveem as Diretrizes Nacionais do Ensino Médio. Diário Oficial da União da República Federativa do Brasil. Brasília, 05 abril. 2019.

GRAEML, Alexandre Reis. **Sistema de Informação: O alinhamento da estratégia de TI com a estratégia corporativa**. São Paulo: Atlas, 2000.

LARMAN, Craig. **Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. 3ª Edição. Bookman, 2007.

MANGAN, Marco Aurélio Souza. **Uma abordagem para o Desenvolvimento de Apoio à Percepção em Ambientes Colaborativos de Desenvolvimento de Software**. 2006. Tese – Universidade Federal do rio de Janeiro, Rio de Janeiro, 2006.

MOTA, Casemiro; DA SILVA, Eduardo; BRAZ, Fernando; RIEGEL, Ivo; MOTA, Joice;

MENDES, Marco André; DE OLIVEIRA, Paulo Cesar. **A experiência do ambiente da Fábrica de Software nas atividades de ensino do curso de Sistemas de Informação do IFC - Campus Araquari**. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI),



22. , 2014, Brasília. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2014 . p. 259-268. ISSN 2595-6175.

SANTOS, Alexandro Sousa dos; NEVES, Matheus; RODRIGUES, Yuri; OLIVEIRA, Nisio H. de; KUHN, Daniel; SANTOS, Gustavo; SILVA, Rodolfo A.. **Experiência do Projeto "Fábrica de Software"** em um Curso de Engenharia de Software. In: ESCOLA REGIONAL DE ENGENHARIA DE SOFTWARE (ERES), 5. , 2021, Evento Online. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021 . p. 89-98. DOI: <https://doi.org/10.5753/eres.2021.18454>.

SANTOS, Giovanna Lorena R; NETO, Edmilson B. Campos. **Um Processo de Desenvolvimento de Software para o Ensino Técnico baseado em uma Fábrica de Software Escola**. VII Congresso Brasileiro de Informática na Educação, CBIE, 2018.

SOMMERVILLE, I. (2011). **Engenharia de software**. Pearson Prentice Hall, 9th edition.

Vanessa Desidério

Professora, graduada em administração e Mestre em Engenharia de Produção atuante em empreendedorismo, núcleo de prática profissional e incubadora tecnológicas.

Hortevan Marrocos Frutuoso

Professor, graduado em sistema de informação e Mestre em Engenharia de Computação atua em Desenvolvimento de Software e inteligência artificial.