

# UM SERVIÇO PARA POSSIBILITAR FLUXOS CONFIGURÁVEIS PARA A REALIZAÇÃO DE ATIVIDADES COOPERATIVAS ON-LINE<sup>1</sup>

Nikolas Bernardes Vieira de Freitas<sup>2</sup>

Rodrigo Prestes Machado<sup>3</sup>

**Resumo:** Ao longo dos anos, os estudos sobre Aprendizagem Cooperativa demonstram alguns benefícios para os estudantes, pois apontam para um aumento na retenção de conhecimentos e também melhorias de aspectos psicológicos. Da mesma forma, trata-se de uma metodologia que demonstra flexibilidade, pois pode ser aplicada em diferentes matérias e em contextos de educação presencial e também on-line. Porém, construir sistemas que apoiem práticas de Aprendizagem Cooperativa se apresenta como um desafio. Primeiro, por se tratarem de aplicações que gerenciam atividades de grupo, apresentam complicações inerentes à implementação de sistemas de *groupware*, como questões de rede e sincronismo de dados. Segundo, como cada prática de Aprendizagem Cooperativa normalmente estabelece um fluxo de trabalho, existe uma complexidade de se generalizar regras de negócios ao ponto de dificultar a implementação de aplicações que suportem mais de uma prática. Assim, o objetivo deste trabalho foi a construção de um micro-serviço que trate das questões de *groupware* e que consiga generalizar regras de negócio a fim de sustentar a implementação de diversos fluxos. O sistema proposto obteve uma alta taxa de cobertura de testes, confiabilidade individual nos testes unitários, e coletiva em testes de integração, nos componentes que o compõem. Foi possível criar micro-executores de tarefas (*steps*) parametrizados e modularizados, tornando o fluxo de atividade flexível e confiável, concretizando o objetivo deste trabalho.

**Palavras-chave:** Aprendizagem cooperativa, ferramenta educacional, sistema para aprendizagem cooperativa, *workflow* flexível.

## 1 INTRODUÇÃO

Considerando a diversidade de metodologias utilizadas em sala de aula, o professor como organizador de grupos e atividades pedagógicas, pode propor aos estudantes que participem na resolução de tarefas em ao menos três modelos distintos: competitivo, individual e cooperativo. No modelo **competitivo** os alunos acabam por possuir uma interdependência negativa, a interação da competição acaba por gerar um ambiente que limita o sucesso de todos, pois para alguns ganharem, conseqüentemente outros terão que perder, ocasionando casos onde alunos não se esforçam ou até mesmo desistem da atividade proposta por não acreditarem que sejam capazes de vencer seus colegas. Outra opção para uma

---

<sup>1</sup> Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas para Internet.

<sup>2</sup> Graduando do Curso Superior de Tecnologia em Sistemas para Internet. E-mail: nikolasbvf@gmail.com.

<sup>3</sup> Professor Orientador do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS). E-mail: rodrigo.prestes@poa.ifrs.edu.br.

atividade é o modelo de resolução **individual**. Nesse modelo o sucesso da atividade não depende da cooperação ou do sucesso de um grupo, por este motivo os alunos não possuem um estímulo ou uma metodologia para se auxiliarem. Se a atividade é proposta no modelo **cooperativo**, os estudantes participantes acabam por depender de seu grupo para o sucesso da atividade, ocasionando uma interdependência positiva entre outras características consideradas positivas para o aprendizado (MACHADO, 2019).

O aprendizado cooperativo, o qual foi o modelo de aprendizagem focado neste artigo, é um modelo em que alunos participam de atividades na forma coletiva, interativa **racional** e coordenada. Quatro elementos formam uma atividade cooperativa: interdependência positiva, interação racional, responsabilidade individual (*accountability*) e habilidades interpessoais (JOHNSON; JOHNSON; HOLUBEC, 1984).

Apesar do crescimento expressivo da utilização de estratégias de aprendizado cooperativo, Slavin (1999) levanta uma preocupação quanto à qualidade na implementação destas estratégias. Para uma atividade em grupo se tornar cooperativa é necessário a existência de um conjunto de requisitos, dentre eles dois acabam por se destacar na literatura: interdependência positiva e responsabilidade individual (MARQUES, 2017). O uso de versões **informais** de estratégias de aprendizagem cooperativa costumam falhar no quesito responsabilidade individual, um dos destaques citados, ocasionando o surgimento de alunos que desempenham papéis de “pensadores” enquanto outros assumem papel de passivos durante as execução de atividades (SLAVIN, 1999).

Muitos confundem a modernização de materiais utilizados em sala de aula com uma mudança no formato de metodologia de aprendizagem. Porém a modernização de materiais como lousas eletrônicas que auxiliam o professor na prática de suas atividades, pode acabar por contribuir com o modelo clássico de relacionamento vertical, onde o professor age como um repassador de informações (CAMARGO; DAROS, 2018). Mas, por meio da inovação tecnológica, podemos criar novas ferramentas com o intuito de auxiliar o professor a viabilizar o uso de mais metodologias de aprendizagem em sala de aula. Uma ferramenta computacional apoiada por fluxos (*workflow*) baseados em modelos de atividades cooperativas estruturadas pré-cadastradas, porém com a possibilidade também de criação ou customização para que a autonomia do professor seja preservada, poderia servir de auxílio ao gerar um ambiente controlado que dificultasse o acesso de usuários às liberdades que levam à falha no uso da técnica de aprendizagem cooperativa.

Apesar do modelo de relacionamento vertical em sala de aula não ser alterado apenas pela modernização de materiais já utilizados pelo professor (CAMARGO; DAROS, 2018), a

utilização de vídeo aulas, que tentam transportar o modelo praticado em salas de aula para o digital, são as ferramentas mais utilizadas no modelo de ensino à distância (EAD) (ABED, 2019). O uso de vídeo aulas auxilia na aprendizagem dos estudantes que não podem ou não possuem interesse em estar presentes em salas de aula presenciais, porém, além de vídeo aulas, observa-se em cursos EADs, uma deficiência em oferecer atividades cooperativas para os estudantes dessa modalidade de ensino. Parte dessa carência pode ser atribuída ao fato de existir uma baixa disponibilidade ou interesse na adesão de ferramentas planejadas exclusivamente para mediar atividades em grupo no EAD. Portanto, desenvolver ferramentas e serviços que coordenam a realização de atividades em grupo (*groupware*) em ambientes EAD, muitas vezes sem a presença de um mediador, se torna imprescindível.

O problema que tange a criação de uma ferramenta computacional com o propósito de criação, coordenação e execução de atividades pedagógicas no formato cooperativo, são as características desejadas nesta ferramenta: (1) flexibilidade na customização e criação de fluxos de atividades; (2) Suporte de diversos fluxos de atividades cooperativas paralelos e simultâneos para abranger maior número de participantes e atividades ao mesmo tempo. Do ponto de vista técnico existem os seguintes desafios: a ferramenta computacional criada necessita ser escalável, confiável e auditável por meio de logs e, quando desejado, que possuam histórico de atividades ou workflows salvos em banco de dados.

O objetivo geral deste trabalho foi criar um serviço que se baseasse nas técnicas de aprendizagem cooperativas, porém que fosse capaz de fornecer a flexibilidade de configuração de fluxo que um sistema *groupware* demanda. Além das regras de negócio esperadas para a flexibilidade de fluxo, o serviço foi arquitetado para se encaixar em um ambiente de microsserviços, possuindo a possibilidade de escalabilidade em múltiplas instâncias, não possuindo dependência em memória para a execução das atividades.

Este texto foi fracionado em cinco partes: a primeira é fundamentação teórica do trabalho e a comparação com trabalhos relacionados, a segunda é descrição do sistema que contém os principais diagramas que estruturam a aplicação, a terceira é a metodologia de pesquisa, na quarta seção se encontram os resultados e a discussão, e a quinta seção contém a conclusão.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 REFERENCIAL TEÓRICO

#### 2.1.1 Diferença entre colaboração ou cooperação

Apesar de conceitos de colaboração e cooperação não terem uma distinção clara ou universal, provavelmente devido ao fato de alguns autores seguirem linhas de pensamento divergentes sobre o assunto, é importante para este trabalho salientar o significado de cooperação segundo a linha de Piaget. Para Piaget (1973) a cooperação se alicerça na interação **racional**, exige o respeito mútuo e reciprocidade, pois é um método interativo e estruturado, que busca pela descentração intelectual através de trocas argumentativas havendo a construção de normas racionais e morais (COGO, 2006; MACHADO, 2019). Quando não existe respeito mútuo, ocorre a inibição da cooperação, pois a consequência é o surgimento da coação, e “a diferença essencial entre coação e cooperação, é que uma impõe regras totalmente elaboradas e a outra impõe um método de elaboração das próprias regras” (RAMOS, 1996, p. 60).

#### 2.1.2 Aprendizagem cooperativa

O modelo de aprendizagem cooperativa segundo Johnson, Johnson e Holubec (1984), o qual é o modelo de aprendizagem foco neste trabalho, é constituída por pelo menos 4 elementos base: (1) interdependência positiva, que se realiza por meio da divisão dos objetivos e das tarefas de trabalho, do compartilhamento de materiais, recursos, informações, e por meio da atribuição de papéis aos membros da equipe; (2) interação, que proporcionará à interdependência e o compartilhamento de ações e de saberes entre os integrantes do grupo; (3) responsabilidade individual, a qual impulsiona a Aprendizagem Cooperativa e descreve o processo de construção do conhecimento efetuado individualmente; (4) habilidades interpessoais, que resultam da necessidade dos estudantes adquirirem aptidões pessoais para a participação em atividades em grupo. Sendo de consenso mais geral por autoridades do assunto, a presença dos dois elementos: interdependência positiva e responsabilidade individual (*accountability*) (ANTIL *et al.*, 1998).

### 2.1.3 Constituição de uma ferramenta de aprendizagem cooperativa

Segundo o artigo publicado pela universidade de Harvard: *What Your Pre-Covid Course Design Was Missing*, de (PESHKAM, 2020) existem 5 elementos-chaves que compõem uma ferramenta de aprendizado formalmente cooperativo: O primeiro elemento é a **Interdependência positiva**, onde cada estudante depende da contribuição, inclusão, e sucesso de outros para completar uma atividade com sucesso; O segundo é a **Responsabilidade Individual**, que cada estudante trabalha, tanto no processo de criação quanto no produto, são visíveis e avaliados por outros. No **Processamento em grupo**, que é o terceiro elemento, cada membro do grupo provê um *feedback* para os outros sobre as suas contribuições. No quarto elemento-chave, deve-se promover a **interação cara a cara (*awareness*)**, onde os estudantes auxiliam no sucesso um do outro e decidem dividir ou juntar seus recursos. As **Habilidades interpessoais**, que são o quinto e último elemento, incluem: ouvir, comunicar, construir confiança, e trabalhar sobre momentos de conflitos.

### 2.1.4 Exemplo de fluxos estruturados de atividade cooperativa

Existem fluxos de trabalho baseados em modelos estruturados, como por exemplo **Círculo de Escritores**, **Revisão por Pares**, **Jigsaw Classroom**, etc. Estes modelos visam promover uma aprendizagem ativa onde alunos participam de maneira interdependente positiva, conectando o sucesso individual ao sucesso do grupo como um todo (JACOBS; POWER; INN, 2016). No **Círculo de Escritores**, por exemplo, existem adaptações planejadas conforme o objetivo da atividade ou a área de aplicação da atividade. Algumas áreas de aplicação abrangem o estudo de linguagens, matemática, ciências, etc.

#### 2.1.4.1 *Círculo de escritores*

Podemos citar dois exemplos práticos sobre a técnica **Círculo de Escritores**: (1) Na aplicação de estudo de **linguagens**, estudantes deverão escrever histórias em conjunto. Cada um inicia individualmente a sua história. Depois de um tempo determinado, o aluno passa a tarefa para o aluno da esquerda que decide como continuar a história. A atividade continua até que a história esteja de novo com o aluno que a iniciou e seja concluída pelo mesmo. (2) No estudo de **matemática**, a estrutura central da dinâmica se mantém, mas se alteram algumas características. Cada membro do grupo da atividade recebe um problema matemático que é

resolvido em múltiplos passos. Após resolver o passo atual, cada aluno entrega o problema para o colega ao lado (todos para o mesmo lado sempre) que resolverá o passo seguinte do problema. A atividade continua até o problema ser resolvido. Caso algum dos estudantes considere que há algo errado na solução proposta pelo colega, o mesmo pede uma pausa na atividade e inicia-se uma discussão em grupo para encontrar um senso sobre uma solução para o momento atual da atividade. A atividade encerra-se quando o grupo chega ao consenso de que todos os passos foram concluídos com êxito (ANEXO 1).

#### *2.1.4.2 Jigsaw Classroom*

Jigsaw Classroom<sup>4</sup> é um modelo de aprendizagem cooperativa que surgiu da necessidade de cessar conflitos e agressões entre alunos por diferenças étnicas após a suprema corte nos Estados Unidos em 1971, decidir pelo fim da segregação racial em escolas. A técnica se mostrou útil na geração de confiança entre alunos durante o aprendizado das matérias escolares. Uma motivação para o uso da técnica foi o fato de não precisar ser aplicada fora do horário programado de aula, motivando os alunos a participarem. Esta técnica é executada em 10 passos: (1) Formação de grupos: grupos de estudantes contendo entre 5 e 6 alunos, os grupos precisam ser diversos em termos de gênero, origem étnica, raça e habilidade para um melhor aproveitamento do modelo; (2) Escolha do líder: Aponta-se um estudante para o papel de líder, geralmente este estudante é o mais maduro do grupo. O líder tem o objetivo de manter a organização do grupo; (3) Divisão de tarefas: Através do tema geral, como por exemplo segunda-guerra mundial, é dividido em 5 ou 6 assuntos, todos os grupos recebem os mesmos assuntos; (4) Atribuição de tarefas: cada membro o grupo fica responsável por um assunto originado da divisão de tarefas, por exemplo o aluno A fica responsável pelo ascensão de Hitler na pré-Alemanha nazista, a aluna B fica responsável por pesquisar sobre o avanço do exército vermelho, etc; (5) Tempo de leitura: Um tempo pré-determinado para que os alunos possam ler sobre o seu tema por pelo menos duas vezes. Não há necessidade para memorização; (6) Grupo de experts: Forma-se novos grupos temporários sobre determinado segmento da tarefa geral, contendo um membro de cada grupo, desde que tenha como atribuição o segmento do novo grupo formado, e iniciam-se discussões sobre o tema por um tempo pré-determinado. Exemplificado, todos os alunos que foram atribuídos ao assunto “avanço do exército vermelho” serão temporariamente colocados no mesmo grupo para discutirem sobre o assunto, o mesmo vale para o resto dos assuntos; (7)

---

<sup>4</sup> Disponível em: <https://www.jigsaw.org/#overview>. Acesso em: 25 dez. 2021.

De volta ao grupo: Todos os alunos retornam ao seu grupo de origem; (8) Apresentação do segmento: Cada aluno apresenta para o seu grupo o seu segmento do tema geral e é esperado que seus colegas possam fazer perguntas; (9) Observância nos grupos pelo professor: Apesar da autonomia dos alunos ser importante, o professor deve observar o andamento da tarefa por grupos e pode intervir quando achar necessário. Mas o objetivo do líder do grupo é cumprir o papel de intervir para manter a dinâmica do grupo a melhor possível, o autor sugere que é preferível que o professor gorjeie ao líder do grupo dicas de como lidar com problemas do grupo; (10) Encerramento da sessão: Como sendo o último passo da tarefa, é aplicado um teste individual para avaliar individualmente os alunos.

### 2.1.5 Verificação de sistema através de testes

Testes são técnicas que procuram garantir a qualidade nas entregas de requisitos do sistema e encontrar falhas no código do programa. Durante a implementação é possível fazer a verificação através de testes unitários, ou quando a modificação no *software* já está considerada pronta para a fase de produção, usam-se testes de integração para verificar como o *software* se comporta em cenários que envolvem a interação com outros sistemas simulando um ambiente de produção. Martin (2008), também conhecido como “Uncle Bob”, sugere que não existe código limpo sem testes, não importando o quão claro ou de fácil manutenção se encontra o código, pois além de garantirem maior segurança e previsibilidade no comportamento da aplicação. Testes quando bem escritos geram informações sobre o comportamento geral do *software*, servindo como um auxílio à documentação. Como guia para melhores práticas de testes, foi criado o acrônimo F.I.R.S.T.: Rápido (*Fast*), testes necessitam executar de forma ágil, de preferência em milissegundos, para que sejam executados mais frequentemente; Isolado (*Isolated, independent*), existe uma divergência no termo, Martin (2008) chama de “*independent*” e na documentação de boas práticas de testes da Microsoft (2021) prefere usar o termo “*isolated*”, porém ambos convergem na ideia de que cada teste é único e não possui dependência ou influência de outro; Repetível (*Repeatable*), testes devem gerar os mesmos resultados independente do ambiente em que rodar. Ambiente de nuvem, produção ou local sem internet não devem ser fatores decisivos para os testes; Auto-validado (*Self-validating, self checking*), os testes devem ter um resultado binário, indicar se o teste passou ou falhou, sem interpretação pós validação, e não devem ter influência humana no processo; Conveniente (*timely*), a escrita do teste não deve demorar muito tempo além da construção da *feature* que está sendo testada. É sugestão do Uncle Bob

que se use a técnica de escrita de testes antes do código, também conhecido como TDD, porém se os testes unitários estão difíceis a documentação de melhores práticas de testes da Microsoft sugere que se considere um design de código mais fácil para ser testado (MARTIN, 2008; MICROSOFT, 2021).

## 2.2 TRABALHOS RELACIONADOS

Não foram encontradas uma grande variedade de ferramentas pedagógicas que juntassem simultaneamente as seguintes características: aprendizagem cooperativa, atividade síncrona e com opção de construção de workflow. Porém serão listadas ferramentas que possuem atributos semelhantes ou sejam colaborativas.

### 2.2.1 Breakout EDU

Breakout EDU<sup>5</sup> é uma plataforma para atividades utilizando principalmente o ambiente virtual, porém possui materiais físicos disponibilizados pela plataforma. A ferramenta libera diversas atividades através de *kits* e assinaturas no site, inclusive materiais físicos como uma caneta “invisível”, alguns tipos de cadeados, entre outros materiais. As atividades são baseadas em jogos pedagógicos, é possível a criação de novos jogos além dos cadastrados bem como a customização de jogos já criados. Apesar de ser uma plataforma digital, as atividades parecem ter maior incentivo para ocorrer presencialmente, assim a plataforma digital serve como guia e o foco é na resolução em sala de aula. No site não foram encontradas muitas informações de aplicação de aprendizagem cooperativa online, apenas é citado que possui atividades que seguem o modelo cooperativo e indícios de que ocorrem na sala de aula física, não virtual.

### 2.2.2 FTD Digital

FTD é uma editora de livros pedagógicos que criou a FTD Digital<sup>6</sup>, que fornece aos alunos e professores ferramentas para aulas online, materiais didáticos virtuais, simulados e uma grande variedade de jogos e atividades digitais. O ambiente é rico em informação, ferramentas de aprendizagem e parcerias com *youtubers* da área de educação, permite também

---

<sup>5</sup> Disponível em: <https://www.breakoutedu.com/>. Acesso em: 28 dez. 2021.

<sup>6</sup> Disponível em: <https://digital.ftd.com.br/>. Acesso em: 28 dez. 2021.

que o professor crie plano de aula e formato de provas. Porém não foi localizada possibilidade de criação de atividades, ferramentas cooperativas ou qualquer tarefa colaborativa.

### **2.2.3 Serviço de atividade cooperativa**

O diferencial mais notável dentre os trabalhos relacionados é a aparente motivação de sua construção. O serviço de atividade cooperativa proposto por este trabalho possuiu desde a sua ideia inicial o objetivo de utilização em ambiente virtual, sendo mais preparado para atividades cooperativas de workflow configurável online do que os comparados por este trabalho.

Em resumo sobre os trabalhos relacionados, Breakout EDU é uma plataforma que objetiva a atividade presencial, sendo a fração online da plataforma apenas um guia para atividades utilizando um *kit* de materiais físicos fornecidos pela empresa ou para atividades virtuais no modelo individual; FTD Digital é uma extensão da editora FTD, provendo recursos adaptados e semelhantes aos utilizados em sala de aula, porém as atividades em grande parte seguem o modelo individual; O serviço implementado neste trabalho, foi projetado para que futuramente seja possível uma integração com outros microsserviços na plataforma Orion Services, e teve como motivação a implementação do modelo de atividades cooperativas síncronas online, sem a utilização de materiais físicos ou materiais didáticos vinculados a alguma editora, sendo apenas necessário algum dispositivo conectado à internet (APÊNDICE 1).

## **3 DESCRIÇÃO DO SISTEMA PROPOSTO**

O Serviço condutor de atividades gerou um ambiente controlado para *groupware*, onde as ações dos alunos foram geridas e limitadas por um fluxo de trabalho (*workflow*) composto por um conjunto de regras pré-definidas e aplicadas conforme determinadas fases da atividade. O usuário que fez o papel de professor pôde escolher durante a criação de uma atividade entre um modelo estruturado pré-cadastrado ou a elaboração de um fluxo novo. O objetivo das atividades pré-cadastradas foi gerar um ambiente controlado que dificulte ao professor cometer erros que o levariam à falha na tentativa de uso da técnica de aprendizagem cooperativa. Contudo, para preservar a autonomia do professor na elaboração de atividades, o sistema proposto disponibilizou a possibilidade de criação e customização de fluxos de

trabalho, não limitando a viabilidade do sistema somente aos modelos cooperativos pré-cadastrados.

### 3.1 ARQUITETURA PARA UTILIZAÇÃO DO SISTEMA

A arquitetura para a utilização do sistema proposto foi composta por diversos microsserviços que operam em conjunto como uma solução completa. As responsabilidades de edição de documento e gerenciamento da atividade cooperativa foram divididas para que os serviços sejam mais modulares e não acumulem funções. O serviço descrito como "Activity" foi responsável por gerenciar a criação, inclusão de alunos, envio de atualização e organização de uma atividade, repassando de forma assíncrona a edição do documento para o serviço de edição. O serviço Editor foi responsável por guardar, atualizar e enviar o documento conforme solicitado. A plataforma Orion Services, fora da *stack*, apoia os serviços internos fornecendo informações e o gerenciamento de dados não produzidos e controlados na *stack*, como dados e cadastros de usuário e serviço de autenticação, por exemplo. Para não haver dúvidas quanto aos serviços produzidos por este trabalho na *stack* apresentada, este trabalho teve o objetivo de produzir **somente** o serviço **Activity**. A apresentação da arquitetura tem o objetivo de servir como uma ilustração de como o serviço proposto por este trabalho foi projetado para trabalhar em uma arquitetura de micro-serviço.

A comunicação interna na *stack* é composta majoritariamente por sistema de filas (*brokers*), ou seja, serviços internos na *stack* enviam mensagens entre si de maneira assíncrona. Para disponibilizar os serviços para a internet geral é definido serviço de *gateway* que se conectará às filas (*broker*) para receber e enviar informações para os serviços internos e abstrair essa comunicação através de um túnel de comunicação bidirecional utilizando *websocket*, assim permitindo que navegadores ou serviços que não possuem integração com fila possam utilizar a API construída (APÊNDICE 2A).

### 3.2 SEQUÊNCIA DE CRIAÇÃO DE UMA ATIVIDADE

O serviço de atividade recebe a intenção de criação da atividade. O processo de criação de atividade também inclui o serviço de editor. O serviço de Atividade através de uma requisição HTTP, requisita ao para o serviço Editor a criação de um documento para edição. tais como proibição de deletar conteúdo geral, deletar conteúdos escritos pelos colegas, edição de linhas não proprietárias, etc (APÊNDICE 2B).

### 3.3 FLUXO DA PRÁTICA DA ATIVIDADE

O fluxo da prática de atividade está ligado ao cliente que possui a intenção de praticar a atividade, muitas vezes essa intenção envolve a edição do documento que está sendo produzido pela atividade cooperativa. Para ocorrer o fluxo, o cliente envia a requisição ao gateway que repassa os dados ao serviço de Atividade. Ao receber os dados e a intenção da prática da tarefa, o serviço executa os *steps* cadastrados no *workflow* da atividade, verificando se o cliente pode efetuar tal transação e, em caso positivo, prossegue para a etapa de formatação da mensagem para solicitar a edição do documento ao serviço Editor. A partir deste momento os serviços trabalham em paralelo, o serviço de Atividade segue processando a atividade e envia uma atualização publicando em uma fila do broker; o serviço de Editor que recebe a mensagem do *broker* com a intenção de edição do documento da atividade, e verifica qual foi o tipo de modificação solicitada pelo cliente. Se o tipo de edição estiver conforme as regras pré estipuladas de edição, o documento é editado e enviado para uma fila gerenciada por um *broker* que envia a atualização do documento (APÊNDICE 3).

## 4 METODOLOGIA DE PESQUISA

Para este trabalho foi utilizado o Design Science Research Process (DSRP). O método proposto e suas 6 etapas para validação de criações de novas tecnologias e trabalhos na área de sistemas de informação (PEFFERS *et al.*, 2006). Este método foi escolhido pois se adequa com a produção e validação do artefato tecnológico proposto por este trabalho.

### 4.1 ETAPAS DA METODOLOGIA DE PESQUISA

Este trabalho utilizou o processo DSRP de 6 etapas: identificação do problema, design e desenvolvimento, demonstração, validação e comunicação. As etapas estão descritas nos itens seguintes.

#### 4.1.1 Identificação do problema e motivação

Durante a pesquisa deste trabalho se constatou a baixa adesão de atividades em modelos cooperativos principalmente em EAD, execuções equivocadas deste modelo de atividades e a baixa oferta de ferramentas tecnológicas para auxílio desta prática. Os

problemas citados e a possibilidade deste trabalho servir como complementação do trabalho do professor Rodrigo (2019) motivaram a criação deste trabalho.

#### **4.1.2 Objetivos para a solução**

Servir como auxílio na execução de atividades no modelo cooperativo, oportunizar a criação de *workflows* configuráveis e flexíveis e assessorar a metodologia cooperativa durante a execução de uma atividade e integrar na plataforma aberta Orion Service, que é uma plataforma de código livre tendo o seu código fonte hospedado no site Github.

#### **4.1.3 Design e desenvolvimento**

O design do artefato foi composto por microsserviços que em conjunto elaboram o cumprimento dos objetivos, dentro de um ambiente fechado também conhecido como *stack* e a abertura para comunicação via internet foi através de um serviço de gateway que também funcionou como um facilitador. Os recursos fornecidos para uma aprendizagem cooperativa através do sistema criado, foram os necessários para a criação e execução de atividades cooperativas. As atividades pré-cadastradas foram majoritariamente com base na obra *The Teacher's Sourcebook for Cooperative Learning* (JACOBS; POWER; INN, 2016). Por ser um projeto *open-source*, a modelagem pode sofrer alterações conforme o andamento da aplicação e as novas descobertas providas do desenvolvimento, inclusive após a entrega deste trabalho. Porém o mesmo teve seus diagramas e descrições do ferramental atualizados até a data de entrega deste artigo.

O desenvolvimento foi produzido seguindo a metodologia ágil, com entregas através de *sprints* que estipulam um prazo definido para uma lista de tarefas, que no contexto deste trabalho possuíram 2 semanas. A metodologia ágil possui 4 premissas fundamentais que foram desfrutadas para organização do desenvolvimento: “indivíduos e interações mais que processos e ferramentas, Software em funcionamento mais que documentação abrangente, Colaboração com o cliente mais que negociação de contratos, Responder a mudanças mais que seguir um plano” (AGILE SOFTWARE DEVELOPMENT, 2001, n.p., tradução nossa) Para o cumprimento da metodologia ágil no desenvolvimento deste trabalho, entende-se como cliente a plataforma Orion Service e a equipe que o compõem.

#### 4.1.4 Demonstração

A demonstração do funcionamento da API foi dada através da execução de testes criados baseados em cenários que descreveram os comportamentos esperados pelos componentes da ferramenta criada por este trabalho.

#### 4.1.5 Avaliação

O sistema foi avaliado pelos testes unitários para garantir o funcionamento de cada unidade que compõem os serviços, e pelos testes de sistema baseados em cenários hipotéticos do uso da ferramenta, seguindo a técnica Behavior Driven Development (BDD).

#### 4.1.6 Comunicação

A comunicação é feita através da publicação deste trabalho na rede social profissional LinkedIn<sup>7</sup>. Será feita uma publicação que incluirá o endereço virtual do código fonte deste trabalho, que se encontra no Github<sup>8</sup>.

### 4.2 CENÁRIOS DOS CASOS DE TESTES

Os cenários planejados para os testes da aplicação foram baseados no fluxo completo de uma atividade: criação de uma atividade, inclusão de estudantes na atividade, controle da atividade conforme o fluxo de trabalho proposto e a criação de fluxos configuráveis.

#### 4.2.1 Casos de teste

Nesta seção será abordado os cenários de teste que avaliam se o sistema proposto por este trabalho atingiu seus objetivos. Em todos os cenários o usuário é considerado devidamente autenticado, visto que o serviço produzido por este trabalho não possui a responsabilidade de autenticação na *stack* de microsserviços onde o mesmo se encontra. (APÊNDICE 4).

---

<sup>7</sup> Disponível em: <https://www.linkedin.com/> Acesso em: 05 jul. 2022.

<sup>8</sup> Disponível em: <https://github.com/orion-services/activity> Acesso em: 05 jul. 2022.

## 5 RESULTADOS E DISCUSSÃO

O micro-serviço produzido foi dividido em camadas para melhor organização dos domínios de cada componente. As camadas do serviço que possuem maior significado para o trabalho são: Service, WorkflowExecutor e API.

Na camada de Service foram desenvolvidos e testados componentes para a criação e utilização de *workflow*, abstraindo a lógica para o uso do fluxo e tornando o serviço utilizável para *groupware*. A cobertura de instruções Java *bytecode* por testes foi de 1.483 de um total de 1.488, totalizando um *coverage* acima de 99%, e para ramificações possíveis (*branches*) para o fluxo de instruções foram cobertas 69 de 70, sendo indicado pela ferramenta um percentual arredondando de 98. A camada WorkflowExecutor foi responsável por centralizar a lógica de um *workflow* configurável. Nela ficaram os executores dos *steps* que podem ser criados, parametrizados, validados e executados. Os componentes presentes nesta camada possuíram a cobertura de testes para instruções Java *bytecode* de 403 de um total de 405, totalizando uma abrangência acima de 99%. A cobertura de *branches* foi de 28 de um total de 30 possíveis, atingindo um *coverage* arredondando de 93%. A API foi a camada responsável pela comunicação REST, garantindo a comunicabilidade entre o micro-serviço produzido e o mundo exterior. Na API foram criados testes *end-to-end* e de integração. Nesta camada foi possível criar testes que iniciaram-se na *request* HTTP e seguiram até a persistência no banco de dados. A ferramenta utilizada (Jacoco) para gerar as métricas de testes considerou para a cobertura de testes apenas as instruções *bytecode*: 111 instruções cobertas de um total de 114. O *coverage* gerado foi de 97%.

Foram criados *steps* para a validação deste trabalho. Ambos *steps* são completamente distintos e possuem funções diferentes. Um dos *steps*, chamado de "UnorderedCircleOfWritersStep", foi projetado para ser executor de um gênero de uma atividade cooperativa onde alunos podem editar um mesmo documento somente uma vez por round, sendo a quantidade de round parametrizados. Outro *step* criado e nomeado como "SendEmailStep", o mesmo também configurável, teve o objetivo de enviar e-mails para o criador da atividade e, também podendo ser estendida para os participantes envolvidos na atividade, possibilitando o envio de mensagens distintas em determinadas fases da atividade. As mensagens de e-mail foram configuradas no momento da criação do *workflow* e foram mapeadas pela fase da atividade.

A cobertura total da aplicação foi aproximadamente de 98% pelas instruções *bytecode* e de 97% para as ramificações possíveis. Foram realizados 118 testes, nenhum apresentou

falha. Com a cobertura de teste alta, a chance de falha nas instruções do serviço se manteve baixa. Dificulta-se um cenário onde pode-se encontrar em um *bug*, e conforme se aumenta a cobertura de testes, menor é a chance deste cenário de falha se tornar real.

## 6 CONCLUSÃO

O serviço concluído apresenta uma alta cobertura de teste, foi posto em prática a criação de *workflow* e cada componente foi verificado, não apenas manualmente, mas também pela ferramenta de testes Junit. Com o Mockito foi possível abdicar do uso de serviços externos, simulando como seria o comportamento da aplicação em uma *stack* de microsserviços real. Estes pontos trazem mais segurança para concluir que foi possível criar uma ferramenta on-line para atividades cooperativas, e ainda ser acrescida pela característica de flexibilidade nos fluxos de atividade, atingindo o objetivo deste trabalho.

A utilização do *Test Driven Development* (TDD) acabou por surpreender positivamente os resultados deste trabalho. Pois ao seguir esta metodologia que consiste em testes escritos antes das funções da aplicação, atingiu-se automaticamente uma cobertura de testes alta. Ao utilizar alguma função recém implementada, dificilmente foi necessário fazer algum ajuste e, mesmo quando necessário, foi algo muito simples. A necessidade do ajuste foi ocasionada pela falta de algum cenário não validado que gerou alguma exceção inesperada.

A partir deste trabalho, espera-se a continuação da implementação das dependências da aplicação para que ela possa ser utilizada como um serviço que possa auxiliar professores de diversos lugares a aproveitarem os benefícios de atividades cooperativas. Dependências como um serviço de edição de documento, usuário e também de gateway são essenciais para que o serviço proposto por este trabalho possa ser efetivo em uma *stack* de microsserviços. Também é sugerido a criação de mais *steps* para aumentar a combinação de *workflows* possíveis e, assim, enriquecer as possibilidades de atividades em *groupware*.

## REFERÊNCIAS

- AGILE SOFTWARE DEVELOPMENT. **Manifesto for Agile Software Development**. 2001. Disponível em: <https://agilemanifesto.org/>. Acesso em: 2 jan. 2022.
- ANTIL, L. R. *et al.* Cooperative Learning: Prevalence, Conceptualizations, and the Relation Between Research and Practice. **American Educational Research Journal**, v. 35, n. 3, 1998.
- ASSOCIAÇÃO BRASILEIRA DE EDUCAÇÃO A DISTÂNCIA (ABED). **CensoEAD.BR 2018**: relatório analítico da aprendizagem a distância no Brasil. Curitiba: InterSaberes, 2019. Disponível em: [http://abed.org.br/arquivos/CENSO\\_DIGITAL\\_EAD\\_2018\\_PORTUGUES.pdf](http://abed.org.br/arquivos/CENSO_DIGITAL_EAD_2018_PORTUGUES.pdf). Acesso em: 26 out. 2021.
- BREAKOUT EDU. Disponível em: <https://www.breakoutedu.com/>. Acesso em: 28 dez. 2021.
- CAMARGO, F.; DAROS, T. **A sala de aula inovadora**: estratégias pedagógicas para fomentar o aprendizado ativo. Porto Alegre: Penso, 2018.
- COGO, A. L. P. Cooperação versus colaboração: conceitos para o ensino de enfermagem em ambiente virtual. **Revista Brasileira de Enfermagem**, v. 59, n. 5, p. 680-683, out. 2006.
- FTD DIGITAL. Disponível em: <https://digital.ftd.com.br/>. Acesso em: 28 dez. 2021.
- JACOBS, G. M.; POWER, M. A.; INN, L. W. **The Teacher's Sourcebook for Cooperative Learning**: Practical Techniques, Basic Principles, and Frequently Asked Questions. New York: Simon & Schuster, 2016.
- Java Persistence Api (JPA). Disponível em: <https://docs.oracle.com/javaee/6/tutorial/doc/bnbpz.html>. Acesso em 26 jun. 2022
- JIGSAW CLASSROOM. Disponível em: <https://www.jigsaw.org/#overview>. Acesso em: 25 dez. 2021.
- JOHNSON, D. W.; JOHNSON, R. T.; HOLUBEC, E. J. **Circles of Learning**: Cooperation in the Classroom. New York: Association for Supervision and Curriculum Development, 1984.
- MACHADO, R. P. **Percepção sonora**: discutindo os limites e as possibilidades de interação e de interdependência positiva de pessoas com deficiência visual em sistemas Web síncronos. Tese (Doutorado em Informática na Educação) – Programa de Pós-Graduação em Informática na Educação, Centro Interdisciplinar de Novas Tecnologias na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2019.
- MARQUES, M. C. E. **A importância da aprendizagem cooperativa na qualidade das interações que se estabelecem entre as crianças**. Dissertação (Mestrado em Educação Pré-Escolar) – Escola Superior de Educação e Comunicação, Universidade do Algarve, Algarve, 2017.
- MARTIN, R. C. **Clean Code**: A Handbook of Agile Software Craftsmanship. Upper Saddle River: Pearson, 2008.

MICROSOFT. **Unit testing best practices with .NET Core and .NET Standard**. 2021. Disponível em: <https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices>. Acesso em: 27 dez. 2021

MOCKITO. Disponível em: <https://site.mockito.org/>. Acesso em 26 jun. 2022

ORION SERVICES. Disponível em: <https://orion-services.dev/>. Acesso em: 29 dez. 2021.

PEFFERS, K. *et al.* **The design science research process: a model for producing and presenting information systems research**. 2006. Disponível em: <https://arxiv.org/ftp/arxiv/papers/2006/2006.02763.pdf>. Acesso em: 13 maio 2022.

PESHKAM, A. 8 Cooperative Learning Practices to Enrich Your Online or Hybrid Classroom. **Harvard Business Publishing Education**, 2020. Disponível em: <https://hbsp.harvard.edu/inspiring-minds/cooperative-learning-practices>. Acesso em: 15 nov. 2021.

PIAGET, J. **Estudos Sociológicos**. Rio de Janeiro: Florense, 1973.

QUARKUS. Disponível em: <https://quarkus.io/>. Acesso em: 26 jun. 2022.

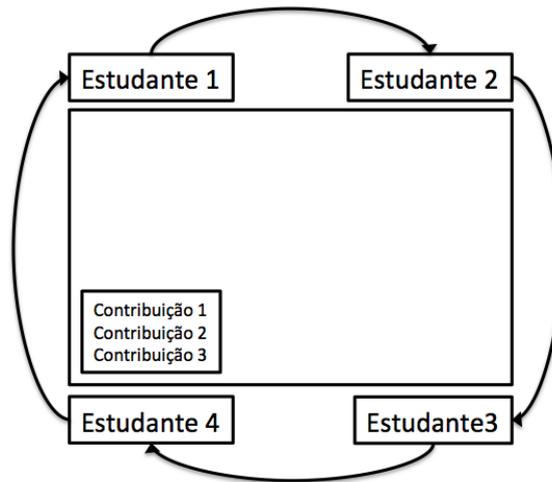
RABBITMQ. Disponível em: <https://www.rabbitmq.com/>. Acesso em 26 jun. 2022.

RAMOS, E. M. F. **Análise ergonômica do sistema hiperNet buscando o aprendizado da cooperação e da autonomia**. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1996.

RESTFUL API. Disponível em: <https://restfulapi.net/>. Acesso em 26 jun. 2022.

SLAVIN, R. E. Comprehensive approaches to cooperative learning. **Theory Into Practice**, v. 38, n. 2, p. 74-79, 1999.

**ANEXO 1 – CÍRCULO DE ESCRITORES: SEGUNDO PASSO DA ATIVIDADE E PRIMEIRA CONTRIBUIÇÃO ENTRE INTEGRANTES DO GRUPO**



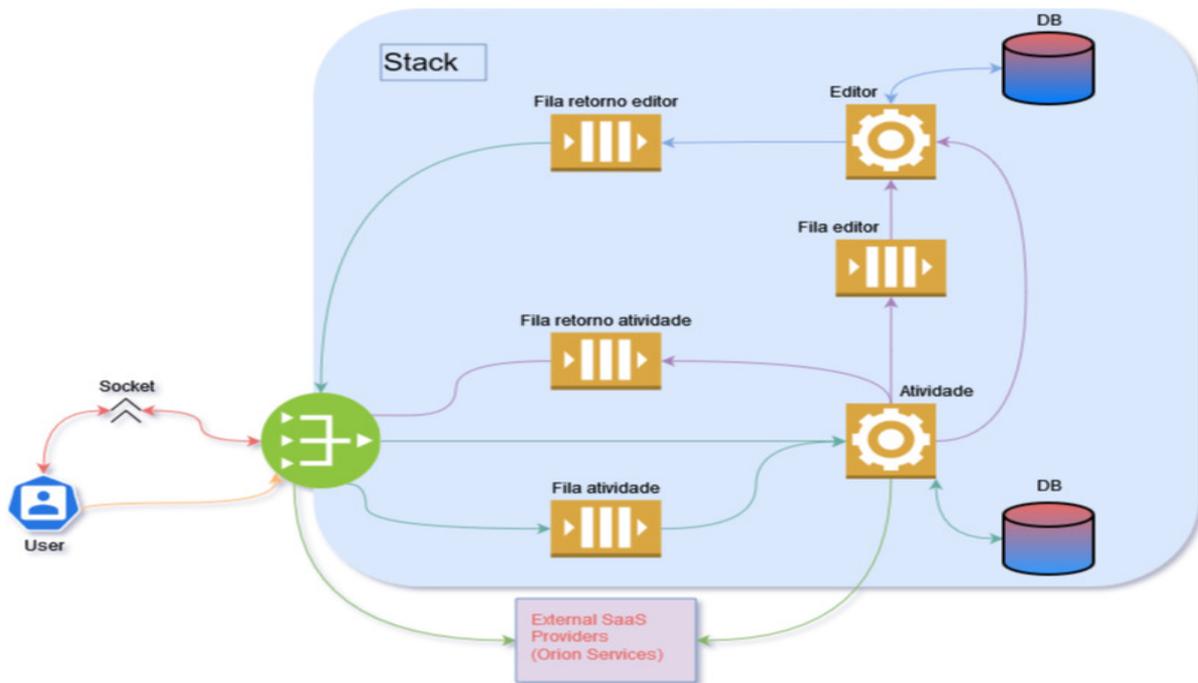
Fonte: Adaptada de Jacobs, Power e Inn (2016).

**APÊNDICE 1 – COMPARATIVO ENTRE OS SISTEMAS ANALISADOS E O SISTEMA PROPOSTO**

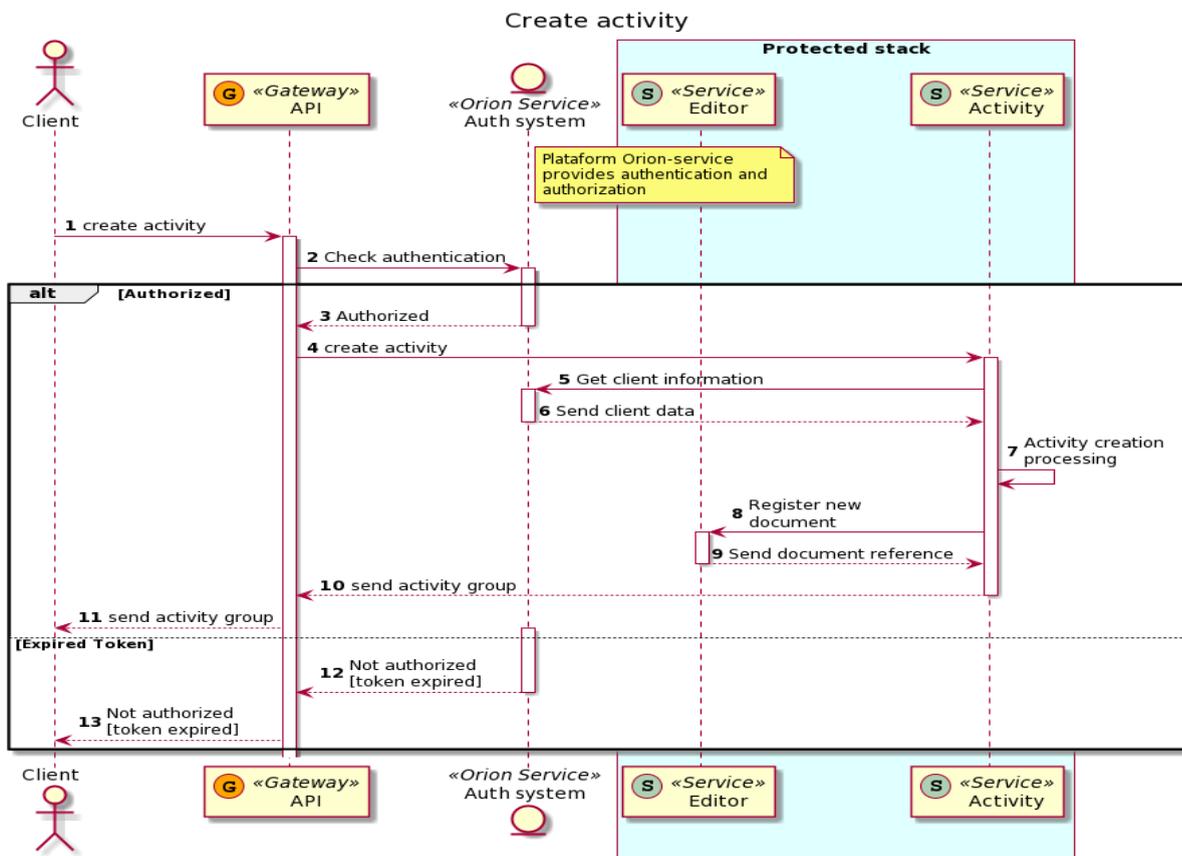
	<b>Breakout EDU</b>	<b>FTD Digital</b>	<b>Sistema proposto</b>
Jogos interativos	Sim	Sim	Não
Editor de Texto colaborativo	Não	Não	Sim
Atividades cooperativas	Sim	Não	Sim
Customização de fluxo de atividade	Sim	Não	Sim
Material de aprendizagem virtual (livros, artigos, vídeos, etc.)	Sim	Sim	Não
Ferramenta focada na aplicação de atividades virtuais	Não	Não	Sim
Atividade virtual - individual	Sim	Sim	Não
Atividade virtual - colaborativa online	Não	Não	Sim

## APÊNDICE 2 – DIAGRAMAS

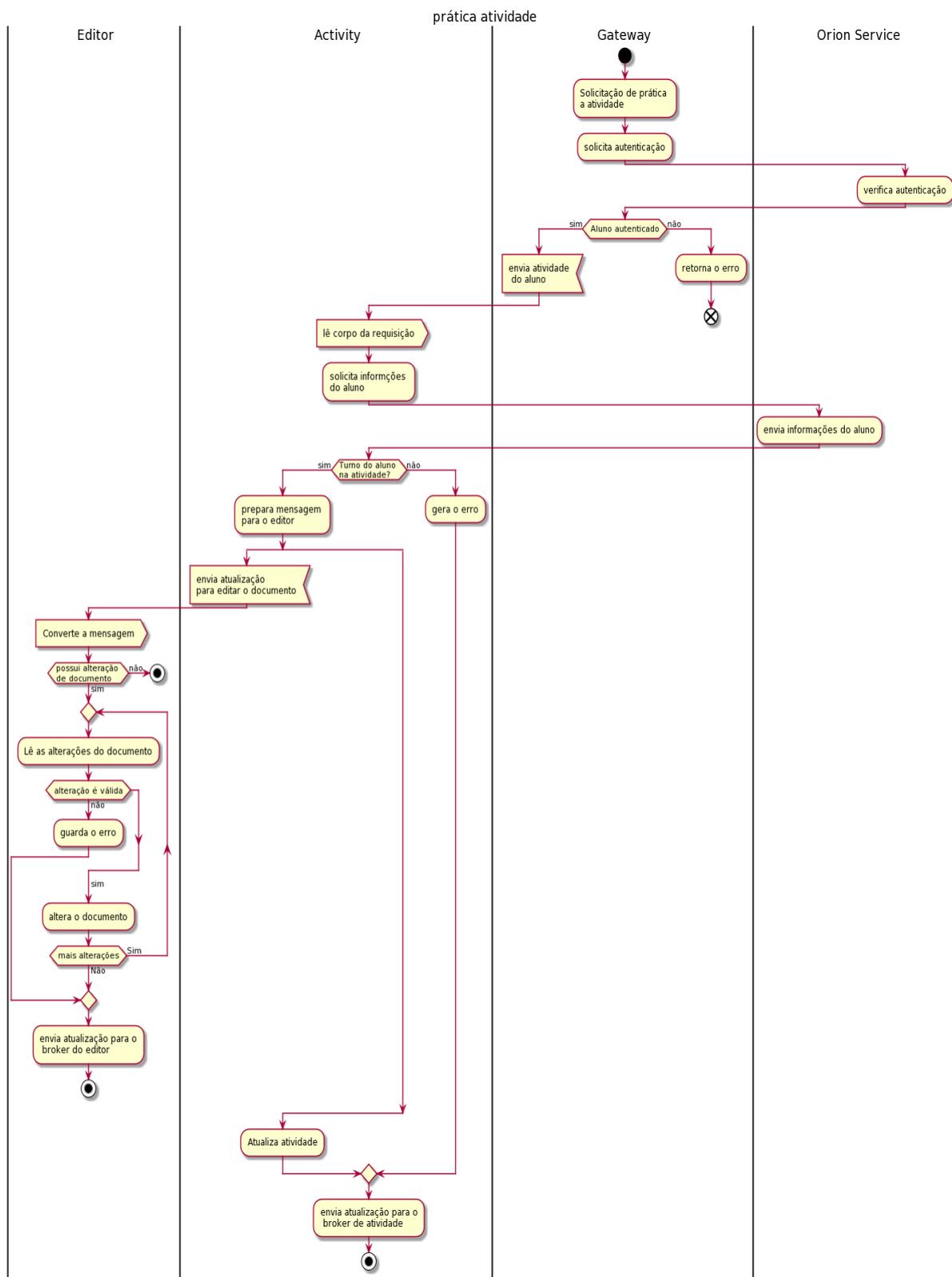
### A – DIAGRAMAS DA ARQUITETURA DA API



### B – DIAGRAMA DE SEQUÊNCIA SOBRE A CRIAÇÃO DA ATIVIDADE



### APÊNDICE 3 – DIAGRAMA DE ATIVIDADE SOBRE A PRÁTICA DA ATIVIDADE



## APÊNDICE 4 – CASOS DE TESTE

	Cenário	Resultado Esperado
<b>CRIAÇÃO DE ATIVIDADE</b>		
1	Usuário com <i>role</i> "creator" cria uma atividade.	O sistema faz uma requisição para a API de usuário e então valida o <i>role</i> do usuário e se está ativo, e então cria a atividade.
<b>INCLUSÃO DE ESTUDANTES NA ATIVIDADE</b>		
1	Usuário adiciona um outro usuário com o <i>role</i> de "participant" em uma atividade.	O sistema faz uma requisição para a API de usuário e então valida o <i>role</i> do usuário e se está ativo. O sistema adiciona o aluno a um grupo da atividade e retorna positivo.
2	Usuário adiciona um aluno de uma atividade em andamento em outra atividade.	O sistema verifica que o aluno já está em uma atividade, não permite que adicione o aluno e retorna negativo.
<b>CONTROLE DA ATIVIDADE CONFORME O FLUXO DE TRABALHO PROPOSTO</b>		
1	Aluno pratica a tarefa <b>conforme</b> o fluxo proposto para a atividade.	O serviço de Atividade executa todos os <i>steps</i> do workflow pertencentes ao estágio adequado, envia a modificação do documento para o serviço de Editor através do <i>broker</i> que possui a fila de edição de documento, e por fim envia a atualização da atividade para a fila que gere a fila de atualização de atividades.
2	Aluno pratica a tarefa de forma <b>divergente</b> do fluxo proposto para a atividade.	O serviço de Atividade executa todos os <i>steps</i> do workflow pertencentes ao estágio adequado, o <i>step</i> na fase de validação emite um erro e não permite a alteração no documento, gera um <i>log</i> de erro e envia a tentativa não permitida para o <i>broker</i> .
<b>EDIÇÃO DO DOCUMENTO</b>		
1	Serviço de atividade solicita alteração de documento para o serviço de edição <b>conforme</b> regras pré estipuladas no cadastro do documento.	O serviço Editor altera o documento da atividade e envia a atualização para o <i>broker</i> .
2	Serviço de atividade solicita alteração de documento para o serviço de edição de forma <b>divergente das</b> regras pré estipuladas no cadastro do documento.	O serviço de Editor não altera o documento e gera um log de erro e envia a tentativa não permitida para o <i>broker</i> .
3	Usuário responsável pela edição não encontrado.	O serviço de Editor gera um <i>log</i> informando a situação.
<b>CRIAÇÃO DE WORKFLOW</b>		
1	Responsável pelo sistema cria manualmente o <i>workflow</i> , inserindo os <i>steps</i> conforme o estágio da atividade desejado para a execução do mesmo.	O serviço de atividades faz a validação da configuração dos <i>steps</i> conforme o estágio escolhido para execução e parâmetros internos. Então salva o <i>workflow</i> e disponibiliza o mesmo para ser utilizado.
2	Responsável pelo sistema cria manualmente o <i>workflow</i> , inserindo os <i>steps</i> conforme o estágio da atividade desejado para a execução do mesmo. Porém configura de forma errônea os	O serviço de atividades faz a validação da configuração dos <i>steps</i> conforme o estágio escolhido para execução e parâmetros internos. Então ao <i>step</i> verificar que a sua

	parâmetros ou o estágio para execução do <i>step</i> .	configuração está incorreta, o serviço não permite o cadastro do <i>workflow</i>
<b>FINALIZAÇÃO DA ATIVIDADE</b>		
1	A atividade se encerra conforme o fluxo estabelecido.	Ao atingir o objetivo a atividade se encerra, informa o encerramento do documento ao serviço de Editor e envia a atualização do estado da atividade pelo <i>broker</i> .
2	O usuário tenta editar uma atividade encerrada.	Sistema retorna erro e gera o log informando a situação.
3	Aluno tenta editar documento de uma atividade encerrada.	Sistema retorna erro e gera o log informando a situação.