

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

**CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE
TECNOLOGIAS**

FACULDADE DE ENGENHARIA DE COMPUTAÇÃO

ANDRÉ MATHEUS NICOLETI LEMOS DA SILVA

**COLEARN – PLATAFORMA DE APRENDIZAGEM
COLABORATIVA ONLINE**

**CAMPINAS
2016**

ANDRÉ MATHEUS NICOLETI LEMOS DA SILVA

**COLEARN – PLATAFORMA DE APRENDIZAGEM
COLABORATIVA ONLINE**

Monografia de Trabalho de Conclusão de Curso apresentada como exigência da disciplina Projeto Final II, ministrada no Curso de Engenharia de Computação, do Centro de Ciências Exatas, Ambientais e de Tecnologias da Pontifícia Universidade Católica de Campinas.

Orientador: Prof. Dr. Carlos Miguel Tobar Toledo

Coorientador: Prof. Dr. Carlos Miguel Tobar Toledo

**PUC-CAMPINAS
2016**

**Pontifícia Universidade Católica de Campinas
Centro de Ciências Exatas, Ambientais e de Tecnologias
Faculdade de Engenharia de Computação**

**NICOLETI, André Matheus Lemos da Silva
CoLearn - Plataforma de aprendizagem colaborativa *online***

**Monografia de Trabalho de Conclusão de Curso
Graduação em Engenharia de Computação**

BANCA EXAMINADORA

Presidente e Orientador Prof. Dr. Carlos Miguel Tobar Toledo
1º Examinador Prof. Dr. Fernando Ernesto Kintschner

Campinas, 28 de novembro de 2016.

À minha mãe, por tudo o que me ensinou, pela atenção, amor, carinho, apoio e, principalmente, por ter tornado esta minha conquista possível.

Aos meus familiares, em especial aos que não estão mais presentes, que foram importantes ao longo dessa minha jornada.

À minha namorada, Larissa, por sempre estar ao meu lado em todos os momentos, pelo apoio, amor e atenção.

AGRADECIMENTOS

À toda a minha família e a minha namorada,
Por todo os ensinamentos e apoio, sem os quais não estaria onde estou.

Ao Pr. Dr. Carlos Miguel Tobar Toledo,
Pelo conhecimento, ajuda e presença durante a realização do Trabalho de Conclusão de Curso
(TCC).

Ao meu amigo Lincoln Moraes de Souza,
Pela participação importante ao longo do meu trabalho.

Ao Pr. Dr. Juan Manuel Adán Coello,
Pela disposição e ajuda indispensáveis para o TCC.

Aos meus amigos,
Por todo apoio e convívio ao longo deste longo ano.

A todos os professores da Faculdade de Engenharia de Computação,
Por proporcionar um excelente ambiente de estudo, que contribuíram para a minha formação
pessoal.

“O sucesso é ir de fracasso em fracasso sem perder o entusiasmo”

Winston Churchill
(1874-1965)

RESUMO

NICOLETI, André Matheus Lemos da Silva. CoLearn – Plataforma de aprendizagem colaborativa *online*. 2016. 85p. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Faculdade de Engenharia de Computação, Campinas, 2016.

Esta monografia relata um TCC em que foi desenvolvido um sistema de *software* denominado CoLearn, que tem como objetivo melhorar o conhecimento dos estudantes que o utilizam. Para cumprir com este objetivo, o autor escolheu implementar uma aplicação *Web* em que estudantes interagem através de perguntas e respostas, para aprenderem de forma colaborativa. O artefato também conta com recomendação personalizada de instrutores para auxiliar estudantes a encontrarem ajuda. O desenvolvimento ocorreu seguindo-se as atividades do método Scrum. Além disso, para que o desenvolvimento ocorresse da melhor forma possível, foi utilizado a IDE Visual Studio 2015, a linguagem de programação C#, o *framework* para desenvolvimento de aplicações *Web* ASP.NET MVC 5, um banco de dados SQL Server, bibliotecas para realizar comunicação em tempo real entre os usuários, e as linguagens HTML, CSS e JavaScript para a interface de usuário *Web*. Para validar o objetivo, foi planejada e executada uma avaliação com um professor universitário experiente, que respondeu a um questionário a fim de perceber se a solução desenvolvida é adequada para estudantes melhorarem seus conhecimentos. A análise dos resultados mostrou que o objetivo do TCC foi atingido.

Palavras-chave: Sistema de apoio à aprendizagem colaborativa. Recomendação personalizada de instrutores. Aplicação *Web*.

ABSTRACT

NICOLETI, André Matheus Lemos da Silva. 2016. 85p. *Capstone Project (Computer Engineering Undergraduate)* – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Faculdade de Engenharia de Computação, Campinas, 2016.

In this monograph it is described a Capstone Project (CP), in which was developed a software system named CoLearn that aims to improve the knowledge of the students that use it. To reach this goal the author has chosen to implement a Web Application to allow students to interact with each other, asking and answering questions, in a collaborative way. The system also provides personalized recommendation of instructors to assist students seeking for help. The development occurred following the activities of the Scrum method. Furthermore, for the best possible development happening, the Visual Studio IDE was used, alongside with the C# programming language, the Web Application development framework ASP.NET MVC 5, a SQL Server database, libraries to real-time communication, and the Web languages HTML, CSS and JavaScript for the user interface. To assess the objective, an evaluation was designed and applied to an experienced professor, who followed a script and answered a questionnaire to respond whether the developed solution is suitable to students improve their knowledge. The results analysis showed that the objective of the CP was achieved.

Descriptors: Collaborative learning support system. Personalized instructors recommendation. Web Application.

LISTA DE FIGURAS

Figura 1. Versão final do diagrama de arquitetura do sistema CoLearn.....	27
Figura 2. Tela para um usuário se cadastrar no site.....	29
Figura 3. Tela inicial de um estudante.....	30
Figura 4. Tela de gerenciamento de tags.....	31
Figura 5. Tela de cadastramento de novas instituições.....	32
Figura 6. Tela para moderar pedidos de cadastramento de instituições.....	33
Figura 7. Sugestão de pergunta similar à que está sendo criada.....	34
Figura 8. Tela para responder uma questão.....	36
Figura 9. Respostas dadas a uma pergunta, ordenadas por avaliação.....	37
Figura 10. Tela da videoconferência entre dois usuários.....	39
Figura 11. Tela de preferências de um estudante.....	44
Figura 12. Diagrama de atividades do algoritmo de cálculo de similaridade.....	47
Figura 13. Primeira versão do diagrama de arquitetura do sistema CoLearn.....	60
Figura 14. Burndown do primeiro semestre.....	61
Figura 15. Burndown do segundo semestre.....	62
Figura 16. Configuração para a geração de backup.....	63
Figura 17. Configuração do esquema de backup.....	63
Figura 18. Primeira etapa da configuração do TFS.....	64
Figura 19. Segunda etapa da configuração do TFS.....	64
Figura 20. Terceira etapa da configuração do TFS.....	65
Figura 21. Quarta etapa da configuração do TFS.....	65
Figura 22. Quinta etapa da configuração do TFS.....	66
Figura 23. Sexta etapa da configuração do TFS.....	66
Figura 24. Sétima etapa da configuração do TFS.....	67
Figura 25. Oitava etapa da configuração do TFS.....	67
Figura 26. Diagrama de casos de uso do sistema CoLearn.....	68
Figura 27. Agenda sendo visualizada no modo semana.....	69
Figura 28. Agenda sendo visualizada no modo mês.....	69
Figura 29. Tela para gerenciamento de níveis acadêmicos.....	70
Figura 30. Pesquisa feita por nome e avaliação mínima.....	71
Figura 31. Pesquisa feita por instituição, nível e avaliação mínima.....	71
Figura 32. Diagrama de atividades do algoritmo de recomendação de instrutores.....	72
Figura 33. Instrutores recomendados a um estudante.....	73

LISTA DE QUADROS

Quadro 1. Comparativo dos trabalhos relacionados	16
---	----

LISTA DE ABREVIATURAS E SIGLAS

ABNT	= Associação Brasileira de Normas Técnicas
API	= <i>Application Programming Interface</i>
AVA	= Ambiente Virtual de Aprendizagem
CP	= <i>Capstone Project</i>
IDE	= <i>Integrated Development Environment</i>
MEC	= Ministério da Educação
MVC	= <i>Model View Controller</i>
SMTP	= <i>Simple Mail Transfer Protocol</i>
TCC	= Trabalho de Conclusão de Curso
TCP	= <i>Transmission Control Protocol</i>
TFS	= Microsoft <i>Team Foundation Services</i>
TFVC	= Microsoft <i>Team Foundation Version Control</i>

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Caracterização do Problema e Objetivo	13
1.2 Proposta do Artefato.....	13
1.3 Trabalhos Relacionados	14
1.4 Organização da Monografia	16
2 DESENVOLVIMENTO.....	18
2.1 Método.....	18
2.1.1 Scrum	18
2.1.2 Gerência do Projeto.....	19
2.2 Ferramentas e Tecnologias	21
2.2.1 ASP.NET MVC 5	21
2.2.2 C#.....	21
2.2.3 Entity Framework.....	21
2.2.4 Microsoft Visual Studio IDE	22
2.2.5 WebRTC	22
2.2.6 JavaScript.....	22
2.2.7 RecordRTC.....	23
2.2.8 FFmpeg	23
2.2.9 Apache Lucene.NET.....	24
2.2.10 FullCalendar	24
2.2.11 WebSockets	24
2.2.12 HTML.....	25
2.2.13 CSS	25
2.2.14 JSON	25
2.2.15 Backup.....	25
2.2.16 Controle de Versões	26
2.3 Arquitetura do CoLearn	26
3 COLEARN	28
3.1 Gerenciamento de Usuários	28
3.1.1 Cadastramento e Edição de Estudantes	28
3.1.2 Cadastramento de Moderadores	29
3.1.3 Autorização e Autenticação de Estudantes e Moderadores	30
3.2 Gerenciamento de Tags	30
3.3 Gerenciamento de Instituições	32
3.3.1 Cadastramento de Instituições	32
3.3.2 Moderação de Instituições.....	33
3.4 Gerenciamento de Perguntas	33
3.4.1 Criação de Perguntas	33
3.4.2 Sugestão de Perguntas Similares.....	34
3.4.3 Filtragem de Perguntas	35
3.5 Sugestão de Materiais de Estudos	35
3.6 Criação e Avaliação de Respostas	36
3.6.1 Criação de Respostas em Texto	36
3.6.2 Avaliação das Respostas	37
3.7 Videoconferência.....	38

3.8 Gerenciamento de Agenda.....	40
3.8.1 Gerenciamento de Horários Disponíveis	40
3.8.2 Agendamento de Videoconferência.....	41
3.9 Gerenciamento de Níveis Acadêmicos	41
3.10 Pesquisa de Estudantes	42
3.11 Preferências para Recomendação	43
3.12 Recomendação Personalizada de Instrutores	44
4 AVALIAÇÃO E ANÁLISE DOS RESULTADOS	48
4.1 Avaliação	48
4.2 Análise dos Resultados	49
5 CONSIDERAÇÕES FINAIS	51
5.1 Grau de Dificuldade - Aspectos de Inovação e Aprimoramento	52
5.1.1 Aprimoramento e Inovação - Inteligência	52
5.1.2 Aprimoramento – Videoconferência.....	52
5.1.3 Aprimoramento - Programação Web	53
5.2 Análise da Qualidade e do Grau de Complexidade	53
5.2.1 Análise de Qualidade.....	53
5.2.2 Análise do Grau de Complexidade	55
5.3 Dificuldades Encontradas	56
5.4 Melhorias Futuras	56
REFERÊNCIAS.....	58
APÊNDICES	60
Apêndice A Diagrama de Arquitetura Elaborado no Primeiro Semestre	60
Apêndice B Burndowns do Desenvolvimento do Projeto.....	61
Apêndice C Configuração da Ferramenta de <i>Backup</i>	63
Apêndice D Configuração do Controlador de Versões	64
Apêndice E Diagrama de Casos de Uso	68
Apêndice F Visualizações da Agenda de um Usuário	69
Apêndice G Tela de Gerenciamento de Níveis Acadêmicos	70
Apêndice H Visualização dos Resultados de Pesquisas de Estudantes	71
Apêndice I Algoritmo de Recomendação de Instrutores.....	72
Apêndice J Resultado de uma Recomendação de Instrutores	73
Apêndice K Dados Inseridos no Sistema para a Avaliação	74
Apêndice L Roteiro Aplicado ao Avaliador do Sistema.....	83
Apêndice M Questionário Aplicado ao Avaliador do Sistema.....	85

1 INTRODUÇÃO

Em uma instituição de ensino superior, os estudantes buscam elucidar suas dúvidas e ampliar seu conhecimento de muitas formas, seja através das aulas, com seus professores, da consulta aos livros, da pesquisa na *Web* e na interação com outros alunos.

Quando os estudantes procuram melhorar seus conhecimentos, visando aumentarem suas notas e/ou se aprofundarem em temas de seus interesses, podem buscar ajuda através da interação com outros alunos que se façam disponíveis. Alguns desses alunos são denominados de instrutores e são encontrados através de mídias sociais, anúncios nos murais da instituição ou por indicação de outros estudantes.

Após a localização dos instrutores, os estudantes podem marcar encontros e trocar conhecimentos, de forma colaborativa.

1.1 Caracterização do Problema e Objetivo

Os conhecimentos de um estudante não são melhorados quando não se encontram instrutores adequados para ajudá-lo a entender um assunto de seu interesse, fazendo com que suas notas não sejam aumentadas ou que este não consiga se aprofundar em um tema de seu interesse.

Sendo assim, o objetivo do Trabalho de Conclusão de Curso (TCC) é melhorar os conhecimentos dos estudantes, no contexto descrito anteriormente.

1.2 Proposta do Artefato

A proposta de artefato inicialmente pensada considerou um sistema, denominado CoLearn, para que estudantes possam fazer e responder perguntas, melhorando seus conhecimentos com outros estudantes de forma colaborativa e totalmente *online*. Para tanto, foi desenvolvida uma aplicação *Web*, que é descrita em detalhes no Capítulo 3.

Todos os módulos do sistema estão descritos em um diagrama de arquitetura desenvolvido ao longo do primeiro semestre, que pode ser visualizado no Apêndice A.

No primeiro semestre, os módulos inicialmente pensados foram: Gerenciar Instituições, Gerenciar Tags, Filtrar Perguntas, Gerenciar Perfil, Gerenciar Perguntas, Gerenciar Respostas, Avaliar Respostas, Gerenciar Agenda, Gerenciar Videoconferências, Sugerir Materiais de Estudo, Sugerir Perguntas Similares, Enviar *Email*, Gerenciar Preferências e Recomendar Instrutores.

Ao longo do desenvolvimento do artefato, observou-se a necessidade de alteração e inclusão de um novo módulo para tornar o sistema mais adequado aos seus usuários. No Capítulo 2 é descrito estas alterações.

Os requisitos do artefato desenvolvido foram definidos pelo cliente do TCC. Este é um profissional da área de TI que atua como *Scrum Master* em um dos projetos de uma empresa de tecnologia, localizada na cidade de Campinas.

Para auxiliar na parte técnica e na tomada de decisão, um professor doutor da faculdade de Engenharia de Computação da Pontifícia Universidade Católica de Campinas (PUC-Campinas) atuou como orientador e coorientador, já que este tem conhecimento em assuntos que foram utilizados no decorrer do desenvolvimento e do gerenciamento do TCC.

1.3 Trabalhos Relacionados

O artefato desenvolvido foi comparado a quatro sistemas descritos em publicações sobre apoio à aprendizagem colaborativa, com pelo menos um atributo similar às características dele. Estes trabalhos são:

- A publicação 1, escrita por Arai e Handayani (2012), apresenta um sistema de apoio à aprendizagem colaborativa para responder às perguntas feitas por estudantes. Este sistema permite que estudantes enviem suas perguntas que serão discutidas e respondidas por outros aprendizes, de forma colaborativa. As questões também são respondidas de forma automática pelo

sistema, que gera uma resposta a partir da base de dados da Wikipédia. Caso o sistema consiga gerar uma resposta, esta é guardada na base de dados do artefato e enviada para os professores avaliarem-na. Se esta não for adequada ou se a geração falhar, os professores respondem à pergunta manualmente, quando estes estiverem *online*, através da própria aplicação.

- A publicação de escrita por Zhang e Hazeyama (2012), denominada publicação 2, apresenta um sistema em que estudantes de engenharia de *software* podem trocar conhecimentos, segundo as teorias da Aprendizagem Cognitiva e da Cognição Distribuída. Os estudantes interagem entre si através de perguntas e respostas ou por meio de comentários feitos às notas de aula, de forma colaborativa, para que os estudantes consigam expressar suas ideias sobre determinados assuntos.
- A publicação 3, de Sasaki et. al (2015), descreve a implementação de uma aplicação para examinar como a comunicação entre dispositivos *tablets*, dentro de uma sala de aula, auxilia na aprendizagem colaborativa dos alunos. A aplicação permite que estudantes troquem informações através de fotos e imagens de mensagens escritas a mão (*handwriting*), dentro e fora da sala de aula.
- A publicação 4 é uma dissertação de mestrado, escrita por Frota (2012), que apresenta o desenvolvimento de um sistema agente para a plataforma de Ambiente Virtual de Aprendizagem (AVA) Moodle. O sistema cria modelos dos alunos utilizando ontologias na plataforma Moodle, a fim de se obter as competências e habilidades destes, bem como as atividades propostas por professores com as quais os estudantes mais tiveram dificuldades. Considerando isto, o sistema recomenda alunos que possuam elevados níveis de habilidades e competências para sanar dúvidas de seus colegas no mesmo curso ou disciplina.

No Quadro 1 é possível observar a comparação entre o sistema CoLearn e os sistemas descritos nas publicações apresentadas anteriormente.

Quadro 1. Comparativo dos trabalhos relacionados.

	Sistema 1	Sistema 2	Sistema 3	Sistema 4	CoLearn
Modo de interação entre os estudantes	Perguntas e respostas	Perguntas e respostas; respostas automáticas são geradas a partir da base de dados da Wikipédia	Troca de conteúdo como imagens e fotos, que são comentados entre os estudantes	Alunos se ajudam na resolução de tarefas propostas por professores	Perguntas e respostas através de textos ou de videoconferências
Formato dos objetos trocados entre os estudantes	Texto e vídeo	Texto	Texto e imagens	Texto, vídeo e imagem	Texto e vídeo (gravação da videoconferência)
Forma de recomendação de ajuda	Nenhuma	Nenhuma	Nenhuma	Uso de ontologias em um AVA para coletar informações dos estudantes (níveis de habilidades e competências, assuntos que estes têm dúvidas)	Instrutores são recomendados para estudantes através de recomendação personalizada baseada nas preferências do estudante, utilizando um algoritmo de similaridade entre perfis (<i>matchmaking</i>)

1.4 Organização da Monografia

Esta monografia possui seu conteúdo distribuído em cinco capítulos:

- No segundo encontra-se a apresentação do método de desenvolvimento escolhido e a gerência do projeto utilizando-o. Também são apresentadas as tecnologias e ferramentas utilizadas no desenvolvimento do sistema, bem como a sua arquitetura.
- O terceiro capítulo apresenta o sistema desenvolvido durante o TCC, com a especificação de todas as suas funcionalidades.

- A avaliação e a análise dos resultados estão presentes no quarto capítulo, no qual é possível verificar se o objetivo do TCC foi atingido.
- O quinto capítulo contém as considerações finais do trabalho. Contém também a análise da qualidade e do grau de complexidade presentes no artefato desenvolvido, os graus de dificuldade encontrados e possíveis melhorias futuras.

2 DESENVOLVIMENTO

Neste capítulo é descrito o método de desenvolvimento escolhido para o planejamento e o gerenciamento do trabalho realizado para projetar o CoLearn, as tecnologias e ferramentas utilizadas no decorrer do desenvolvimento deste e a sua arquitetura.

2.1 Método

O método de desenvolvimento escolhido foi o *Scrum* (RUBIN, 2012). Suas atividades e os relatos do que aconteceu durante a gerência do trabalho são descritos a seguir.

2.1.1 Scrum

Segundo Rubin, o *Scrum* permite a priorização do *backlog* – lista de funcionalidades e outros itens necessários para o desenvolvimento de um produto -, permitindo que as partes mais prioritárias do sistema sejam entregues primeiro, além de contar com encontros entre a equipe técnica e o cliente, para que este último acompanhe o desenvolvimento do *software*. Reuniões periódicas entre a equipe para definir o que será desenvolvido durante um intervalo fixo de tempo, chamado *sprint*, também foram um diferencial que pesou na escolha deste método.

O método *Scrum*, adaptado ao contexto do TCC, apresenta as seguintes atividades:

- Definição do *product backlog*, que é a definição do que deve ser feito. Esta atividade é realizada junto ao cliente e tem como resultado um conjunto de histórias do usuário.
- *Sprint Planning*, que é realizada no primeiro dia da *sprint* para se definir o que deve ser feito durante este novo ciclo, junto com o cliente. Ou seja, nesta atividade é definido o *sprint backlog*.

- *Sprint Execution*, que é a execução das tarefas planejadas (*sprint backlog*) para a *sprint* atual.
- *Sprint Review Demonstration (demo)*, que é o encontro realizado ao final de uma *sprint*, para que o incremento de *software* feito possa ser demonstrado e avaliado pelo cliente.

Inicialmente, foram definidos que seriam necessárias oito *sprints* para o desenvolvimento do sistema.

2.1.2 Gerência do Projeto

A gerência do projeto iniciou-se com a definição do *product backlog*. Todas as histórias foram especificadas e o tempo máximo de um *sprint* foi estipulado de acordo com a história de maior duração. Após isso, foi possível definir o número necessário de *sprints*, com quatorze dias de duração cada.

Antes do início de cada *sprint* era realizada uma reunião para a definição do *sprint backlog*. O *product backlog* era analisado e, de acordo com as prioridades de cada história e sua duração, era gerada uma lista de atividades a serem desenvolvidas no *sprint* corrente.

Durante a execução das *sprints*, surgiram necessidades de reuniões com o cliente para sanar dúvidas sobre o que estava sendo implementado. As principais dúvidas que surgiram ao longo do desenvolvimento estavam relacionadas à interface do sistema.

Após o término dos *sprints* aconteciam as *demos*. As histórias completadas eram apresentadas ao cliente que, em alguns momentos, dava sugestões de melhorias para que o *software* atendesse às suas expectativas.

Algumas *demos* não foram realizadas ao final de algumas *sprints*, devido à indisponibilidade do cliente ou a não conclusão das tarefas do círculo em questão. As reuniões com o cliente foram remarcadas e as histórias não completadas retornaram ao *product backlog*, para que fosse avaliada a necessidade das mesmas fazerem parte do próximo *sprint*.

Os principais atrasos se deram no desenvolvimento do módulo da videoconferência, principalmente pelo fato de que o autor não conhecia a API WebRTC, responsável pela comunicação entre os participantes da videoconferência. Também houve problemas para salvar a videoconferência no servidor, uma vez que o autor levou alguns dias para definir como essa funcionalidade seria implementada, pois a ideia inicial não apresentou resultados satisfatórios com relação à performance. Foi necessário um *sprint* a mais do que o planejado para a conclusão desse módulo.

O módulo de recomendação de estudantes também foi responsável por atrasos. Este módulo foi desenvolvido em dois *sprints* e não em um, como foi inicialmente planejado, devido ao atraso na definição do algoritmo a ser utilizado pelo autor.

Algumas alterações no sistema foram solicitadas pelo orientador do TCC, a fim de melhorar a qualidade do artefato. Para realizar estas alterações não planejadas, um *sprint* adicional foi necessário e as melhorias foram desenvolvidas nele.

Além destes atrasos, o autor também sentiu muita dificuldade com a programação Web, principalmente na parte do *layout* do site. Criar uma interface que funcionasse em aparelhos com telas de diferentes tamanhos foi um grande desafio enfrentado durante o desenvolvimento do artefato. Porém, esta dificuldade não gerou atrasos no cronograma, mas sim atrasos de algumas atividades que não terminaram dentro do *sprint*.

Como encaminhamento para os problemas encontrados, foi necessário a criação de *sprints* adicionais, além da necessidade de aumentar o número de horas dedicadas ao desenvolvimento do artefato. Mesmo fazendo isto, o autor não foi capaz de terminar todas as atividades no prazo inicialmente definido. Foram necessárias seis semanas a mais (três *sprints*).

Os *burndowns* para cada semestre podem ser vistos no Apêndice B. Cada *burndown* mostra as atividades realizadas durante um *sprint* específico. É possível visualizar as horas estimadas e a quantidade de horas trabalhadas em cada atividade.

2.2 Ferramentas e Tecnologias

Nas subseções seguintes são descritas as ferramentas e tecnologias utilizadas durante o desenvolvimento do CoLearn.

2.2.1 ASP.NET MVC 5

ASP.NET MVC 5 (ASP.NET, 2016) é um *framework* para a criação de *websites*. Este separa uma aplicação em três principais camadas: a camada modelo, a camada visão e a camada controladora. Com esse *framework* foi possível implementar parte do código do cliente e toda a parte *Web* do servidor.

Este *framework* foi escolhido por ser muito completo para desenvolvimento *Web*, além de aumentar a produtividade do programador em pouco tempo.

2.2.2 C#

C# (MICROSOFT DEVELOPER NETWORK, 2016a) é uma linguagem de programação orientada a objetos de propósito geral, que roda no .NET Framework, um componente que inclui um sistema de execução virtual, denominado *Common Language Runtime (CLR)*, e um conjunto unificado de bibliotecas de classes.

Essa linguagem foi utilizada para a codificação do CoLearn, pelo fato de ser a linguagem de programação compatível com o ASP.NET MVC 5 que o autor tem mais conhecimento, além de ser muito produtiva, devido às inúmeras funções utilitárias que já estão implementadas e prontas para uso.

2.2.3 Entity Framework

O Entity Framework (SANTOS, 2016) é uma ferramenta de mapeamento objeto relacional (ORM – *Object Relational Management*), que mapeia as classes utilizadas na aplicação do usuário em tabelas relacionais em um banco de dados.

O autor escolheu este *framework* por ele ser fácil de ser configurado e utilizado, e também por ser o mais utilizado no desenvolvimento de aplicações ASP.NET MVC 5, o que faz com que ele tenha muitos exemplos e tutoriais disponíveis para consulta.

2.2.4 Microsoft Visual Studio IDE

Microsoft Visual Studio IDE (MICROSOFT DEVELOPER NETWORK, 2016b) é um conjunto de ferramentas para a criação de *software*, especialmente para aqueles que utilizam o .NET *Framework*. Esta IDE suporta, por padrão, as linguagens de programação C#, C e C++, JavaScript, F # e Visual Basic, além de oferecer suporte à criação de *websites* em ASP.NET MVC 5.

O autor escolheu esta IDE por ela apresentar uma versão gratuita e por ser a ferramenta para desenvolvimento de *software* que o autor tem mais conhecimento.

2.2.5 WebRTC

WebRTC (WEBRTC, 2016) é um projeto aberto que fornece comunicação em tempo real para aplicações *Web* e móveis através de APIs simples, que permitem implementar soluções de comunicação em tempo real, como videoconferências, através de um conjunto de protocolos comuns.

Essa tecnologia foi a escolhida para a implementação da funcionalidade de videoconferência entre dois estudantes, por ser a tecnologia que os principais navegadores *Web* modernos mais oferecem suporte.

2.2.6 JavaScript

JavaScript (MOZILLA DEVELOPER NETWORK, 2016a) é uma linguagem de programação leve e interpretada, conhecida por ser a linguagem de *scripts* para páginas *Web*.

Essa linguagem foi utilizada para realizar a comunicação entre as páginas *Web* e o servidor. Com ela foi possível se comunicar com o servidor de forma assíncrona, realizar validações e implementar a videoconferência, utilizando a API *WebRTC*.

2.2.7 RecordRTC

RecordRTC (RECORDRTC, 2016) é uma biblioteca para gravação de mídias, escrita em JavaScript. Com ela é possível gravar os diferentes tipos de mídias que são suportadas pela tecnologia *WebRTC*.

Tal biblioteca foi utilizada para que fosse possível a captura dos áudios de ambos os participantes, assim como da tela do instrutor, incluindo os vídeos de suas câmeras, em tempo real. A escolha dessa biblioteca se justifica devido ao fato dela ser a única que o autor encontrou para gravação de uma videoconferência *WebRTC*.

2.2.8 FFmpeg

FFmpeg (FFMPEG, 2016) é um *framework* multimídia capaz de trabalhar com diversos formatos de áudio e vídeo, sendo altamente portável, podendo ser compilado e executado em uma grande variedade de ambientes e configurações.

Esta ferramenta foi utilizada para manipular os dados de áudio e vídeo enviados ao servidor durante uma videoconferência, pois estes são recebidos de forma separada e precisam ser unidos para se ter um único arquivo com toda a gravação da videoconferência.

O autor escolheu este *framework* devido à facilidade de uso e ao grande número de exemplos na *Web* sobre como juntar *frames* de áudio e vídeo em um único arquivo.

2.2.9 Apache Lucene.NET

Apache Lucene.NET (APACHE, 2016) é uma biblioteca de busca de texto escrita em C#. Com esta biblioteca foi possível realizar buscas por perguntas existentes na base de dados do sistema, através de palavras-chave.

Foi escolhida esta biblioteca para implementar a funcionalidade de mostrar perguntas existentes com título similar ao que um estudante está criando, além de ser fácil de utilizar e retornar resultados confiáveis e rápidos.

2.2.10 FullCalendar

O FullCalendar (FULLCALENDAR, 2016) é um *plugin* JavaScript *open source*, utilizado para gerenciamento de eventos, na forma de um calendário, que permite altos níveis de customização.

Esta ferramenta foi utilizada para implementar a interface de gerenciamento de horários disponíveis dos usuários. Como o autor não conhecia *plugins* deste tipo, várias outras ferramentas foram pesquisadas, mas esta foi a que apesentou a melhor documentação e a maior facilidade de uso.

2.2.11 WebSockets

WebSockets (MOZILLA DEVELOPER NETWORK, 2016b) permite a criação de um canal de comunicação TCP entre um servidor e o navegador *Web* do usuário. Com WebSockets é possível que um servidor envie notificações para um cliente sem a necessidade de realizar *pool* no servidor para obter uma resposta.

Esta tecnologia foi utilizada para enviar notificações aos clientes. Estas notificações enviadas são: nova pergunta, nova resposta, nova *tag* para moderação e nova instituição para moderação. Além disto, esta tecnologia foi utilizada para o controle da videoconferência, uma vez que era necessário que o servidor se comunicasse com os clientes quando uma nova videoconferência está se iniciando ou terminando, a fim de sincronizá-los.

2.2.12 HTML

Hyper Text Markup Language (W3C, 2016), ou HTML como é mais conhecida, é uma linguagem de marcação padrão usada para a criação de páginas Web.

Como o artefato é um website que implementa o padrão de desenvolvimento MVC, a linguagem HTML foi utilizada na camada view, para a criação das páginas Web.

2.2.13 CSS

Cascading Style Sheets (W3SCHOOLS, 2016), ou CSS, é uma linguagem que descreve como os elementos HTML serão exibidos na tela.

Essa linguagem foi utilizada na personalização do layout da tela, fazendo com que todo o site tenha o mesmo padrão de estilo em todas as suas páginas.

2.2.14 JSON

JavaScript Object Notation (JSON, 2016), ou JSON, é um formato leve para troca de dados de fácil leitura e escrita para humanos e máquinas, além de ser totalmente independente de linguagem de programação.

O JSON foi utilizado para a troca de informações entre o cliente e o servidor, e foi escolhido por ser um formato de fácil manipulação tanto pelo JavaScript (cliente) como pelo ASP.NET MVC 5 (servidor).

2.2.15 Backup

Para a realização periódica de backups completos e parciais, foi utilizada a ferramenta EaseUS *Todo Backup Free* 9.1. As configurações da ferramenta são apresentadas no Apêndice C (Figuras 14 e 15).

2.2.16 Controle de Versões

Para o controle de versões do código fonte e dos documentos do TCC foi utilizada a ferramenta Microsoft *Team Foundation Version Control* (TFVC) (MICROSOFT DEVELOPER NETWORK, 2016c), que é um controlador de versões integrado ao *Team Foundation Services* (TFS). Uma nova versão de um arquivo é criada através de um *check-in*, realizado através do Visual Studio ou do *site* do TFS.

Cada *check-in* resulta em um *changeset*, que é um conjunto de mudanças no código fonte. Cada *changeset* contém uma nova versão de um ou mais arquivos. Cada uma dessas novas versões é salva no servidor do TFVC, e as mesmas podem ser recuperadas a qualquer momento.

Os passos necessários para configurar o controlador de versões TFS são mostradas no Apêndice D (Figuras 16 a 23).

2.3 Arquitetura do CoLearn

A arquitetura do sistema CoLearn, em relação à versão do primeiro semestre (apresentada na Seção 1.2), sofreu algumas mudanças no decorrer do segundo semestre. Essas mudanças foram:

- Inclusão do módulo “Gerenciar Níveis Acadêmicos”, para que os estudantes possam visualizar os níveis acadêmicos dos instrutores recomendados para eles, além de poder usar a informação de nível como uma preferência para a recomendação.
- Modificação no módulo “Gerenciar Perguntas”, para melhorar o modo como as perguntas são listadas para um estudante, na página inicial.
- Inclusão da API RecordRTC no módulo responsável pelo controle da videoconferência, para que fosse possível a gravação da mesma entre dois usuários do sistema, com um desempenho satisfatório.
- Modificação do módulo “Avaliar Resposta”, para melhorar a confiança da avaliação das respostas, como é explicado na Subseção 3.6.2.

Na versão final do diagrama, que pode visualizada na Figura 1, é possível visualizar os módulos desenvolvidos pelo autor, as APIs utilizadas, a separação dos módulos segundo a arquitetura MVC e a separação da aplicação cliente e da aplicação servidor. Estas se comunicam através do protocolo HTTP/1.1 (RFC 2616), na forma de requisição e resposta. Também é mostrada no diagrama a base de dados utilizada e o servidor de *email*s que o artefato se comunica, através do protocolo SMTP, para enviar mensagens.

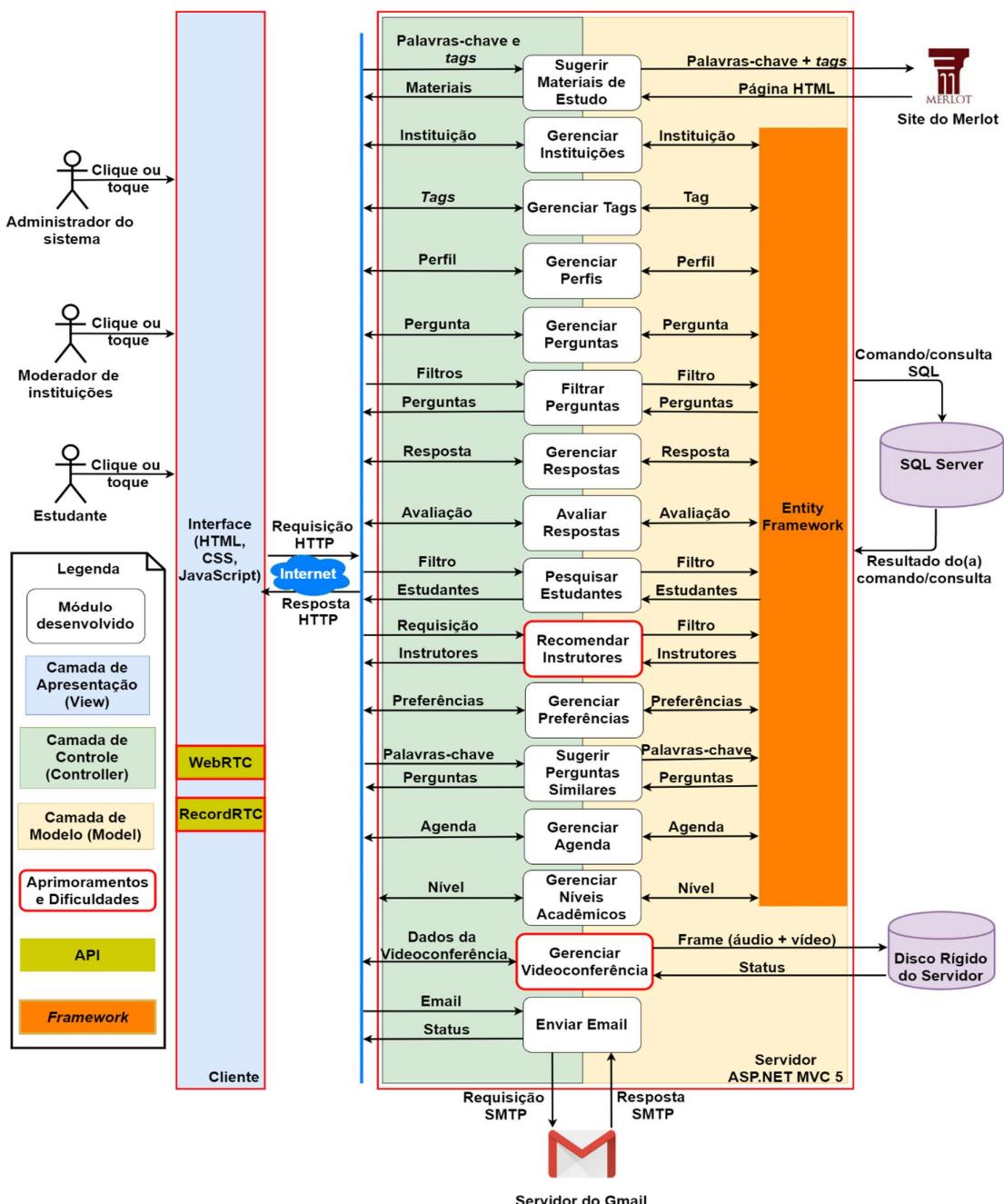


Figura 1. Versão final do diagrama de arquitetura do sistem CoLearn.

3 COLEARN

Neste capítulo é apresentado o artefato desenvolvido durante o TCC. Ele possui funcionalidades que são acessíveis via interação do usuário, através de páginas/telas desenvolvidas em HTML, CSS e JavaScript, enquanto outras foram implementadas do lado do servidor, sendo acessíveis apenas através de requisições HTTP, que são feitas pelos navegadores *Web* dos usuários.

O sistema oferece funcionalidades específicas para três tipos de usuários: o administrador do *site*, moderadores de *tags* e estudantes. No Apêndice E é apresentado um diagrama de casos de uso que mostra as funcionalidades para cada tipo de usuário.

3.1 Gerenciamento de Usuários

O sistema CoLearn conta com um módulo para o gerenciamento de usuários, que é responsável por manter os perfis destes e autenticar suas permissões nos outros módulos do sistema. O cadastramento de usuários é apresentado com detalhes na Subseção 3.1.1 e 3.1.2, sendo que a primeira também descreve a funcionalidade de edição de estudantes. O modo como o sistema autoriza e autentica os usuários é mostrado na Subseção 3.1.3.

3.1.1 Cadastramento e Edição de Estudantes

Os estudantes se cadastram no *site* informando seu nome, um endereço de *email*, seu grau acadêmico atual e uma senha, como pode ser visto na Figura 2.

Após completar o registro, o sistema informa ao estudante que ele deve definir a sua instituição e o redireciona para a página de seleção de uma instituição. Caso a entidade que o estudante procura não esteja na lista, ele pode solicitar para que esta seja incluída no *site*. Neste caso, a solicitação poderá ser aceita ou não pelo administrador do sistema, e o autor da solicitação será notificado via *email*.

Todas as informações usadas no cadastro podem ser alteradas. Além disso, um estudante também pode informar quais disciplinas ele está apto a ensinar

para outros estudantes, através de *tags*. Se alguma *tag* for informada, o estudante passará a ser também um instrutor, o que permite a ele responder perguntas e avaliar respostas de outros instrutores. Na Seção 3.2 é apresentado como as *tags* podem ser escolhidas por um estudante.

The screenshot shows a registration form for the CoLearn platform. The title 'Crie uma nova conta' is at the top. Below it, a note says '*Campos obrigatórios'. There are six input fields: 'Nome' (Name), 'Sobrenome' (Last Name), 'E-mail' (Email), 'SELECIONE SUA FORMAÇÃO ATUAL' (Select your current education level), 'Senha' (Password), and 'Confirme a senha' (Confirm password). A blue 'CRIAR CONTA' (Create Account) button is at the bottom. A link 'Já tem uma conta?' (Already have an account?) is at the very bottom.

Figura 2. Tela para um usuário se cadastrar no site.

3.1.2 Cadastramento de Moderadores

Os moderadores são cadastrados e associados a uma ou mais instituições pelo administrador do site. Os moderadores têm a responsabilidade de aceitar ou rejeitar novas disciplinas, em formato de *tags*, criadas por outros usuários, assim como cadastrar, editar, excluir e juntar *tags*. Estas funcionalidades são descritas com detalhes na Seção 3.2.

Eles também podem ser estudantes que foram promovidos pelo administrador do sistema à função de moderador, não precisando ser cadastrados por este. Quando isto acontece, um email é enviado ao usuário informando que ele se tornou um moderador de uma instituição. Percebe-se então que estes usuários também podem criar perguntas e associar *tags* aos seus perfis para se tornarem instrutores.

3.1.3 Autorização e Autenticação de Estudantes e Moderadores

No sistema CoLearn, cada tipo de usuário tem permissões diferentes, dependendo da sua função. Um usuário com função de moderador tem as mesmas permissões que um usuário com função de estudante, com a diferença de que o primeiro pode acessar o módulo de moderação de *tags* das instituições atribuídas a ele.

Quando um usuário realiza seu cadastramento no *site*, seu perfil é criado com permissões de moderador ou de estudante. Estas permissões são verificadas para cada página navegada pelo usuário. Caso o sistema verifique que o usuário não tem permissão para acessar uma página específica, todo o conteúdo desta é substituído por uma mensagem de erro.

O módulo de gerenciamento de estudantes também é responsável por validar as credenciais inseridas por um usuário na tela de *login*. Após uma autenticação bem-sucedida, o sistema verifica qual é a função do usuário, e redireciona-o para páginas específicas de usuários daquela função. A Figura 3 mostra a página de perguntas relacionadas ao perfil de um estudante/moderador.

The screenshot shows the CoLearn platform interface. At the top, there's a blue header bar with the CoLearn logo and a user profile for 'André'. Below the header, there's a search bar labeled 'FILTRAR PERGUNTAS' (Filter Questions). The main content area displays a grid of questions from different users:

- Sara Nicoletti** (PUC-Campinas, 10/20/2016): "Transmissões simplex podem suportar tráfego half-duplex ou full-duplex?" Tags: computação, redes de computadores. 2 responses.
- Pedro Lemos** (PUC-Campinas, 10/12/2016): "O que é transparência de distribuição?" Tags: computação, sistemas distribuídos. 2 responses.
- Pedro Lemos** (PUC-Campinas, 10/12/2016): "Qual o papel de um middleware em um sistema distribuído?" Tags: computação, sistemas distribuídos. 2 responses.
- Sophia Monteiro** (PUC-Campinas, 10/12/2016): "Por que usar APIs ao invés de chamadas de sistemas?" Tags: computação, sistemas operacionais. 2 responses.
- Sophia Monteiro** (PUC-Campinas, 10/12/2016): "No contexto dos Sistemas Operacionais, qual o propósito de se ter chamadas de sistema?" Tags: computação, sistemas operacionais. 2 responses.
- Larissa Ferreira** (PUC-Campinas, 6/13/2016): "Qual é a definição de Aprendizado de Máquina (Machine Learning)??" Tags: computação, Inteligência artificial. 3 responses.
- André Nicoletti** (PUC-Campinas, 6/13/2016): "Belchior Lauer" (This entry is incomplete).
- Belchior Lauer** (PUC-Campinas, 6/13/2016): "Larissa Ferreira" (This entry is incomplete).

Figura 3. Tela inicial de um estudante.

3.2 Gerenciamento de Tags

O artefato desenvolvido contém um módulo responsável pelo gerenciamento de *tags*. Esta parte do sistema é acessível apenas por moderadores

e tem como responsabilidade incluir, excluir, moderar, renomear e juntar novas *tags*.

Um moderador pode acessar as funcionalidades descritas anteriormente através de uma única página, mostrada na Figura 4. As *tags* são agrupadas por instituição e cada um destes grupos divide as *tags* em dois subgrupos: *tags* aprovadas e *tags* pendentes.

Figura 4. Tela de gerenciamento de *tags*.

As *tags* pendentes são aquelas que foram pedidas por estudantes ou instrutores e precisam ser moderadas pelo moderador da instituição. As *tags* aprovadas são aquelas que já foram confirmadas e podem ser excluídas, renomeadas e juntadas com as *tags* que ainda não foram moderadas.

Novas *tags* podem requisitadas através da criação de perguntas, funcionalidade que é detalhada na Subseção 3.4.1, ou através da edição do perfil de um estudante, no caso de instrutores informando ao sistema quais disciplinas eles podem ajudar os estudantes. Em ambos os casos, as *tags* não moderadas são listadas na tela de moderação de *tags* e ficam disponíveis para moderação. As *tags* só aparecem nas perguntas ou perfis quando são aprovadas por algum moderador.

Para controlar *tags* criadas com erros de digitação no momento de incluir uma *tag* em uma pergunta ou em um perfil, os moderadores podem juntar estas *tags* erradas com outras já aprovadas, fazendo com que as perguntas ou perfis de instrutores associados a estas sejam atualizados com uma considerada correta.

3.3 Gerenciamento de Instituições

No sistema CoLearn, cada instituição cadastrada contém perguntas e respostas associadas a ela. Elas são gerenciadas exclusivamente pelos administradores do sistema, que podem incluir, aceitar ou rejeitar novas instituições pedidas pelos estudantes.

3.3.1 Cadastramento de Instituições

O administrador do site cobra novas instituições informando o nome, a sigla e o código da instituição, conforme é ilustrado na Figura 5. Estes dados podem ser obtidos através do portal do MEC.

Figura 5. Tela de cadastramento de novas instituições.

Após o cadastro, o sistema pede para que seja associado um moderador à instituição cadastrada. O administrador pode digitar o *email* do usuário, e o sistema automaticamente preenche os campos de nome e sobrenome.

No caso de moderadores que ainda não estão cadastrados no sistema, é enviada uma mensagem para o endereço de *email* informado, convidando a

pessoa a se cadastrar no *site* e se tornar um moderador. A instituição só é de fato incluída no sistema quando este processo terminar.

3.3.2 Moderação de Instituições

O sistema também permite que o administrador aprove o cadastro de novas instituições pedidas pelos estudantes, a partir de uma lista de pedidos pendentes que mostram o código, o nome e a sigla das instituições pendentes de aprovação, como pode ser visto na Figura 6.



The screenshot shows a table titled 'CoLearn' with a header row containing '#', 'Nome completo', 'Sigla', 'Código', and 'Ação'. There are three rows of data:

#	Nome completo	Sigla	Código	Ação
1	PUC	PUC	1	APROVAR/RECUSAR
2	Pontifícia Universidade Católica do Rio de Janeiro	PUC-RIO	528	APROVAR/RECUSAR
3	Universidade Federal do Amazonas	UFAM	4	APROVAR/RECUSAR

A dropdown menu is open over the third row's 'Ação' button, showing options 'Aprovar' and 'Recusar'.

Figura 6. Tela para moderar pedidos de cadastramento de instituições.

Caso um pedido seja aprovado, o fluxo será idêntico ao do cadastramento de uma nova instituição. Se o pedido não for aprovado, o administrador descreve o motivo da não aprovação e um *email* com esta explicação é enviado ao estudante que solicitou a inclusão da instituição.

3.4 Gerenciamento de Perguntas

No sistema CoLearn existe um módulo responsável por gerenciar as perguntas feitas por estudantes. Este módulo é acessível a todos os usuários, com exceção do administrador do *site*.

3.4.1 Criação de Perguntas

Para criar uma pergunta, um estudante precisa preencher três campos: título, *tags* e descrição, sendo este último opcional. Durante o preenchimento do

primeiro campo, o sistema sugere outras perguntas que foram criadas previamente, como é descrito na Seção 3.4.2.

É permitido que um estudante crie uma pergunta com *tags* que não existem na base de dados. Caso todas as *tags* escolhidas sejam novas, a pergunta não é listada para outros estudantes até que, pelo menos, uma das *tags* seja aprovada pelo moderador. Se a pergunta contiver *tags* existentes e não existentes, a pergunta é criada e aparece para outros usuários; porém, as *tags* que não estão aprovadas não aparecem na questão, até que um moderador as aprove.

3.4.2 Sugestão de Perguntas Similares

Conforme o título de uma pergunta é digitado, o sistema procura por perguntas que contêm títulos similares ao que está sendo digitado, oferecendo ao estudante uma forma de encontrar questionamentos equivalentes feitos por outros usuários, com tolerância a erro de digitação. A tolerância a erros permite ao sistema sugerir perguntas com palavras exatamente ou parcialmente iguais as que estão sendo digitadas, como ilustra a Figura 7.

The screenshot shows the CoLearn platform's question creation interface. At the top, there's a blue header bar with the CoLearn logo and user profile information. The main area has a light gray background. A text input field asks 'Qual a sua dúvida?'. Below it, a message says 'Após descrever sua dúvida, adicione tags para que outros estudantes encontrem sua pergunta mais facilmente.' An asterisk indicates it's an optional field. A suggestion 'O que é uma chav publica?' appears in the input field, preceded by an 'O*' icon. Below the input field is a button labeled 'Escolha suas tags'. At the bottom of the interface is a rich text editor toolbar with various formatting options like bold, italic, underline, etc. A large text area for writing the question details is at the bottom, with a 'PERGUNTAR!' button at the very bottom right.

Figura 7. Sugestão de pergunta similar à que está sendo criada.

Cada sugestão feita pode ser clicada e uma página com os detalhes da pergunta associada à sugestão é carregada, em uma nova guia do navegador Web, para que o usuário visualize os detalhes da pergunta, incluindo a descrição e as respostas da mesma.

3.4.3 Filtragem de Perguntas

Na página inicial, os estudantes visualizam as perguntas relacionadas aos seus perfis. Estas perguntas são selecionadas para os usuários através de um filtro, que considera as *tags* utilizadas em seus perfis e nas perguntas respondidas por estes, no caso de instrutores. As *tags* das perguntas criadas pelos próprios estudantes também são consideradas no filtro.

Também é possível utilizar filtros customizados, para filtrar perguntas específicas. O sistema suporta a filtragem por título e/ou *tags*. Além disto, o usuário também pode escolher se quer ver apenas as suas perguntas, assim como optar por visualizar questões exclusivamente de sua instituição ou de todas as cadastradas no site.

3.5 Sugestão de Materiais de Estudos

Para estudantes que criaram perguntas com *tags* novas, é sugerida uma lista de materiais de estudo sobre o assunto perguntado, para que o autor da questão tenha algum material de apoio para consultar enquanto aguarda à aprovação das *tags* do seu questionamento.

Estes materiais são consultados no repositório de objetos de aprendizagem MERLOT (MERLOT, 2016). Este sistema oferece um modo de pesquisa que consulta diversas bases que contêm objetos deste tipo, além de seu próprio catálogo.

A pesquisa no MERLOT é feita através de palavras-chave, que são extraídas do título da pergunta utilizando a biblioteca Apache Lucene.NET e depois combinadas com as *tags* utilizadas para a criação da questão. No final, estas palavras-chave são passadas ao MERLOT, que retorna uma página HTML com os

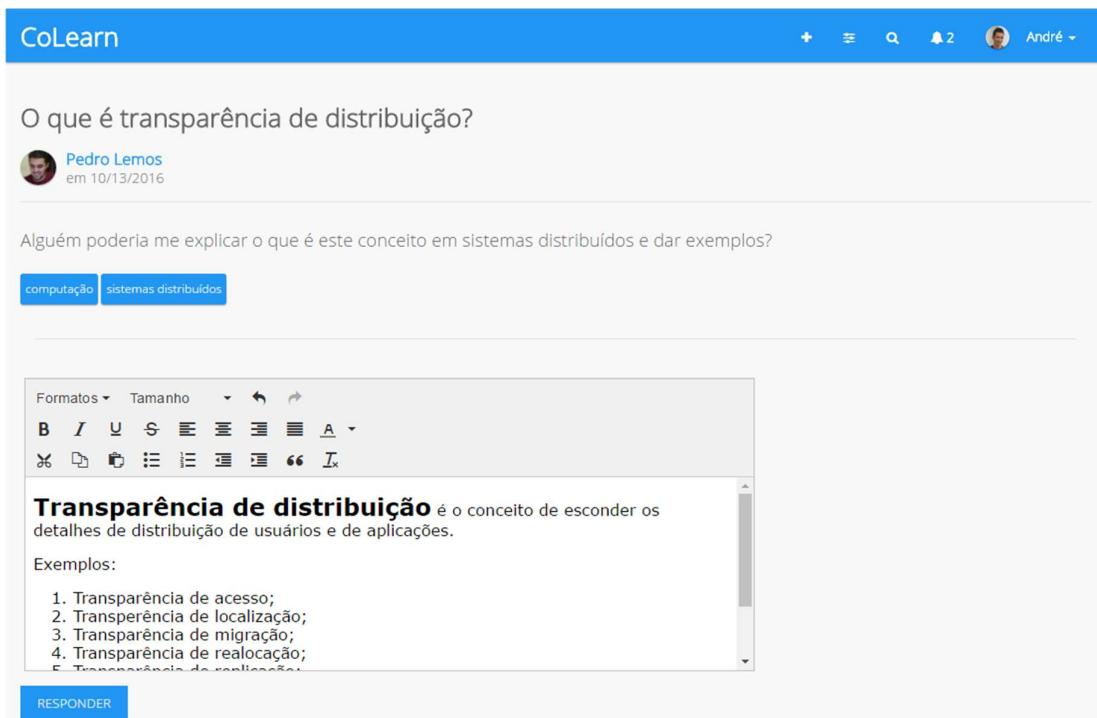
resultados. O sistema então procura pelos *links* dos materiais através de *tags* HTML específicas e exibe ao estudante os materiais com título, *link* e o nome do *site* que contém este material. Ao clicar em um material sugerido, o sistema redireciona o estudante à página *Web* que contém este item.

3.6 Criação e Avaliação de Respostas

Perguntas criadas por estudantes podem ser respondidas por instrutores de duas formas: texto ou videoconferência. A segunda forma é abordada na Seção 3.7. Além da criação de respostas, também é possível a avaliação destas.

3.6.1 Criação de Respostas em Texto

Para responder às perguntas, instrutores podem criar respostas em texto que serão visíveis a todos os outros usuários que acessarem estas questões. Como pode ser visto na Figura 8, é possível a formatação do texto das respostas, para que estas possam ser criadas de maneira mais organizadas, dependendo da criatividade do respondedor.



The screenshot shows a user interface for a learning platform. At the top, there's a blue header bar with the logo 'CoLearn' on the left and user profile icons on the right. Below the header, a question is displayed: 'O que é transparência de distribuição?'. This question was posted by a user named 'Pedro Lemos' on October 13, 2016. Underneath the question, there's a text input field containing the following text:

Alguém poderia me explicar o que é este conceito em sistemas distribuídos e dar exemplos?

Below the text input, there are two blue buttons: 'computação' and 'sistemas distribuídos'. A rich text editor toolbar is visible above the text area, featuring various formatting options like bold, italic, underline, and alignment. The text in the editor is formatted as follows:

Transparência de distribuição é o conceito de esconder os detalhes de distribuição de usuários e de aplicações.

Exemplos:

1. Transparência de acesso;
2. Transparência de localização;
3. Transparência de migração;
4. Transparência de realocação;
5. Transparência de replicação;

At the bottom of the text area, there's a blue 'RESPOSTER' button.

Figura 8. Tela para responder uma questão.

As novas respostas são exibidas para os usuários assim que elas são criadas, ou seja, quando uma pergunta é respondida, o servidor notifica todos os clientes para que não seja necessário recarregar a página para ver novas respostas a um questionamento. Isto foi possível pela utilização da tecnologia WebSockets, que mantém uma conexão TCP aberta com o servidor durante todo o tempo em que uma página estiver sendo exibida.

3.6.2 Avaliação das Respostas

Para que outros estudantes saibam quais são as melhores e mais confiáveis respostas a uma pergunta, apenas instrutores de disciplinas compatíveis com as *tags* da questão podem avaliar as respostas desta. Ou seja, uma dúvida contendo as *tags* “matemática” e “álgebra linear” só pode ter suas respostas avaliadas por instrutores destas disciplinas (que adicionaram estas *tags* em seus perfis). Na Figura 9 é ilustrada a tela de respostas, na qual um instrutor consegue escolher as melhores respostas, avaliando-as. Também é possível observar, na figura, que as respostas avaliadas têm suas estrelas preenchidas de acordo com a sua avaliação média.

The screenshot shows a list of three responses to a question. Each response includes a user profile picture, the user's name, the date of the post, and a video icon. Below each response is a rating bar indicating the average score and the number of reviews.

- Sara Nicoleti** em 6/13/2016

O termo se refere a um enorme conjunto de técnicas que visam construir sistemas computacionais cujo comportamento seja definido com base em dados existentes. Como o comportamento do sistema não seria diretamente programado, mas sim adaptado de algum "conhecimento" previamente adquirido, essa abordagem teria similaridade com a forma como animais (entre eles, nós humanos) aprendem com a experiência.

4.5 (8)

- Belchior Lauer** em 6/13/2016

Machine Learning ou Aprendizado de Máquina é um método de análise de dados que automatiza o desenvolvimento de modelos analíticos. Usando algoritmos que aprendem interativamente a partir de dados, o aprendizado de máquinas permite que os computadores encontrem insights ocultos sem serem explicitamente programados para procurar algo específico.

4.1 (4)

- Lucas Silva** em 10/20/2016

É o processo pelo qual uma máquina aprende a partir de dados pré-definidos.

2 (1)

Figura 9. Respostas dadas a uma pergunta, ordenadas por avaliação.

Para um instrutor dar uma nota para uma resposta, ele pode clicar sobre uma das cinco estrelas. As estrelas são então atualizadas para refletir a nova avaliação média e a ordem da pergunta avaliada pode mudar, caso sua nova avaliação média seja maior que a da resposta logo acima ou menor que a da resposta logo abaixo. Para que a ordem das respostas não fique mudando a cada nova avaliação de um instrutor, elas são ordenadas apenas quando a página é carregada, ou seja, os usuários precisam recarregar a página que mostra as respostas para que a ordem destas sejam atualizadas.

O valor da avaliação vai de um (muito ruim) a cinco (muito bom), sendo válidos valores inteiros e valores decimais (de meio em meio). Para o usuário, estes valores são representados por estrelas e, para cada resposta, são mostradas a avaliação média e o número total de avaliações. Além dessas informações, também é mostrado, para cada resposta, um botão ao lado do nome do instrutor que criou a resposta, para que um estudante inicie uma videoconferência com ele. A funcionalidade de videoconferência é apresentada na Seção 3.7.

Como uma forma de ajudar os estudantes a encontrarem as melhores respostas, estas são ordenadas por avaliação, sendo as mais bem avaliadas mostradas primeiro. Deste modo, as respostas consideradas mais corretas tendem a serem visualizadas pelos usuários antes das outras.

3.7 Videoconferência

Caso o estudante precise de uma explicação melhor sobre o assunto, ele pode requisitar uma videoconferência a um instrutor. Os participantes podem visualizar uns aos outros, além de poderem utilizar uma lousa para desenhos, como é ilustrado na Figura 10. Todas as videoconferências são gravadas e disponibilizadas aos outros usuários do sistema, em forma de resposta.

A gravação da videoconferência é transmitida ao servidor em tempo real, com o auxílio da biblioteca RecordRTC. Ao invés de salvá-la no dispositivo do usuário e depois transmitir toda a gravação de uma única vez, o que pode levar horas, dependendo da duração da videoconferência, os arquivos de vídeo e de áudio são gravados durante intervalos específicos de tempo e enviados ao servidor separadamente. Este processo se repete até que a videoconferência termine.

O servidor então salva os *frames* de áudio e de vídeo recebidos em pastas separadas e, ao final da transmissão, junta todos os pedaços de vídeo e de áudio que foram gravados, com ajuda da ferramenta FFmpeg, que foi utilizada para juntar os quadros de áudio e vídeo em um único arquivo. Após o processamento dos dados, a gravação completa da videoconferência é disponibilizada para ser assistido por outros usuários do sistema.

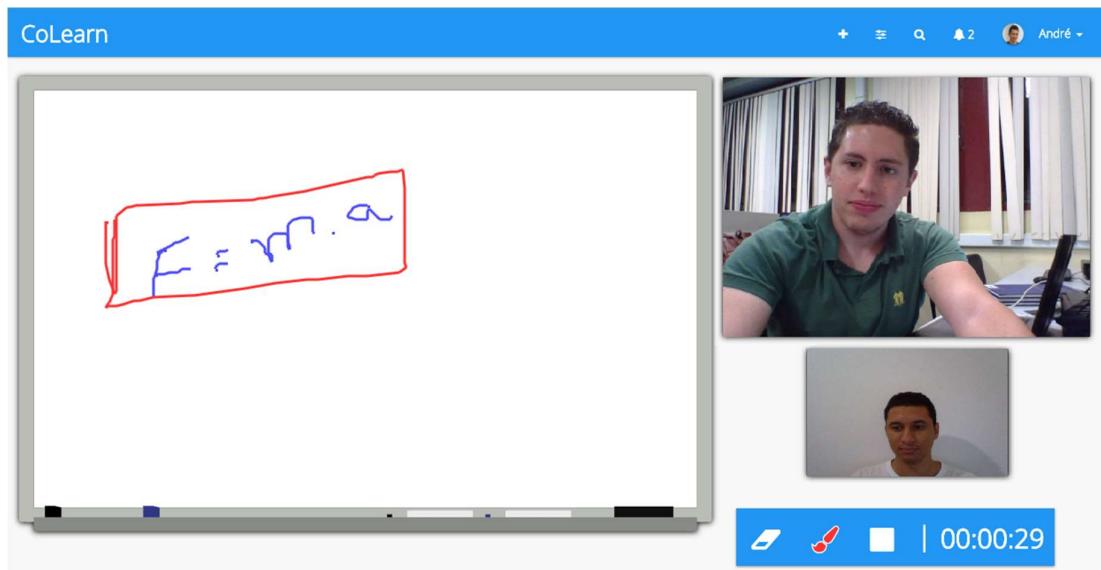


Figura 10. Tela da videoconferência entre dois usuários.

Uma videoconferência pode ser iniciada a partir de uma resposta ou a partir do perfil de um instrutor. No primeiro caso, a videoconferência é associada à pergunta em que uma resposta está contida.

Se um estudante ligar para outro através do perfil deste, será pedido, ao final da chamada, que o estudante que iniciou a conferência crie uma pergunta referente ao assunto tratado na mesma.

Após o término da videoconferência, independente da forma que esta foi iniciada, será requisitado ao estudante que iniciou a chamada que ele avalie como foi a explicação do instrutor. O estudante que iniciou a videoconferência será direcionado para uma página onde ele deve dar uma nota, de um a cinco, representada por estrelas, como na avaliação das respostas em texto. O estudante deve clicar sobre uma estrela para informar a sua nota para então ser direcionado para página de criação de uma nova pergunta, no caso de uma videoconferência

iniciada através do perfil de um instrutor, ou para a página inicial, no caso de uma videoconferência iniciada através da resposta de um instrutor.

Assim que o estudante avaliar a videoconferência e o processo de codificação da gravação for concluído, o sistema disponibiliza o vídeo para outros estudantes assistirem e avaliarem, da mesma forma que é descrita na Seção 3.6.

3.8 Gerenciamento de Agenda

Para realizar uma videoconferência, um estudante precisa antes agendar um horário com o outro participante. Este último precisa definir em seu calendário os horários que estará disponível para a realização de uma videoconferência com outro estudante. Estas funcionalidades estão presentes no módulo “Gerenciar Agenda” e podem ser acessadas por meio de uma interface de calendário.

Os estudantes podem visualizar as suas agendas de duas formas: visão do mês e visão da semana. A navegação entre as semanas e os meses, anteriores e posteriores, também é suportada pelo sistema. As formas de visualização podem ser observadas nas figuras contidas no Apêndice F.

3.8.1 Gerenciamento de Horários Disponíveis

Para um estudante informar aos outros os seus horários livres, este deve acessar a sua agenda. O usuário então pode clicar em um local em branco, que representa os horários que ele está indisponível, para abrir a tela de criação de um novo horário disponível, onde é possível definir o início e o fim do horário sendo criado, além da frequência de repetição deste evento: todos os dias, toda semana ou todo o mês. Um horário disponível é representado por um bloco verde com comprimento relativo ao horário informado na criação do horário, como mostrado no Apêndice F (Figura 24), e pode ser removido ou editado.

Existe também, na página de gerenciamento da agenda, uma interface para um instrutor configurar a duração máxima de uma videoconferência agendada

com ele, forçando os outros estudantes a não agendar mais horas com este instrutor do que o configurado por ele.

3.8.2 Agendamento de Videoconferência

Um estudante pode navegar até a agenda de outro usuário para agendar uma videoconferência com este. Ao clicar em algum horário disponível, o usuário então descreve o assunto que irá ser abordado e escolhe um intervalo de tempo, respeitando a duração máxima configurada pelo autor da agenda, como foi descrito na Subseção 3.8.1.

Após cada agendamento feito, o calendário então é atualizado e os agendamentos são mostrados com a cor azul, para diferenciar dos horários não agendados. Além disto, as marcações feitas no calendário podem ser canceladas ou editadas.

3.9 Gerenciamento de Níveis Acadêmicos

Os usuários podem informar seu nível acadêmico a outros usuários. Os níveis contêm valores que indicam quais são melhores, que são definidos de acordo com a sua posição na lista de níveis acadêmicos disponíveis.

Os administradores, que são os únicos usuários que podem gerenciar os níveis acadêmicos, podem editar o nome de níveis existentes, assim como adicionar novos, informando um nome através de um campo de texto disponível na página de gerenciamento de níveis acadêmicos, como é ilustrado no Apêndice G. Também é possível definir a posição de um nível, arrastando-o pela lista. A posição de um nível acadêmico é o valor que o quantifica em relação aos outros níveis.

Quando um novo nível é criado por um administrador, ele é listado junto aos existentes, ao final da lista, e pode ser ordenado, isto é, pode ter sua posição modificada. Quanto mais no topo da lista de níveis acadêmicos um nível está, menor é a sua posição e melhor o nível é. Por exemplo, se o nível acadêmico “mestrado” é listado antes do nível “graduação”, o primeiro é considerado melhor que o segundo, pois a posição do nível “mestrado” é menor que a do outro nível.

Toda vez que o sistema percebe que não existe nenhum nível cadastrado, situação que ocorre apenas quando o banco de dados é novo e não contém nenhum dado, os seguintes níveis são criados: pós-doutorado, doutorado, mestrado, especialização e graduação. A posição do primeiro nível é um, a do segundo é dois e assim por diante.

3.10 Pesquisa de Estudantes

A pesquisa de estudantes é feita segundo várias características, como nome, instituição, nível acadêmico e avaliação média.

O usuário pode pesquisar por estudantes informando um nome, ou parte deste. Ele também pode escolher uma ou mais instituições que desejar. Caso nenhuma instituição seja selecionada, o sistema procura por estudantes independente da instituição.

Caso o usuário escolha um nível acadêmico, o sistema procura por estudantes com nível igual ou melhor que o nível selecionado, utilizando a posição dos níveis para compará-los. O usuário pode escolher entre os níveis disponíveis que foram cadastrados pelo administrador do *sítio*, conforme é apresentado na Seção 3.9.

Também é possível pesquisar por usuários utilizando a avaliação média das respostas destes para que o sistema procure por usuários com avaliação média igual ou maior ao valor informado. Ou seja, o sistema permite que o usuário procure por usuários que tenham um valor de avaliação média mínima de duas estrelas, por exemplo.

Os estudantes que apresentarem ao menos um atributo igual ao fornecido pelo usuário na pesquisa serão listados. Caso nenhuma característica seja preenchida ao se realizar a busca, todos os estudantes serão listados. O Apêndice H mostra alguns resultados de pesquisas feitas com diferentes filtros.

3.11 Preferências para Recomendação

O sistema CoLearn permite aos estudantes definirem suas preferências de recomendação. Essas preferências são utilizadas pelo sistema para recomendar instrutores de forma personalizada, como descrito na Seção 3.12.

As preferências que um estudante pode configurar são:

- Avaliação média das respostas dos instrutores, que é comparada à avaliação média de cada instrutor. Se o valor da avaliação média do instrutor for menor que o valor escolhido pelo estudante, o algoritmo de cálculo de similaridade retorna zero. Caso contrário, o valor da avaliação média do instrutor é dividido por cinco, que é o valor máximo de uma avaliação, fazendo o algoritmo retornar um valor entre zero e um. Ou seja, quanto maior for a avaliação de um instrutor em relação ao valor escolhido pelo estudante, maior será o valor da similaridade para a preferência de avaliação média.
- Instituições, que são comparadas à instituição do instrutor. Caso a instituição do instrutor seja igual à selecionada pelo estudante, o algoritmo de cálculo de similaridade retorna um. Caso contrário, o valor retornado é zero.
- Nível acadêmico mínimo dos instrutores, que é comparado ao nível de cada instrutor. Se a posição do nível acadêmico do instrutor for menor que a do nível escolhido pelo estudante, o algoritmo de cálculo de similaridade retorna zero. Caso contrário, o algoritmo calcula o inverso multiplicativo da posição do nível do instrutor, como mostra a Equação (1), onde “s” é o valor de similaridade para um nível “n” com posição “p”. Desse modo, quanto menor a posição do nível acadêmico de um instrutor em relação à posição do nível selecionado pelo estudante, maior será o valor de similaridade para a preferência de nível acadêmico. A Seção 3.9 descreve o que é a posição de um nível e como ela é definida pelo administrador do site.

$$s_n = \frac{1}{p_n} \quad (1)$$

Além das preferências apresentadas, o estudante pode definir, na tela de preferências mostrada na Figura 11, um valor mínimo de similaridade entre o seu perfil e o dos instrutores recomendados. O estudante pode selecionar um valor entre um e cem. Instrutores com similaridade menor que o valor mínimo definido não serão listados.

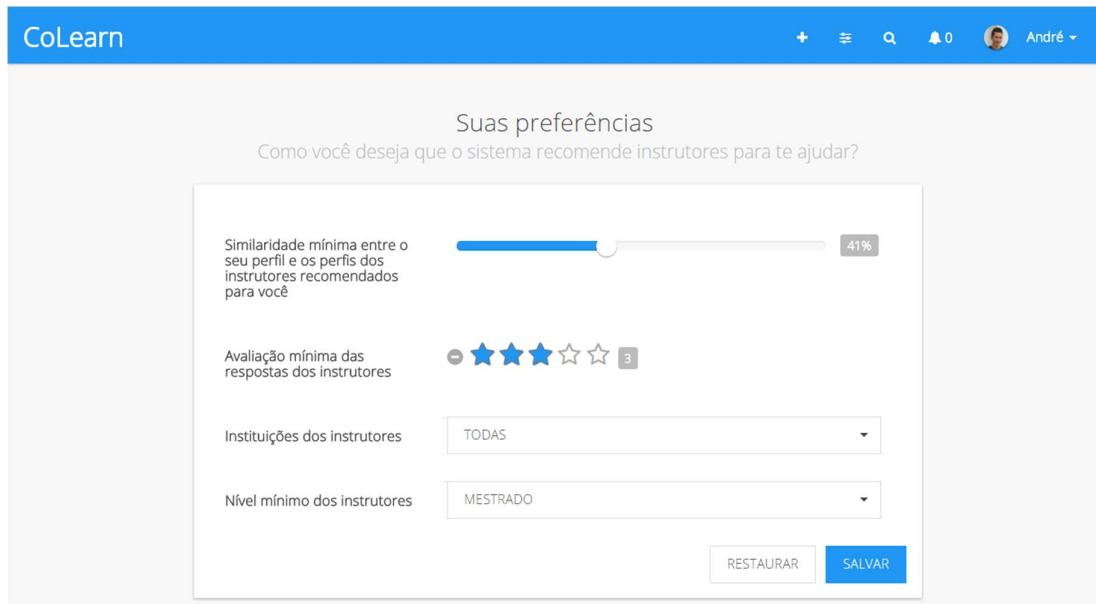


Figura 11. Tela de preferências de um estudante.

Na Seção 3.12 é explicado como o valor de similaridade mínima é utilizado pelo algoritmo de recomendação personalizada.

Para lidar com o problema de estudantes não terem preferências para a recomendação (*cold start*), valores padrão de preferências são configuradas pelo administrador do site através da mesma página que os estudantes e os instrutores definem as suas preferências. Desse modo, todos os estudantes do sistema começam com preferências iguais, até que sejam customizados da forma desejada.

3.12 Recomendação Personalizada de Instrutores

A busca por ajuda também é feita através de recomendações personalizadas de bons instrutores. O sistema procura por instrutores relacionados com as *tags* das perguntas criadas pelos estudantes, realizando um casamento das preferências dos estudantes com os perfis dos instrutores. Dessa forma, um

estudante consegue encontrar um instrutor capaz de ajudá-lo em assuntos que ele tenha demonstrado interesse.

O algoritmo de recomendação RECON (PIZZATO *et al*, 2010) foi adaptado e implementado. Nele, a recomendação é feita através de um valor de similaridade calculado entre um usuário “x” e um usuário “y” e, depois, outro valor de similaridade é calculado entre “y” e “x”, e então é feita uma média harmônica entre as similaridades calculadas. Desse modo, a recomendação feita é reciproca (bilateral).

A principal adaptação feita no algoritmo RECON é a recomendação de instrutores de forma unilateral e não de forma reciproca, como o autor da publicação descreve. Ou seja, a recomendação implementada pelo autor considera apenas o valor de similaridade entre um estudante “a” e um instrutor “b”, enquanto que a similaridade entre “b” e “a” não é calculada.

Para realizar o cálculo de similaridade de um estudante com um instrutor, o algoritmo inicialmente obtém a lista de instrutores cadastrados no sistema. Em seguida, as preferências do estudante que receberá as recomendações são recuperadas, junto com as *tags* que este usuário utilizou para fazer as suas perguntas. As *tags* informam ao sistema quais as disciplinas para as quais o estudante procura por ajuda.

Se nenhuma disciplina for encontrada, ou seja, se o usuário não fez alguma pergunta com *tags* válidas, o algoritmo não faz a recomendação e nenhum resultado é retornado, pois não há informações sobre quais disciplinas o estudante mostrou ter dúvidas. Caso contrário, o algoritmo utiliza as preferências do estudante, incluindo as *tags* utilizadas na criação de suas perguntas, para calcular um valor de similaridade para cada preferência, de forma diferente para cada uma, como é apresentada na Seção 3.11. Os valores calculados estão definidos no intervalo entre zero (nenhum pouco similar) e um (muito similar).

Para as *tags* dos estudantes, o algoritmo calcula a similaridade das mesmas com o perfil de um instrutor. O cálculo é feito contando quantas *tags* do estudante são iguais às de um instrutor e depois divide a contagem pelo número total de *tags* do estudante, obtendo um valor entre zero e um.

Após o cálculo do valor de similaridade para uma preferência, a mesma é multiplicada pelo seu peso (um ou dois), que é obtido de acordo com o tipo da preferência. As *tags* e o valor médio da avaliação de um instrutor são as preferências com maior peso, ou seja, peso dois. Elas são consideradas de forma diferente pelo algoritmo para que instrutores com *tags* compatíveis e boa avaliação sejam listados primeiro que outros que só apresentem as outras preferências similares. As preferências restantes têm peso um.

O valor de similaridade final é a média aritmética de todas as similaridades calculadas individualmente, como mostra a Equação (2), onde “s” é o valor da similaridade de um estudante “e” com um instrutor “i”, “n” é o número de preferências do estudante, “v” é o valor de similaridade de uma preferência “p” no perfil do instrutor e “w” é o número de preferências com peso dois. O último valor é somado à quantidade de preferências, pois existem aquelas que têm seu valor de similaridade multiplicado por dois, como explicado anteriormente. Ou seja, para que o valor calculado pela média aritmética esteja sempre entre zero e um, a inclusão de “w” se faz necessário.

$$s_{e,i} = \frac{\sum v_{p_{e,i}}}{n + w} \quad (2)$$

A partir do valor calculado, é possível filtrar apenas os perfis com valor de similaridade maior ou igual ao número configurado na tela de preferências do estudante, para então ordenar a lista restante segundo a similaridade de cada instrutor. Todo este processo é mostrado no Apêndice I, através de um digrama de atividades, enquanto que o algoritmo de similaridade é ilustrado pela Figura 12.

Os resultados de uma recomendação incluem o valor da similaridade dos instrutores com as preferências do estudante, as *tags* dos instrutores compatíveis com as *tags* que o estudante usou para a criação de suas perguntas, avaliação média da comunidade das respostas dadas, instituição e grau acadêmico. No Apêndice J é mostrado alguns instrutores recomendados para um estudante que criou questões referentes às disciplinas “computação” e “sistemas operacionais”.

Um usuário pode navegar pelo perfil de cada instrutor, onde é possível visualizar as contribuições deste, que são as perguntas feitas e respondidas por ele. Deste modo, um estudante consegue avaliar como um instrutor responde aos questionamentos de outros usuários, entrando em contato com o qual ele achar que responde de maneira mais clara sobre um assunto em específico.

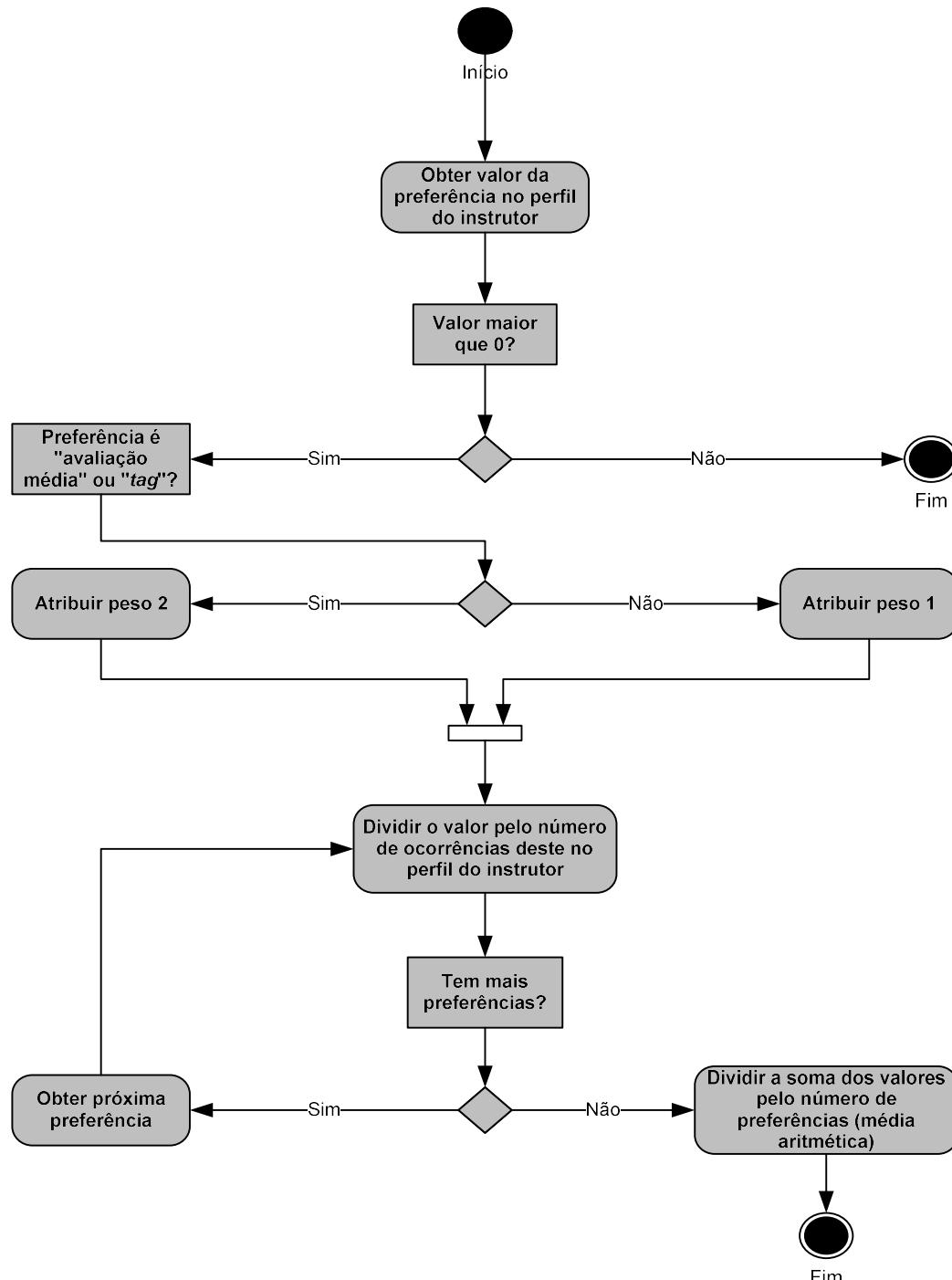


Figura 12. Diagrama de atividades do algoritmo de cálculo de similaridade.

4 AVALIAÇÃO E ANÁLISE DOS RESULTADOS

O artefato desenvolvido no TCC visa auxiliar estudantes a encontrar ajuda para esclarecer suas dúvidas por meio da aprendizagem colaborativa, com o objetivo de aprimorar o conhecimento destes. Para tanto, foi elaborado um roteiro de avaliação e um questionário para análise dos resultados obtidos.

4.1 Avaliação

A avaliação do artefato foi realizada por um professor da Pontifícia Universidade Católica de Campinas. Ele possui graduação em Ciência da Computação, mestrado e doutorado em Engenharia Elétrica. Tem experiência nas áreas de Ciência e Engenharia de Computação, com ênfase em Sistemas Inteligentes. Já realizou trabalhos na área de aprendizagem colaborativa, sendo este o motivo que mais pesou na escolha deste profissional para a realização da avaliação.

O ambiente de avaliação foi preparado pelo próprio autor. Foram utilizados dois computadores com o navegador *Web Google Chrome* instalado, para que fosse possível o acesso ao *site*, tanto pelo avaliador quanto pelo autor do TCC.

O sistema continha em sua base de dados perguntas e respostas previamente criadas pelo autor do sistema. Estes dados podem ser encontrados no Apêndice K.

A avaliação foi feita através de uma análise exploratória pelo *site*, incluindo a navegação pelas perguntas e suas respostas, e a busca por bons instrutores que pudessem ajudar os estudantes nos assuntos que eles tivessem demonstrado interesse (perguntas feitas). O avaliador seguiu os passos do roteiro contido no Apêndice L para avaliar o sistema.

A avaliação ocorreu no dia 25 de outubro de 2016 no período da manhã. O local escolhido para a sua realização foi a sala do professor, localizada no campus I da universidade que este atua. Esta localidade foi definida pelo próprio

avaliador e a data e o horário foram definidos de acordo com a disponibilidade do mesmo.

Os resultados obtidos com esta avaliação são as respostas das questões fornecidas pelo avaliador do sistema. Os resultados e as análises destes estão presentes a seguir.

4.2 Análise dos Resultados

O questionário fornecido ao avaliador contém três perguntas que cobrem todo o roteiro contido no Apêndice L.

Na primeira pergunta, o avaliador teve que dizer se, dado a experiência dele na avaliação, o sistema permite que estudantes melhorem seus conhecimentos em assuntos de seus interesses. A resposta foi: “Sim. A possibilidade de examinar as várias questões e respostas relativas a um tópico de interesse ajuda a entender os principais conceitos a ele relacionados. Os mecanismos de busca por palavras contidas nas questões respondidas e por etiquetas (*tags*) associadas às questões facilitam a identificação do material relevante”.

Na segunda questão, o avaliador teve que dizer se a forma de interação entre estudantes e instrutores, através de perguntas e respostas, favorece a melhora de conhecimento dos estudantes. A resposta para esta questão foi: “Sim. A disponibilidade de uma comunidade ampla de estudantes fazendo perguntas e de instrutores fornecendo as respostas permite que os estudantes percebam nuances que os ajudam a solidificar conceitos. Isto é favorecido pela possibilidade de que vários instrutores respondam às mesmas perguntas, propiciando a depuração das respostas”.

Na terceira e última questão, o avaliador foi perguntado se, segundo as respostas dadas pelos instrutores, que podem ser vistas no Apêndice K, o sistema ajuda um estudante a encontrar bons instrutores, que possam ajudá-lo a melhorar seus conhecimentos em assuntos que eles tenham demonstrado ter interesse, através das perguntas que o estudante criou. O avaliador respondeu: “Sim. Ao examinar as respostas dos instrutores recomendados para várias questões, o

estudante tenderá a identificar aqueles que se expressam de uma maneira que lhe é mais clara, assim como aqueles que têm mais credenciais em cada assunto”.

As respostas dadas pelo avaliador demonstram que o sistema é capaz de melhorar o conhecimento dos estudantes, dada a forma como é feita a interação entre estes e os instrutores. Além disso, percebe-se que a maneira como a busca por ajuda de instrutores é feita colabora de forma positiva para que os aprendizes possam aperfeiçoar seus conhecimentos.

Portanto, após a análise dos resultados da avaliação, é possível afirmar que o objetivo foi atingido. Isso significa dizer que, utilizando o sistema CoLearn, um estudante consegue melhorar os seus conhecimentos.

5 CONSIDERAÇÕES FINAIS

Esta monografia apresenta informações sobre o desenvolvimento de um artefato computacional, denominado CoLearn, cujo objetivo é melhorar o conhecimento dos estudantes em assuntos que eles tenham interesse, considerando o contexto descrito no Capítulo 1.

Como foi descrito anteriormente na Seção 4.2, o objetivo foi atingido por meio de uma análise feita às respostas obtidas a partir de um questionário respondido por um professor experiente.

Durante o decorrer deste trabalho, o autor se encontrou em situações com as quais ainda não tinha experiência, como entrevistar um cliente para extrair requisitos funcionais, planejar cronogramas e gerenciar o seu tempo disponível para o trabalho, realizar encaminhamentos para atividades que sofreram atrasos, priorizar tarefas e estimar o tempo necessário para a realização de cada uma destas, documentar os detalhes de desenvolvimento de um *software* e usar ferramentas desconhecidas.

Como o sistema desenvolvido é um *website* que pode ser visto como um sistema distribuído, o autor precisou aprender ferramentas para desenvolver o sistema servidor e outras ferramentas diferentes para o sistema cliente, o que aumentou o conhecimento do mesmo sobre o desenvolvimento de sistemas *Web*.

O método de desenvolvimento utilizado foi o Scrum, adaptado ao contexto deste trabalho. Este método se mostrou bastante vantajoso para o autor, porque facilitou a comunicação com o cliente, permitiu uma boa flexibilidade na alocação das tarefas e se mostrou ágil no gerenciamento do projeto em geral. As dificuldades encontradas foram superadas, porém o cronograma inicial não foi respeitado e o artefato foi entregue com algumas semanas de atraso.

5.1 Grau de Dificuldade - Aspectos de Inovação e Aprimoramento

Nesta seção são apresentados os graus de dificuldade, ou seja, os aspectos de inovação e aprimoramento envolvidos no desenvolvimento do sistema CoLearn.

5.1.1 Aprimoramento e Inovação - Inteligência

O sistema desenvolvido conta com inteligência para recomendar instrutores a estudantes. A implementação desta parte do sistema foi especialmente complexa, devido ao desconhecimento do autor de algoritmos de recomendação, especialmente os personalizados, além de necessitar de muito estudo para definir qual algoritmo utilizar.

Após uma série de publicações lidas na área de recomendação de pessoas para encontros *online*, além de reuniões com o coorientador do TCC, foi escolhido implementar o algoritmo RECON.

A forma como o algoritmo foi aplicado no sistema CoLearn, descrita na Seção 3.12, não é encontrada nos sistemas publicados consultados pelo autor. Devido a isso, a inteligência implementada no sistema é considerada, além de um aprimoramento, também uma inovação.

5.1.2 Aprimoramento – Videoconferência

A implementação do módulo de videoconferência, responsável por realizar e gravar a comunicação dos estudantes, em tempo real, foi uma das maiores dificuldades encontradas pelo autor ao longo do desenvolvimento do sistema. Além da falta de conhecimento das APIs WebRTC e RecordRTC, o autor nunca havia desenvolvido algo relacionado com comunicação em tempo real utilizando áudio e vídeo, o que dificultou ainda mais implementação do módulo em questão.

Para integrar estas APIs ao sistema CoLearn e superar as dificuldades encontradas, foi necessária a leitura de vários tutoriais, assim como da

documentação existente, para que se pudesse aprender a utilizar estas APIs da melhor maneira possível.

5.1.3 Aprimoramento - Programação Web

A programação *Web* foi um dos grandes desafios enfrentados ao longo do desenvolvimento do sistema. O autor tinha pouca experiência com esse tipo de programação, fazendo com que o mesmo tivesse que estudar e seguir tutoriais para as ferramentas, *frameworks* e tecnologias envolvidas na programação do CoLearn.

Também foi um desafio a implementação do padrão de arquitetura MVC. Este padrão exigiu do autor algumas horas dedicadas ao estudo do mesmo, para que a implementação do código fosse feita nas camadas adequadas, o que facilitará futuras manutenções do *software*.

O fato do artefato poder ser utilizado através de um navegador *Web*, independentemente do tamanho da tela do dispositivo sendo usado, também foi algo novo aprendido pelo autor. Como este tinha pouco experiência com o desenvolvimento de *websites* responsivos, este item foi de complexidade relevante ao longo do projeto.

5.2 Análise da Qualidade e do Grau de Complexidade

Nesta subseção é feita uma análise da qualidade e do grau de complexidade do sistema CoLearn.

5.2.1 Análise de Qualidade

Para a descrição da qualidade do artefato de *software* desenvolvido no TCC, foi feita a análise do CoLearn com relação a uma norma técnica (ABNT, 2003) sobre qualidade de *software*. Essa norma define os seguintes atributos de qualidade: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.

A funcionalidade foi constatada, uma vez que o sistema desenvolvido atende a todos os requisitos definidos pelo cliente do TCC no início e no decorrer do projeto. O cliente acompanhou a evolução do sistema, por meio das reuniões com este, como é especificado pelo método de desenvolvimento escolhido.

A confiabilidade pode ser verificada no CoLearn, pois ele foi desenvolvido com tolerância a falhas e exceções, como a inserção de dados fora de um intervalo estabelecido ou o não preenchimento de campos obrigatórios. Os dados dos usuários estão armazenados de forma confiável, através de criptografia, e são acessados mediante autenticação. Além disto, os dados dos usuários são armazenados em um banco de dados SQL Server toda vez que eles são atualizados. O banco de dados é executado em um servidor da plataforma Azure, que oferece suporte à recuperação de dados em caso de falha do banco.

O artefato computacional desenvolvido contempla o item de usabilidade. Houve grande preocupação com a usabilidade do CoLearn durante seu planejamento e desenvolvimento, levando-se em conta que o *software* foi feito para ser manipulado por diversos tipos de usuários, incluindo os que são leigos na utilização de sistemas informatizados. Isto pôde ser comprovado pelo *feedback* recebido do avaliador no momento de avaliação do sistema, além da observação da facilidade de como ele utilizou o *site*, comprovando, então, que o sistema atende à característica de usabilidade.

A eficiência do sistema pôde ser evidenciada, porque ele se mantém estável em um nível de desempenho muito bom e não apresenta problemas de recursos alocados. Porém, o mesmo não passou por testes de grandes cargas de dados para uma melhor análise de seu desempenho, e a funcionalidade de videoconferência não apresenta a mesma performance do restante do sistema, o que faz com que o mesmo não satisfaça de forma plena esta característica.

A manutenibilidade foi uma das características mais trabalhadas pelo autor do sistema. O CoLearn apresenta o padrão arquitetural MVC e segue, de forma muito similar, seus conceitos e regras, o que facilita a manutenção do *software* por terceiros, pois o código foi separado nas camadas corretas, facilitando mudanças e diagnósticos em partes específicas do sistema. Também foram

empregadas boas práticas de programação na maioria dos módulos do sistema, além dos comentários nas partes mais complexas do sistema. É possível dizer que o sistema atende à característica de manutenibilidade.

Por ser um sistema *Web*, o CoLearn pode ser acessado de qualquer dispositivo que possua um navegador *Web* e acesso à Internet. Porém, para ser executado, é necessária que a máquina possua o sistema operacional Windows, porque a versão do ASP.NET MVC utilizada não é portável para outros sistemas operacionais. Logo, o sistema CoLearn não é portável.

5.2.2 Análise do Grau de Complexidade

Segundo Toscani e Veloso (2012), para medir a quantidade de trabalho realizado por um algoritmo, costuma-se escolher uma operação, denominada operação fundamental e então contar o número de execuções desta operação na execução do algoritmo, dispensando outras medidas. Portanto, a análise do grau de complexidade do CoLearn foi feita escolhendo-se o algoritmo mais complexo do sistema, o algoritmo de cálculo de similaridade, utilizado na recomendação personalizada de instrutores.

Apesar de complexo, a videoconferência não foi considerada para análise porque para seu desenvolvimento foram utilizadas várias bibliotecas e APIs, enquanto o algoritmo de recomendação foi desenvolvido por inteiro pelo autor.

A recomendação implementada no *software* desenvolvido apresenta duas principais iterações: o algoritmo sempre percorre todos os perfis dos instrutores para manipulá-los um a um e calcular a similaridade de cada um individualmente; para cada instrutor, os atributos de seus perfis são comparados com cada preferência do usuário, para que seja possível calcular o valor de similaridade para cada uma destas preferências, individualmente. No cálculo de similaridade, o algoritmo itera preferência por preferência contida na lista de preferências.

Analisando os passos descritos acima, é possível identificar que a operação fundamental, o cálculo de similaridade, está contida dentro de dois *loops* encadeados. Verifica-se então que a funcionalidade de recomendação é

influenciada diretamente pelo número de instrutores e pelas preferências dos estudantes. Como ambos os números podem crescer indefinidamente, conclui-se que a complexidade deste algoritmo é $O(n \times m)$, onde "n" é o número de instrutores e "m" é o número de preferências do estudante. Ou seja, a complexidade do algoritmo é quadrática.

5.3 Dificuldades Encontradas

As principais dificuldades encontradas foram aprender sobre o desenvolvimento de aplicações *Web*, como descrito na Seção 5.1, e gerenciar o tempo dedicado ao TCC.

Gerenciar o tempo disponível do autor foi um grande obstáculo encontrado ao longo do projeto. O autor sentiu dificuldade em estimar as tarefas planejadas de forma correta e em seguir o planejamento feito em cada *sprint*. Isto causou atrasos, que foram solucionados com o replanejamento das atividades e, em alguns casos, o cronograma teve que ser modificado. Horas extras dedicadas ao TCC também foram necessárias.

5.4 Melhorias Futuras

Durante as reuniões com o cliente do TCC, surgiram algumas ideias para melhorar o artefato. Porém, alguns destes conceitos não foram priorizados e, portanto, não foram implementados no sistema CoLearn. Algumas das ideias que podem ser implementadas futuramente para deixar a solução mais completa são:

- Comentários nas respostas dadas pelos instrutores, para que estes recebam *feedback* da comunidade e melhorem a qualidade de suas respostas.
- Exibir a média da avaliação das respostas de um instrutor separadas por tags, para que outros usuários do sistema vejam em quais áreas do conhecimento este instrutor possuí melhor avaliação.

- Migrar a solução para ASP.NET Core MVC 6 para que o *software* possa ser portado para diferentes ambientes, fazendo este atender à característica de portabilidade, analisada na Seção 5.2.1.
- Permitir que as instituições de ensino superior sejam cadastradas diretamente do portal do MEC, sem a necessidade dos administradores do sistema inserirem os dados manualmente.
- Permitir que os instrutores aceitem ou rejeitem os agendamentos para videoconferência feitos pelos estudantes.

REFERÊNCIAS

ABNT, ABNT NBR ISO/IEC 9126-1:2003 *Engenharia de software - Qualidade de produto Parte 1: modelo de qualidade*, 2003.

APACHE. *Apache Lucene.Net*. Disponível em: <<https://lucenenet.apache.org>>. Acesso em: 01 dez. 2016.

ARAI, K.; HANDAYANI, A. N. Question answering system for an effective collaborative learning. *IJACSA Journal*, v. 3, p. 60-64, n. 1, 2012.

ASP.NET. *ASP.NET MVC Overview*. Disponível em: <<https://www.asp.net/mvc/overview/older-versions-1/overview/asp-net-mvc-overview>>. Acesso em: 03 out. 2016.

FFMPEG. Disponível em: <<https://www.ffmpeg.org/about.html>>. Acesso em: 04 out. 2016.

FROTA, V. B. *Sistema inteligente de recomendação baseado no modelo do aluno*. 2012.

FULLCALENDAR. *FullCalendar – JavaScript Event Calendar*. Disponível em: <<https://fullcalendar.io>>. Acesso em: 01 dez. 2016.

JSON. *Introducing JSON*. Disponível em: <<http://www.json.org>>. Acesso em: 03 out. 2016.

MERLOT. Disponível em: <<https://www.merlot.org/merlot/index.htm>>. Acesso em: 18 out. 2016.

MICROSOFT DEVELOPER NETWORK. *Introduction to the C# Language and the .NET Framework*. Disponível em: <<https://msdn.microsoft.com/pt-br/library/z1zx9t92.aspx>>. Acesso em: 03 out. 2016a.

MICROSOFT DEVELOPER NETWORK. *Visual Studio IDE*. Disponível em: <<https://msdn.microsoft.com/pt-br/library/dn762121.aspx>>. Acesso em: 03 out. 2016b.

MICROSOFT DEVELOPER NETWORK. *TFVC - Team Foundation Version Control*. Disponível em: <<https://msdn.microsoft.com/pt-br/library/cc580642.aspx>>. Acesso em: 04 out. 2016c.

MOZILLA DEVELOPER NETWORK. *JavaScript*. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>. Acesso em: 03 out. 2016a.

MOZILLA DEVELOPER NETWORK. *WebSockets*. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/WebSockets>>. Acesso em: 04 out 2016b.

PIZZATO, L; REJ, T; CHUNG, T; KOPRINSKA, I; KAY, J. RECON: a reciprocal recommender for online dating. *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010. p. 207-214.

RECORDRTC. Disponível em: <<http://recordrtc.org>>. Acesso em: 04 out. 2016.

RUBIN, K. S. *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley, 2012.

SANTOS, C. dos. *Fundamentos do Entity Framework 4.* Disponível em: <<https://msdn.microsoft.com/pt-br/library/jj128157.aspx>>. Acesso em: 03 out. 2016.

SASAKI, N. et al. Collaborative learning support system in a classroom using ad hoc network of tablet devices. *2015 International Conference on Learning and Teaching in Computing and Engineering. IEEE.* p. 142-143, 2015.

TOSCANI, L. V; VELOSO, P. A. *Complexidade de algoritmos.* 3 Ed. Sagra-Luzzatto, 2012. 202p.

ZHANG, L.; HAZEYAMA, A. A Collaborative Learning Support System for Software Engineering Education. *Proc. 6th Conference on Collaboration Technologies (Collabtech2012).* p. 123-126, 2012.

W3C. *W3C HTML.* Disponível em: <<http://www.w3.org/html/>>. Acesso em: 03 out. 2016.

W3SCHOOLS. CSS *Introduction.* Disponível em: <http://www.w3schools.com/css/css_intro.asp>. Acesso em: 03 out. 2016.

WEBRTC. Disponível em: <<https://webrtc.org/>>. Acesso em: 04 out. 2016.

APÊNDICES

Apêndice A Diagrama de Arquitetura Elaborado no Primeiro Semestre

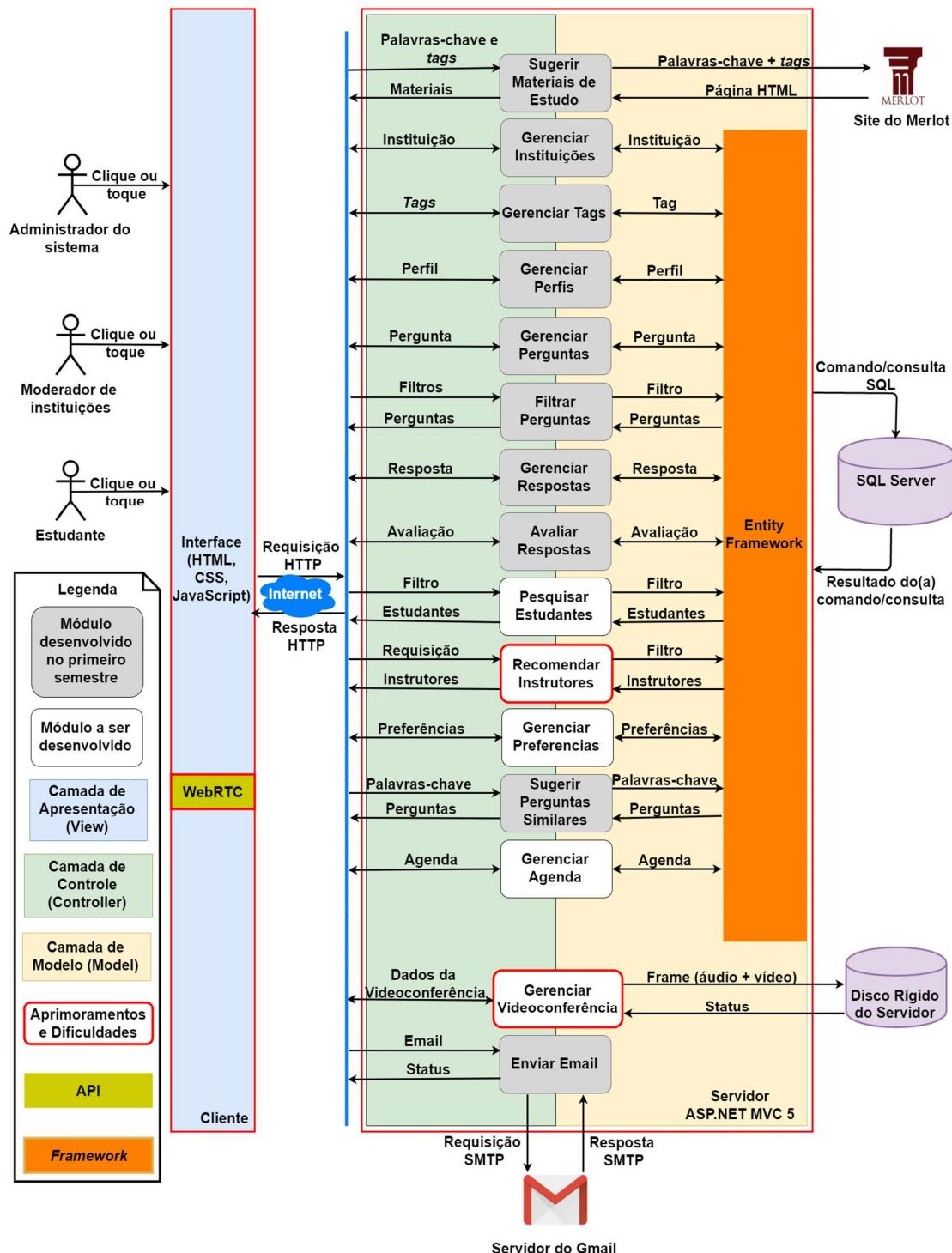


Figura 13. Primeira versão do diagrama de arquitetura do sistema CoLearn.

Apêndice B Burndowns do Desenvolvimento do Projeto

Tipo	Título	Progresso	Sprint	Horas Estimadas	Horas Trabalhadas
História	Como estudante/instrutor, quero realizar login no site	100	Sprint 1	12	12
Atividade	Criar tela de login	100	Sprint 1	3	3
Atividade	Controlar requisição de login	100	Sprint 1	5	5
Atividade	Controlar acesso do estudante	100	Sprint 1	4	4
História	Como estudante, quero me cadastrar no site	100	Sprint 1	12	14,5
Atividade	Criar tela um estudante se cadastrar no site	100	Sprint 1	1,5	3
Atividade	Controlar requisição de cadastro ou manutenção do perfil	100	Sprint 1	4	4
Atividade	Criar camada para persistir e recuperar dados do perfil do estudante	100	Sprint 1	4	4
Atividade	Criar tela para um estudante manter seu perfil	100	Sprint 1	1,5	2,5
Atividade	Criar serviço de e-mail	100	Sprint 1	1	1
Como administrador do site, quero criar, aprovar e/ou rejeitar cadastro de novas instituições					
História	Instituições	100	Sprint 1	15	20
Atividade	Criar tela de cadastros pendentes	100	Sprint 1	2	2
Atividade	Criar tela para cadastrar um moderador para uma instituição	100	Sprint 1	2	2
Atividade	Controlar requisição de aprovação ou rejeição de uma nova instituição	100	Sprint 1	2	4
Atividade	Criar camada para persistir ou recuperar cadastros de moderadores de instituições	100	Sprint 1	2	2
Atividade	Controlar requisição de cadastro de um novo moderador de uma instituição	100	Sprint 1	3	3
Atividade	Criar tela para o moderador de uma instituição cadastrar sua senha	100	Sprint 1	2	2
Atividade	Criar camada de acesso para os pedidos pendentes	100	Sprint 1	2	5
Como moderador de uma instituição, quero moderar novas disciplinas (tags) requisitadas pelos estudantes/instrutores					
História	pelos estudantes/instrutores	100	Sprint 2	27	30
Atividade	Criar tela para manter tags	100	Sprint 2	4	5
Atividade	Criar tela para mostrar tags não moderadas	100	Sprint 2	2	2
Atividade	Controlar requisição de criação ou deleção de uma tag	100	Sprint 2	7	7
Atividade	Criar camada de acesso para tags	100	Sprint 2	4	4
Atividade	Criar tela para aprovar, rejeitar ou juntar uma tag	100	Sprint 2	6	6
Atividade	Controlar requisição de merge de duas tags	100	Sprint 2	4	6
Como administrador do site, quando eu aceitar ou rejeitar o cadastro de uma instituição, quero enviar um e-mail para o autor desse cadastro					
História		100	Sprint 2	3	2,5
Atividade	Criar modal com uma caixa de texto a ser preenchida com a justificativa	100	Sprint 2	1,5	1,5
Atividade	Enviar e-mail notificando o autor do cadastro da instituição	100	Sprint 2	1,5	1
História	Como estudante, eu quero pedir o cadastro da minha instituição	100	Sprint 2	9	10
Atividade	Controlar requisição de nova instituição por um estudante	100	Sprint 2	4	4
Atividade	Criar tela para um estudante requisitar o cadastro de sua instituição	100	Sprint 2	2	3
Atividade	Notificar administrador do site	100	Sprint 2	3	3
Como estudante/instrutor, quero fazer uma pergunta associada a uma ou mais disciplinas					
História		100	Sprint 4	39	43
Atividade	Criar tela para um estudante criar sua pergunta	100	Sprint 3	6	6
Atividade	Controlar requisição de nova pergunta	100	Sprint 3	8	8
Atividade	Enviar tags novas para o moderador da instituição moderar	100	Sprint 3	2	2
Atividade	Criar tela para exibir materiais de estudos	100	Sprint 3	4	4
Atividade	Mostrar perguntas semelhantes	100	Sprint 3	6	6
Atividade	Criar tela para mostrar perguntas feitas pelos estudantes	100	Sprint 3	5	5
Atividade	Criar mecanismo para listar materiais de estudo	100	Sprint 4	9	12
História	Como instrutor, quero informar aos estudantes quais disciplinas eu posso ajudá-los	100	Sprint 4	12	11
Atividade	Criar tela para estudante informar as tags (disciplinas)	100	Sprint 4	2	2
Atividade	Controlar requisição de adição/deleção de tags de ajuda	100	Sprint 4	4	3
Atividade	Criar tela para mostrar perfil do estudante/instrutor	100	Sprint 4	4	4
Atividade	Persistir tags de ajuda	100	Sprint 4	2	2
História	Como estudante/instrutor, quero responder a perguntas (texto)	100	Sprint 4	15	15
Atividade	Criar tela para mostrar respostas feitas por outros estudantes	100	Sprint 4	4	4
Atividade	Criar campo para escrever nova resposta	100	Sprint 4	3	3
Atividade	Controlar requisição de nova resposta	100	Sprint 4	4	4
Atividade	Notificar autor da pergunta quando sua pergunta for respondida	100	Sprint 4	4	4
História	Como estudante/instrutor, quero avaliar uma resposta feita a uma pergunta (texto)	100	Sprint 4	12	12
Atividade	Customizar componente para que este tenha o padrão de layout do site	100	Sprint 4	2	2
Atividade	Controlar requisição de avaliação	100	Sprint 4	4	4
Atividade	Persistir avaliação	100	Sprint 4	3	3
Atividade	Adicionar componente para um estudante avaliar a questão	100	Sprint 4	3	3

Figura 14. Burndown do primeiro semestre.

Tipo	Título	Progresso	Sprint	Horas Estimadas	Horas Trabalhadas
História	Como um estudante, quero realizar uma videoconferência com um instrutor	100	Sprint 5	39	51
Atividade	Criar tela da videoconferência	100	Sprint 5	3	3
Atividade	Criar canvas para desenhos durante a videoconferência	100	Sprint 5	4	4
Atividade	Controlar inicio da videoconferência	100	Sprint 5	6	6
Atividade	Controlar a persistência dos dados da videoconferência no servidor de arquivos	100	Sprint 5	10	16
Atividade	Controlar a transmissão (WebRTC)	100	Sprint 5	10	10
Atividade	Controlar exibição do vídeo salvo	100	Sprint 6	6	12
História	Como um estudante/instrutor, quero informar as datas que estou disponível para videoconferências	100	Sprint 6	12	12
Atividade	Criar tela para mostrar os dias que um instrutor está disponível para videoconferências	0	Sprint 6	3	3
Atividade	Controlar requisição de alterações nos horários disponíveis para videoconferência	0	Sprint 6	3	3
Atividade	Persistir datas de disponibilidade	0	Sprint 6	2	2
Atividade	Criar tela para um instrutor controlar suas datas disponíveis para videoconferência	0	Sprint 6	4	4
História	Como estudante, quero agendar uma videoconferência com um instrutor	100	Sprint 6	15	13
Atividade	Criar view para agendar videoconferência	100	Sprint 6	5	4
Atividade	Controlar requisição de agendamento de videoconferência	100	Sprint 6	3	3
Atividade	Notificar estudante que um agendamento foi realizado em sua agenda	100	Sprint 6	4	3
Atividade	Persistir agendamento	100	Sprint 6	3	3
História	Como estudante/instrutor, quero avaliar uma resposta feita por videoconferência	100	Sprint 7	12	15
Atividade	Criar view para avaliar videoconferência	100	Sprint 7	5	6
Atividade	Controlar requisição de avaliação da videoconferência	100	Sprint 7	4	6
Atividade	Persistir avaliação	100	Sprint 7	3	3
História	Como um estudante, quero filtrar perguntas por título e/ou por tags	100	Sprint 8	39	39
Atividade	Criar partial view para usuário selecionar e aplicar filtros	100	Sprint 7	6	6
Atividade	Controlar requisição de filtros	100	Sprint 7	15	15
Atividade	Recuperar dados da base de acordo com os filtros	100	Sprint 8	6	6
Atividade	Aplicar filtros na tela de perguntas (filtrar perguntas)	100	Sprint 8	12	12
	Como um estudante, quero visualizar uma lista de instrutores recomendados para mim, de acordo com minhas preferências	100	Sprint 9	39	63
História	Criar tela para mostrar os estudantes recomendados	100	Sprint 10	6	7
Atividade	Filtrar perfis de estudantes (algoritmo + preferências)	100	Sprint 10	21	30
Atividade	Criar campos de busca de estudantes específicos na tela de recomendação	100	Sprint 10	3	3
Atividade	Recuperar estudantes de acordo com as informações da busca	100	Sprint 10	6	5
Atividade	Criar tela de preferências para a recomendação	100	Sprint 8	6	6
Atividade	Mantar preferências dos estudantes	100	Sprint 8	6	6
Atividade	Criar tela para administrador cadastrar preferências padrões	100	Sprint 9	6	6
	Como um estudante, quero configurar o número máximo de horas que um outro estudante pode agendar conigo para uma videoconferência	100	Sprint 9	9	9
Atividade	Mantar configuração	100	Sprint 9	3	3
Atividade	Criar tela para configuração	100	Sprint 9	3	3
Atividade	Controlar requisição de configuração	100	Sprint 9	3	3
História	Implementar melhorias	100	Sprint 11	24	21,5
	Permitir que apenas estudantes que responderam a uma pergunta sejam capazes de avaliar as outras respostas dessa pergunta	100	Sprint 11	6	5
Atividade	Criar novo item no meu de opções de uma tag pendente para abrir a pergunta associada a requisição	100	Sprint 11	3	3
Atividade	Atualizar tela inicial para que apenas as perguntas feitas pelo estudante autenticado sejam mostradas	100	Sprint 11	2	2
Atividade	Notificar estudante quando tags relacionadas a suas perguntas mudarem	100	Sprint 11	7	6,5
Atividade	Melhorar a usabilidade do menu do estudante na barra de navegação	100	Sprint 11	3	2
Atividade	Criar texto na tela de perfil de um estudante informando se este está online ou não	100	Sprint 11	3	3

Figura 15. Burndown do segundo semestre.

Apêndice C Configuração da Ferramenta de *Backup*

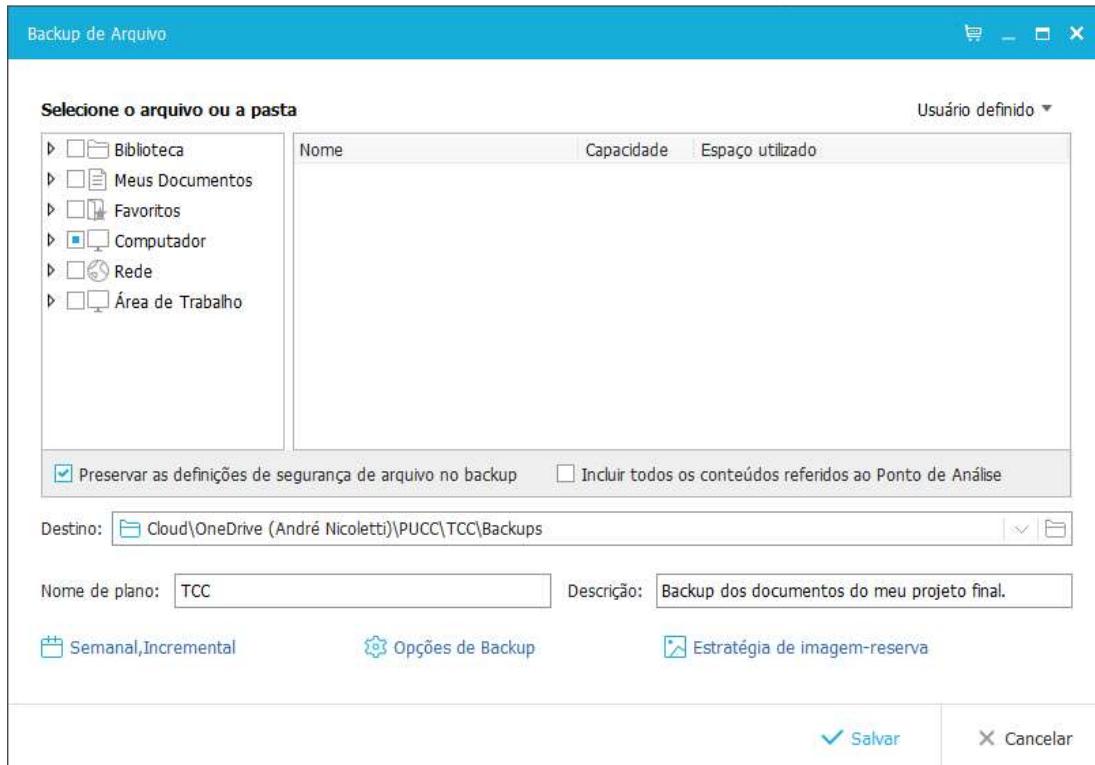


Figura 16. Configuração para a geração de *backup*.

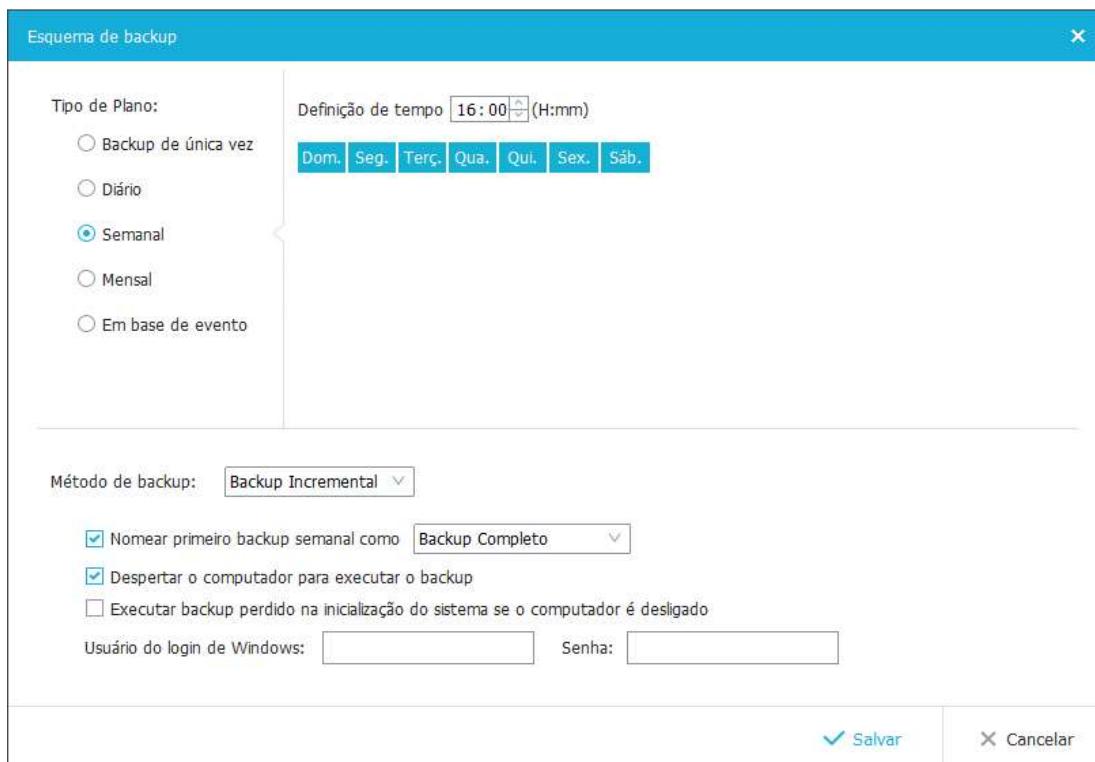


Figura 17. Configuração do esquema de *backup*.

Apêndice D Configuração do Controlador de Versões

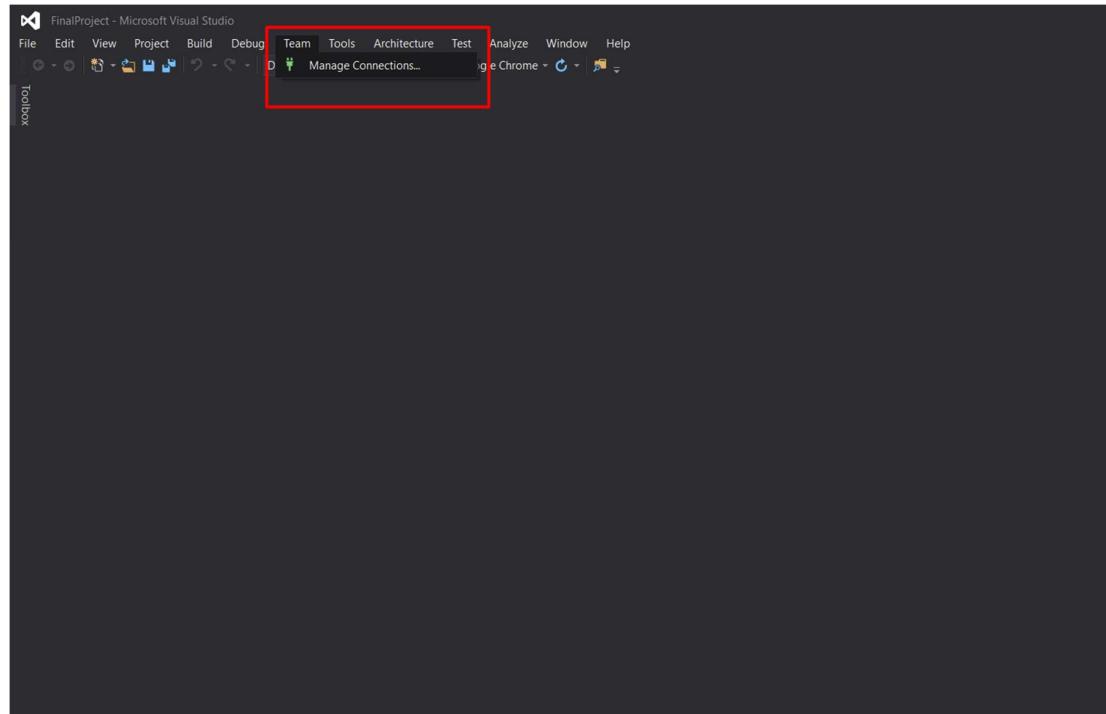


Figura 18. Primeira etapa da configuração do TFS.

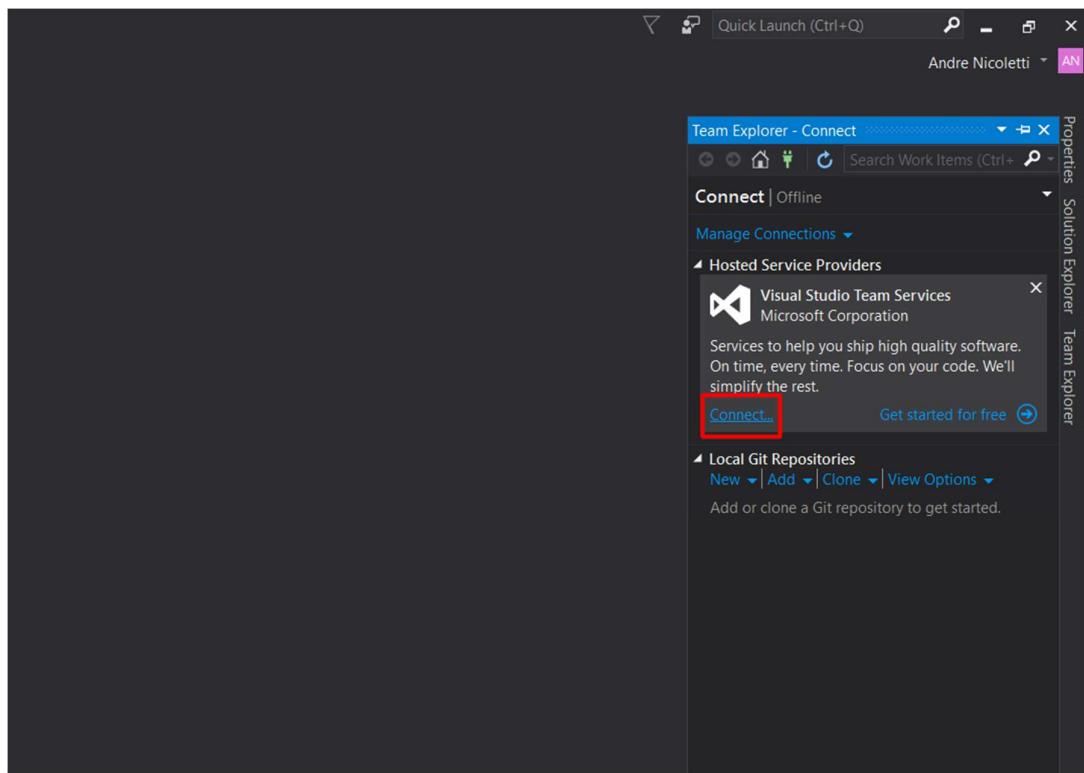


Figura 19. Segunda etapa da configuração do TFS.

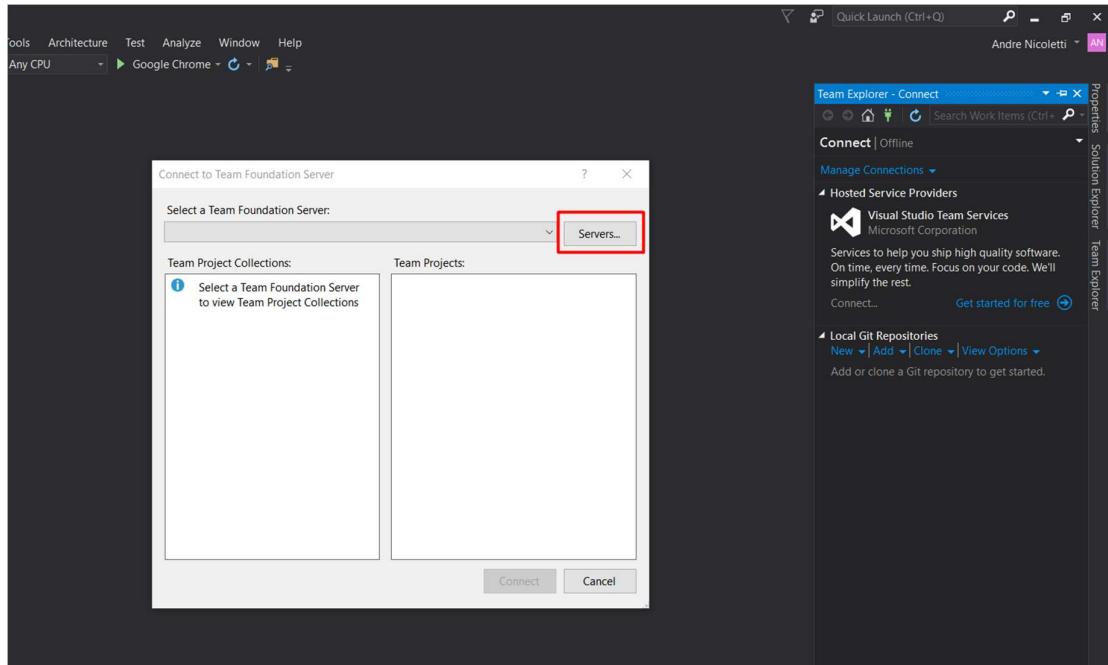


Figura 20. Terceira etapa da configuração do TFS.

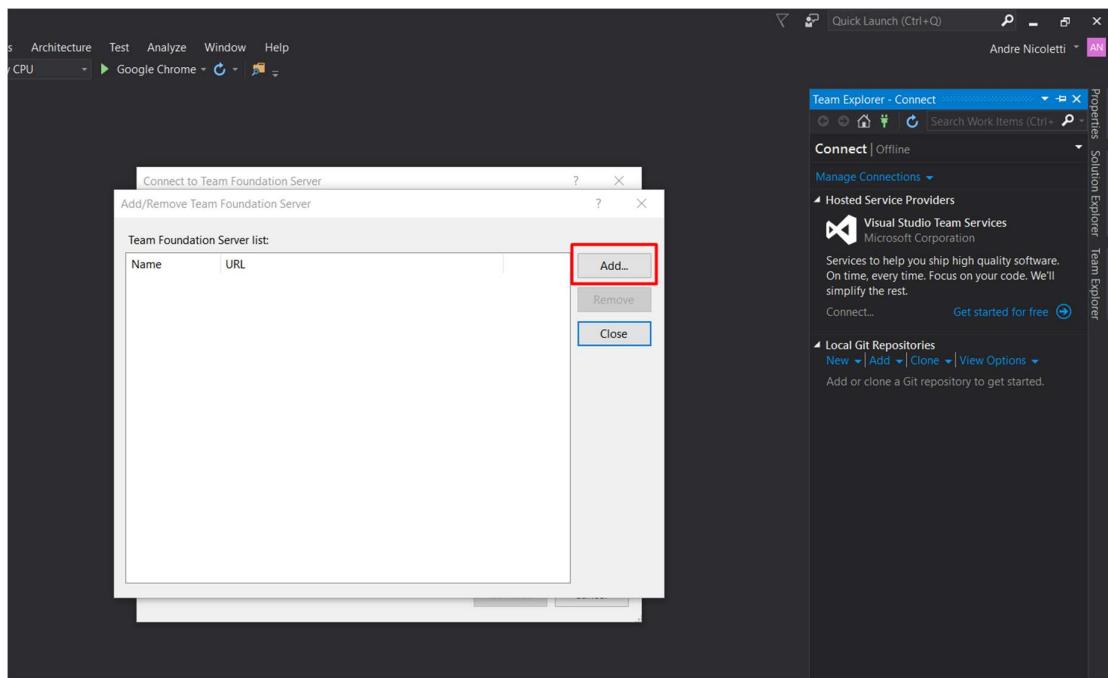


Figura 21. Quarta etapa da configuração do TFS.

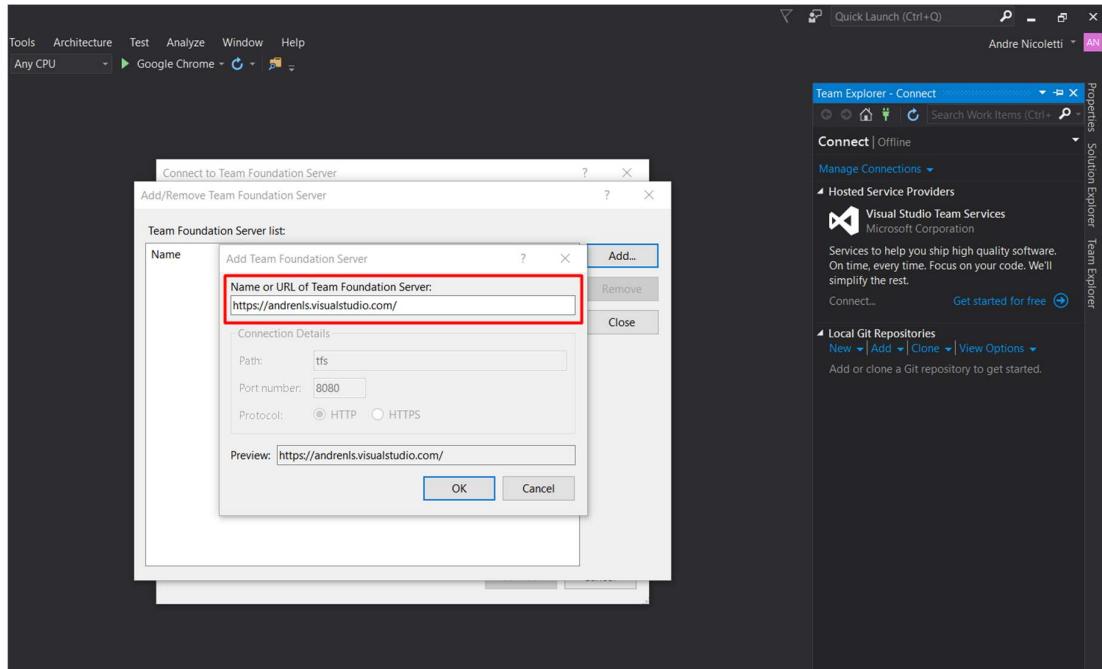


Figura 22. Quinta etapa da configuração do TFS.

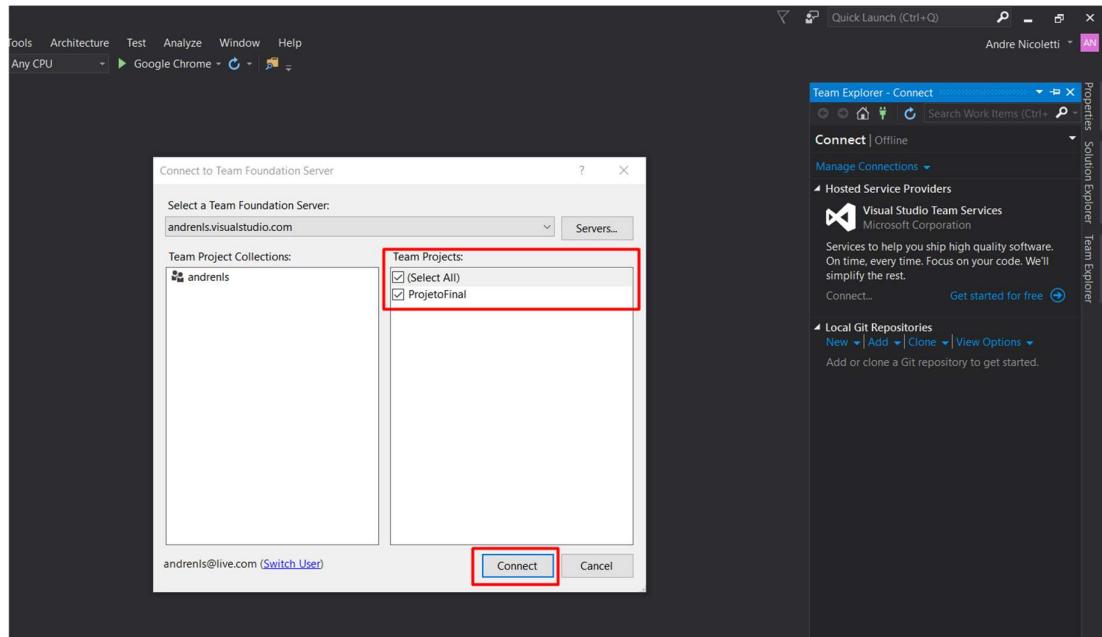


Figura 23. Sexta etapa da configuração do TFS.

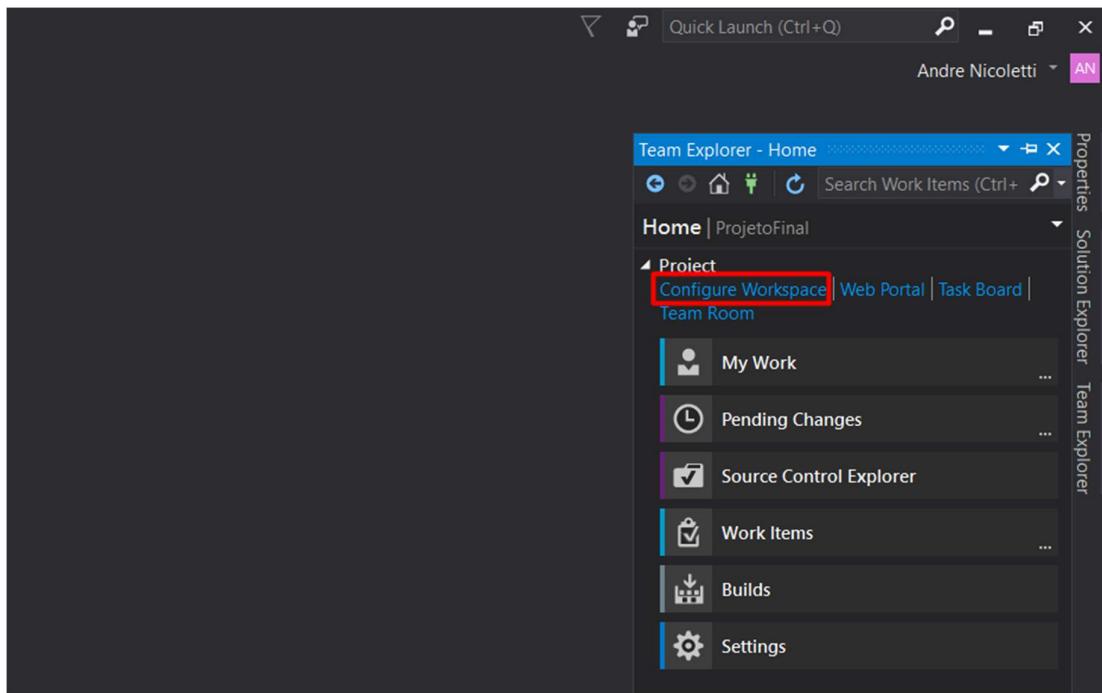


Figura 24. Sétima etapa da configuração do TFS.

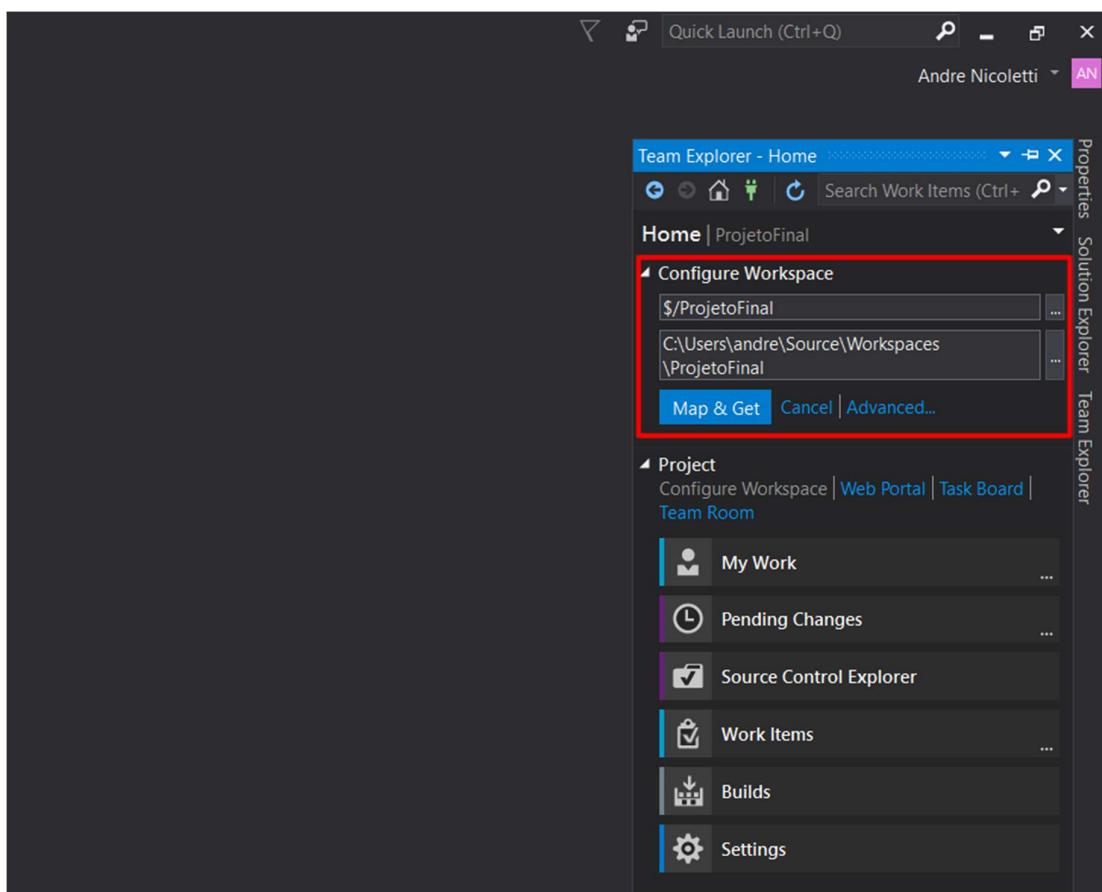


Figura 25. Oitava etapa da configuração do TFS.

Apêndice E Diagrama de Casos de Uso

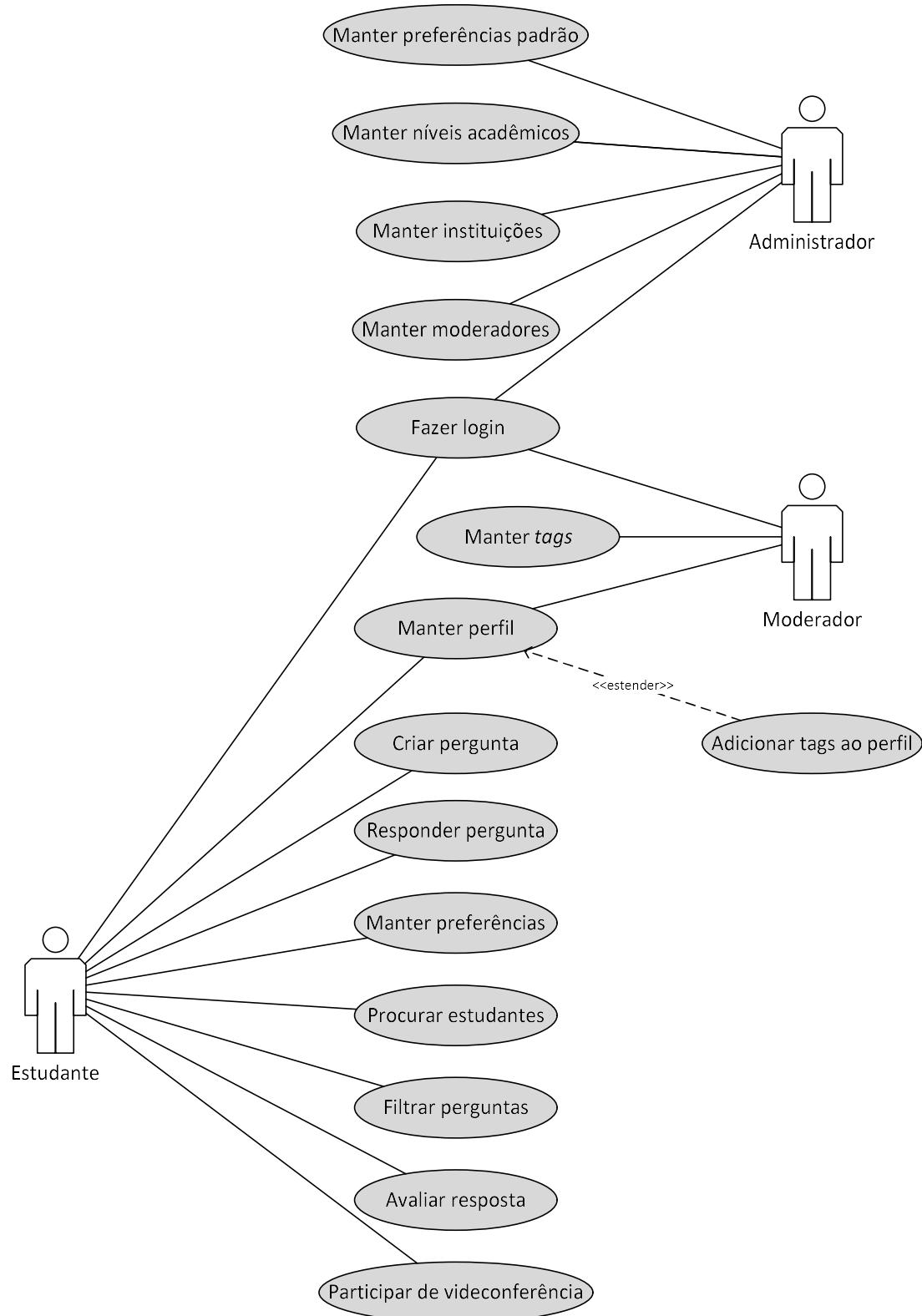


Figura 26. Diagrama de casos de uso do sistema CoLearn.

Apêndice F Visualizações da Agenda de um Usuário

The screenshot shows a weekly agenda view for the week of October 16 to 22, 2016. The interface includes a header with the CoLearn logo and user information (André). The main area displays a grid where each row represents a day of the week and each column represents a day of the month. The grid cells are color-coded: blue for booked slots, green for available slots, and yellow for specific events like 'Segurança de Redes' on Friday evening.

	Dom 16/10	Seg 17/10	Ter 18/10	Qua 19/10	Qui 20/10	Sex 21/10	Sáb 22/10
dia inteiro							
09							
10	10:00 - 12:30 Pilha TCP/IP (Larissa Ferreira)						
11							
12	12:30 - 14:00 Disponível			12:00 - 15:00 Disponível	12:00 - 15:00 Disponível	12:00 - 13:00 Disponível	12:00 - 15:00 Disponível
13						13:00 - 15:00 Criptografia em Redes de Computadores (Belchior Lauer)	
14							
15							
16							
17							
18					18:00 - 19:30 Segurança de Redes (Belchior Lauer)		
19							

Figura 27. Agenda sendo visualizada no modo semana

The screenshot shows a monthly agenda view for October 2016. The interface includes a header with the CoLearn logo and user information (André). The main area displays a grid where each row represents a week of the month and each column represents a day of the week. The grid cells are color-coded: blue for booked slots, green for available slots, and yellow for specific events like 'Segurança de Redes' on Friday evening.

Dom	Seg	Ter	Qua	Qui	Sex	Sáb
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
10 Pilha TCP/IP (Larissa Ferreira)	12 Disponível	12 Disponível	12 Disponível	12 Disponível	12 Disponível	12 Disponível
12:30 Disponível				18 Segurança de Redes (Belchior Lauer)	13 Criptografia em Redes de Computadores (Belchior Lauer)	
23	24	25	26	27	28	29
12 Disponível	12 Disponível	12 Disponível	12 Disponível	12 Disponível	12 Disponível	12 Disponível
30	31	1	2	3	4	5
12 Disponível	12 Disponível	12 Disponível	12 Disponível	12 Disponível	12 Disponível	12 Disponível

Figura 28. Agenda sendo visualizado no modo mês

Apêndice G Tela de Gerenciamento de Níveis Acadêmicos

The screenshot shows a web application interface for managing academic levels. At the top, there is a blue header bar with the text "CoLearn". On the right side of the header, there are icons for notifications (with a count of 1) and user profile, followed by the text "Administrador". Below the header, the main content area has a light gray background. The title "Níveis Acadêmicos" is centered at the top of the content area, with the subtitle "Gerencie os níveis disponíveis para os usuários." directly below it. A horizontal line separates this from the main content. Below the line, a note states: "Os níveis podem ser ordenados arrastando-os para cima ou para baixo. Quanto mais no topo um nível está, maior o grau acadêmico dos usuários que o utilizam." This is followed by a list of five academic levels, each in a separate row of a table-like structure:

1.	Pós-doutorado
2.	Doutorado
3.	Mestrado
4.	Especialização
5.	Graduação

At the bottom left of the content area, there is a button labeled "Formação". To its right is a small rectangular button with a plus sign (+). The entire interface is contained within a white rectangular frame.

Figura 29. Tela para gerenciamento de níveis acadêmicos.

Apêndice H Visualização dos Resultados de Pesquisas de Estudantes

The screenshot shows the CoLearn platform interface. At the top, there is a blue header bar with the CoLearn logo and user profile information. Below the header, a search form is displayed with the following fields:

- Pesquise por estudantes: A text input field containing "Sophia".
- Instituição: A dropdown menu set to "TODAS AS INSTITUIÇÕES".
- Nível: A dropdown menu set to "TODOS OS NÍVEIS".
- Avaliação mínima: A rating scale from 1 to 5 stars, with 5.0 selected.
- Buttons: "LIMPAR" and "PROCURAR" (Search).

Below the search form, the results are displayed under the heading "Resultados da pesquisa (1)". There is one result card for "Sophia Monteiro":

- Sophia Monteiro**
- sophia.monteiro@mail.com
- PUC-Campinas - Graduação
- Avaliação média da comunidade: 5.0 (0 avaliações)
- Pode ajudar em: computação, redes de computadores

Figura 30. Pesquisa feita por nome e avaliação mínima.

The screenshot shows the CoLearn platform interface. At the top, there is a blue header bar with the CoLearn logo and user profile information. Below the header, a search form is displayed with the following fields:

- Pesquise por estudantes: A text input field containing "Nome".
- Instituição: A dropdown menu set to "PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS (PUC-CAMPINAS)".
- Nível: A dropdown menu set to "DOUTORADO, PÓS-DOUTORADO".
- Avaliação mínima: A rating scale from 1 to 5 stars, with 3.0 selected.
- Buttons: "LIMPAR" and "PROCURAR" (Search).

Below the search form, the results are displayed under the heading "Resultados da pesquisa (2)". There are two result cards:

- Vitor Moraes**
- vitor.moraes@mail.com
- PUC-Campinas - Pós-doutorado
- Avaliação média da comunidade: 5.0 (0 avaliações)
- Pode ajudar em: redes de computadores, física, mecânica
- Lorenzo Carvalho**
- lorenzo.carvalho@mail.com
- PUC-Campinas - Doutorado
- Avaliação média da comunidade: 3.6 (0 avaliações)
- Pode ajudar em: sistemas operacionais, programação orientada a objetos, paradigmas de programação

Figura 31. Pesquisa feita por instituição, nível e avaliação mínima.

Apêndice I Algoritmo de Recomendação de Instrutores

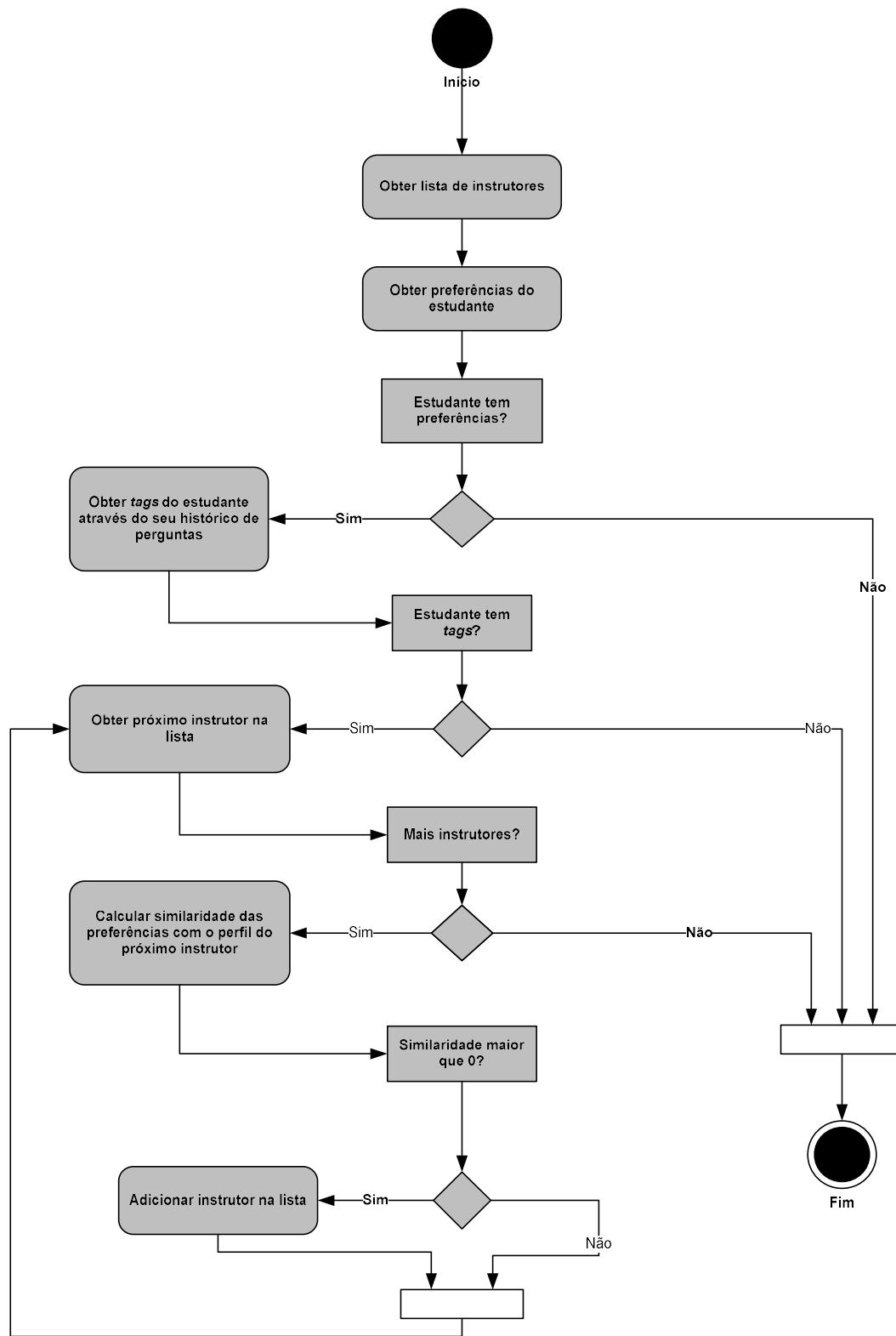


Figura 32. Diagrama de atividades do algoritmo de recomendação de instrutores.

Apêndice J Resultado de uma Recomendação de Instrutores

CoLearn

Instrutores recomendados para você

Estes instrutores foram recomendados para você de acordo com suas preferências

 Larissa Ferreira andrenlsl@gmail.com PUC-Campinas • Graduação	 Rodrigo Silveira teste@teste.com PUC-Campinas • Graduação	 Sophia Monteiro sophia.monteiro@mail.com PUC-Campinas • Graduação
Similaridade com seu perfil 81.81%	Similaridade com seu perfil 79.44%	Similaridade com seu perfil 75%
Avaliação média da comunidade ★★★★☆ 4.8 (14 avaliações)	Avaliação média da comunidade ★★★★☆ 4.4 (8 avaliações)	Avaliação média da comunidade ★★★★☆ 5 (3 avaliações)
Pode ajudar em computação sistemas operacionais	Pode ajudar em computação redes de computadores sistemas operacionais sistemas distribuídos	Pode ajudar em computação redes de computadores
 Sara Nicoletti eduarda@eduarda.com PUC-Campinas • Mestrado	 Valentina Souza valentina.souza@mail.com PUC-Campinas • Graduação	 Lorenzo Carvalho lorenzo.carvalho@mail.com PUC-Campinas • Doutorado
Similaridade com seu perfil 73.33%	Similaridade com seu perfil 66.39%	Similaridade com seu perfil 65.83%
Avaliação média da comunidade ★★★★☆ 4.8 (5 avaliações)	Avaliação média da comunidade ★★★★☆ 3.7 (3 avaliações)	Avaliação média da comunidade ★★★★☆ 3.6 (3 avaliações)
Pode ajudar em computação inteligência artificial	Pode ajudar em redes de computadores sistemas operacionais sistemas distribuídos	Pode ajudar em sistemas operacionais programação orientada a objetos paradigmas de programação
 Lucas Silva lucas@lucas.com PUC-Campinas • Graduação	 Isabella Pinto isabella.pinto@mail.com PUC-Campinas • Graduação	 Alice Mendes alice.mendes@mail.com PUC-Campinas • Graduação
Similaridade com seu perfil 50%	Similaridade com seu perfil 50%	Similaridade com seu perfil 41.67%
Avaliação média da comunidade ★★☆☆☆ 2 (1 avaliação)	Avaliação média da comunidade ☆☆☆☆ Nenhuma avaliação	Avaliação média da comunidade ☆☆☆☆ 1 (3 avaliações)
Pode ajudar em computação redes de computadores sistemas operacionais sistemas distribuídos	Pode ajudar em computação redes de computadores sistemas operacionais	Pode ajudar em computação redes de computadores
 João Pedro Fogaça joao.fogaca@mail.com PUC-Campinas • Pós-doutorado	 Maria Clara Ferreira maria.clara.ferreira@mail.com PUC-Campinas • Mestrado	 Pedro Lemos pedro.lemos@mail.com PUC-Campinas • Pós-doutorado
Similaridade com seu perfil 41.67%	Similaridade com seu perfil 41.67%	Similaridade com seu perfil 41.67%
Avaliação média da comunidade ★☆☆☆☆ 1 (2 avaliações)	Avaliação média da comunidade ★☆☆☆☆ 1 (3 avaliações)	Avaliação média da comunidade ★☆☆☆☆ Nenhuma avaliação
Pode ajudar em redes de computadores sistemas operacionais matemática física sistemas distribuídos	Pode ajudar em redes de computadores sistemas operacionais geometria sistemas distribuídos	Pode ajudar em computação inteligência artificial
 Gustavo Martins gustavo.martins@mail.com PUC-Campinas • Especialização	 Pedro Henrique da Cruz pedrohenrique.cruz@mail.com PUC-Campinas • Graduação	
Similaridade com seu perfil 41.67%	Similaridade com seu perfil 41.67%	
Avaliação média da comunidade ☆☆☆☆ Nenhuma avaliação	Avaliação média da comunidade ☆☆☆☆ Nenhuma avaliação	
Pode ajudar em redes de computadores sistemas operacionais	Pode ajudar em computação	

Figura 33. Instrutores recomendados a um estudante.

Apêndice K Dados Inseridos no Sistema para a Avaliação

Título: Qual a diferença entre um programa cliente e um programa servidor?

Descrição: Nenhuma

Tags: computação, redes de computadores

Resposta de Larissa Ferreira, 5 estrelas:

Geralmente, um programa de rede tem dois programas, cada um rodando em um hospedeiro diferente, e eles se comunicam. O programa que inicia a comunicação é denominado cliente. Normalmente, o programa cliente requisita e recebe serviços do programa servidor.

Resposta de Bruno Pereira, 4 estrelas:

Um programa cliente é aquele que faz requisições a um outro programa, que é denominado servidor. Exemplo: o aplicativo do Facebook que você utiliza no seu celular é um programa cliente. Este faz requisições para os servidores do Facebook, que processa a requisição e devolve uma resposta.

Título: O que é um protocolo hand-shaking?

Descrição: Nenhuma.

Tags: computação, redes de computadores

Resposta de André Nicoletti, 4,7 estrelas:

Um protocolo de apresentação é aquele que os dois lados da comunicação trocam pacotes de controle antes de trocar dados um com outro. O protocolo SMTP utiliza hand-shaking, enquanto HTTP não.

Resposta de Vitor Moraes, 5 estrelas:

Exemplo de como o protocolo funciona:

Cliente: Servidor, estou enviando a mensagem 100 (Número de sequência do cliente). Dá para sincronizar (SYN)?

Servidor: Claro, sincroniza a mensagem 200 (Número de sequência do servidor) que estou enviando (SYN). Prossiga com a mensagem 101 (ACK).

Cliente: Ok, estou enviando a mensagem 101. Prossiga com a mensagem 201 (ACK).

O cliente e o servidor, possuem números de sequência distintos, por este motivo faz-se necessária a sincronização em ambos os sentidos.

Feita a sincronização, começam a troca de pacotes com base em números de sequência, que tem o objetivo de enumerar os pacotes de cada um.

Resposta de Theo Teixeira, 1 estrela, ERRADA:

É um protocolo para um computador saber informações de rede do outro.

Título: Por que HTTP, FTP e SMTP rodam sobre TCP e não sobre UDP?

Descrição: Estava pensando se esses protocolos não ficariam mais leves e rápidos se rodassem sobre UDP e não TCP. Isso é verdade?

Tags: computação, redes de computadores

Resposta de André Nicoletti, 4,7 estrelas:

Esses protocolos necessitam que os dados da aplicação sejam recebidos em ordem e sem perdas. TCP suporta isso, enquanto UDP não. Portanto, por mais que o UDP seja mais leve, ele não seria adequado para esses protocolos por não ser confiável.

Resposta de Theo Teixeira, 1 estrela, ERRADA:

Por que o protocolo TCP é mais rápido que o UDP.

Título: Como a camada de rede sabe qual o protocolo da camada superior?

Descrição: Suponha que um hospedeiro A envie ao hospedeiro B um segmento TCP encapsulado em um datagrama IP. Quando B recebe o diagrama de A, como a camada de rede de B sabe que deve passar o segmento para TCP e não UDP ou outro protocolo?

Tags: computação, redes de computadores

Resposta de Bruno Pereira, 5 estrelas:

No cabeçalho IP há um campo, de 8 bits, que contém a informação para qual protocolo da camada de transporte o hospedeiro de destino deve passar o segmento.

Resposta de Rodrigo Silva, 3,5 estrelas:

Existe um campo no cabeçalho IP que identifica o protocolo de transporte utilizado na comunicação entre a fonte e o destino.

Título: Se um nó tem uma conexão com uma rede sem fio, este nó é móvel?

Descrição: Suponha que um usuário, portanto um laptop ande com ele por sua casa, e sempre acesse a Internet pelo mesmo ponto de acesso. Este usuário é móvel, em relação à rede? Explique.

Tags: computação, redes de computadores

Resposta de André Nicoletti, 5 estrelas:

Não. Um nó móvel é aquele que muda seu ponto de acesso de tempos em tempos (como uma rede celular 3G/4G). Como o usuário da pergunta está sempre acessando a Internet através do mesmo ponto de acesso, ele não é móvel.

Resposta de Valentina Souza, 1 estrela, ERRADA:

Sim, pois ele está se movendo, então consideramos o nó e o usuário como um moveis.

Título: Qual a principal diferença entre um sistema de chave simétrica e um sistema de chave pública?

Descrição: Nenhuma.

Tags: computação, criptografia

Resposta de Bruno Pereira, 4,3 estrelas:

A principal diferença entre esses sistemas é que no sistema de chaves simétricas, a fonte e o destino devem conhecer a mesma chave secreta. No sistema de chaves públicas, as chaves de encriptação e decriptação são diferentes. A chave de encriptação é conhecida por todos (por isso é pública), mas a chave de decriptação é conhecida apenas pelo destino (a chave é privada).

Título: Por que pelo menos uma parte dos manipuladores de interrupção dos computadores atuais é escrita em Assembly e não em uma linguagem de alto nível como C?

Descrição: Nenhuma.

Tags: computação, sistemas operacionais

Resposta de Larissa Ferreira, 5 estrelas:

Geralmente, linguagens mais alto nível, como C, não permitem acesso a certas instruções da CPU que são necessárias para manipular interrupções. Sem falar que esse tipo de processamento deve ser feito o mais rápido possível e,

normalmente, Assembly é mais rápido que outras linguagens de programação mais alto nível.

Resposta de Lorenzo Carvalho, 2,3 estrelas, ERRADA (parcialmente):

Uma implementação em Assembly executa mais rápido que uma implementação escrita em uma linguagem de alto nível, como é o caso da linguagem C.

Título: Quais serviços são oferecidos por um sistema operacional, que tornam o sistema de computação mais conveniente para os usuários?

Descrição: Nenhuma.

Tags: computação, sistemas operacionais

Resposta de Larissa Ferreira, 5 estrelas:

Gerenciamento de processos: Criação, fechamento, escalonamento, prioridades e comunicação entre eles; Gerenciamento da memória principal: Alocação, desalocação, proteção e abstração da memória virtual; Gerenciamento dos sistemas de I/O; Conexão em rede: Conexão com os dispositivos e implementação dos protocolos de rede; Sistema de proteção (autorização a usuários); Implementação de sistema de arquivos. Todos esses serviços ajudam muito o desenvolvimento de uma nova aplicação para um SO, pois o desenvolvedor não precisa se preocupar com detalhes de "baixo nível". Caso contrário, cada programa teria que ter seu próprio gerenciamento de memória, de rede etc.

Título: No contexto dos Sistemas Operacionais, qual o propósito de se ter chamadas de sistema?

Descrição: Nenhuma.

Tags: computação, sistemas operacionais

Resposta de Larissa Ferreira, 4 estrelas:

Chamadas de sistemas permitem que os processos de nível de usuário requisitem serviços ao sistema operacional, como manipulação de arquivos (open, read, write etc), serviços de rede etc.

Reposta de Lorenzo Carvalho, 5 estrelas:

Complementando a resposta da Larissa:

As chamadas de sistemas são funções (interfaces) usadas pelos aplicativos para solicitar a execução de algum serviço ao kernel do sistema operacional. Por isso,

as chamadas de sistemas são instruções com maior privilégio quando comparadas às outras instruções.

Com as chamadas de sistemas é possível, por exemplo, definir acesso a recursos de baixo nível como alocação de memória, periféricos e arquivos. Além disso, são as chamadas de sistemas que permitem a criação e a finalização de processos. Quando a execução de uma chamada de sistema é solicitada, o sistema operacional salva todo o contexto do processo (para continuar mais tarde de onde parou), verifica as permissões envolvidas no pedido e autoriza (se for o caso) o processador a executar o serviço solicitado.

Quando o processador termina a execução da chamada de sistema, o sistema operacional retorna o controle para o processo, colocando-o novamente na fila de processos prontos para a execução.

Título: Por que usar APIs ao invés de chamadas de sistemas?

Descrição: Nenhuma.

Tags: computação, sistemas operacionais

Resposta de Rodrigo Silveira, 4,8 estrelas:

- Uma API pode ser desenvolvida para compilar e rodar em diferentes sistemas (portabilidade). Chamadas de sistemas diferem de plataforma para plataforma;
- Se uma chamada de sistema for atualizada com uma nova versão, uma API que a utiliza pode ser atualizada para suportar essa mudança e quem utiliza a API vai poder usufruir dessa nova chamada de sistema, sem nem ao menos saber o que mudou no SO;
- Uma API pode suportar múltiplas versões de um SO. Se chamadas de sistemas forem utilizadas, o programa do usuário deve conter alguma lógica para identificar a versão do SO e utilizar a chamada correta, aumentando a complexidade da aplicação.

Resposta de Larissa Ferreira, 4,6 estrelas:

Complementando a resposta do Rodrigo:

- APIs são mais fáceis de trabalhar do que realizar a interface com o SO manualmente;

- APIs podem ter funções e ferramentas para realizar tarefas comuns com uma chamada simples que normalmente precisaria de várias linhas de código se fossem ser realizadas por meio das chamadas de sistemas.

Título: O que é condição de corrida?

Descrição: Uma condição de corrida pode gerar deadlocks?

Tags: sistemas operacionais

Resposta de Larissa Ferreira, 5 estrelas: Condição de corrida é uma situação em que dois ou mais processos leem e escrevem dados compartilhados e o resultado final depende da ordem de quem precisamente executa, e quando. Para evitar isso, podemos travar regiões do código (regiões críticas) que podem ser acessadas por um ou mais processos (ou threads).

Caso um processo bloqueie uma região e tente acessá-la sem antes desbloqueá-la, podemos ter uma situação de espera permanente, ou deadlock, onde o processo fica esperando que alguém desbloqueie a região crítica, mas todos estão travados esperando o desbloqueio.

Resposta de Valentina Souza, 5 estrelas:

É uma situação que pode ocorrer em todos os casos onde um determinado recurso computacional tem acesso concorrente. O melhor resumo é que a situação precisa contar que algo está em determinado estado e vai fazer algo pressupondo este estado, mas o estado muda por outra execução entre o tempo que você obteve o estado e o momento que vai executar algo nele. Em outras palavras é a situação onde o tempo que os eventos ocorrem podem influenciar sua execução.

Resposta de João Pedro Fogaça, 1 estrela, ERRADA:

Uma condição de corrida é quando sistemas não são executados devido a um ser mais prioritário que o outro. No final, ambos podem ficar indefinidamente esperando para serem executados, causando deadlock.

Título: Qual é a definição de Aprendizado de Máquina (Machine Learning)?

Descrição: Pelo o pouco conhecimento que tenho a respeito do assunto, aprendizagem de máquina é uma forma de uma máquina "aprender" a ter um comportamento a partir de dados pré-definidos.

Tags: computação, inteligência artificial

Resposta de Eduarda Alves Silva, 4,5 estrelas:

O termo se refere a um enorme conjunto de técnicas que visam construir sistemas computacionais cujo comportamento seja definido com base em dados existentes. Como o comportamento do sistema não seria diretamente programado, mas sim adaptado de algum "conhecimento" previamente adquirido, essa abordagem teria similaridade com a forma como animais (entre eles, nós humanos) aprendem com a experiência.

Resposta de Bruno Pereira, 4,1 estrelas:

Machine Learning ou Aprendizado de Máquina é um método de análise de dados que automatiza o desenvolvimento de modelos analíticos. Usando algoritmos que aprendem interativamente a partir de dados, o aprendizado de máquinas permite que os computadores encontrem insights ocultos sem serem explicitamente programados para procurar algo específico.

Resposta de Lucas Silva, 2 estrelas, **ERRADA**:

É o processo pelo qual uma máquina aprende a partir de dados pré-definidos.

Título: O que é Rede Neural?

Descrição: Qual seria a melhor definição para esta área de estudos da Inteligência Artificial?

Tags: computação, inteligência artificial

Resposta de Eduarda Alves Silva, 5 estrelas:

A Rede Neural Artificial estuda formas de modelar matematicamente o modo como o cérebro humano funciona, permitindo que uma máquina pense e aprenda do mesmo jeito que nós humanos fazemos - tornando as máquinas capazes de reconhecer fala, objetos e animais, assim como nós.

Resposta de Nicole Rezende, 3,8 estrelas:

Outra definição:

Redes Neurais Artificiais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência.

Título: Qual o papel de um *middleware* em um sistema distribuído?

Descrição: nenhuma

Tags: computação, sistemas distribuídos

Resposta de Rodrigo Silveira, 5 estrelas:

Um middleware tem o papel de aumentar a transparência de distribuição. Ou seja, um middleware visa melhorar a visão única que um sistema distribuído deve ter.

Por ser um software de conectividade, consiste de um conjunto de serviços disponíveis que permite que múltiplos processos, executando em uma ou mais máquinas, interajam através de uma rede.

Resposta de Valentina Souza, 4 estrelas:

Middleware é um programa de computador que faz a mediação entre um software e demais aplicações. É utilizado para mover ou transportar informações e dados entre programas de diferentes protocolos de comunicação, plataformas e dependências do sistema operacional.

Título: O que é transparência de distribuição?

Descrição: Alguém poderia me explicar o que é este conceito em sistemas distribuídos e dar exemplos?

Resposta de Valentina Souza, 4,8 estrelas:

Transparência de distribuição é o conceito de esconder os detalhes de distribuição de usuários e de aplicações.

Exemplos:

1. Transparência de acesso;
2. Transparência de localização;
3. Transparência de migração;
4. Transparência de realocação;
5. Transparência de replicação;
6. Transparência de concorrência;
7. Transparência de falhas;
8. Transparência de persistência.

Resposta de Maria Clara Ferreira, 1 estrela, **ERRADA**:

Transparência de distribuição é fazer com que um sistema distribuído não tenha que se importar com detalhes de distribuição.

Título: Transmissões simplex podem suportar tráfego half-duplex ou full-duplex?

Descrição: Nenhuma.

Tags: computação, redes de computadores

Resposta de Sophia Monteiro, 5 estrelas:

Não. Uma aparente exceção seria na utilização de múltiplos meios físicos combinados, como nos cabos de fibra ótica, onde cada interligação é baseada em um par, no entanto, este tipo de interligação se classificaria como half-duplex.

Resposta de Alice Mendes, 1 estrela, **ERRADA**:

Sim, pois transmissões simplex suportam envio e recebimento de dados simultaneamente.

Apêndice L Roteiro Aplicado ao Avaliador do Sistema

1. Acesse o endereço “<https://colearnproject.azurewebsites.net>” pelo navegador Web Google Chrome.
2. No campo “Email” digite “avaliador@avaliacao.com”, e no campo “Senha” digite “Pass#1”.
3. Quando a página inicial carregar, clique no botão “+”, localizado no canto superior direito da página, para criar uma pergunta.
4. Na caixa de texto “Digite sua pergunta” escreva “Qual a diferença entre um sistema hospedeiro e um sistema final?”.
5. Note que questões com títulos similares aparecem abaixo da caixa de texto;
6. No campo “Escolha suas tags”, adicione as tags “computação” e “redes de computadores”. Para adicionar uma tag clique no botão “+”, ao lado direito da caixa de texto.
7. Clique no botão “Perguntar!”.
8. Leia a resposta em texto que irá aparecer.
9. Clique no nome do *site*, localizado no canto superior esquerdo, para voltar à tela inicial.
10. Considerando as perguntas e respostas contidas no Apêndice K, navegue pelas perguntas e compare as respostas certas e erradas com as repostas contidas no sistema, observando que as respostas são ordenadas pelo valor de suas avaliações (melhores avaliadas primeiro), sendo as respostas erradas menos bem avaliadas que as corretas.
11. Agora, clique no ícone de preferências, ao lado esquerdo do ícone para criar uma pergunta (“+”).
12. Defina suas preferências e clique em “Salvar”.

13. Clique no botão de pesquisa de estudantes (ícone de lupa), na barra de navegação superior, para visualizar instrutores recomendados.
14. Como foi feita uma pergunta com as tags “computação” e “redes de computadores”, instrutores que podem ajudar nestas categorias serão listados, obedecendo as preferências definidas no passo 12.
15. Navegue pelos instrutores recomendados, observando as características de cada um. Considere o documento anexado.
16. Volte à tela inicial e responda, certo e errado, questões de sua preferência. Para responder a uma questão, clique nela e digite a resposta na caixa de texto localizada abaixo das tags da questão.
17. Como o sistema não permite que um usuário avalie as próprias respostas, clique no nome do perfil autenticado, no canto superior direito da página e depois clique em “Sair”.
18. Autentique-se novamente, mas desta vez utilize o endereço de *email* “avaliador2@avaliacao.com”. A senha é a mesma usada no passo 3 (“Pass#1”).
19. Procure pelas perguntas respondidas no passo 16 e avalie as respostas criadas por você, clicando nas estrelas abaixo de cada resposta. Observação: uma nota mais alta significa uma resposta boa e correta, enquanto que uma avaliação baixa indica que a resposta não foi boa ou está errada.
20. Após avaliar as respostas de uma questão, recarregue a página para que as respostas sejam ordenadas de acordo com a avaliação dada, e observe como as suas contribuições foram tratadas pelo sistema.
21. Responda ao questionário presente no Apêndice M.

Apêndice M Questionário Aplicado ao Avaliador do Sistema

O questionário a seguir busca fornecer dados para que se possa analisar se o objetivo, que é melhorar o conhecimento dos estudantes, foi ou não entendido.

- I. Considerando sua experiência nesta avaliação, o sistema permite que estudantes melhorem seus conhecimentos em assuntos de seus interesses? Justifique.
- II. A interação entre estudantes e instrutores através de perguntas e respostas favorece a melhora de conhecimento dos estudantes? Justifique.
- III. Segundo as respostas dadas pelos instrutores, o sistema ajuda um estudante a buscar por bons instrutores, que possam ajudá-lo a melhorar seus conhecimentos em assuntos que tenha demonstrado ter dúvidas? Justifique.