

Um Modelo de Ensino Virtual de Engenharia de Software Orientado por Competências, Times e Projetos

Johnny Marques

johnny@ita.br

Instituto Tecnológico de Aeronáutica

Karla Fook

karla@ita.br

Instituto Tecnológico de Aeronáutica

RESUMO

Devido à pandemia COVID-19, muitas disciplinas oferecidas presencialmente em várias instituições educacionais precisaram ser repensadas para uma emergência e oferta virtual. Este trabalho apresenta um relato de experiência na aplicação de um modelo de ensino virtual de Engenharia de Software, envolvendo a seleção das Metodologias Ativas como a Aprendizagem Baseada em Projetos (PjBL), a Aprendizagem Baseada em Equipes (TBL) e a Aprendizagem Baseada em Competências (CBL), com foco em competências, equipes e projeto aplicado na disciplina Fundamentos de Engenharia de Software do Instituto Tecnológico de Aeronáutica (ITA) no segundo semestre de 2020. O modelo apresentado contém um modelo de concepção e um modelo de execução. Também foi realizada uma análise de aderência do modelo de ensino virtual para dois importantes referenciais de ensino em Engenharia de Computação, como o Exame Nacional de Desempenho de Estudantes (ENADE) e a Sociedade Brasileira de Computação (SBC).

PALAVRAS-CHAVE

Educação de computação, Aprendizagem baseada em projetos, Aprendizagem baseada em times, Metodologias ativas, software

1 INTRODUÇÃO

O Instituto Tecnológico de Aeronáutica (ITA) instalou-se na cidade de São José dos Campos em 1950, transformando-a em um dos maiores polos de desenvolvimento do país [10]. Apesar de sua vocação natural para a Engenharia Aeronáutica, o ITA oferece cursos de engenharia em outras 5 modalidades, incluindo o curso de Engenharia de Computação.

A formação qualificada e a capacitação de profissionais são cada vez mais necessárias [8]. Os cursos de curta duração, graduação ou pós-graduação, precisam formar bons profissionais em suas instituições de ensino. Especificamente no ensino de Engenharia de Software (ES), a qualidade dos profissionais está diretamente relacionada à sua formação, embora também existam outros fatores que também contribuem para isto [2].

Dentre as disciplinas de cursos de Engenharia de Computação, a Engenharia de Software apresenta-se com grande destaque. No entanto, a indústria de software se queixa de que os cursos de

graduação não ensinam as competências necessárias para que os alunos desempenhem suas atividades com eficiência [25].

O objetivo deste trabalho é apresentar um relato de experiência da aplicação de um modelo de ensino virtual de Engenharia de Software, envolvendo uma seleção de Metodologias Ativas, com foco em competências, times e projetos. Este relato envolve a disciplina *Fundamentos de Engenharia de Software*, disciplina obrigatória do 6º semestre do curso de graduação em Engenharia da Computação do ITA.

O modelo foi classificado como virtual, porque refere-se ao modo de oferta da disciplina, excepcionalmente no segundo semestre de 2020, por conta da pandemia da COVID-19, onde todas as atividades de ensino aconteceram no formato à distância.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Ensino de Engenharia de Software

O professor é um dos principais responsáveis pelo ensino, pesquisa e gerenciamento do processo educacional, sendo responsável pela seleção dos métodos a serem utilizados e visando a atração e atenção dos seus alunos [20]. O professor deve utilizar instrumentos didáticos que priorizem a participação do aluno, gerenciando suas expectativas e habilidades. Existem duas categorias principais de métodos de ensino: as centradas no professor e as centradas no aluno.

Através de uma pesquisa de opinião com professores no Brasil[9] mapeou dificuldades no ensino de Engenharia de Software. Os inúmeros ciclos existentes, a quantidade de processos e tecnologias dificultam o ensino completo sendo o tempo e carga horária, um limitador para o bom aprendizado. Adicionalmente, os 5 (cinco) problemas apresentados na literatura para o processo de ensino e aprendizagem em Engenharia de Software são [17] :

- Problema 1: A existência de uma grande distância entre o que se ensina e a realidade existente no mercado de trabalho;
- Problema 2: Os discentes têm pouco interesse pelas aulas maciçamente teóricas;
- Problema 3: É dada pouca ênfase ao trabalho em grupo;
- Problema 4: Em geral, o que se ensina e aprende em um semestre ou disciplina, não é apropriado pela comunidade como um patrimônio intelectual para uso futuro; e
- Problema 5: Há pouca ou nenhuma interação entre professores e alunos fora da sala de aula.

Neste contexto, as Metodologias Ativas são notadamente centradas no aluno. Um dos grandes desafios em Engenharia de Software é utilizar métodos eficazes de ensino e aprendizagem de modo que os conceitos teóricos sejam abordados através de aplicações práticas [14]. A Engenharia de Software é repleta de conceitos, muitos deles extremamente teóricos, o que pode comprometer a atenção

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'22, Abril 24-29, 2022, Feira de Santana, Bahia, Brasil (On-line)

© 2022 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

do aluno em sala de aula, podendo ter como consequência perda de estímulo e motivação.

2.2 Metodologias Ativas

As Metodologias Ativas referem-se à aprendizagem construtiva, colaborativa, interdisciplinar, contextualizada, reflexiva, crítica, investigativa, humanista, motivadora e desafiadora [16]. Assim, ganham destaque metodologias como o ensino baseado em projetos, ou *Project-Based Learning* (PjBL), com lastro no desenvolvimento de competências, na aprendizagem colaborativa e na interdisciplinaridade. Na Tabela 1 apresenta um sumário das principais metodologias ativas.

2.3 Trabalhos Correlatos

Dentre as abordagens que se interceptam com o modelo de ensino virtual aplicado neste relato de experiência pode-se citar um relato de experiência de aprendizagem baseada em projetos (PjBL) em engenharia de software, em uma universidade espanhola [21]. A experiência se baseia na formação de pequenas equipes heterogêneas, que enfrentam as fases iniciais de uma metodologia de software durante o desenvolvimento de um projeto próximo da realidade. Os autores ressaltam que a avaliação adotada considera a avaliação dos pares dos alunos, além das avaliações do professor. Os resultados apontam para o efeito positivo do uso do PjBL na aquisição de diferentes habilidades por parte dos alunos, futuros engenheiros de software.

Ainda nesta linha, o método “Multi-Role Project” (MRP) [26], proposto para ser aplicado no contexto de um curso de Ciências, Tecnologia, Engenharia e Matemática (STEM). O relato é sobre a aplicação do método em um curso de ensino de engenharia de software, linguagem UML e gerenciamento de projetos. Cinco princípios fornecem uma estrutura para orientar as práticas de trabalho das equipes de alunos: distribuição de responsabilidades; interações e solicitações regulares dentro da equipe; antecipação e melhoria contínua; interdependência positiva e alternância de trabalho individual/coletivo; e comunicação aberta e gerenciamento de conteúdo. Segundo os autores, os resultados mostraram que o método permitiu que os alunos do curso adquirissem conhecimentos e habilidades profissionais ao vivenciar realidades profissionais quase reais.

Existe um relato envolvendo a elaboração de um portfólio de ensino para auxiliar os professores a maximizar os benefícios da aprendizagem combinada (presencial e online) dentro da aplicação da metodologia ativa *Team-Based Learning* (TBL) [18]. O portfólio de ensino combinado foi criado para um curso de Gestão Internacional de Recursos Humanos, contemplando informações como: visão geral do curso, competência de graduação, um plano de estudos, recursos materiais do curso, um cenário de ensino, um teste de garantia de leitura, exames intermediários / finais, atribuições do aluno, avaliação da aprendizagem resultados e uma folha de melhoria da qualidade do curso. Uma meta apontada pelos autores é aumentar a motivação dos alunos no tocante à leitura do material do curso antes das aulas.

Outros relatos de uso de Aprendizagem Baseada em Projetos (PjBL) foram realizadas no âmbito de cursos de pós-graduação no Instituto Tecnológico de Aeronáutica [5–7, 11, 22]. As abordagens adotam atividades e ações similares entre si e corroboram pontos positivos

do uso de metodologias ativas para melhorar a formação dos alunos no que se refere a habilidades requeridas por parte dos engenheiros de software, tanto no trabalho em equipe e quanto na forma autônoma.

3 MODELO DE ENSINO VIRTUAL PROPOSTO

Dentre as 7 (sete) metodologias ativas identificadas na literatura e sintetizadas na Seção 2, os professores responsáveis pela disciplina decidiram incorporar ao modelo de ensino virtual a PjBL e a TBL.

A escolha da PjBL e TBL foi orientada para tentar eliminar, ou mesmo reduzir, os 5 (cinco) problemas identificados por Paiva & Medeiros (2011) [17] para o processo de ensino e aprendizagem em Engenharia de Software.

Os professores entenderam que o uso da PjBL reduziria os impactos dos problemas 1, 2 e 5. Para o problema 1, entendeu-se que a aprendizagem baseada em projetos diminuiu a distância entre o que se ensina e a realidade do mercado de trabalho. Também foi vislumbrado que a PjBL mitigaria o problema 2, reduzindo levemente a carga de aulas maciçamente teóricas. Por fim, entendeu-se também que a PjBL reduziria o problema 5, especialmente por ser um modelo de ensino virtual, que aumenta muito as oportunidades de redução da interação entre alunos e professores. Com a PjBL, os professores foram consultados em diversos momentos pelos alunos.

Já na escolha da TBL, os professores entenderam que esta reduziria os impactos dos problemas 3 e 4. Com o fomento do trabalho em times, os professores entenderam que isso aumentaria a capacidade dos alunos trabalharem em equipe, especialmente em um momento de ensino virtual, onde os encontros presenciais em sala de aula estavam suspensos. Para a redução do impacto do problema 4, os professores acreditaram a TBL fomentou a capacidade dos alunos de trabalharem nos seus projetos de maneira distribuída e assíncrona e que seria um ativo importante para uso futuro.

Para a escolha da CBL, nosso modelo procurou incorporar as *Competências* existentes em referenciais de ensino, como o Exame Nacional de Desempenho de Estudante (ENADE) [4] e da Sociedade Brasileira de Computação (SBC) [27].

Assim, o modelo de ensino virtual proposto e relatado neste trabalho fomenta a integração de Teoria e Prática. Consequentemente, os professores entenderam que a aplicação das metodologias ativas *Project-Based Learning* (PjBL), *Team-Based Learning* (TBL) e *Competency-Based Learning* (CBL) seriam adequadas para aumentar o estímulo e interesse dos alunos na integração teórica e prática da disciplina.

O modelo de ensino apresentado neste trabalho foi virtualizado no segundo semestre de 2020 por conta da pandemia da COVID-19. O modelo envolveu duas abstrações:

- *Modelo de Concepção*: Este modelo tem como centro o projeto que envolverá a incorporação de competências previstas em referenciais de ensino em graduação em Engenharia de Computação; e
- *Modelo de Execução*: Este modelo envolve a execução do Modelo de Concepção com todas as etapas previstas no projeto desenvolvimento pelos alunos na disciplina. É um modelo detalhado que pode ser instanciado a cada oferta da disciplina.

Tabela 1: Sumário das Metodologias Ativas

| Metodologia Ativa | Descrição |
|---|---|
| Aprendizagem Baseada em Problemas (<i>Problem-Based Learning</i> - PBL) | O PBL é uma metodologia ativa em que os alunos são confrontados com problemas contextualizados e pouco estruturados e para os quais se empenham em encontrar soluções significativas [1]. |
| Aprendizagem por Estudo de Caso (<i>Case Study-Based Learning</i> - CSBL) | O CSBL é uma metodologia ativa em que os alunos são colocados no papel de pessoas que enfrentaram decisões difíceis em algum momento no passado. Os alunos discutem se a decisão passada foi correta ou não [23]. |
| Aprendizagem Baseada em Projetos (<i>Project-Based Learning</i> - PjBL) | O PjBL é uma metodologia ativa em que os alunos resolvem problemas autênticos e devem produzir resultados através de projetos. A PjBL é uma especialização do PBL [3]. |
| Sala de Aula Invertida (<i>Flipped Classroom</i> - FC) | O FC é uma metodologia ativa em que inverte-se a lógica de organização da sala de aula. Com ela, os alunos tem prévio acesso ao material do curso, impresso ou on-line, e podem discutir o conteúdo com o professor e os demais colegas. Nessa perspectiva, a sala de aula se transforma em um espaço dinâmico e interativo, permitindo a realização de atividades em grupo, estimulando debates e discussões, e enriquecendo o aprendizado do estudante a partir de diversos pontos de vista [24]. |
| Aprendizagem Baseada em Equipes (<i>Team-Based Learning</i> - TBL) | O TBL é uma metodologia ativa que consiste em módulos que podem ser ensinados em um ciclo de três etapas: preparação, teste de garantia de prontidão em sala de aula e exercícios focados em aplicações do conhecimento adquirido. Por fim, os alunos trabalham em equipes para resolver problemas que lhes permitem aplicar e expandir o conhecimento que acabaram de aprender e testar [15]. |
| Aprendizagem Baseada em Gamificação (<i>Gamification-Based Learning</i> - GBL) | O GBL é uma metodologia ativa que usa de técnicas de design de jogos que utilizam mecânicas e pensamentos orientados a jogos para enriquecer contextos diversos [13]. |
| Aprendizagem Baseada em Competências (<i>Competency-Based Learning</i> - CBL) | O CBL é uma metodologia ativa em que os alunos geralmente são avaliados em várias competências e em várias etapas durante uma disciplina ou curso e geralmente têm a oportunidade de tentar uma determinada competência várias vezes e receber <i>feedback</i> contínuo dos professores e demais alunos [12]. |

3.1 Modelo de Concepção

Por conta da pandemia da COVID-19, houve a necessidade de incorporação de um modelo de ensino virtual que viabilizasse o processo de ensino-aprendizagem em Engenharia de Software. Nosso Modelo de Concepção, ilustrado na Figura 1, é composto de 8 partes:

- (1) Competências;
- (2) Videoaulas;
- (3) Encontros Telepresenciais;
- (4) Roteiros de Laboratórios;
- (5) Organização dos alunos em Times;
- (6) Projeto;
- (7) Subprodutos; e
- (8) Protótipo de Software.

Para a execução desse modelo de ensino virtual, os alunos contam com: 1) as *Videoaulas*, semanalmente postadas, abordam toda a ementa prevista e já apresentada; 2) os *Encontros Telepresenciais* semanais; 3) os *Roteiros de Laboratórios* (RL) que planejam as entregas dos subprodutos ao longo do projeto; e 4) a formação de *Times* de 4 a 6 alunos para cada projeto.

Além disso, os professores adotaram uma das ferramentas institucionais disponíveis, o *Google Classroom*®. Neste ambiente, os professores foram disponibilizando semanalmente os conteúdos teóricos por meio de *Videoaulas* que foram construídas com base no material já existente, do modelo presencial, com adaptações necessárias para a mudança da forma de oferta para o ensino virtual. As *Videoaulas* são estruturadas de forma a disponibilizar aos alunos todos os conteúdos de forma assíncrona. Elas fornecem apoio para

que os alunos, organizados em *Times*, desenvolvam seus *Projetos* e utilizem os conceitos abordados no ensino virtual. Os *Times* foram organizados pelos próprios alunos, seguindo o requisito de número de participantes estipulados pelos professores (4 a 6 alunos por times).

Os *Encontros Telepresenciais* semanais são sempre ao vivo, e por videoconferência entre os professores e alunos para tratamento das possíveis dúvidas das *Videoaulas* e dos *Projetos*. Nesta oportunidade, os professores passam a ter um contato síncrono com os alunos, favorecendo o relacionamento aluno-professor, tão necessário em sala de aula, mas que precisou ser reinventado com a chegada da pandemia, já que tanto alunos, quanto professores deste curso, eram acostumados ao ensino presencial.

Os *Roteiros de Laboratórios* (RL) foram divididos em 4 entregas principais, envolvendo subprodutos desenvolvidos nos *Projetos*. Cada entrega envolveu a apresentação de *Subprodutos* dos processos de desenvolvimento de software abordados nas *Videoaulas*. A Tabela 2 apresenta o escopo de cada uma das 4 entregas aplicáveis, guiadas pelos *Roteiros de Laboratório* (RL) de 1 até 4.

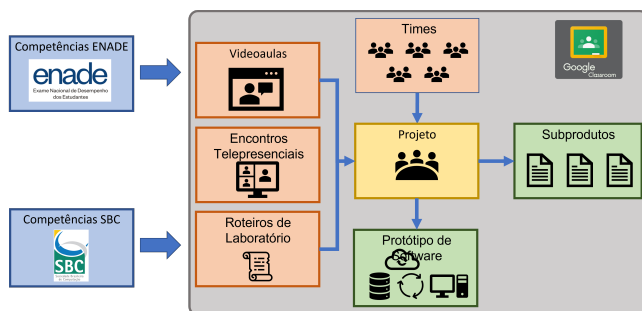
O *Projeto* pode ser escolhido livremente pelos alunos que compõem seus *Times*. Os alunos podem escolher o desenvolvimento de um *Protótipo de Software* totalmente do novo, construído desde o começo, ou ainda, propor um conjunto significativo de novas funcionalidades de um produto de software já existente.

A entrega do Protótipo de Software segue a filosofia de um *Minimum Viable Product* (MVP). Conforme a própria terminologia, o

Tabela 2: Entregas de Laboratório

| Laboratório | Escopo |
|-------------|--|
| RL1 | Subproduto 1: Definição dos requisitos funcionais e não funcionais e Arquitetura de cada projeto. |
| RL2 | Subproduto 2: Descrição de cada projeto, contemplando: a) O modelo de processo proposto; b) Os diagramas previstos na Unified Modeling Language (UML) e de banco de dados; c) As competências e dedicações de cada um dos integrantes no projeto; e d) O cronograma para o desenvolvimento do protótipo de SI. |
| RL3 | Subproduto 3: Apresentação para acompanhamento do desenvolvimento do Protótipo. |
| RL4 | Subproduto 4: Definição dos testes de cada projeto, incluindo: a) Os níveis de teste; b) As técnicas propostas; b) Os métodos a serem utilizados; c) Os casos de testes gerados; e d) Os resultados de testes obtidos. |

MVP deve apresentar uma versão reduzida e simplificada do produto que se pretende ofertar. É importante lembrar que ele precisa ter, no mínimo, todas as funções que o produto final terá, para que as análises sejam confiáveis [19].

**Figura 1: Modelo de Concepção**

3.2 Modelo de Execução

Nosso Modelo de Execução contém 7 etapas, conforme apresentado na Figura 2.

Na etapa 1, é realizada a divisão dos alunos em times, com 4 a 6 integrantes cada. A disciplina não envolve o ensino de ferramentas específicas para modelagem, codificação, testes e outras tarefas técnicas associadas à Engenharia de Software. Por ser a primeira disciplina de Engenharia de Software do nosso curso, que oferece duas, o objetivo desta disciplina é apresentar todos os conceitos de Engenharia de Software e seus processos. Assim, os alunos devem ser motivados a buscarem livremente as ferramentas e plataformas necessárias para construção dos seus projetos. Ainda nesta etapa 1, os alunos são incentivados a avaliarem suas competências técnicas, definindo o papel de cada aluno no seu time. Alguns alunos notadamente possuem mais vontade de trabalhar na codificação, outros

na especificação de requisitos, e assim os times tinham seus papéis baseados em competências.

Na etapa 2, os alunos selecionam seus projetos e possuem liberdade na proposta de seus projetos, passando pelo crivo dos professores envolvidos. Após aceite, os alunos desenvolvem todas as atividades propostas e requeridas de seus projetos para atenderem os propósitos estabelecidos na disciplina, integrando de maneira incremental aspectos teóricos observados nas videoaulas.

Na etapa 3, os alunos iniciam a identificação dos requisitos e da arquitetura a ser utilizada no projeto do protótipo de software. Aqui já prevista a primeira entrega de subprodutos do projeto. Conforme apresentado na Tabela 2, o Roteiro de Laboratório 1 (RL 1) é desenvolvido e entregue pelos alunos para tarefa no ambiente Google Classroom.

Na etapa 4, os alunos devem propor seus modelos de processo. Os modelos de processo consistem na representação de vários processos de Software, desde a especificação de requisitos até os testes. Nossas videoaulas apresentam os principais modelos de processo em forma de ciclos de vida, como:

- Ciclo de Vida em Cascata;
- Ciclo de Vida em V;
- Ciclo de Vida Incremental e Iterativo; e
- Ciclo de Vida Evolutivo.

Ainda nesta etapa 4, os alunos são incentivados a definirem seu próprio modelo, podendo escolher entre os apresentados em videoaulas ou existentes na literatura. Além dos modelos de processo, nessa mesma etapa, os alunos também: (i) constroem os diagramas previstos na Unified Modeling Language (UML) e de banco de dados; (ii) apresentam os papéis e responsabilidades de cada aluno, orientado pelas competências e dedicações de cada um dos integrantes no projeto; (iii) e apresentam o cronograma para o desenvolvimento do protótipo. Toda a etapa 4 é guiada pelo Roteiro de Laboratório 2 (RL 2).

Na etapa 5, os alunos fazem a codificação do protótipo de software como parte da entrega do projeto. O Roteiro de Laboratório 3 (RL 3) prevê apenas uma apresentação aos professores envolvidos do protótipo em construção. Os professores também dão um *feedback* avançado sobre o protótipo enquanto o mesmo ainda está em construção.

Na etapa 6, os alunos fazem os testes sobre o protótipo de software. O Roteiro de Laboratório 4 prevê que os alunos façam a definição dos testes de cada projeto, incluindo: a) Os níveis de teste; b) As técnicas propostas; b) Os métodos a serem utilizados; c) Os casos de testes gerados; e d) Os resultados de testes obtidos. Nas videoaulas são apresentadas os métodos e técnicas de testes usualmente abordados na literatura consolidada de Engenharia de Software.

Por fim, na etapa 7, os alunos realizam na última semana de curso suas apresentações finais, considerando um Roteiro de Exame Final (REF) disponibilizado pelos professores. Cada grupo deve fazer uma apresentação virtual de até 15 minutos. Nessa etapa 7, os professores avaliam os alunos nos seguintes critérios:

- Interesse, envolvimento, responsabilidade e compromisso;
- Planejamento e organização da apresentação;
- Domínio do conteúdo apresentado; e
- Adequação ao uso do tempo de apresentação.

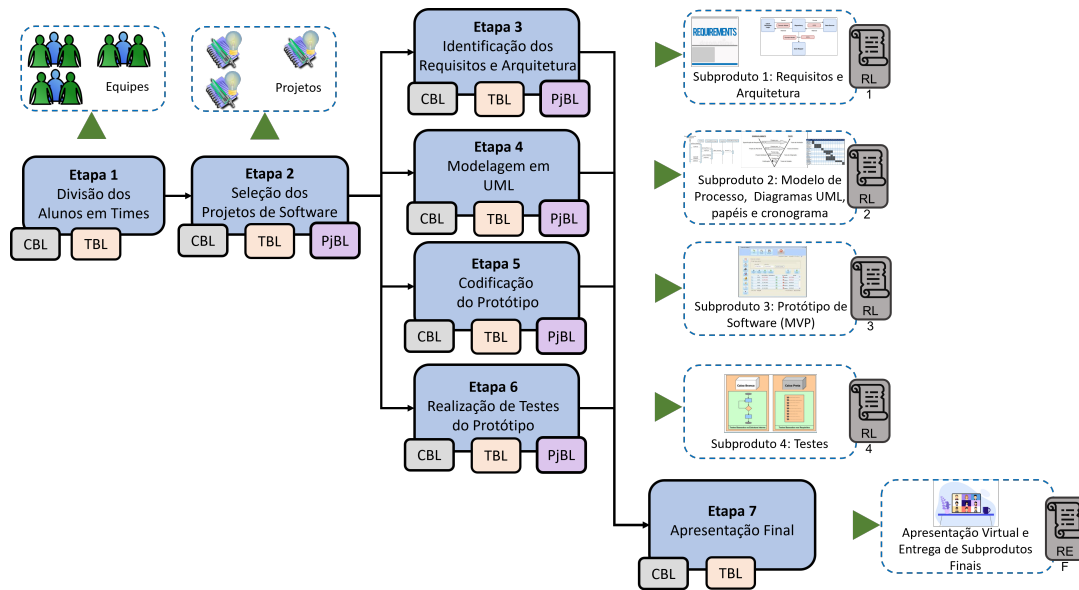


Figura 2: Modelo de Execução

Além da apresentação, os grupos devem entregar os subprodutos gerados durante o projeto, contendo as informações atualizadas e revisadas, em sua versão final, sendo eles:

- Documento de Requisitos e Diagrama de Blocos da Arquitetura - versão final do Subproduto entregue na RL 1;
- Relatório Descritivo com Modelo de Processo, Diagramas UML, papéis e cronograma - versão final do Subproduto entregue na RL 2;
- Os códigos-fonte atualizados - versão final do Protótipo entregue na RL3;
- Plano e Resultados de Testes - versão final do Subproduto entregue na RL 4;
- Slides da Apresentação; e
- Protótipo.

Os alunos são incentivados a manterem todos os subprodutos atualizados até o final do projeto, mesmo que um subproduto já tenha sido entregue aos professores, na etapa 7, os professores recebem os subprodutos novamente com as possíveis mudanças de conteúdo necessárias e identificadas pelos alunos, ao longo do desenvolvimento. O curso semestral foi realizado em 16 semanas, que é o padrão no Instituto Tecnológico de Aeronáutica. As etapas foram distribuídas no tempo, conforme a Figura 3.

4 SÍNTESE DOS PROJETOS DESENVOLVIDOS

O modelo de ensino virtual de Engenharia de Software apresentado neste trabalho foi aplicado no 2º Semestre de 2020. A turma contou com 22 alunos. Os dois professores se reversaram na confecção de videoaulas e na criação dos roteiros.

4.1 OnlineParty

A proposta do projeto consistia em desenvolver um jogo, *multi-player*, *online* e 3D, onde os jogadores adicionam um apelido *nick-name* de usuário e, em seguida, selecionam a sala que desejam entrar. Além disso, durante o jogo, um usuário poderiam interagir com outros usuários por meio do sistema de chat. O jogo foi concebido com as *engines* Unity e Photon com auxílio do *package* Photon Unity Networking (PUN). Uma tela do protótipo concebido pelos alunos é apresentada na Figura 4. Os 4 integrantes tiveram papéis específicos no desenvolvimento do projeto, conforme apresentado na Tabela 3.

Tabela 3: Distribuição dos Papéis pelos Alunos - OnlineParty

| Aluno | Papel | Responsabilidade |
|---------|-----------------------------------|---|
| Aluno 1 | Desenvolvedor | Projeto de design <i>low-level</i> do software e Implementação. |
| Aluno 2 | Engenheiro de Requisitos e Testes | Definição de Requisitos e Criação dos Casos de Testes. |
| Aluno 3 | Arquiteto de Software | Projeto de design <i>high-level</i> e a interação entre os módulos. |
| Aluno 4 | Gerente de Projeto | Liderança da equipe e intermédio com os clientes (professores). |

4.2 SafePoll

O protótipo de software SafePoll consiste em uma aplicação web que será utilizada para criação e realização de eleições. As eleições serão destinadas a pequenos grupos de indivíduos, podendo ser personalizadas segundo o interesse do grupo. O sistema será utilizado

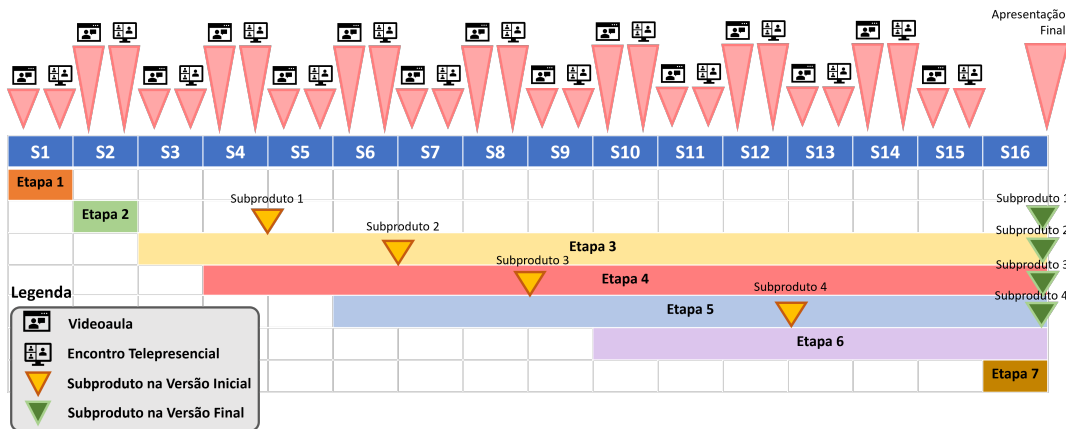


Figura 3: Etapas Distribuídas nas 16 Semanas

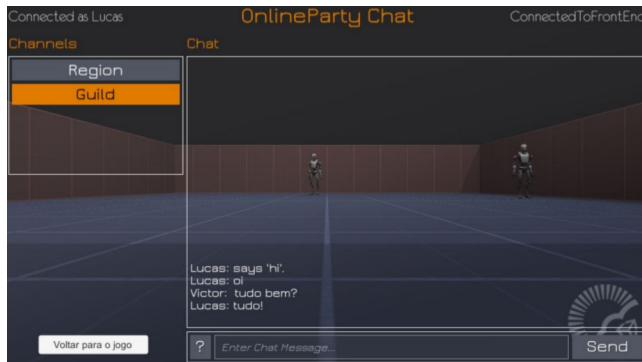


Figura 4: OnlineParty

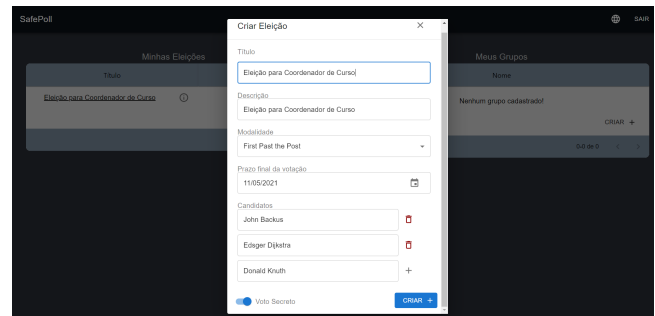


Figura 5: SafePoll

por grupos pequenos de pessoas que necessitam realizar eleições com certa frequência, permitindo a criação de grupos permanentes no próprio servidor do aplicativo de modo a facilitar o processo. O sistema também poderá ser utilizado por grupos que buscam realizar eleições pontuais, sem a necessidade de estar cadastrado em algum desses grupos.

Aquele que criar a eleição, é o administrador da eleição, podendo personalizá-la de acordo com seu interesse e definirá o grupo de usuários que poderão participar da eleição. Cada participante da eleição, denominado eleitor, tem acesso a um *link* que o permite votar. Ao término da eleição, os resultados são disponibilizados para os usuários participantes do processo. Uma tela do protótipo concebido pelos alunos é apresentada na Figura 5. Os 6 integrantes tiveram papéis específicos no desenvolvimento do projeto, conforme apresentado na Tabela 4.

4.3 FarmacITA

O objetivo do projeto é desenvolver um sistema independente, de uma farmácia independente, ou seja, que não faz parte de uma rede de farmácias, para otimizar o sistema de controle de estoque de remédios e de caixa e manter o registro de cadastro de usuários e de representantes dos laboratórios.

Tabela 4: Distribuição dos Papéis pelos Alunos - SafePoll

| Aluno | Papel | Responsabilidade |
|--------------|--------------------------|--|
| Alunos 1 e 2 | Desenvolvedor | Desenvolvimento da interface front-end. |
| Alunos 3 e 4 | Desenvolvedor | Desenvolvimento da interface back-end. |
| Aluno 5 | Engenheiro de Requisitos | Criação, gestão e controle dos requisitos. |
| Aluno 6 | Engenheiro de Testes | Criação dos Casos de Testes. |

Uma das principais funcionalidades deste projeto é conseguir consultar a quantidade de remédios em estoque, para ser avaliada a necessidade de uma nova compra e também verificar sua data de vencimento para poder retirá-lo de possível venda. Outra funcionalidade é que para os remédios de uso restrito haverá a necessidade da inserção do código do farmacêutico para ele poder ser disponibilizado ao cliente. O sistema também contabiliza os preços de cada venda realizada, tendo estes valores obtidos pela identificação do código ordem de venda, sendo possível obter assim um controle de caixa atualizado da operação de venda de medicamentos. Uma tela do protótipo concebido pelos alunos é apresentada na Figura 6.

Os 6 integrantes tiveram papéis específicos no desenvolvimento do projeto, conforme apresentado na Tabela 6.



| Id | Nome do Cliente | CPF | Telefone | Cargo | Admissão | Demissão |
|----|-------------------|----------------|-----------------|--------------------|---------------|----------|
| 1 | roberto galvão | 123.456.789-09 | (74) 75254-5128 | Gerente Financeiro | Nov. 30, 2020 | |
| 2 | adriano | 123.456.789-09 | (74) 37854-5487 | Balancista | Dec. 7, 2020 | |
| 3 | luis carla | 123.456.789-09 | (75) 46499-4634 | Caixa | Dec. 7, 2020 | |
| 4 | luis farmaceutico | 123.456.789-09 | (74) 67891-4634 | Farmacêutico | Dec. 6, 2020 | |

Figura 6: FarmacITA

Tabela 5: Distribuição dos Papéis pelos Alunos - FarmacITA

| Aluno | Papel | Responsabilidade |
|--------------|--|---|
| Aluno 1 | Engenheiro de Requisitos e Desenvolvedor | Criação, gestão e controle dos requisitos e Desenvolvimento da interface front-end. |
| Alunos 2 e 3 | Desenvolvedor | Desenvolvimento da interface back-end. |
| Aluno 4 | Desenvolvedor | Desenvolvimento da interface front-end. |
| Aluno 5 | Gerente de Projeto | Liderança da equipe e intermédio com os clientes (professores). |
| Aluno 6 | Engenheiro de Testes | Criação dos Casos de Testes. |

4.4 CasdVest

O objetivo do projeto CasdVest é o de promover uma extensão com novas funcionalidades no portal CasdVest, utilizado para gestão acadêmica dos alunos do Curso Alberto Santos Dumont (CASD). Este projeto visa a oferta de um curso preparatório de pré-vestibular para alunos de baixa renda. Os alunos deste time decidiram, ao invés de começar um projeto do zero, elaborar uma melhoria significativa no portal, com a inclusão, implementação e testes de um conjunto adicional de novos requisitos de software. Dentre as diversas melhorias introduzidas nessa nova versão do portal, os alunos deste time implementaram uma interface de administração e um módulo para inscrições para o processo seletivo de novos alunos. Uma tela do protótipo concebido pelos alunos é apresentada na Figura 7. Os 6 integrantes tiveram papéis específicos no desenvolvimento do projeto, conforme apresentado na Tabela 6.



Figura 7: CasdVest

Tabela 6: Distribuição dos Papéis pelos Alunos - CasdVest

| Aluno | Papel | Responsabilidade |
|--------------|--|---|
| Aluno 1 | Engenheiro de Requisitos e Desenvolvedor | Criação, gestão e controle dos requisitos. |
| Alunos 2 e 3 | Desenvolvedor | Desenvolvimento da interface back-end. |
| Aluno 4 e 5 | Desenvolvedor | Desenvolvimento da interface front-end. |
| Aluno 6 | Gerente de Projeto | Liderança da equipe e intermédio com os clientes (professores). |
| Todos | Gerente de Projeto | Criação dos Casos de Testes. |

5 AVALIAÇÕES E DISCUSSÕES DE RESULTADOS

5.1 Avaliação do Uso das Metodologias Ativas

O modelo de ensino virtual aplicado neste relato de experiência envolveu a completa integração de Teoria e Prática, consequentemente, os professores entenderam que a aplicação das metodologias ativas: *Project-Based Learning* (PjBL), *Team-Based Learning* (TBL) e *Competency-Based Learning* (CBL) seriam adequadas para aumentar o estímulo e interesse dos alunos na integração teórica e prática da disciplina.

No uso do TBL, conforme apresentado na etapa 1, a turma é dividida em times, com 4 a 6 integrantes cada. Assim, com a motivação dos alunos a buscarem livremente as ferramentas e plataformas necessárias para construção dos seus projetos, a TBL é importante para os alunos desenvolverem sua capacidade de trabalho em times e de forma integrada e assíncrona.

A CBL mostrou-se integrada em todas as etapas. Os papéis e atividades de cada membro do time foram atribuídos conforme as competências técnicas de cada aluno. Desde a etapa 1, os alunos são incentivados a avaliarem suas competências, definindo o papel de cada aluno no seu time. Alguns alunos notadamente possuem mais vontade de trabalhar na codificação, outros na especificação de requisitos, e assim os times tinham seus papéis baseados em competências. Assim:

- Nas competências comportamentais: alguns alunos possuíam comunicação marcante e liderança; e
- Nas competências técnicas: alguns alunos se mostraram mais aptos para escrita de requisitos, outros para modelagem da arquitetura e uso do UML, enquanto outros preferiram a codificação e testes.

Já o PjBL foi usado na escolha dos projetos (etapa 2) elaborados e executados por cada time. Os alunos tiveram liberdade na proposta de seus projetos, passando pelo crivo dos professores envolvidos. Após aceite, os alunos desenvolveram todas as atividades propostas e requeridas de seus projetos para atenderem os propósitos estabelecidos na disciplina, integrando de maneira incremental aspectos teóricos observados nas videoaulas com a prática no projeto.

5.2 Survey

Ao término do semestre letivo, os professores aplicaram um *Survey* para avaliar o *feedback* dos alunos quanto ao Modelo de Ensino Virtual de Engenharia de Software. O *Survey* foi composto de 6 itens, com uso da escala Likert. Afirmativas foram apresentadas e os alunos foram convidados a apresentar o seu grau de concordância com uma afirmação. Para isso, cada aluno marcou, na escala, a resposta que mais traduz sua opinião e considerando a escala original de 5 pontos, a saber: 1) discordo totalmente; 2) discordo; 3) neutro; 4) concordo; e 5) concordo totalmente. O *Survey* teve uma adesão de 18 alunos dos 22 da turma completa, representando uma amostragem de 81,82% dos alunos matriculados na disciplina. A Tabela 7 apresenta os 6 Itens do *Survey* (IS). A Figura 8 apresenta um gráfico com as respostas obtidas em percentual.

No item IS1, os alunos precisavam avaliar a afirmativa “*Eu acredito que a realização de um projeto permitiu a aplicação do conteúdo teórico em situação prática*”. Neste item, 28% das respostas foram 5 - *Concordo permanente* e 72% das respostas com 4 - *Concordo*. Assim, entende-se que 100% dos alunos que responderam, concordaram que o uso de um projeto permitiu a aplicação do conteúdo teórico. No entanto, apesar o aspecto positivo deste item, alguns alunos reportaram que precisaram recorrer a outros materiais para realizar o desenvolvimento do protótipo de software. Isso porque a disciplina não aborda com profundidade os inúmeros ambientes, linguagens e *frameworks* existentes sendo incentivado que os alunos buscassem livremente esses recursos para viabilizarem seus projetos na disciplina.

No item IS2, os alunos precisavam avaliar a afirmativa “*Eu acredito que o conteúdo abordado no curso foi utilizado no projeto*”. Neste item, 33% das respostas foram dadas com 5 - *Concordo permanente* e 67% das respostas dadas com 4 - *Concordo*. Assim, entende-se que 100% dos alunos concordaram que o uso de um projeto permitiu a aplicação do conteúdo teórico.

No item IS3, os alunos precisavam avaliar a afirmativa “*Eu acredito que a execução do projeto auxiliou na assimilação das práticas da Engenharia de Software*”. Neste item, 61% das respostas foram 5 - *Concordo permanente* e 39% das respostas com 4 - *Concordo*. Assim, entende-se que 100% dos alunos concordaram que o uso de um projeto permitiu a aplicação do conteúdo teórico.

No item IS4, os alunos precisavam avaliar a afirmativa “*Eu acredito que o balanceamento de atividades síncronas e assíncronas foi adequado, proporcionando acompanhamento da evolução dos Times*”.

Tabela 7: Itens do Survey

| IS | Item |
|-----|---|
| IS1 | Considerando o modelo de ensino virtual adotado, como você avaliaria a afirmação: “ <i>Eu acredito que a realização de um projeto permitiu a aplicação do conteúdo teórico em situação prática</i> ”. |
| IS2 | Durante a realização do Projeto, como você avaliaria a afirmação: “ <i>Eu acredito que o conteúdo abordado no curso foi utilizado no projeto</i> ”. |
| IS3 | Durante a realização do Projeto, como você avaliaria a afirmação: “ <i>Eu acredito que a execução do projeto auxiliou na assimilação das práticas da Engenharia de Software</i> ”. |
| IS4 | Considerando o modelo híbrido de encontros síncronos e disponibilização de conteúdo assíncrono (videoaulas, laboratórios, etc.), como você avaliaria a afirmação: “ <i>Eu acredito que o balanceamento de atividades síncronas e assíncronas foi adequado, proporcionando acompanhamento da evolução dos Times</i> ”. |
| IS5 | Considerando a viabilidade do desenvolvimento do projeto com os integrantes das equipes trabalhando remotamente, como você avaliaria a afirmação: “ <i>Eu acredito que o desenvolvimento de projetos com os integrantes das equipes trabalhando remotamente foi viável</i> ”. |
| IS6 | Sobre a escolha dos projetos, como você avaliaria a afirmação: “ <i>Eu acredito que os grupos terem a autonomia para definirem seus projetos foi adequada</i> ”. |

Neste item, 33% das respostas foram 5 - *Concordo permanente*; 39% das respostas com 4 - *Concordo*; e 17% das respostas com 3 - *Neutro*. Assim, entende-se que 83% dos alunos concordaram que o uso de um projeto permitiu a aplicação do conteúdo teórico, no entanto, nenhum aluno discordou, o que é um bom resultado. Em conversa posterior com os alunos, os professores receberam alguns *feedbacks* que suportam este resultado. Os alunos reportaram percepções diferentes e algumas conflitantes. Alguns alunos disseram preferirem encontros telepresenciais quinzenais, ao invés de semanais, já que as etapas 3, 4, 5 e 6 demoravam mais de uma semana. Outros alunos reportaram preferir que algumas videoaulas fossem ministradas em modo telepresencial.

No item IS5, os alunos precisavam avaliar a afirmativa “*Eu acredito que o desenvolvimento de projetos com os integrantes das equipes trabalhando remotamente foi viável*”. Neste item, 61% das respostas foram dadas com 5 - *Concordo permanente*; 33% das respostas dadas com 4 - *Concordo*; e 6% das respostas dadas (1 aluno) com 2 - *Discordo*. Assim, entende-se que 94% dos alunos concordaram que o trabalho remoto foi viável e apenas um aluno discordou. Em conversa com o aluno, este reportou que teve problema de sincronismo com outros integrantes de seu time com a forma remota.

No item IS6, os alunos precisavam avaliar a afirmativa “*Eu acredito que os grupos terem a autonomia para definirem seus projetos foi adequada*”. Neste item, 67% das respostas foram 5 - *Concordo permanente* e 33% das respostas com 4 - *Concordo*. Assim, entende-se que 100% dos alunos concordaram que a autonomia na seleção de projetos foi adequada.

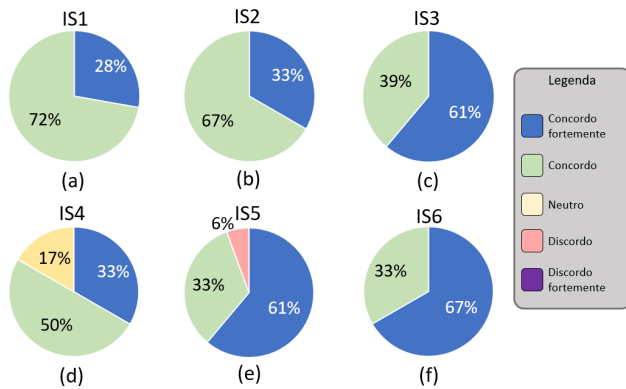


Figura 8: Resultados do Survey

5.3 Análise quanto aos Referenciais de Ensino

Existem as competências de caráter específico indicadas pelo Exame Nacional de Desempenho de Estudantes (ENADE) [4] e pela Sociedade Brasileira de Computação (SBC) para o bacharelado em Engenharia de Computação [27]. As competências foram avaliadas com 3 (três) possibilidades: (i) Incorporada, quando a competência é completamente prevista no modelo de ensino virtual; (ii) Parcialmente incorporada, quando apenas uma parte dos aspectos da competência é prevista no modelo de ensino virtual; e (iii) Não incorporada, quando nenhum aspecto da competência é previsto no modelo ensino virtual. Quanto às competências do ENADE [4] podemos destacar as seguintes correlações com o modelo de ensino virtual apresentado.

- (1) Conceber, especificar, projetar, construir, testar, verificar e validar sistemas de computação: **Incorporada**. Os alunos exercitaram no Projeto todas as ações previstas nessa competência;
- (2) Compreender e explicar as dimensões quantitativas de um problema: **Incorporada**. Os alunos exercitaram essa competência com a escolha de seus projetos, visando assim solucionar algum problema ou dificuldade existente com a concepção de seu protótipo de software;
- (3) Interpretar e resolver problemas computacionais, empregando recursos lógicos e/ou matemáticos: **Parcialmente incorporada**. A questão do problema foi abordada, mas não necessariamente empregando recursos matemáticos;
- (4) Implementar e gerenciar a segurança de sistemas de computação: **Parcialmente incorporada**: Alguns projetos previram a questão de segurança com a criação de telas de *login*, níveis de permissão e outros aspectos de segurança importantes;
- (5) Desenvolver sistemas integrados, incluindo o desenvolvimento de software para esses sistemas: **Não incorporada**. Os alunos desenvolveram projetos independentes;
- (6) Analisar, avaliar, desenvolver e otimizar software para arquiteturas, plataformas computacionais e sistemas de comunicação: **Parcialmente incorporada**. Os projetos tinham uma necessidade arquitetural prevista e com isso os alunos

precisaram definir as plataformas computacionais adequadas para o projeto. No entanto, não foi dado enfoque em sistemas de comunicação;

- (7) Desenvolver, implantar e configurar aplicações de software e/ou serviços em plataformas de hardware: **Incorporada**. Os times desenvolveram, implantaram e, quando aplicável, configuraram os protótipos de software, no entanto, sem um enfoque no hardware; e
- (8) Realizar estudos de viabilidade técnica, social e econômica de projetos, produtos e/ou serviços na área de computação: **Parcialmente incorporada**. Os alunos foram incentivados a realizar uma pesquisa e elicitação de requisitos com os potenciais clientes de seus protótipos de software. No entanto, o foco foi mais direcionado apenas para a concepção dos requisitos, sem uma análise mais profunda e direcionada para a parte social e econômica.

Já a Sociedade Brasileira de Computação (SBC) descreve vários referenciais para o bacharelado em Engenharia de Computação com competências de caráter específico [27] com forte interseção com a Engenharia de Software:

- (1) Especificar e validar os requisitos, projetar, implementar, verificar, implantar e documentar soluções de software baseadas no conhecimento apropriado de teorias, modelos e técnicas: **Incorporada**. Os alunos exercitaram no Projeto todas as ações previstas nessa competência;
- (2) Compreender as estruturas organizacionais e os papéis relacionados ao desenvolvimento de projetos, serviços e experimentos de Engenharia de Computação: **Parcialmente incorporada**. Os alunos tiveram autonomia, mas foram cobrados quanto aos papéis de cada um em seus projetos, conforme apresentado nas tabelas da Seção 4. No entanto, por serem projetos independentes, a parte de estruturas organizacionais não foi explorada. Também não houve um viés voltado para experimentação.

6 CONCLUSÃO

O objetivo deste trabalho foi apresentar um relato de experiência da aplicação de um modelo de ensino virtual de ensino de Engenharia de Software, envolvendo uma seleção de Metodologias Ativas, com foco em competências, times e projetos. Este relato envolve a disciplina *Fundamentos de Engenharia de Software*, disciplina obrigatória do 5º semestre do curso de graduação em Engenharia da Computação do ITA.

Como principais contribuições deste trabalho, destacam-se:

- O modelo de concepção de Engenharia de Software orientado por competências, times e projetos apresentados na Seção 3.1;
- O modelo de execução apresentado na Seção 3.2, que apresenta a operacionalização do modelo de concepção;
- A integração de 3 (três) metodologias ativas PjBL, TBL e CBL; e
- A análise de aderência do modelo de ensino virtual para dois importantes referenciais de ensino em Engenharia de Computação, como o ENADE; e o da SBC, ambos apresentados na seção 5.3.

Dentre os principais problemas apresentados na literatura [17], os professores dessa disciplina procuraram: (i) Aprimorar o processo de ensino e aprendizagem com o desenvolvimento dos protótipos no mesmo compasso do conteúdo teórico, com o uso de PjBL; (ii) Aproximar o que se ensina em termos de desenvolvimento de software da realidade existente no mercado de trabalho, explorando as competências de cada aluno, com o uso do CBL; e (iii) Enfatizar a importância do trabalho em grupo e iteração entre alunos e professores, com o uso de TBL.

Um ponto importante a considerar é que a disciplina foi ofertada com dois professores, que dividiram a carga docente em partes iguais, tanto na parte teórica, quanto de laboratório. Essa vantagem obviamente favoreceu uma melhor estruturação do modelo de ensino virtual apresentado neste trabalho, facilitando sua concepção e execução no momento delicado de ensino virtual em virtude da pandemia de COVID-19.

Como um primeiro trabalho futuro, os professores pensam em continuar aplicando o modelo de ensino virtual enquanto a pandemia de COVID-19 continuar e as aulas presenciais continuarem suspensas. Após o término da pandemia, os professores adaptarão o modelo para o ensino presencial. Como o curso já conta com as videoaulas com os tópicos da ementa já gravados, os professores pensam em não desperdiçar esses conteúdos gerados e analisarão como integrá-los ao ensino presencial. Existe uma expectativa de integração do *Flipped Classroom* (FC), onde os alunos poderão ter acesso antecipado às videoaulas e nos encontros presenciais viabilizar uma discussão do conteúdo com os professores e demais alunos, estimulando debates e discussões.

Um segundo trabalho futuro interessante, seria a aplicação do modelo de ensino virtual com projetos integrados, ao invés de projetos independentes. Isso poderia dar uma visão de integração em maior escala, fomentando a comunicação e disseminação das práticas através de subprojetos de um projeto maior e único, envolvendo todos os alunos da disciplina. Este trabalho futuro poderá incorporar, parcialmente ou mesmo integralmente, a competência “Desenvolver sistemas integrados, incluindo o desenvolvimento de software para esses sistemas”, prevista no referencial de competências do ENADE e não incorporada conforme analisado na Seção 5.3.

REFERÊNCIAS

- [1] Terry Barrett and Sarah Moore. 2010. *New Approaches to Problem-based Learning: Revitalising Your Practice in Higher Education* (1a ed.). Routledge.
- [2] Kathy Beckman, Neal Coulter, Soheil Khajenoori, and Nancy R. Mead. 1997. Collaborations: Closing the Industry–Academia Gap. *IEEE Software* 14, 6, 49–57.
- [3] Willian N. Bender. 2014. *Aprendizagem Baseada em Projetos: Educação Diferenciada para o Século XXI* (1a ed.). Penso.
- [4] Brasil. 2019. Portaria Nº 497, de 31 de Maio 2019. *Diário Oficial da União - Secão 1*.
- [5] Caio Henrique Coutinho, Johnny Cardoso Marques, Sarasuaty Megume Hayashi Yelisetty, Samoel Mirachi, Alexandre Lima Possebon Ribeiro, Marcelo Amaral da Silva, Adilson Marques da Cunha, and Luiz Alberto Vieira Dias. 2012. Developing a Smart Grids System as a PBL with Agile Method on Computer Engineering Courses. In *2012 Ninth International Conference on Information Technology - New Generations*. 325–330.
- [6] Daniela America da Silva, Fabio Kfourri, Samara Cardoso dos Santos, Luiz Henrique Coura, Wilson Cristoni, Gildarcio Sousa Gonçalves, Leonardo Guimarães dos Santos, Jose Crisostomo Ozorio Junior, Breslei Max Reis da Fonseca, Jean Carlos Lourenco Costa, Juliana Pasquini, Alexandre Nascimento, Johnny Marques, Luiz Alberto Vieira Dias, Adilson Marques da Cunha, Paulo Marcelo Tasinaffo, Beatriz Perondi, Anna Miethke-Morais, Amanda Cardoso Montal, Solange Regina Giglioli Fusco, and Thiago Sakamoto. 2019. Urgent and Emergency Care: An Academic Application System Case Study. In *16th International Conference on Information Technology-New Generations (ITNG 2019)*, Shahram Latifi (Ed.). Springer International Publishing, Cham, 143–152.
- [7] Rafael Augusto da Silva Coelho, Adilson Marques da Cunha, Anderson Alves Gomes, Eliezer Rodrigues Segeti, Johnny Cardoso Marques, Leonardo Miranda Vicente, Luiz Alberto Vieira Dias, Milton Luiz Abrunhosa, Roger Nobuyuki Kamoi, Samoel Mirachi, Tiago Josue Diedrich, and Valdir da Costa Guerra. 2014. Developing a CDS with Scrum in an interdisciplinary academic project. In *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*. 5D6–1–5D6–13.
- [8] Delcia Enricone. 2002. *Ser Professor* (3a ed.).
- [9] Thais Ferreira, Davi Viana, Juliana Fernandes, and Rodrigo Santos. 2018. Identifying Emerging Topics and Difficulties in Software Engineering Education in Brazil. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering (Sao Carlos, Brazil) (SBES '18)*. Association for Computing Machinery, New York, NY, USA, 230–239.
- [10] Decio Fischetti. 2010. *Instituto Tecnológico de Aeronáutica, 60 anos*.
- [11] Gildarcio Sousa Gonçalves, Glaydson Luiz Bertoze Lima, Rene Esteves Maria, Ramiro Tadeu Wisniewski, Mayara Valeria Morais dos Santos, Manasseis Alves Ferreira, Alexandre Chaves da Silva, Andre Olimpio, Andre Gomes Lamas Otero, Luiz Eduardo Guarino de Vasconcelos, Luciano Yukio Coutinho Sato, Henrique Nunweiler Angelim Silva, Johnny Cardoso Marques, André Luiz Pierre Mattei, Adilson Marques da Cunha, Luiz Alberto Vieira Dias, and Osamu Saotome. 2015. An interdisciplinary academic project for spatial critical embedded system agile development. In *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. 8C3–1–8C3–11.
- [12] Karin J. Hess, Rose L. Colby, and Daniel A. Joseph. 2020. *Deeper Competency-Based Learning: Making Equitable, Student-Centered, Sustainable Shifts*. Corwin.
- [13] Sangkyun Kim, Kibong Song, Barbara Lockee, and John Burton. 2018. *Gamification in Learning and Education: Enjoy Learning Like Gaming*. Springer.
- [14] José Lima, Mozart Alves Júnior, Andres Moya, Ricardo Almeida, Patricia Anjos, Maria Lencastre, Roberta Fagundes, and Fernanda Alencar. 2019. As Metodologias Ativas e o Ensino em Engenharia de Software: uma revisão sistemática da literatura. In *Anais do XXV Workshop de Informática na Escola (Brasília)*. SBC, Porto Alegre, RS, Brasil, 1014–1023.
- [15] Larry K Michaelsen, Arletta Bauman Knight, and L Dee Fink. 2004. *Team-Based Learning: A Transformative Use of Small Groups in College Teaching*. Stylus Publishing.
- [16] Andreza Mourão. 2017. Uma Proposta da Eficiência do Uso da Metodologia Ativa Baseada em Problemas, Utilizando Dojo de Programação, Aplicada na Disciplina de Lógica de Programação. In *XXIII Workshop de Informática na Escola*. 667–676.
- [17] Severino Paiva and Álvaro Medeiros. 2011. ESCollab : Uma metodologia colaborativa voltada para o ensino de Engenharia de Software. In *Anais do XXII Simpósio Brasileiro de Informática na Educação*. 630–639.
- [18] Bens Pardamean, Harjanto Prabowo, Hery Muljo, Teddy Suparyanto, Eryadi Masli, and Jerome Donovan. 2017. The Development of Blended-Learning Teaching Portfolio Course Using TBL Approach. *International Journal of Virtual and Personal Learning Environments* 7, 30–43.
- [19] Jurgen Pott. 2019. *Professional MVP Development: things to consider when developing a minimum viable product*. Amazon Pub.
- [20] Rafael Prikladnicki, Adriano Bessa Albuquerque, Christiane G. von Wangenheim, and Reinaldo Cabral. 2009. Ensino de Engenharia de Software: Desafios, Estratégias de Ensino e Lições Aprendidas. In *Anais do II Fórum de Educação em Engenharia de Software*.
- [21] Beatriz Pérez and Angel Rubio. 2020. A Project-Based Learning Approach for Enhancing Learning Skills and Motivation in Software Engineering. 309–315.
- [22] Lais S. Siles, Mayara V. M. Santos, Romulo A. Rodrigues, Lineu A. L. Filho, João P. T. Siles, Renê Esteves Maria, Johnny C. Marques, Luiz A. V. Dias, and Adilson M. da Cunha. 2018. An Integrated Academic System Prototype Using Accidents and Crises Management as PBL. In *Information Technology - New Generations*, Shahram Latifi (Ed.). Springer International Publishing, Cham, 419–427.
- [23] Todd Stanley. 2019. *Case Studies and Case-Based Learning: Inquiry and Authentic Learning That Encourages 21st-Century Skills*. Prufrock Press.
- [24] Fundação Getulio Vargas. 2015. Sala de Aula Invertida. *El! Ensino Inovativo*.
- [25] Christiane Gresse von Wangenheim and Djoni Antonio da Silva. 2009. Qual conhecimento de Engenharia de Software é importante para um profissional de software?. In *Anais do II Fórum de Educação em Engenharia de Software*.
- [26] Bruno Warin, Omar Talbi, Christophe Kolski, and Frédéric Hoogstoel. 2016. Multi-Role Project (MRP): A New Project-Based Learning Method for STEM. *IEEE Transactions on Education* 59, 2, 137–146.
- [27] Avelino Zorzo, Daltro Nunes, Eivaldo Matos, Igor Steinmacher, Jair Leite, Renata Araujo, Ronaldo Correia, and Simone Martins. 2017. *Referenciais de Formação para os Cursos de Graduação em Computação*. Technical Report. Sociedade Brasileira de Computação.