

UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE CIÊNCIA E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

THAYNARA ALEXANDRE CARDOSO

**SACPD - SISTEMA COLABORATIVO PARA COMPARTILHAMENTO DE
PRÁTICAS DIDÁTICAS EM ENGENHARIA DE SOFTWARE**

Rio das Ostras
2018

THAYNARA ALEXANDRE CARDOSO

**SACPD - SISTEMA COLABORATIVO PARA COMPARTILHAMENTO DE
PRÁTICAS DIDÁTICAS EM ENGENHARIA DE SOFTWARE**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação do Instituto de Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Bacharel. Área de concentração: Engenharia de Software.

Orientadora:

D.Sc. ADRIANA PEREIRA DE MEDEIROS

Rio das Ostras
2018

Ficha catalográfica automática - SDC/BRO
Gerada com informações fornecidas pelo autor

C268s Cardoso, Thaynara Alexandre
SACPD - Sistema colaborativo para compartilhamento de
práticas didáticas em engenharia de software / Thaynara
Alexandre Cardoso ; Adriana Pereira de Medeiros, orientador.
Niterói, 2018.
69 f. : il.

Trabalho de Conclusão de Curso (Graduação em Ciência da
Computação)-Universidade Federal Fluminense, Instituto de
Ciência e Tecnologia, Rio das Ostras, 2018.

1. Engenharia de software. 2. Práticas didáticas. 3.
Ensino. 4. Produção intelectual. I. Medeiros, Adriana
Pereira de, orientador. II. Universidade Federal Fluminense.
Instituto de Ciência e Tecnologia. III. Título.

CDD -

THAYNARA ALEXANDRE CARDOSO

**SACPD - SISTEMA COLABORATIVO PARA COMPARTILHAMENTO DE
PRÁTICAS DIDÁTICAS EM ENGENHARIA DE SOFTWARE**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação do Instituto de Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Bacharel. Área de concentração: Engenharia de Software.

Aprovada em 12 de dezembro de 2018.

BANCA EXAMINADORA

Prof. D. Sc. ADRIANA PEREIRA DE MEDEIROS – Orientadora
UFF

Prof. D. Sc. CARLOS BAZILIO MARTINS
UFF

Prof. M. Sc. EDUARDO MARQUES
UFF

Rio das Ostras
2018

AGRADECIMENTOS

Agradeço primeiramente à Deus, que em todas as suas formas e manifestações, fez com que eu chegasse até aqui.

Aos meus pais que, sempre com seu carinho e amor incondicionais, fizeram seu melhor rompendo todas as barreiras para que eu pudesse chegar onde cheguei. Certamente eu não poderia ter sido presenteada com pais mais incríveis.

Aos meus irmãos Viviane, Bruno e Mayara que sempre me incentivaram a seguir meus sonhos, que em momento nenhum deixaram de acreditar em mim e me motivam diariamente para continuar seguindo e realizando meus sonhos.

À minha família, Yan, Liz e Igor, Reginaldo Mosca e Joanna Misurelli que com suas formas únicas de ser e se preocuparem e expressarem seu afeto, sempre demonstrando cuidado quando precisei Minha gratidão..

Aos amigos e irmãos que a vida me presenteou durante minha caminhada durante a graduação, Luis Felipe Fernandes, Lisandra Almeida, Lucas Maximiliano, Marilia Almeida, Maria Carolina Dalmazo, Marlon Pierre, Sara Sotte, Thiago Bulhosa e tantos que passaram por mim neste tempo e me ajudaram a tornar a caminhada mais leve com colo, ouvidos, sorrisos e algumas lágrimas. Sem os quais seria muito mais difícil chegar até aqui.

Agradeço à minha orientadora, Adriana Medeiros, por acreditar em mim, e desde o tempo de monitoria dedicar tempo e esforço para me auxiliar o meu desenvolvimento acadêmico e profissional, revisando e ajudando neste trabalho.

RESUMO

Engenharia de software é uma disciplina difícil de ser ensinada para os alunos, pois tem uma grande quantidade de conteúdo teórico, poucas propostas inovativas disponibilizadas e muitos níveis de abstração necessários. Aplicar o conteúdo teórico para produzir as especificações e modelos de um software é um dos desafios que o aluno encontra durante seu curso. Além disso, o uso de aulas expositivas torna o ensino desse conteúdo cansativo e pouco efetivo. A partir da observação dessas nuances no aprendizado nasce esse trabalho, com a proposta de uma estratégia de ensino a partir de práticas didáticas e a então, desenvolver um sistema que apoia o compartilhamento dessas práticas didáticas e seus respectivos materiais para auxiliar professores de ES a tornar suas aulas mais dinâmicas e incentivar os alunos a participar ativamente da construção do seu conhecimento.

Palavra chave: Engenharia de Software; Ensino; Práticas Didáticas

ABSTRACT

Software engineering is hard to teach students, because many levels of abstraction are required. In addition to, it has a large amount of theoretical resources but a small innovations content are proposed. Applying the theoretical content to produce specifications and software models in practice is the first challenge that students encounter during their course. Moreover, the use of expository classes makes teaching content tiring and small effective. Watching these problems at the learning process, this work proposes to support software engineering teachers to try solve it. With a proposed strategy of teaching using didactic practice this system was developed to support the sharing of these didactic practice and the own didacts resources to support the software engineering teachers to turn the dynamic classes and to encourage students to actively participate in building their knowledge.

Keywords: Software Engineering; Teach; Didactic practices.

LISTA DE FIGURAS

Figura 1 – Diagrama de Casos de Uso do sistema	28
Figura 2 - Diagrama de classes de análise de domínio.....	39
Figura 3 - Modelo entidade relacionamento dos dados.....	40
Figura 4a - Diagrama de classes de projeto – Controladores e Interface (camadas <i>control</i> e <i>view</i>)	42
Figura 4b - Diagrama de classes de projeto – Cclasses de negócio (camadas <i>control</i> e <i>view</i>).....	42
Figura 5 – Tela exibindo Listagem (vazia) de roteiros criados pelo professor “Teste_Usuario”	50
Figura 6 - Tela para criação de novo roteiro	51
Figura 7 - Tela de listagem de roteiros atualizada.....	51
Figura 8 - Tela de criação da prática didática.....	52
Figura 9 - Tela de criação de atividades	53
Figura 10 - Tela de roteiro criado	54
Figura 11 - Todos roteiros criados pelo usuário " Usuario2_Comum"	55
Figura 12 - Tela listagem de roteiros públicos	55
Figura 13 - Tela do usuário "Administrador" - Gerenciar Instituições	56
Figura 14 - Tela de editar uma instituição	56
Figura 15 - Tela de listagem de instituições (atualizada)	57
Figura 16 - Tela com listagem de instituições atualizada.....	57
Figura 17 - Tela de Gerenciar Usuários.....	58
Figura 18 - Tela de edição dos dados de um usuário.....	58
Figura 19 - Tela de listagem de usuário atualizada	59

LISTA DE QUADROS

Quadro 1 - Exemplo da estrutura do Guia e do roteiro	19
Quadro 2 - Exemplo de uma Prática Didática	21
Quadro 3 - Exemplo de roteiro específico para atividade complementar	24
Quadro 4 - Requisitos funcionais do sistema	26
Quadro 5 - Regras de negócio do sistema	27
Quadro 6 - Descrição de caso de uso "Criar Roteiro"	29
Quadro 7 - Operação da classe "RoteiroModel" para carregar os dados de um roteiro já cadastrado	46
Quadro 8 - Classes da camada modelo do sistema	46
Quadro 9 - Estrutura das classes de negócio "AtividadeModel.php" e "PraticaDidaticaModel.php"	47
Quadro 10 - Estrutura da classe de negócio "RoteiroModel.php"	47
Quadro 11 - Trecho de código do controller "PraticaDidaticaController.php"	48
Quadro 12 - Trecho de código da view "listarPraticasDidaticas.phtml"	49

SUMÁRIO

1. INTRODUÇÃO	12
1.1. MOTIVAÇÃO	13
1.2. OBJETIVOS	14
1.3. JUSTIFICATIVA	14
2. REFERENCIAL TEÓRICO	15
2.1. SISTEMAS COLABORATIVOS	15
2.2. ENGENHARIA DE SOFTWARE	16
3. GUIA DE APOIO AO ENSINO DE ES	18
3.1. ROTEIROS	18
3.2. PRÁTICAS DIDÁTICAS	21
3.3. ATIVIDADE COMPLEMENTAR	23
4. O SACPD - SISTEMA DE APOIO AO COMPARTILHAMENTO DE PRÁTICAS DIDÁTICAS	25
4.1. DESCRIÇÃO DO SISTEMA	25
4.2. ESPECIFICAÇÃO E MODELAGEM	26
4.3. TECNOLOGIAS E FERRAMENTAS UTILIZADAS	44
4.4. IMPLEMENTAÇÃO	45
5. AVALIAÇÃO DO SISTEMA	50
5.1. CENÁRIO DE USO 1	50
5.2. CENÁRIO DE USO 2	55
5.3. CENÁRIO DE USO 3	55
5.4. CENÁRIO DE USO 4	57
5.5. CASOS DE TESTE	59
6. TRABALHOS RELACIONADOS	60
7. CONCLUSÕES E TRABALHOS FUTUROS	61
REFERÊNCIAS BIBLIOGRÁFICAS	62
APÊNDICE A – CASOS DE TESTES E RESULTADOS	64
1.1. Criação de um novo roteiro	64
1.2. Cadastro de uma nova prática didática	64
1.3. Cadastro de uma nova atividade	65
1.4. Realizar download de material de apoio de uma atividade	66
1.5. Cadastro de uma nova instituição	66
1.6. Editar uma instituição já cadastrada	67

1.7.	Excluir uma instituição já cadastrada	67
1.8.	Criar um usuário administrador	67
1.9.	Editar os dados de um usuário	68
1.10.	Excluir um usuário	68
1.11.	Cadastro de uma nova área de atuação	68
1.12.	Editar uma área de atuação já cadastrada	69
1.13.	Excluir uma área de atuação já cadastrada	69

1. INTRODUÇÃO

Mudanças nas práticas de ensino, sejam elas em aspectos de avaliação ou reprodução de conhecimento, são importantes para a evolução do processo de ensino e aprendizagem. Essas mudanças devem privilegiar a construção coletiva do conhecimento, tendo a tecnologia como mediadora, na qual o professor tem o papel de orientar e intermediar esta construção (ENRICONE, 2004). Um fator determinante para a necessidade dessas mudanças foi o surgimento da chamada Geração Y, formada por pessoas que nasceram a partir de 1978 e cresceram em contato com as tecnologias de informação (VELOSO *et al.*, 2008). Esta geração possui grande capacidade de trabalhar em rede e, devido à internet, tem acesso a um grande volume de informação à sua disposição. Assim, já chega à escola conectada com o mundo e desafia diariamente as estratégias pedagógicas, pois já construiu diferentes formas de pensar e de aprender (CASTANHA e CASTRO, 2010). Esse desafio também é percebido no ensino superior, no qual ela é preparada para atuar profissionalmente em um novo paradigma econômico e produtivo, onde o fator mais importante é o uso intensivo de conhecimento e informação (BERNHEIM e CHAUI, 2008).

Como uma das propostas da universidade é preparar seus alunos para entrar no mercado de trabalho, ela deve então propor técnicas que alcancem os diferentes tipos de alunos tendo sempre que revisar suas técnicas de ensino. Engenharia de software trata-se de um dos principais assuntos nos currículos de graduação em Computação e Sistemas de Informação. A qualidade do ensino em Engenharia de Software contribui para a melhoria dos profissionais que trabalham com desenvolvimento de software na indústria (SILVA *et al.* 2011). No entanto, o ensino de Engenharia de Software tradicional, ou seja, focado em aulas puramente expositivas, deve ser transformado para melhor preparar os alunos para as atividades práticas que serão exercidas no mercado de trabalho (BAKER *et al.*, 2005). O professor é o principal responsável pelo processo de ensino e cabe a ele escolher os recursos didáticos mais apropriados para incentivar a participação dos alunos nas aulas, gerenciando suas expectativas e habilidades (PRIKLADNICKI *et al.* 2009). Esta não é uma tarefa simples, pois o professor precisa ir além do formato de aula expositiva e criar novas técnicas de ensino que incentivem o aluno a participar ativamente da construção do seu conhecimento.

Este trabalho propõe um sistema para apoiar professores a compartilhar práticas didáticas que possam tornar o ensino de Engenharia de Software mais prático e dinâmico. Este sistema tem como base um Guia criado para apoiar o ensino de modelagem de software contendo sugestões de práticas didáticas que permitam ao aluno ter maior autonomia sobre a construção do seu conhecimento.

1.1. MOTIVAÇÃO

Ao se observar as dificuldades que os alunos têm em acompanhar aulas expositivas e de pouca interação como é o modelo há tempos utilizado no ensino, em um contexto onde estão todos conectados com grande quantidade de informação presente na rede, surge o desafio de repensar a forma como o professor deve passar o conteúdo da sua disciplina. O professor deixa de estar na posição de ser apenas a pessoa que vai transmitir o conhecimento e passa a se colocar na posição de guiar o aluno na descoberta do próprio conhecimento, levantando questões e motivações para que os alunos se aprofundem nos temas por conta própria através de pesquisa em grupo, estímulo à dúvidas dinâmicas.

Assim, foi criado um guia para auxiliar o professor na elaboração de aulas de engenharia de software, área que apesar de ter algumas propostas de ensino diferenciadas disponibilizadas na web, suas práticas nem sempre são de fácil acesso ou reuso. Desta forma, este guia tem como principal estratégia a utilização de práticas didáticas trabalhando muitas vezes com analogias e metáforas quanto ao tema abordado na aula e os elementos presentes na prática.

Após a utilização das práticas contidas no guia em sala de aula, foram levantadas considerações a respeito desta abordagem, como observações, limitações, e uma forma de evoluir a ideia e tornar mais democrático o acesso às práticas e possibilitar o compartilhamento desta prática entre professores, educadores e facilitadores. Sendo esta evolução a maior motivação para que este trabalho tenha sido realizado.

O sistema visa possibilitar a troca de experiências entre a comunidade acadêmica, isto é, apoiar os professores na criação dessas práticas didáticas, disponibilizar essas práticas para consulta de outros professores, consultar práticas de outros professores e reutilizar estas práticas, facilitando a aplicação em diferentes realidades de cursos, instituições e turmas.

1.2. OBJETIVOS

Este trabalho tem como objetivo documentar e descrever uma solução que permita professores de Engenharia de Software criarem e disponibilizarem suas práticas didáticas para outros professores de forma que elas possam ser utilizadas para compor diferentes roteiros de aula.

1.3. JUSTIFICATIVA

O sistema proposto neste trabalho visa expandir e conectar a rede de professores e educadores. Tratando de um trabalho escalável, isto é, podendo ser evoluído a se tornar uma plataforma de compartilhamento de práticas didáticas de quaisquer áreas e domínios, possui um significativo papel no processo de desenvolvimento do ensino propondo uma nova abordagem que incentiva a construção do próprio conhecimento.

Para a área de Engenharia de Software, pouco material de apoio foi encontrado disponibilizado na web com propostas de aulas atuais, dinâmicas e práticas. Do pouco material encontrado, em sua grande maioria trata-se de propor a documentação de um sistema para um domínio específico. O sistema tem a visão de atuar como um grande repositório de ideias de práticas independentes a serem desenvolvidas em sala dando dinamismo e possibilitando a mudança nos paradigmas de ensino.

2. REFERENCIAL TEÓRICO

Por se tratar de um sistema colaborativo no domínio de Engenharia de Software, as definições desses dois tópicos foram levantadas e contextualizadas de acordo com o objetivo deste trabalho.

2.1. SISTEMAS COLABORATIVOS

Sistemas Colaborativos são ferramentas de software utilizadas em redes de computadores para facilitar a execução de trabalhos em grupos. Essas ferramentas devem ser especializadas o bastante, a fim de oferecer aos seus usuários formas de interação, facilitando o controle, a coordenação, a colaboração e a comunicação entre as partes envolvidas que compõe o grupo, tanto no mesmo local, como em locais geograficamente diferentes e que as formas de interação aconteçam tanto ao mesmo tempo ou em tempos diferentes. Percebe-se com isso que o objetivo dos Sistemas Colaborativos é diminuir as barreiras impostas pelo espaço físico e o tempo (CAMARGO, KHOURI, GIAROLA, 2005).

Existem vários termos para designar Sistemas Colaborativos, porém a idéia principal ou objetivo desses sistemas continuam sendo os mesmos, que é o suporte e a promoção da colaboração (OLIVEIRA, 2016).

Segundo Colleman (1997), existem sete taxionomias para sistemas colaborativos, isto é, ferramentas que utilizam do conceito de sistemas colaborativos com finalidades diferentes e plataformas diferentes. São elas:

- sistemas colaborativos de gerenciamento de conteúdo: ferramentas para publicação automatizada com a participação de diversas pessoas e grupos na elaboração do conteúdo;
- sistemas colaborativos de gestão do conhecimento: ferramentas de armazenamento, indexação, avaliação e distribuição de conhecimento tácito e explícito;
- real time collaboration tools (RTC) (áudio/vídeo/data conferencing) : ferramentas de colaboração síncronas que usam áudio, vídeo e dados;

- virtual Team Tools (DPM, virtual team and process-oriented tools) : ferramentas para grupos de trabalho. Dividem-se em três classes, a classe de gerenciamento distribuído de projetos, a classe de local de trabalho virtual e a classe de processos e workflow
- customer resource management (CRM) : ferramentas para auxílio a processos de venda e atendimento a clientes;
- portais e comunidades on-line : ferramentas para comunidade virtuais para troca de informações e idéias;
- ferramentas e infra-estrutura para colaboração Wireless: ferramentas para mensagens em dispositivos wireless. Normalmente se integram com as demais soluções de colaboração.

Segundo as definições dadas, este trabalho encaixa-se como um sistema colaborativo de gestão de conhecimento podendo ter características de outros tipos de sistemas colaborativos também.

De acordo com Tijiboy et al. (1999), as ferramentas de colaboração (sistemas colaborativos) são classificadas de acordo com o lugar das interações (presenciais ou à distância) e o tempo (síncronas ou assíncronas). Ferramentas síncronas são aquelas que requerem tempo de resposta imediato. Já as ferramentas assíncronas não necessitam de um tempo de resposta curto ou imediato que é justamente a categoria a qual este trabalho se encaixa.

2.2. ENGENHARIA DE SOFTWARE

Segundo (IEEE Computer Society, 2004) define Engenharia de Software como a aplicação de abordagens sistemáticas, disciplinadas e quantificáveis ao desenvolvimento, operação e manutenção de software, além do estudo dessas abordagens.

Já para Sommerville (2011), engenharia de software é uma disciplina de engenharia cujo foco está em todos os aspectos da produção de software, desde os estágios iniciais da especificação do sistema até sua manutenção, quando o sistema já está sendo usado. Os aspectos de produção de software incluem atividades como gerenciamento de projeto de software e desenvolvimento de ferramentas, métodos e teorias para apoiar a produção de software.

Um software não é apenas um programa ou um conjunto de programas; ele inclui também a documentação, que é composta por diferentes artefatos dependendo do modelo de desenvolvimento de software adotado.

Segundo Sommerville (2011), um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software. Essas atividades podem envolver o desenvolvimento de software a partir do zero em uma linguagem padrão de programação. Existem muitos processos de software diferentes, mas todos devem incluir quatro atividades fundamentais para a engenharia de software: 1. Especificação de software: A funcionalidade do software e as restrições a seu funcionamento devem ser definidas. 2. Projeto e implementação de software: O software deve ser produzido para atender às especificações. 3. Validação de software: O software deve ser validado para garantir que atenda às demandas do cliente. 4. Evolução de software: O software deve evoluir para atender às necessidades de mudança dos clientes. Em cada etapa, um conjunto de artefatos é produzido.

A engenharia de software também se ocupa em tomar decisões gerenciais de projeto, como gerenciamento de riscos, gerenciamento de pessoas, recursos, escopos, testes e outras variáveis que atingem ou são atingidas direta ou indiretamente pelo projeto.

3. GUIA DE APOIO AO ENSINO DE ES

O Guia utilizado neste trabalho é o resultado de um projeto de monitoria desenvolvido em 2015 para produção e compartilhamento de diferentes práticas didáticas visando auxiliar professores a tornar o ensino de modelagem de software mais dinâmico. Ele tem como base experiências práticas no ensino de Engenharia de Software realizadas em sala de aula. O Guia está organizado em quatro módulos independentes, definidos a partir do conteúdo programático da disciplina de Engenharia de Software I do curso de Ciência da Computação ministrada na Universidade Federal Fluminense em Rio das Ostras. São eles: Módulo 1 – Conceitos Básicos em Engenharia de Software; Módulo 2 – Engenharia de Requisitos; Módulo 3 – Análise de Requisitos; e Módulo 4 – Projeto de Software. Cada módulo é dividido em roteiros que contêm as práticas didáticas para apoiar o ensino dos temas que serão abordados nas aulas. O Guia também propõe uma atividade complementar, na qual os alunos fazem a modelagem de um produto de software genérico para um problema real. O objetivo é dar aos alunos a oportunidade de aprender a trabalhar em equipe e vivenciar algumas situações comuns do dia a dia dos profissionais de software.

3.1. ROTEIROS

Um roteiro corresponde a duas aulas de cinquenta minutos cada e descreve: os objetivos específicos que se pretende alcançar ao final daquele roteiro; o conteúdo básico contendo um resumo dos conceitos que serão abordados nas aulas; a prática didática com as atividades que podem ser realizadas com os alunos para o ensino desse conteúdo; o material de apoio disponível para a realização das práticas; os recursos básicos para a execução do roteiro (p. ex. data show, folha sulfite, cola, etc.); as atividades extraclases (pesquisa, listas de exercícios, etc.) sugeridas aos alunos como complemento à aprendizagem do conteúdo; e referências (livros, artigos, sítios web, etc.) que podem ser consultadas para complementar o conteúdo visto em sala de aula.

No Quadro 1 é possível visualizar o objetivo geral do Módulo 1 e seus roteiros. Também é possível observar a estruturação do Roteiro 3, exceto sua prática didática que será abordada na seção 3.2 deste trabalho. Neste Roteiro, a ideia é introduzir os conceitos de abstração, modelagem e UML, então todo o seu conteúdo estará direcionado para estes temas.

I. Módulo 1 – Conceitos Básicos

Objetivo Geral: Apresentar os principais conceitos relacionados à Engenharia de Software e à Modelagem de Software utilizando a linguagem UML.

1.1 Roteiro 1 – Introdução à Engenharia de Software

1.2 Roteiro 2 – Processo de Desenvolvimento de Software

1.3 Roteiro 3 – Modelagem de Sistemas de Software e Apresentação da UML

Objetivos Específicos:

- Introduzir o conceito de modelagem e sua importância no desenvolvimento de software.
- Apresentar o conceito de abstração em diferentes níveis.
- Abordar o histórico do surgimento da linguagem UML e suas principais características.

Conteúdo Básico:

- Modelagem é uma prática comum compartilhada pela maioria das disciplinas de engenharia. Na Engenharia de Software, a modelagem consiste na utilização de notações gráficas e textuais com o objetivo de construir modelos que representam as partes essenciais de um sistema, considerando-se diversas perspectivas diferentes e complementares.
- Um modelo é uma abstração que representa um ou mais aspectos de um sistema, tais como sua função, comportamento, estrutura e propriedades. Assim, modelos ajudam a gerenciar a complexidade inerente ao desenvolvimento de software, pois revelam as características essenciais de um sistema ignorando detalhes que não são relevantes e que só aumentariam a complexidade do problema a ser tratado. Além disso, modelos permitem realizar uma previsão do comportamento futuro do sistema e avaliar diferentes soluções para os problemas relacionados ao desenvolvimento do software. Desta

forma, a modelagem pode ajudar a reduzir os custos no desenvolvimento, uma vez que a correção dos erros identificados em modelos é menos custosa do que após a implementação do mesmo. Por conter grande parte das informações relativas ao sistema, modelos facilitam a comunicação entre as pessoas envolvidas na sua construção, servindo como um ponto de referência comum [1].

- A UML (*Unified Modeling Language*) é uma linguagem visual utilizada para modelar softwares baseados no paradigma da Orientação a Objetos. É uma linguagem de propósito geral que pode ser aplicada a todos os domínios de aplicação [2]. A UML não é uma linguagem de programação, mas sim uma linguagem de modelagem. Como tal, ela fornece uma notação para criação de diversos documentos, que podem ser gráficos ou textuais. Na terminologia da UML, esses documentos são denominados artefatos de software, ou simplesmente artefatos. Os artefatos gráficos podem ser produzidos pela utilização dos diagramas da UML. Um diagrama é uma coleção de elementos gráficos que possuem uma forma (sintaxe) e um significado predefinido (semântica)[1].

Prática didática:

-

Material de Apoio:

- Modelos, que não sejam da UML, de um mesmo objeto, com diferentes abstrações.
- Diagramas da UML para um mesmo domínio.
- Formulário sobre a UML.

Recursos: Projetor Multimídia; Computador.

Atividade Extraclasse: Resolução da Lista de Exercícios 2.

Referências:

1. Bezerra E. Princípios de Análise e Projeto de Sistemas com UML, 2ª Edição. Editora Campus. Capítulo 1
2. Guedes, G. T. A. UML 2 – Uma Abordagem Prática, 2ª Edição, Editora Novatec, 2011. Capítulo 1.

Esta estrutura dos roteiros ilustrada no Quadro 1 é semelhante à estrutura dos planos de aula tradicionalmente utilizados pelas Instituições de Ensino Superior (IES). No entanto, o foco na elaboração desses roteiros foi a sugestão de práticas didáticas que possibilitem a construção do conhecimento pelos alunos, e não apenas a organização das aulas como geralmente ocorre nos planos de aula. Nos roteiros, essas práticas descrevem atividades elaboradas com o objetivo de despertar o interesse dos alunos pelo conteúdo abordado e incentivar a interação entre eles para a construção coletiva do conhecimento, minimizando a exposição de conteúdos nas aulas.

3.2. PRÁTICAS DIDÁTICAS

Uma prática didática define uma abordagem para o professor trabalhar o conteúdo de um roteiro em sala de aula de forma dinâmica, ou seja, de forma a promover a interação entre os alunos. Cada roteiro define uma prática didática, que é organizada em: motivação - atividade introdutória com o objetivo de despertar o interesse dos alunos pelo tema; desenvolvimento - uma ou mais atividades práticas relacionadas ao conteúdo, tendo material de apoio próprio; conclusão - revisão dos conceitos principais abordados no roteiro; e avaliação - sugestões para o professor avaliar se a execução do roteiro foi bem sucedida.

O Quadro 2 mostra um exemplo de prática didática, definida para o roteiro 3 do Quadro representado na seção 3.1 deste trabalho, tratando os conceitos de abstração, modelagem e linguagem UML.

Quadro 2 - Exemplo de uma Prática Didática

Motivação: Apresentar diferentes modelos com diferentes abstrações de um mesmo objeto que pode ser, por exemplo, uma casa ou um automóvel.

Solicitar aos alunos que destaquem quais aspectos cada modelo representa e discutam o motivo pelo qual existem diferentes modelos para representar o mesmo objeto.

DESENVOLVIMENTO:

Atividade 1:

- Apresentar exemplos de modelos de software representados por diferentes diagramas UML, previamente definidos para um mesmo domínio.
- Solicitar que os alunos, por meio da observação, descrevam quais aspectos do software cada modelo representa.
- Destacar o uso das diferentes notações gráficas e pedir aos alunos para tentarem descobrir a semântica (significado) de cada notação.
- Pedir que os alunos definam o que é um modelo e o que é um diagrama.

Atividade 2:

- Pedir que os alunos façam uma pesquisa rápida na Web, ou nas referências, sobre a linguagem UML, com foco principalmente nos tópicos: história do surgimento da linguagem; quantos e quais são os diagramas da UML; e qual a função de cada um deles.
- Com base nos resultados obtidos pedir que os alunos preencham o formulário sobre a linguagem UML.
- Pedir que os alunos criem uma definição para a linguagem UML com base na pesquisa realizada.

Conclusão: Destacar a importância da modelagem do sistema durante o desenvolvimento de software e os aspectos que tornam a UML diferente das linguagens de programação.

Avaliação: A aula será considerada bem-sucedida se os alunos, durante as atividades, forem capazes de compreender o conceito de abstração e a importância da modelagem no desenvolvimento de software.

As atividades sugeridas na prática didática visam auxiliar o aluno a aprender a buscar as informações relevantes sobre o conteúdo previsto no roteiro e a construir o seu conhecimento a partir dessas informações e da sua interação com os outros alunos. O professor tem o papel de orientar as atividades e ajudar o aluno a relacionar os conceitos. Para a realização das atividades são definidos materiais de apoio (reportagens, vídeos, diagramas, formulários, jogos, etc.) que serão utilizados em sala de aula, como mostra no Quadro 1. É interessante observar que nas atividades sugeridas não há a exposição tradicional de conteúdo. A construção do conhecimento acontece a partir dos exemplos fornecidos e das pesquisas e definições elaboradas pelos próprios alunos.

3.3. ATIVIDADE COMPLEMENTAR

A Atividade Complementar consiste em utilizar as técnicas de modelagem e especificação aprendidas em aula para documentar os requisitos e fazer a modelagem de um produto de software genérico, simulando o trabalho de uma equipe de desenvolvimento real. Trata-se de uma atividade totalmente prática na qual os próprios alunos definem o domínio que será apoiado pelo software (p.ex.: veterinária, farmácia, imobiliária, etc.) Dependendo do tamanho da turma, mais de um domínio pode ser definido. Neste caso, são definidas diferentes equipes, uma para cada domínio. O domínio deve ser escolhido de forma que o sistema possa ser organizado em módulos.

Inicialmente os alunos devem realizar o levantamento de requisitos e elaborar o documento de requisitos para o sistema que será modelado. Com base nesse documento, os alunos devem modelar a arquitetura inicial do sistema agrupando os requisitos em módulos e definir equipes menores que ficarão responsáveis pela modelagem de cada módulo. Durante a modelagem, as equipes são incentivadas a trocar informações para encontrar soluções de projeto que facilitem a posterior integração dos módulos no sistema.

Alguns roteiros são específicos para a realização de parte da Atividade Complementar em sala de aula para que as equipes possam trabalhar juntas, validando as soluções escolhidas em cada módulo, identificando redundâncias, conflitos, possibilidades de reúso de componentes e esclarecendo dúvidas entre si, com o professor e com o monitor da disciplina.

O Quadro 3 mostra um exemplo de roteiro específico para a realização da atividade complementar.

Quadro 3 - Exemplo de roteiro específico para atividade complementar

Roteiro 6 – Aula Prática: Documentação dos casos de uso da Atividade Complementar

Objetivos Específicos:

- Revisar o Diagrama de Casos de Uso da Atividade Complementar.
- Abordar a documentação dos casos de uso da Atividade Complementar.

Prática Didática:

- Esclarecer possíveis dúvidas sobre o documento de requisitos e o diagrama de casos de uso da atividade complementar. ;
- A partir do diagrama construído no Roteiro 5, e das regras de negócio levantadas no Roteiro 4, solicitar aos alunos que iniciem a descrição dos casos de uso primários, considerando todas as informações coletadas até o momento.

Atividade Extraclasse: Resolução da Lista de Exercícios 3; Pedir que os alunos preparem uma descrição das informações obtidas sobre o tema definido para a Atividade Complementar.

- Solicitar aos alunos que apresentem as informações obtidas sobre o tema da Atividade Complementar.

Na prática descrita nesse roteiro, a atividade sugerida tem como objetivo incentivar os alunos a definirem juntos os pacotes necessários para organizar visualmente os requisitos funcionais do sistema que estão trabalhando como atividade complementar, e decidirem qual é a melhor forma de alocar os casos de uso definidos a esses pacotes.

4. O SACPD - SISTEMA DE APOIO AO COMPARTILHAMENTO DE PRÁTICAS DIDÁTICAS

A ideia do Guia de Aprendizagem Participativa em Engenharia de Software teve como evolução a possibilidade de que as práticas didáticas não ficassem restritas a apenas IES específicas, mas que fosse criada uma grande rede de compartilhamento dessas práticas.

O código do projeto encontra-se disponível para visualização, obtenção e evolução na plataforma GitHub em: <https://github.com/thaynaraCardoso/sacPD>.

4.1. DESCRIÇÃO DO SISTEMA

O Sistema de Compartilhamento de Práticas Didáticas, ainda em suas fases iniciais foi pensado de forma que o professor realize seu cadastro inserindo dados como seu nome, e-mail, instituição a qual pertence e área de atuação. Desses dados inseridos, o e-mail deve ser único no sistema e após a realização deste cadastro, o usuário tenha acesso ao sistema e possa criar seus próprios roteiros, práticas didáticas e atividades. O usuário poderá também consultar e usar práticas e roteiros criados por outros usuários.

A criação de um roteiro ou prática didática será feita seguindo as mesmas estruturas utilizadas no Guia para apoiar a Aprendizagem Participativa em Engenharia de Software, ilustradas nas seções 3.1 e 3.2 deste trabalho. A ideia é que o sistema reflita exatamente a organização das informações proposta no Guia.

É possível ainda a realização de upload ou download de arquivos de até 15MB nos formatos .mp4, .jpeg, .png, .pdf, .ppt, .docx e .xls como material de apoio para quaisquer atividades. Além de ser possível a inserção de links para acesso direto à web. Desta forma, cada atividade que compõe o desenvolvimento de uma prática didática poderá ter ou não um material de apoio necessário, podendo ser posteriormente consultado.

O sistema conta ainda com alguns recursos de segurança, como controle de sessão e criptografia da senha dos usuários em Message-Digest algorithm 5 (MD5), função de dispersão criptográfica unidirecional. O MD5 transforma um dado de entrada (no caso a senha inserida pelo usuário) em um hash, mas que não é possível transformar este hash gerado para

se obter a senha original. Desta forma, o método de verificação é então, feito pela comparação de duas hash (mensagem enviada e mensagem recebida) (TERADA, 2011).

4.2. ESPECIFICAÇÃO E MODELAGEM

O sistema foi desenvolvido seguindo o modelo em cascata de desenvolvimento de software, assim foram realizadas as fases de Levantamento de Requisitos, Análise e Projeto antes de iniciar a implementação do sistema. Em cada fase foram criadas diferentes abstrações e também diferentes representações, que constituem a documentação do sistema.

O primeiro passo no processo de desenvolvimento do sistema é justamente entender qual o problema a ser atacado e quais as funcionalidades devem estar presentes, estas funcionalidades representam os requisitos funcionais do sistema. Para possibilitar a rastreabilidade de cada artefato, seja requisitos, regras de negócio ou casos de uso, deve haver um identificador associado a cada um.

No Quadro 4 estão listados os requisitos funcionais do sistema. Na coluna “Identificador” estão os códigos referentes a cada requisito, facilitando assim a rastreabilidade nas descrições de caso de uso. Na coluna “Nome” estão localizados os nomes dos requisitos seguido da coluna “Descrição” que é justamente a funcionalidade ou o requisito descrito.

Quadro 4 - Requisitos funcionais do sistema

Identificador	Nome	Descrição	Caso de Uso
RF001	Realizar Login	O sistema deve permitir que usuários previamente cadastrados realizem login e tenham acesso às funcionalidades	CSU007
RF002	Realizar Cadastro	O sistema deve permitir a realização de cadastro de professores, solicitando: Nome, E-mail, Área de atuação e instituição	CSU008 CSU010
RF003	Criar Roteiro	O sistema deve permitir que usuários autenticados criem seus próprios roteiros contendo os dados solicitados (título, objetivos específicos, conteúdo básico, prática didática)	CSU004
RF004	Criar Prática Didática	O sistema deve permitir que usuários autenticados criem suas próprias práticas didáticas inserindo os dados solicitados (motivação, conclusão, avaliação e desenvolvimento)	CSU005
RF005	Criar Atividade	O sistema deve permitir que usuários autenticados criem suas próprias atividades com descrição e material de apoio específicos	CSU006
RF006	Consultar Roteiro	O sistema deve permitir que usuários consultem seus roteiros ou roteiros públicos de outros usuários	CSU001

RF007	Consultar Prática Didática	O sistema deve permitir que usuários consultem suas práticas didáticas ou práticas didáticas públicas de outros usuários	CSU002
RF008	Gerenciar instituição	O sistema deve permitir que usuários do tipo “administrador” criem, editem ou excluam quaisquer instituições cadastradas no sistema	CSU 009
RF009	Gerenciar usuários	O sistema deve permitir que quaisquer usuários autenticados encerrem sua seção realizando logout no sistema	CSU010
RF010	Gerenciar áreas de atuação	O sistema deve permitir que quaisquer usuários do tipo administrador gerenciem as áreas de atuação cadastradas no sistema, podendo inserir uma nova área de atuação, editar ou excluir alguma presente	CSU011
RF011	Consultar Atividades	O sistema deve permitir que os usuários consultem suas atividades ou as atividades de quaisquer prática didática pública, incluindo descrição e material de apoio	CSU003
RF012	Encerrar sessão	O sistema deve permitir que quaisquer usuários com acesso ao sistema encerrem sua sessão realizando logout no sistema em qualquer momento	CSU001, CSU002, CSU003, CSU004, CSU005, CSU006, CSU007, CSU008, CSU009, CSU010, CSU011

Como medida de segurança, o sistema deve assegurar que todo usuário que esteja acessando, possa encerrar sua sessão e conseqüentemente encerrar suas atividades no sistema até o próximo login. Desta forma, é possível visualizar o requisito funcional RF012, responsável por realizar justamente essa funcionalidade que deve estar presente em todas as telas do sistema. No Quadro 5 estão listadas as regras de negócio e os casos de uso que estão associados a cada uma delas.

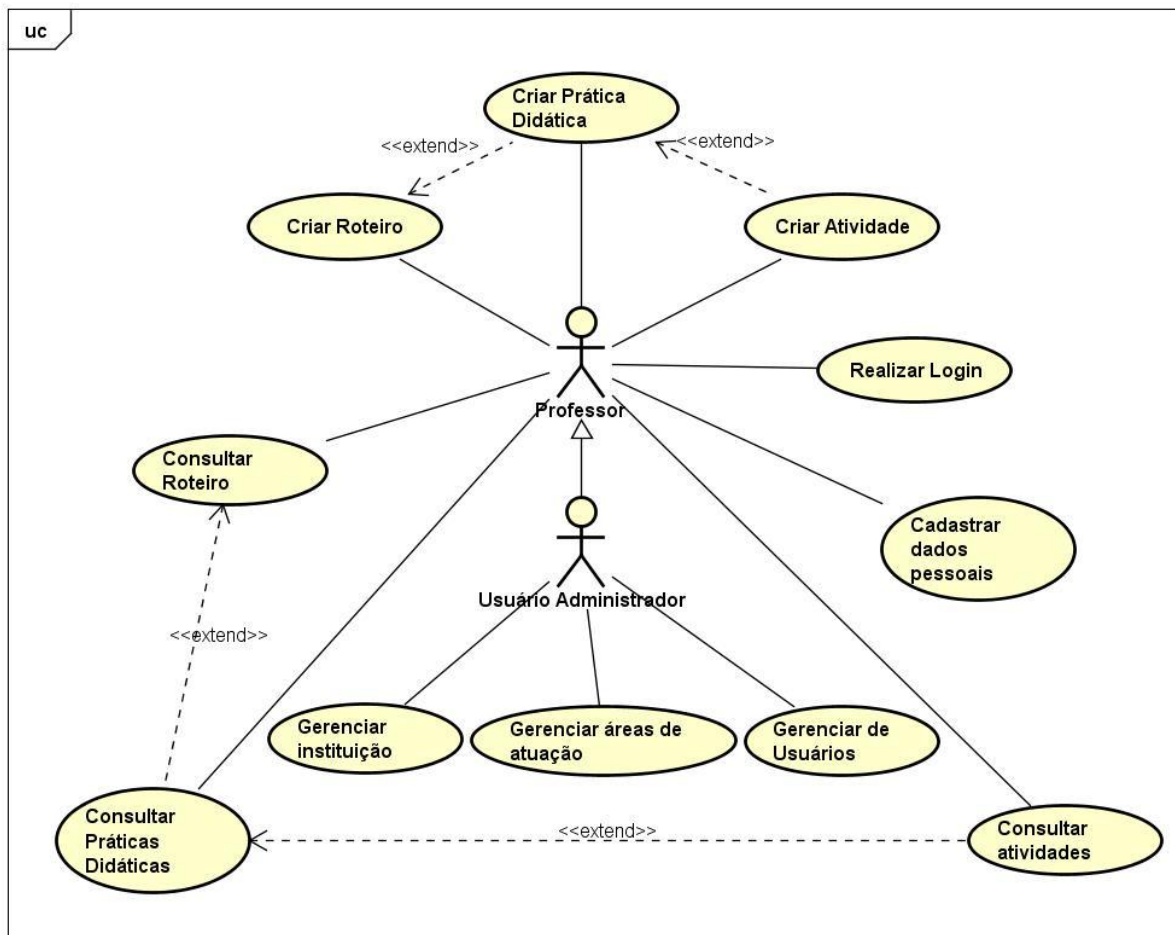
Quadro 5 - Regras de negócio do sistema

Identificador	Nome	Descrição	Caso de Uso
RN001	Visualização de conteúdo	O sistema só deve exibir roteiros ou práticas didáticas que são do próprio usuário ou que são do tipo “público”	CSU001, CSU002,
RN002	Conteúdo do material de apoio	O material de apoio pode ser dos seguintes formatos de arquivo: .docx, .pdf, .png, .jpeg, .ppt, .xls, .txt de tamanho até 30MB	CSU003, CSU006
RN003	Dado único	Os dados do tipo “e-mail”, “área de atuação”, e “instituição” devem ser únicos no sistema	CSU008, CSU009, CSU010, CSU011

É importante destacar que todos esses artefatos sofreram modificação desde o início do processo de desenvolvimento e sempre será necessário revisá-los e atualizá-los à medida que o processo de desenvolvimento do sistema vai evoluindo.

Na Figura 1 é possível visualizar o diagrama de casos de uso onde as principais funcionalidades do sistema são representadas utilizando as notações da linguagem UML 2.5 (OMG, 2017).

Figura 1 – Diagrama de Casos de Uso do sistema



As funcionalidades são representadas como casos de uso (elipses) que serão acessados pelo ator professor, papel exercido pelo principal usuário no sistema. Há interação entre algumas funcionalidades como, por exemplo, ao se criar um roteiro, o usuário pode ou não criar uma prática didática, e por isso o uso da relação “<<extend>>” entre os casos de uso Criar Roteiro e Criar Prática Didática.

Para cada caso de uso, há uma descrição, requisitos funcionais, requisitos não-funcionais e regras de negócio associados. No Quadro 6 é possível visualizar a descrição do caso de uso que descreve a rotina de criar um novo roteiro.

Quadro 6 - Descrição de caso de uso "Criar Roteiro"

CSU 004 – Criar Roteiro

Descrição: O ator cria um novo roteiro com suas próprias práticas didáticas ou com práticas didáticas já existentes no sistema

Pré-condições: Usuário estar cadastrado no sistema.

Ator principal: Professor

Fluxo Principal

1. O ator seleciona a opção de visualizar todos os roteiros criados;
2. Sistema exibe uma listagem contendo todos os roteiros criados pelo ator;
3. O ator seleciona a opção de criar um novo roteiro;
4. Sistema solicita o título, objetivo específico, conteúdo básico, palavras-chave e a visibilidade do roteiro;
5. Ator insere os dados solicitados e conclui ação;
6. Sistema armazena os dados inseridos retorna para a listagem exibida no passo 2 do Fluxo de Eventos, agora atualizada, de roteiros criados e o caso de uso termina.

Fluxos Alternativos

A1 - Fluxo alternativo (passo 6) - Ator deseja criar uma nova prática didática associada ao roteiro criado

1. Ator seleciona a opção de criar prática didática;
2. Sistema executa o caso de uso “CSU 005 - Criar Prática Didática” e o caso de uso termina.

A2 - Fluxo alternativo (passo 6) - Ator deseja associar uma prática didática de outros usuários a um roteiro específico

1. Ator seleciona a opção de adicionar prática didática de terceiros a um roteiro que está sendo criado;
2. Ator insere uma palavra-chave para busca;
3. Sistema exibe uma listagem contendo todas as práticas didáticas públicas que contenham a palavra-chave de busca;

4. Ator seleciona a prática didática desejada e adiciona ao roteiro selecionado.
5. Sistema armazena os dados, atualiza a listagem e o caso de uso termina.

Fluxos de Exceção

Não se aplica

Como parte da especificação do sistema, as descrições de casos de uso tem a principal função de dar entendimento ao comportamento do sistema para cada funcionalidade. Assim, as descrições dos casos de uso ficaram da seguinte forma:

CSU 001 – Gerenciar roteiro

Descrição: O ator realiza uma busca por algum tema dentre os seus roteiros e os roteiros públicos já cadastrados no sistema, editar ou apagar seus roteiros.

Pré-condições: Professor estar identificado no sistema.

Ator principal: Professor

Fluxo Principal

1. O professor solicita a consulta de roteiros públicos;
2. O sistema exibe o título, objetivos específicos e o conteúdo básico de todos os roteiros criados por outros professor definidos como “público”; (A2)
3. Professor deseja visualizar a prática didática associada a um roteiro da lista
4. O professor seleciona a opção de visualizar a prática didática associada a um roteiro da lista;
5. Sistema executa o caso de uso “CSU 002 – Consultar práticas didáticas” e o caso de uso termina.

Fluxo Alternativo

A1 - Fluxo Principal (passo 1) – O professor deseja visualizar seus próprios roteiros

1. O sistema exibe o título, objetivos específicos e o conteúdo básico de todos os roteiros que o usuário já criou e o caso de uso termina

A2 – Fluxo Principal (passo 2) Professor deseja visualizar a prática didática associada a um roteiro da lista

1. O professor seleciona a opção de visualizar a prática didática associada a um roteiro da lista;
2. Sistema executa o caso de uso “CSU 002 – Consultar práticas didáticas” e o caso de uso termina.

A2 – Fluxo Alternativo (passo 2 do fluxo alternativo A1) – Professor deseja editar o um dos roteiros exibidos

1. O professor seleciona a opção de editar o roteiro;
2. Sistema torna os dados do roteiro selecionado editáveis;
3. Usuário altera um ou mais dados e confirma;
4. Sistema atualiza os dados e o caso de uso termina;

A3 – Fluxo Alternativo (passo 2 do fluxo alternativo A1) - Professor deseja excluir um dos roteiros exibidos

1. O professor seleciona a opção de excluir um dos roteiros exibidos;
2. Sistema realiza ação de exclusão, exibe a listagem de roteiros atualizada e o caso de uso termina.

Fluxo de Exceção

Não se aplica

CSU 002 – Consultar prática didática

Descrição: O ator visualiza suas próprias práticas didáticas ou as práticas didáticas públicas criadas por outros usuários

Pré-condições: Professor estar cadastrado no sistema.

Ator principal: Professor

Fluxo Principal

1. O professor solicita a consulta de práticas didáticas públicas; (A1);
2. O sistema exibe a motivação, conclusão e avaliação da prática didática e o caso de uso termina; (A2)

Fluxo Alternativo

A1 – O professor deseja visualizar suas próprias práticas didáticas

1. O sistema exibe a motivação, conclusão e avaliação de todas as suas práticas didáticas e o caso de uso termina. (A2) (A3) (A4)

A2 – Professor deseja visualizar as atividades associadas a uma prática didática específica

3. O professor seleciona a opção de visualizar atividades associadas a uma prática didática da lista;
4. Sistema executa o caso de uso “CSU 003 – Consultar atividades” e o caso de uso termina.

A3 – Professor deseja editar uma de suas práticas didáticas

1. Professor seleciona a opção de editar uma prática didática;
2. Sistema torna todos os dados da prática didática editáveis;
3. Professor edita os dados desejados e salva;
5. Sistema atualiza os dados e o caso de uso termina;

A4 – Professor deseja excluir uma das práticas didáticas exibidas

1. O professor seleciona a opção de excluir uma das práticas didáticas exibidas;
2. Sistema realiza ação de exclusão, exibe a listagem de práticas didáticas atualizada e o caso de uso termina.

CSU 003 – Consultar atividades

Descrição: O ator visualiza as atividades associadas a uma determinada prática didática.

Pré-condições: Usuário estar cadastrado no sistema

Ator principal: Professor

Fluxo Principal

1. Usuário seleciona a opção de visualizar as atividades associadas a uma prática didática;
2. Sistema exibe a descrição e o material de apoio de cada atividade e o caso de uso termina.

Fluxos Alternativos

Não se aplica

Fluxos de Exceção

Não se aplica

CSU 004 – Criar Roteiro

Descrição: O ator cria um novo roteiro com suas próprias práticas didáticas ou com práticas didáticas já existentes no sistema

Pré-condições: Usuário estar cadastrado no sistema.

Ator principal: Professor

Fluxo Principal

7. O ator seleciona a opção de criar um novo roteiro;
8. Sistema solicita o título, objetivo específico, conteúdo básico, palavras-chave e a visibilidade do roteiro;
9. Ator insere os dados solicitados e conclui ação;
10. Sistema armazena os dados inseridos retorna para sua listagem, agora atualizada, de roteiros criados e o caso de uso termina. (A1)

Fluxos Alternativos

A1 - Usuário deseja criar uma nova prática didática associada ao roteiro criado

3. Usuário seleciona a opção de criar prática didática;
4. Sistema executa o caso de uso “CSU 005 - Criar Prática Didática” e o caso de uso termina.

A2 – Usuário deseja associar uma prática didática de outros usuários a um roteiro específico

6. Usuário seleciona a opção de adicionar prática didática de terceiros a um roteiro específico;
7. Sistema exibe uma listagem contendo todas as práticas didáticas públicas;
8. Usuário seleciona a prática didática desejada e adiciona ao roteiro selecionado.
9. Sistema armazena os dados, atualiza a listagem e o caso de uso termina.

Fluxo Alternativo

Não se aplica

CSU 005 – Gerenciar Prática Didática

Descrição: O ator cria uma nova prática didática com suas próprias práticas didáticas ou com práticas didáticas já existentes no sistema

Pré-condições: Usuário estar cadastrado no sistema.

Ator principal: Professor

Fluxo Principal

1. O ator seleciona a opção de criar uma nova prática didática associada a um roteiro;
2. Sistema solicita os dados (motivação, conclusão e avaliação) da nova prática didática;
3. Usuário insere os dados solicitados; (A1)
4. Sistema armazena os dados inseridos, atualiza a tabela de práticas didáticas e o caso de uso termina.

Fluxos Alternativos

A1 – Usuário deseja criar atividades associadas à prática didática criada

1. Usuário seleciona a opção de criar atividades associadas à prática didática selecionada;
2. Sistema executa o caso de uso “CSU 006 – Criar Atividade”

Fluxos de Exceção

Não se aplica

CSU 006 – Gerenciar Atividade

Descrição: Usuário poderá criar uma nova atividade, editar ou excluir alguma atividade já criada

Pré-condições: Usuário estar cadastrado no sistema. Nova atividade deve estar associada a uma prática didática já existente.

Ator principal: Professor

Fluxo Principal

1. Usuário seleciona a opção de criar uma nova atividade; (A1) (A2) (A3)
2. Sistema solicita que o usuário insira a descrição da atividade e o conteúdo do material de apoio da atividade;
3. Usuário insere a descrição da atividade e seleciona a opção de anexar um arquivo;
4. Sistema abre janela de busca de arquivos no computador;
5. Usuário navega até arquivo desejado e seleciona o arquivo desejado; (A3)
6. Sistema armazena a nova atividade associada à prática didática selecionada e o caso de uso termina;(A4)

Fluxo Alternativo

A1 – Usuário deseja editar uma atividade já criada

1. Sistema exibe uma listagem com todas as atividades associadas à uma prática didática selecionada;
2. Usuário seleciona a atividade a ser alterada;
3. Sistema exibe sua descrição e material de apoio;
4. Usuário altera os dados desejados e salva; (A3)
5. Caso de uso segue no passo 6 do Fluxo de Eventos.

A2 – Usuário deseja excluir uma atividade já criada

1. Sistema exibe uma listagem com todas as atividades associadas à prática didática selecionada;
2. Usuário seleciona a atividade desejada;
3. Sistema exibe os dados da atividade desejada;
4. Usuário seleciona a opção de excluir atividade;
5. Sistema excluir atividade selecionada, atualiza a listagem de atividades e o caso de uso termina; (A4)

A3- Violação da Regra Negócio RN002

1. Sistema não aceita ação e solicita novo arquivo e o caso de uso segue no passo 3 do Fluxo de Eventos.

A4 - Usuário deseja adicionar mais uma atividade à prática didática;

1. Usuário seleciona opção de adicionar atividade e o caso de uso retorna ao passo 4 do Fluxo de Eventos.

Fluxo de Exceção

Não se aplica

CSU 007 – Realizar Login

Descrição: Usuário realiza login e passa a ter acesso ao sistema

Pré-condições: Usuário estar cadastrado no sistema.

Ator principal: Professor

Fluxo Principal

1. Usuário insere seu e-mail e senha de acesso ao sistema;
2. Sistema valida os dados inseridos e libera acesso às funcionalidades do sistema; (A1)

Fluxos Alternativos

A1 – E-mail e senha não coincidem com os dados presentes na base do sistema

1. Sistema exibe mensagem de erro e caso de uso segue no passo 1 do Fluxo de Eventos.

Fluxos de Exceção

Não se aplica.

CSU 008 – Cadastrar dados pessoais

Descrição: Usuário cadastra dados pessoais para ter acesso ao sistema

Pré-condições: -

Ator principal: Professor

Fluxo Principal

1. Usuário seleciona a opção de cadastrar dados pessoais para obter acesso ao sistema;
2. Sistema solicita os dados (nome completo, instituição, área de atuação, e-mail e senha de acesso ao sistema);
3. Usuário insere os dados solicitados e salva;
4. Sistema valida e armazena os dados inseridos, cria o novo usuário, retorna à tela de login e o caso de uso termina; (A1).

Fluxos Alternativos

A1 – E-mail já cadastrado no sistema

1. Sistema não finaliza ação e o caso de uso segue no passo 2 do Fluxo de Eventos.

CSU 009 – Gerenciar instituição

Descrição: Usuário gerencia as instituições que poderão ser selecionadas neste campo

Pré-condições: -

Ator principal: Usuário administrador

Fluxo Principal

1. Usuário seleciona a opção de gerenciar as instituições presentes no sistema;
2. Sistema exibe uma listagem com as instituições presentes no sistema;
3. Usuário seleciona a opção de editar uma instituição presente; (A1) (A2)
4. Sistema torna o nome da instituição selecionada editável;
5. Usuário altera o dado desejado e salva;

6. Sistema atualiza o dado a tabela de instituições e o caso de um termina;

Fluxos alternativos

A1 – Usuário deseja excluir uma instituição já presente no sistema

1. Usuário seleciona a instituição que deseja excluir;
2. Sistema exclui instituição selecionada e atualiza lista de instituições e o caso de uso termina

A2 – Usuário deseja criar uma nova instituição

1. Usuário seleciona a opção de adicionar nova instituição;
2. Sistema solicita o nome da nova instituição a ser criada;
3. Usuário insere o nome da nova instituição;
4. Sistema valida o dado inserido, armazena na base de dados e o caso de uso termina;

Fluxo de Exceção

Não se aplica

CSU 010 – Gerenciar usuários

Descrição: Usuário do tipo administrador gerencia todos os usuários cadastrados no sistema

Pré-condições: Usuário ser do tipo “administrador” com acesso ao sistema

Ator principal: Usuário administrador

Fluxo de Eventos:

1. Ator seleciona a opção de Gerenciar usuários;
2. Sistema exibe uma listagem contendo todos os dados (nome completo, e-mail de acesso, instituição, área de atuação e tipo*) de todos os usuários cadastrados.
3. Ator seleciona a opção de criar um novo usuário; (A1) (A2)
4. Sistema solicita os dados do novo usuário;
5. Ator insere os dados e salva;
6. Sistema armazena os dados inseridos e atualiza a lista e o caso de uso termina.

Fluxos Alternativos:

A1 – Ator deseja excluir um usuário da listagem

1. Ator localiza o dado (usuário) que deseja excluir, e seleciona a opção excluir;
2. Sistema realiza ação, atualiza a base e a listagem exibidas e o caso de uso termina.

A2 – Ator deseja editar os dados de algum usuário presente na listagem

1. Ator localiza o dado (usuário) que deseja alterar e seleciona a opção alterar;
2. Sistema torna os dados do usuário editáveis;
3. Ator atualiza as informações necessárias e confirma;
4. Sistema realiza ação, atualiza a base e a listagem exibidas e o caso de uso termina.

*tipo: Usuário administrador ou usuário comum

CSU011 – Gerenciar áreas de atuação

Descrição: Usuário do tipo administrador gerencia as áreas de atuação que poderão ser utilizadas no cadastro

Pré-condições: Usuário ser do tipo “administrador” com acesso ao sistema

Ator principal: Usuário administrador

Fluxo de Eventos:

1. Ator seleciona a opção de Gerenciar Áreas de Atuação;
2. Sistema exibe uma listagem ordenada por ordem alfabética, todas as áreas de atuação cadastradas;
3. Ator seleciona a opção de criar uma nova área de atuação; (A1) (A2)
4. Sistema solicita o nome da nova área de atuação;
5. Ator insere o nome da nova área de atuação;
6. Sistema valida o dado inserido, armazena, retorna à listagem das áreas de atuação e o caso de uso termina. (A3)

Fluxos Alternativos

A1 – Ator deseja editar uma área de atuação já presente na listagem

1. Ator localiza o dado (nome da área de atuação) a ser alterado e seleciona a opção de editar;
2. Sistema torna o dado editável;
3. Ator insere o novo dado e salva e o caso de uso segue no passo 6 do Fluxo de Eventos;

A2 – Ator deseja excluir uma área de atuação já presente na listagem

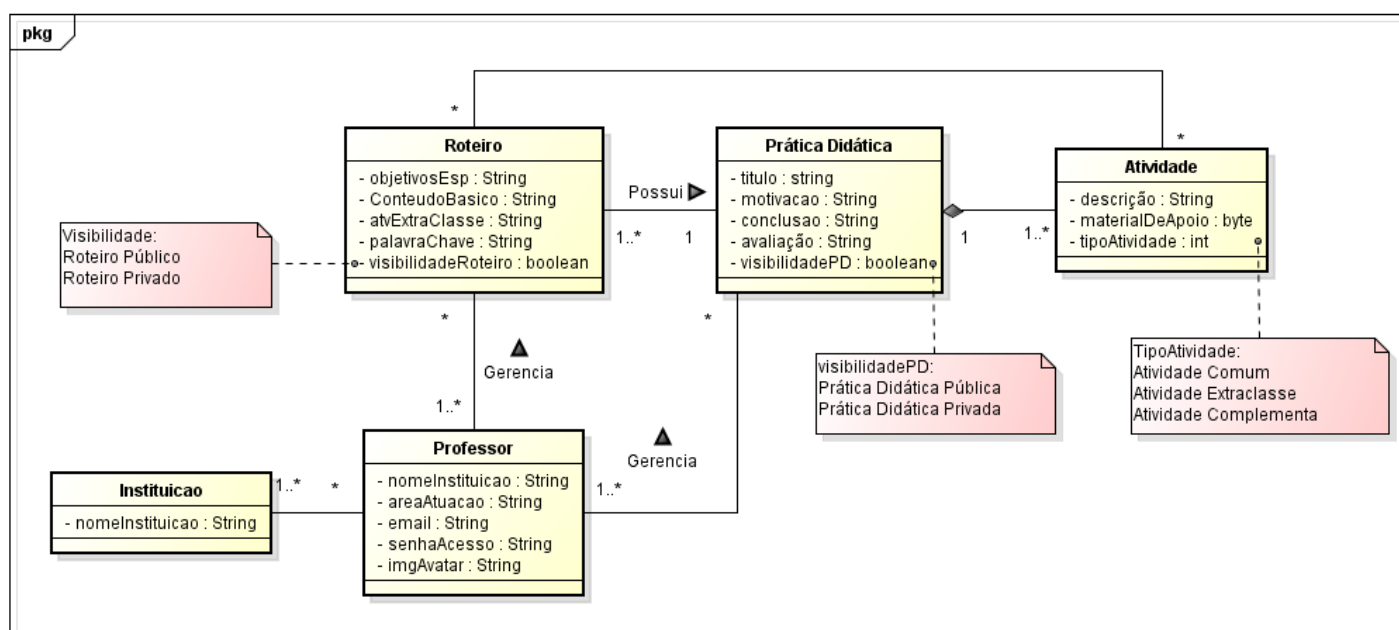
4. Ator localiza o dado (nome da área de atuação) a ser excluído e seleciona a opção de excluir;
5. Sistema apresenta mensagem de confirmação da ação;
6. Usuário confirma;
7. Sistema realiza ação e o caso de uso segue no passo 6 do Fluxo de Eventos;

A3 – Dados já existentes na base do sistema (violação da RN003)

1. Sistema apresenta uma mensagem de que o dado já consta na base;
2. Usuário confirma e o caso de uso segue no passo 6 do Fluxo de eventos.

Uma vez validadas as funcionalidades que devem estar presentes no sistema através das descrições dos casos de uso, é definido o diagrama de classes de análise, onde é possível visualizar as classes de domínio, sem atrelar necessariamente uma linguagem ou tecnologia de desenvolvimento. Tem como finalidade o entendimento do domínio apoiado pelo sistema. Na Figura 2 é possível visualizar o diagrama de classes obtido a partir dos requisitos e entendimento do domínio. O usuário tem um roteiro que está associado a uma prática que se trata de um objeto com dados exclusivos, mas que também armazena uma coleção de objetos do tipo atividade, termo associado a cada atividade presente no desenvolvimento da prática didática.

Figura 2 - Diagrama de classes de análise de domínio

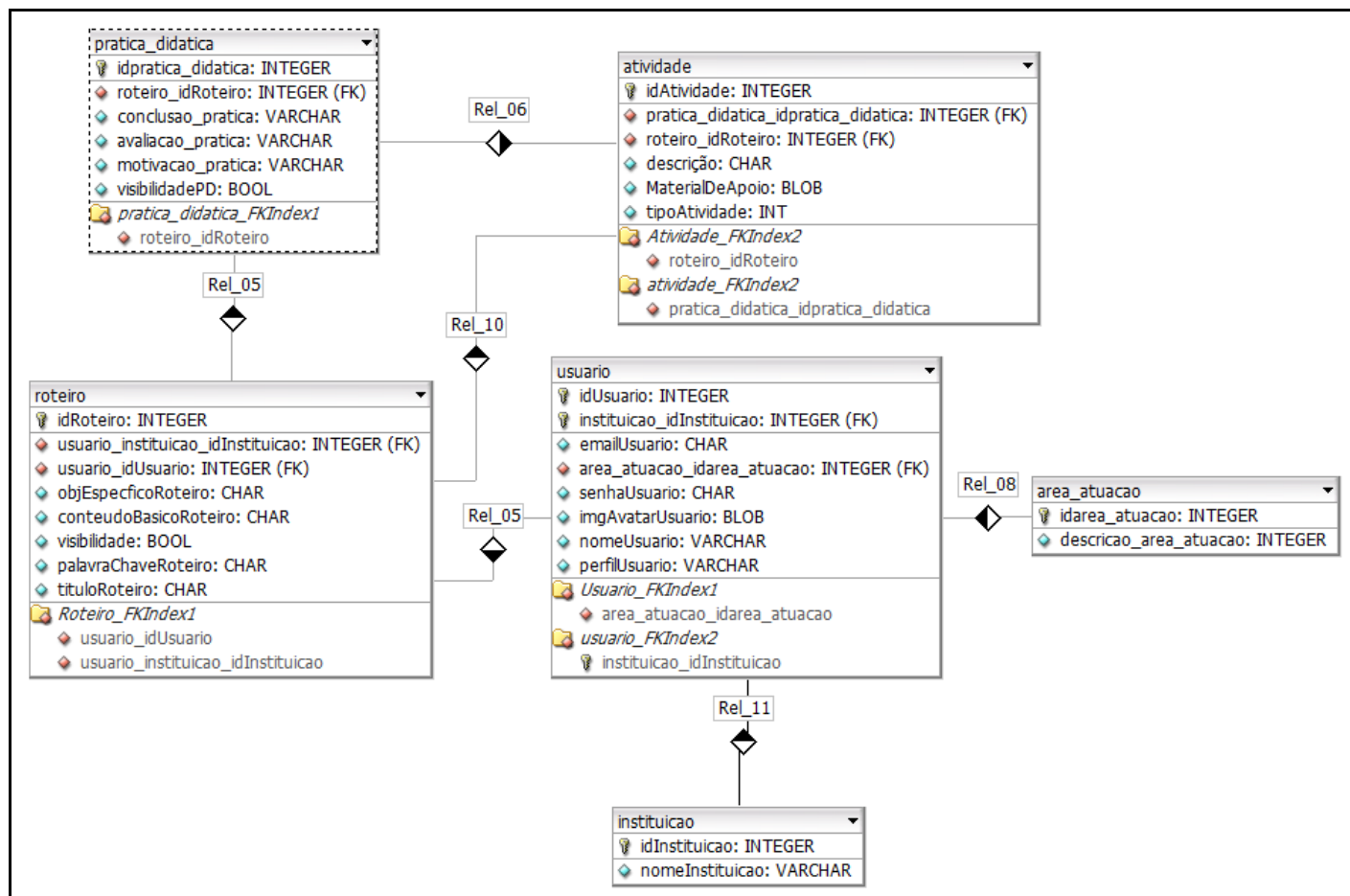


A classe “Professor” representa cada usuário que irá interagir com o sistema. A classe “AtividadeModel” representa cada atividade que será realizada durante o desenvolvimento da Prática Didática. Ela pode conter um arquivo como material de apoio e uma descrição. Ela também poder ser de 3 tipos: Atividade Complementar, referente à atividade citada na Seção 3.3; Atividades Extraclasse que são as atividades que os alunos devem executar fora de aula, e

Atividades Comuns, que são as atividades que normalmente compõem o desenvolvimento da prática didática dentro de sala de aula.

Na Figura 3 é possível observar o modelo entidade relacionamento dos dados.

Figura 3 - Modelo entidade relacionamento dos dados

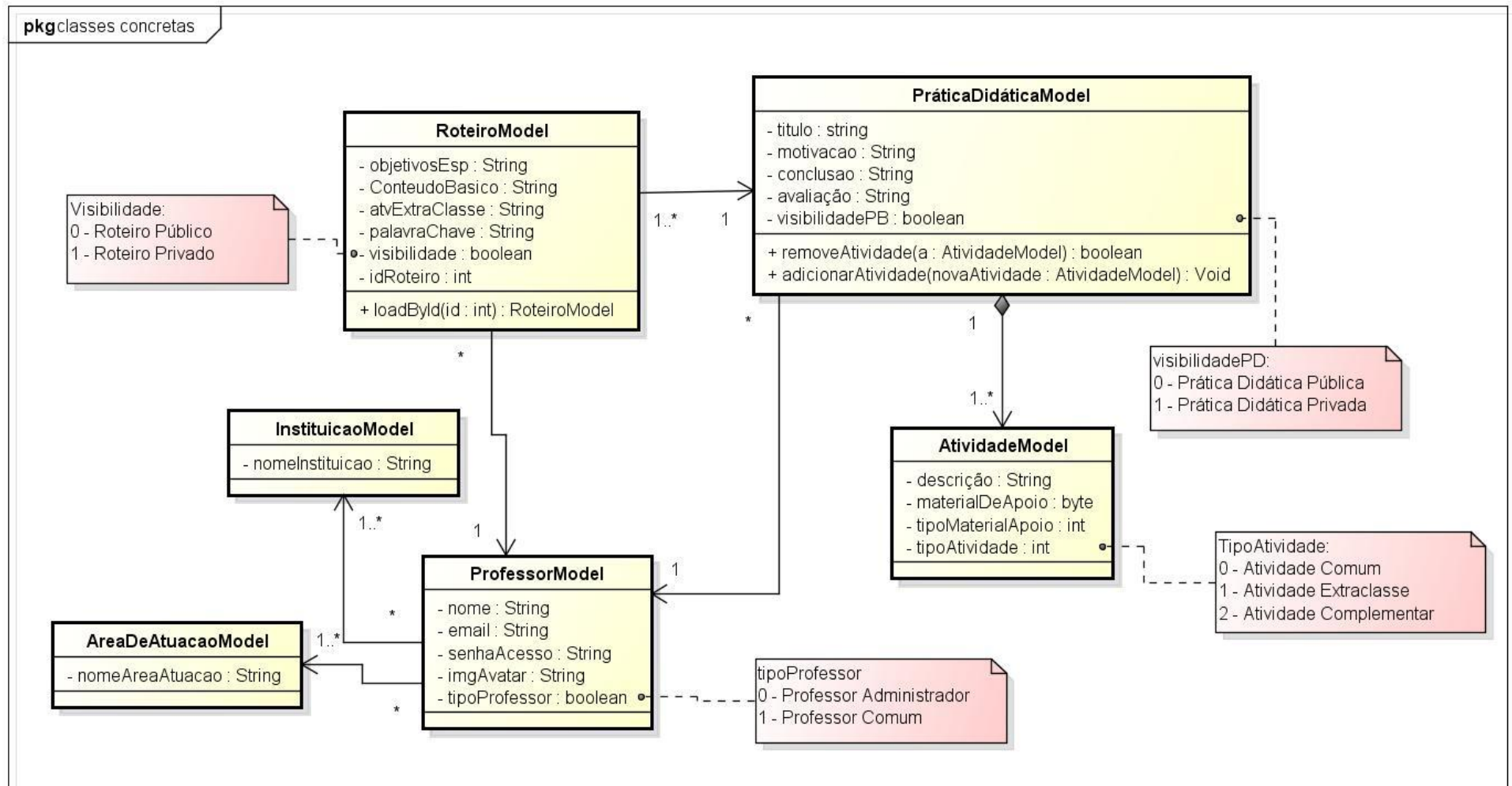


As tabelas “instituicao” e “area_atuacao” armazenam valores padrão para que o usuário possa selecionar durante o cadastro, evitando assim que usuários diferentes insiram nomes diferentes para uma mesma instituição por exemplo.

Uma vez que o domínio foi analisado, chega o momento de projetar o sistema visando sua execução prática. Assim, nas figuras 4a e 4b é possível visualizar a estrutura de como os objetos foram organizados agora considerando tecnologias e decisões de projeto. Para facilitar a leitura e visualização dos objetos, as operações de get e set foram ocultadas do modelo assim como a separação do diagrama em duas imagens, de forma que na Figura 4a estão

representadas as classes de controle e as classes de fronteira, e na Figura 4b estão as classes de negócio, onde são definidos os atributos dos objetos concretos da camada de modelo.

Figura 4b - Diagrama de classes de projeto – Classes de negócio (camada de *model*)



Na Figura 4a a classe abstrata “ObjetoController” assina operações abstratas que serão implementadas em cada classe concreta herdeira, possibilitando o polimorfismo abstraindo as origens de cada objeto. Cada classe controle, representadas pelo esteriótipo “<<control>>” tem uma associação de dependência com exatamente uma classe contida dentro do pacote “Classes de negócio”, representadas na Figura 4b. Exemplo, “RoteiroController“ tem relação de dependência com a classe “RoteiroModel”, assim como “PraticaDidáticaController” tem associação de dependência com a classe “PraticaDidaticaModel” e assim por diante.

Além disso, ainda no diagrama representado na Figura 4a é possível visualizar as classes de fronteira (view), representadas pela notação específica da UML. Essas classes fazem a ponte entre o usuário que está interagindo com o sistema e os controladores que realizam as ações e buscam os dados necessários direto nas classes de modelo, mantendo assim o modelo MVC de desenvolvimento.

No diagrama da Figura 4b é possível visualizar as classes de modelo “RoteiroModel”, “ProfessorModel”, “PráticaDidáticaModel”, “InstituiçãoModel”, “ÁreaDeAtuaçãoModel” e “AtividadeModel”.

4.3. TECNOLOGIAS E FERRAMENTAS UTILIZADAS

Em cada fase do sistema, diferentes ferramentas e diferentes tecnologias foram usadas para se obter cada artefato. Para a produção dos diagramas de casos de uso e diagrama de classes foi utilizada a ferramenta CASE Astah v7.2 (ASTAH, 2017) Astah é uma ferramenta CASE que possui os principais diagramas presentes na linguagem UML 2.5, funcional, intuitiva, de fácil utilização e que possui licença para estudantes.

Ainda na fase de definição do sistema, o diagrama de dados entidade-relacionamento foi desenvolvido na ferramenta DBDesigner 4 (DBDESIGNER, 2018), ferramenta de criação de modelos entidade- elacionamento, passando pela primeira, segunda e terceira forma normal, simplificando assim a estrutura do modelo e diminuindo as dependências.

O sistema foi desenvolvido utilizando a linguagem de programação PHP v7.2 para controle das classes de negócio e comportamentos de controladores. PHP é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML (MANUAL, 2009) .

Como sistema gerenciador de banco de dados relacional foi utilizado o MySQL 8.0 e como servidor o wamp 3.1.4 apache.A IDE utilizada foi a Eclipse PHP 2018-09 por ter uma

interface intuitiva e não gerar lixo no código, deixando a cargo do desenvolvedor inserir apenas arquivos e códigos pertinentes ao projeto.

4.4. IMPLEMENTAÇÃO

O sistema desenvolvido para demonstrar a proposta do trabalho inclui as funcionalidades de criação, edição, e exclusão de roteiros, criação, edição e exclusão de práticas didáticas, assim como a criação de atividades com suas descrições e material de apoio, podendo ser de qualquer dos formatos dentre .docx, .pdf, .png, .jpeg, .ppt, .xls, .txt de tamanho até 15MB. No protótipo também foram implementadas operações de gestão de usuários, de áreas de atuação e instituição.

O padrão de arquitetura MVC (*Model-View-Controller*) (KRASNER e POPE 1998) é bastante utilizado no desenvolvimento de aplicações para dispositivos móveis, pois determina a separação de uma aplicação em três elementos. O *Model* é formado por entidades que representam os dados da aplicação. A *View* tem por objetivo realizar a apresentação destes dados e capturar os eventos do usuário sendo representada pelas telas. O *Controller* faz a ligação entre o *Model* e a *View*, realizando o tratamento dos eventos, atuando sobre o *Model* e alterando os elementos da *View* para representar a nova forma dos dados. Toda a implementação do sistema foi feita com base no modelo MVC de desenvolvimento, tendo o papel de cada camada muito bem definido facilitando assim a rastreabilidade de erros e diminuindo as dependências das classes.

No Quadro 7 é possível visualizar o trecho de código da operação “loadById” presente na classe RoteiroModel que tem a função de retornar todos os dados de um roteiro específico de id = “in_idroteiro” solicitado por uma classe controladora.

Quadro 7 - Operação da classe “RoteiroModel” para carregar os dados de um roteiro já cadastrado

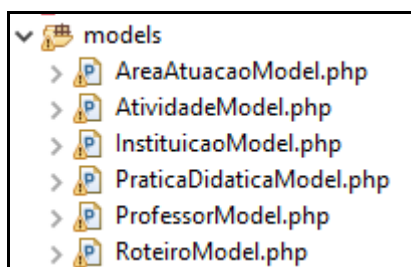
```
/**
 * Retorna os dados de um roteiro referente
 * a um determinado Idroteiro
 * @param integer $in_idroteiro
 * @return RoteiroModel
 */
public function loadById( $in_idroteiro )
{

    $st_query = "SELECT * FROM roteiro WHERE idroteiro = $in_idroteiro";
    $o_data = $this->o_db->query($st_query);
    $o_ret = $o_data->fetchObject();
    $this->setIdRoteiro($o_ret->idroteiro);
    $this->setTituloRoteiro($o_ret->tituloroteiro);
    $this->setObjEspecificoRoteiro($o_ret->objespecificoroteiro);
    $this->setConteudoBasicoRoteiro($o_ret->conteudobasicoroteiro);
    $this->setPalavrachaveRoteiro($o_ret->palavrachaveroteiro);
    $this->setIdVisibilidade($o_ret->idvisibilidade);

    return $this;
}
```

A camada de modelo, onde as classes de negócio estão implementadas, conta com a definição de seis classes de modelo como demonstrado no Quadro 8.

Quadro 8 - Classes da camada modelo do sistema



As classes “*AreaAtuacaoModel.php*” e “*InstituicaoModel.php*” tratam diretamente de dados referentes aos objetos da classe “*UsuarioModel.php*”. No entanto, a decisão de colocá-las em classes separadas surgiu para que um usuário administrador possa cadastrar instituições e áreas de atuação padrão e no momento que quaisquer outro usuário precisar inserir sua área de atuação e instituição, basta selecioná-las dentre as já presentes na listagem, garantindo assim a escalabilidade dos dados.

Como classe de domínio, ficou definido que todo roteiro deve estar associado a exatamente uma prática didática. Uma prática didática contém quantas atividades o usuário desejar. Para que isto fosse implementado da melhor maneira, basta que a prática didática

conheça quais atividades estão associadas a ela, e a prática didática conheça o roteiro a que pertence. No Quadro 9 e Quadro 10 é possível visualizar a estrutura das classes de negócio “AtividadeModel.php”, PraticaDidaticaModel.php” e “RoteiroModel.php”. Em cada uma contém um atributo no formato “\$in_(...)id” como uma referência ao objeto a qual pertence.

Quadro 9 - Estrutura das classes de negócio “AtividadeModel.php” e “PraticaDidaticaModel.php”.

<pre> 15 class AtividadeModel extends PersistModelAbstract 16 { 17 private \$in_idatividade; 18 private \$in_praticadidaticaid; 19 private \$in_descricaoatividade; 20 private \$in_materialapoioatividade; 21 private \$in_tipomaterialapoio; 22 private \$in_nomematerialapoio; </pre>	<pre> 16 class PraticaDidaticaModel extends PersistModelAbstract 17 { 18 private \$in_idpraticadidatica; 19 private \$inroteiroid; 20 private \$in_conclusaopratica; 21 private \$in_avaliacaopratica; 22 </pre>
--	--

A estrutura da classe de negócio “RoteiroModel.php” contém o atributo “\$st_idusuario” como uma referência ao professor que criou o objeto e o atributo “\$st_idvisibilidade” que é responsável por controlar a visibilidade do roteiro, isto é, se ele será público e qualquer outro usuário poderá acessá-lo ou se será privado e apenas o próprio usuário criador do roteiro terá acesso à ele.

Quadro 10 - Estrutura da classe de negócio "RoteiroModel.php"

```

17 class RoteiroModel extends PersistModelAbstract
18 {
19     private $in_idroteiro;
20     private $st_tituloroteiro;
21     private $st_objespecificoroteiro;
22     private $st_conteudobasicoroteiro;
23     private $st_palavrachaveroteiro;
24     private $st_idvisibilidade;
25     private $st_idusuario;

```

O sistema também conta com recursos de segurança como o armazenamento da senha de usuário utilizando MD5, função hash criptográfica unidirecional. E para isso, foi necessário utilizar uma função já nativa do php onde é possível criptografar uma string que retorna a hash resultante de 32 caracteres.

As classes da camada de controle, responsáveis por obter as requisições vindas da view, processarem e manipularem os dados contidos nas classes da camada de negócio, precisam ser o máximo coesas possível, isto é, realizarem apenas ações cujas responsabilidades sejam

da própria classe. Isto porque se trata de uma camada com muitas requisições e muitas trocas entre os objetos, tornando confuso acompanhar o estado de cada objeto.

No Quadro 11 é possível observar um trecho de código da classe “PraticaDidaticaController” da camada de controle onde é definido o método para listar a prática didática referente a um roteiro específico representado pelo atributo “*in_idroteiro*”, fazendo referência ao objeto roteiro o qual a prática didática pertence. Esta operação ainda chama outro arquivo da view “*listarPraticasDidaticas.phtml*” para encaminhar os dados obtidos através dos atributos “*\$o_roteiro*” e “*\$v_praticasdidaticas*”.

Quadro 11 - Trecho de código do controller “PraticaDidaticaController.php”

```
20 public function listarObjetoAction()
21 {
22     //caso o usuário tente acessar diretamente a URL
23     if (!isset($_SESSION['idUsuario'])) {
24         header("location: index.php?controle=Usuario&acao=loginUsuario");
25     }
26     if( isset($_REQUEST['in_idroteiro']) )
27         if( DataValidator::isNumeric($_REQUEST['in_idroteiro']) )
28         {
29             $o_roteiro = new RoteiroModel();
30             $o_roteiro->loadById($_REQUEST['in_idroteiro']);
31
32             $o_praticadidatica = new PraticaDidaticaModel();
33             $v_praticasdidaticas = $o_praticadidatica->list($_GET['in_idroteiro']);
34             $o_view = new View('views/listarPraticasDidaticas.phtml');
35             $o_view->setParams(array('o_roteiro' => $o_roteiro, 'v_praticasdidaticas' => $v_praticasdidaticas));
36             $o_view->showContents();
37         }
38 }
```

A camada de view foi integralmente implementada em phtml, um híbrido de PHP e HTML, e se ocupa basicamente por obter requisições vindas do usuário, e disponibilizá-las para acesso dos controladores ou então exibir os dados também passados pelos controladores. No Quadro 12 é possível visualizar um trecho da view “*listarPraticasDidaticas.phtml*”, arquivo chamado na linha 34 do código mostrado no Quadro 11. No trecho de código mostrado ocorre essa troca entre html e php, delimitada pela presença do conjunto de caracteres “*<?php*” que é quando se inicia o trecho em php e “*?>*” quando finaliza o mesmo trecho. Fora isso, todo resto funciona com o html normalmente.

Quadro 12 - Trecho de código da view “listarPraticasDidaticas.phtml”

```
77 <?php
78
79 foreach($v_praticasdidaticas AS $o_praticadidatica)
80 { $tempratica = 1;
81   ?>
82   <tr>
83
84     <td align="center">
85       <?php echo $o_praticadidatica->getConclusaoPratica()?>
86     </td>
87     <td align="center">
88       <?php echo $o_praticadidatica->getAvaliacaoPratica()?>
89     </td>
90     <td align="center">
91       <a href='?controle=PraticaDidatica&acao=apagarPraticaDidatica&in_idroteiro=
92         <?php echo $oroteiro->getIdRoteiro()?&in_idpraticadidatica=
93         <?php echo $o_praticadidatica->getIdPraticaDidatica()?>'
94         class='bt_red'>Apagar</a>
95     </td>
96     <td align="center">
97       <a href='?controle=PraticaDidatica&acao=manterPraticaDidatica&in_idroteiro=
98         <?php echo $oroteiro->getIdRoteiro()?&in_idpraticadidatica=
99         <?php echo $o_praticadidatica->getIdPraticaDidatica()?>'
100        class='bt_blue'>Editar</a>
101     </td>
102   </tr>
103 <?php
104 }
105 ?>
```

A camada de front-end, aquela a qual o usuário tem acesso diretamente, não conta com muitos recursos avançados como HTML5, por exemplo. Conta com um html e uma base de CSS simples. Por isso o sistema tem uma interface bem simplificada de cores e fontes, não eliminando a possibilidade de um futuro incremento do mesmo. Uma vez que todo o back-end está finalizado e rodando corretamente, basta setar as variáveis corretas no novo layout.

5. AVALIAÇÃO DO SISTEMA

Nesta seção serão apresentados dois cenários da funcionalidade de uso de um usuário comum criar roteiro, a principal funcionalidade do sistema SACPD, dois cenários de um usuário administrador gerenciar as instituições e usuários cadastrados e os resultados dos casos de teste utilizados na avaliação. Primeiramente, será apresentado um cenário onde o usuário comum deseja criar com sucesso um roteiro contendo duas atividades em sua prática didática. No segundo cenário, a sessão iniciada no primeiro cenário será encerrada, outro login acessará o sistema e visualizará todos os roteiros públicos cadastrados no sistema, inclusive o roteiro cadastrado no Cenário de Uso 1. No terceiro cenário o usuário administrador irá excluir uma instituição cadastrada no sistema, e no quarto cenário o usuário administrador irá editar os dados de algum outro usuário cadastrado no sistema.

5.1. CENÁRIO DE USO 1

Neste primeiro cenário, o professor irá criar um novo roteiro, criará uma nova prática didática e nova atividade. Na Figura 5 é possível visualizar que o usuário “Teste_Usuario” não possui nenhum roteiro cadastrado.

Figura 5 – Tela exibindo Listagem (vazia) de roteiros criados pelo professor “Teste_Usuario”



Ao clicar na opção “Adicionar Novo Roteiro”, o sistema solicita os dados do novo roteiro, como representado na Figura 6.

Figura 6 - Tela para criação de novo roteiro

SACPD - Sistema de apoio ao compartilhamento de práticas didáticas

Olá Teste_Usuario!

Home Meus Roteiros Roteiros Públicas Encerrar Sessão

Roteiro

TÍTULO

INTRODUÇÃO À ENGENHARIA DE SOFTWARE

VISIBILIDADE

Roteiro Público
 Roteiro Privado

AÇÕES

Cancelar Salvar

OBJETIVOS ESPECÍFICOS

- Introduzir Os Conceitos Básicos De Engenharia De Software.
- Explicar O Desenvolvimento Profissional De Software.
- Apresentar As Principais Características E Atributos De Um Bom Software.
- Identificar Os Principais Problemas E Desafios Relacionados Ao Desenvolvimento Profissional De Software.

CONTEÚDO BÁSICO

A "Crise Do Software" Foi Um Termo Cunhado Para Descrever As Dificuldades Enfrentadas No Desenvolvimento De Software No Fim Da Década De 60. Ele Surgiu Durante A Conferência Da OTAN Sobre Engenharia De Software (NATO Software Engineering Conference) Em Garmisch Na Alemanha Em 1968. Essa Conferência Tinha Como Principal Objetivo Estabelecer Práticas Mais Maduras Para O Processo De Desenvolvimento E Deu Início À Disciplina De Engenharia De Software [1].

PALAVRAS-CHAVE

Engenharia De Software

Ao concluir a ação e selecionar a opção de salvar, o sistema atualiza a base de dados e retorna para a listagem de roteiros criados, como mostrado na Figura 7.

Figura 7 - Tela de listagem de roteiros atualizada

SACPD - Sistema de apoio ao compartilhamento de práticas didáticas

Olá Teste_Usuario!

Home Meus Roteiros Roteiros Públicas Encerrar Sessão

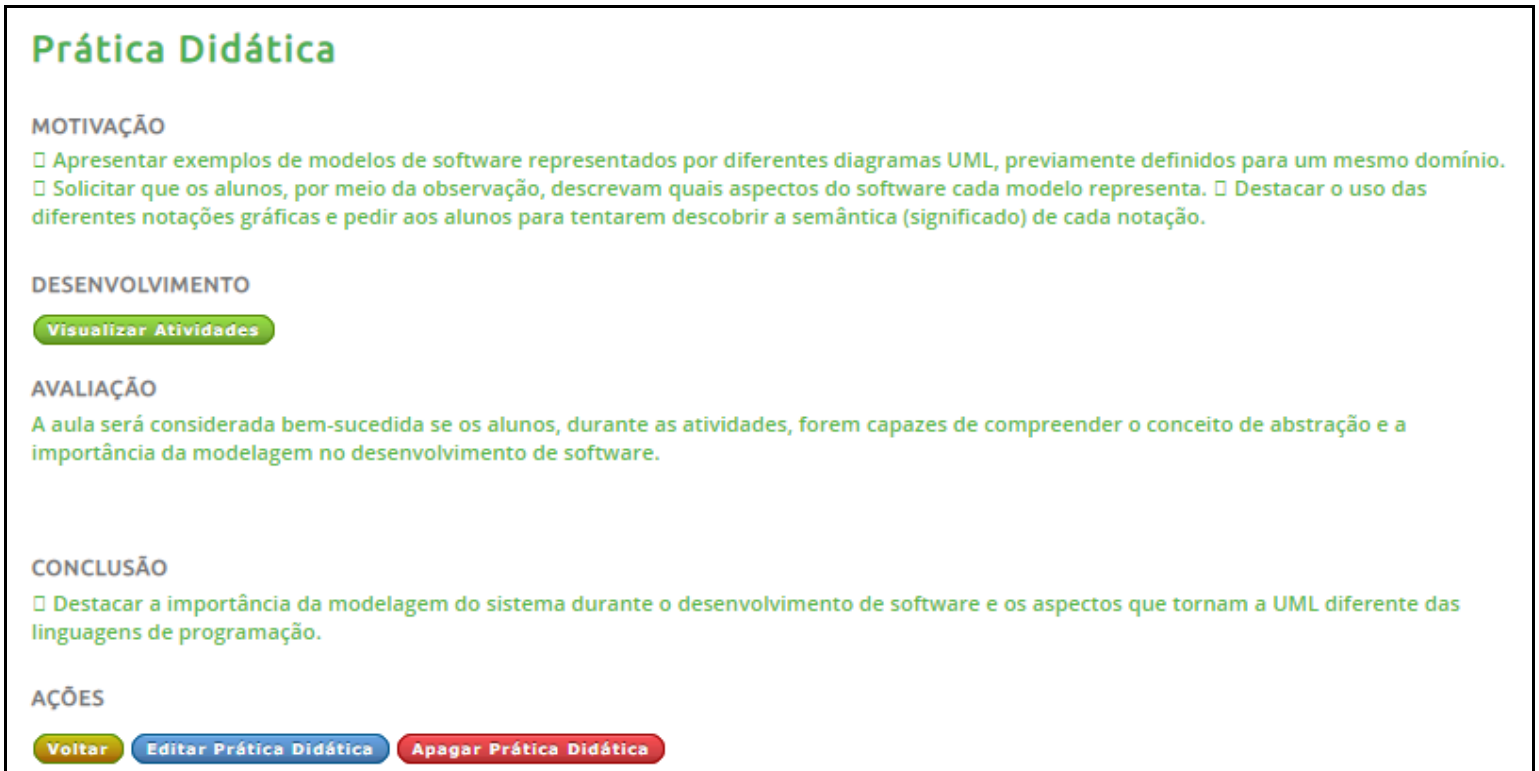
Título do Roteiro	Visibilidade	Ações
INTRODUÇÃO A ENGENHARIA DE SOFTWARE	Roteiro Público	Práticas Didáticas Editar Apagar

Adicionar Novo Roteiro

Após criar o roteiro, o professor tem a opção de continuar a descrição do roteiro criando uma prática didática ou apenas deixar nesse estado, finalizar em outro momento e continuar a navegar no sistema.

Neste Cenário de Uso, o professor seleciona a opção de criar prática didática, e o sistema solicita os dados da prática didática. Após a criação da prática didática o sistema atualiza os dados como mostrado na Figura 8.

Figura 8 - Tela de criação da prática didática



Prática Didática

MOTIVAÇÃO

- Apresentar exemplos de modelos de software representados por diferentes diagramas UML, previamente definidos para um mesmo domínio.
- Solicitar que os alunos, por meio da observação, descrevam quais aspectos do software cada modelo representa. □ Destacar o uso das diferentes notações gráficas e pedir aos alunos para tentarem descobrir a semântica (significado) de cada notação.

DESENVOLVIMENTO

[Visualizar Atividades](#)

AVALIAÇÃO

A aula será considerada bem-sucedida se os alunos, durante as atividades, forem capazes de compreender o conceito de abstração e a importância da modelagem no desenvolvimento de software.

CONCLUSÃO

- Destacar a importância da modelagem do sistema durante o desenvolvimento de software e os aspectos que tornam a UML diferente das linguagens de programação.

AÇÕES

[Voltar](#) [Editar Prática Didática](#) [Apagar Prática Didática](#)

Após a criação da motivação, conclusão e avaliação da prática didática, é necessário criar o desenvolvimento desta prática, ou seja definir as atividades que serão realizadas em sala de aula, ou fora dela, onde é possível adicionar material de apoio para cada atividade, como mostra a Figura 9.

Figura 9 - Tela de criação de atividades

Prática Didática

MOTIVAÇÃO

Apresentar exemplos de modelos de software representados por diferentes diagramas UML, previamente definidos para um mesmo domínio.

Solicitar que os alunos, por meio da observação, descrevam quais aspectos do software cada modelo representa. Destacar o uso das diferentes notações gráficas e pedir aos alunos para tentarem descobrir a semântica (significado) de cada notação.

DESENVOLVIMENTO

DESCRIÇÃO DA ATIVIDADE

Pedir Que Os Alunos Façam Uma Pesquisa Rápida Na Web, Ou Nas Referências, Sobre A Linguagem UML, Com Foco Principalmente Nos Tópicos: História Do Surgimento Da Linguagem; Quantos E Quais São Os Diagramas Da UML; E Qual A Função De Cada Um Deles.

Com Base Nos Resultados Obtidos Pedir Que Os Alunos Preencham O Formulário

TIPO DE ATIVIDADE

Atividade de Classe

Atividade Extraclasse

Atividade Complementar

AÇÕES

MATERIAL DE APOIO

anexo.PNG

AValiação

A aula será considerada bem-sucedida se os alunos, durante as atividades, forem capazes de compreender o conceito de abstração e a importância da modelagem no desenvolvimento de software.

CONCLUSÃO

Destacar a importância da modelagem do sistema durante o desenvolvimento de software e os aspectos que tornam a UML diferente das linguagens de programação.

Para material de apoio o sistema aceita arquivos das extensões .pdf, .jpeg, .png, .docx, .xls, .ppt e .mp4 de até 15MB. Desta forma, o professor fica livre para usar quaisquer recursos que desejar, podendo ser vídeo, documento ou imagem. Assim como o tipo de atividade pode ser definido no momento de sua criação, isto é, se trata de uma atividade para ser executada em sala de aula (Classe), executada fora do horário da aula (Extraclasse) ou se trata da atividade complementar (Atividade Complementar). É possível ainda, realizar o download dos arquivos anexados a cada atividade. Assim, o professor tem acesso à todo material de apoio.

Como conclusão deste Cenário, agora é possível agora visualizar todo o roteiro criado com seus campos preenchidos, como mostrado na Figura 10.

Figura 10 - Tela de roteiro criado

Prática Didática

MOTIVAÇÃO

□ Apresentar exemplos de modelos de software representados por diferentes diagramas UML, previamente definidos para um mesmo domínio. □ Solicitar que os alunos, por meio da observação, descrevam quais aspectos do software cada modelo representa. □ Destacar o uso das diferentes notações gráficas e pedir aos alunos para tentarem descobrir a semântica (significado) de cada notação.

DESENVOLVIMENTO

DESCRIÇÃO DA ATIVIDADE

□ Pedir que os alunos façam uma pesquisa rápida na Web, ou nas referências, sobre a linguagem UML, com foco principalmente nos tópicos: história do surgimento da linguagem; quantos e quais são os diagramas da UML; e qual a função de cada um deles.

MATERIAL DE APOIO
anexo.PNG

TIPO DA ATIVIDADE
Classe

Apagar

DESCRIÇÃO DA ATIVIDADE

□ Pedir que os alunos criem uma definição para a linguagem UML com base na pesquisa realizada

MATERIAL DE APOIO
Texto_Tela_Aguarde.PNG

TIPO DA ATIVIDADE
Classe

Apagar

AVALIAÇÃO

A aula será considerada bem-sucedida se os alunos, durante as atividades, forem capazes de compreender o conceito de abstração e a importância da modelagem no desenvolvimento de software.

CONCLUSÃO

□ Destacar a importância da modelagem do sistema durante o desenvolvimento de software e os aspectos que tornam a UML diferente das linguagens de programação.

AÇÕES

[Voltar](#) [Editar Prática Didática](#) [Apagar Prática Didática](#)

5.2. CENÁRIO DE USO 2

Para este cenário de teste, consideramos que o professor (identificado no sistema como Usuario2_Comum) possui três roteiros criados, sendo dois deles públicos e um privado, ou seja, que apenas o próprio professor tem acesso, como ilustra a Figura 11. Neste cenário, um segundo professor acessa o sistema realizando login como usuário “Usuario3_Comum”.

Figura 11 - Todos roteiros criados pelo usuário " Usuario2_Comum"

Título do Roteiro	Visibilidade	Ações		
Título 2	Roteiro Público	Práticas_Didáticas	Editar	Apagar
Título 2	Roteiro Público	Práticas_Didáticas	Editar	Apagar
INTRODUÇÃO À ENGENHARIA DE SOFTWARE	Roteiro Privado	Práticas_Didáticas	Editar	Apagar

Adicionar Novo Roteiro

Ao realizar login, o professor “Usuario3_Comum” visualiza os roteiros públicos cadastrados no sistema como mostra a Figura 12..

Figura 12 - Tela listagem de roteiros públicos

Título do Roteiro	Visibilidade	Ações
INTRODUÇÃO À ENGENHARIA DE SOFTWARE	Roteiro Público	Práticas Didáticas
PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	Roteiro Público	Práticas Didáticas

Após essa tela, o professor Usuário3_Comum poderá ter acesso à prática didática dos roteiros públicos exibidos na listagem, assim como as atividades e materiais de apoio.

5.3. CENÁRIO DE USO 3

Neste cenário, o usuário do tipo administrador, ou seja que tem acesso à todas as funcionalidades do sistema que inclui “Gerenciar de Usuários” onde o administrador pode visualizar, editar, criar ou excluir quaisquer usuário cadastrado no sistema, além da

funcionalidade “Gerenciar Áreas de atuação” e “Gerenciar Instituições”, que assim como a funcionalidade anterior, o administrador poderá realizar quaisquer ações sobre os dados referentes à área de atuação e instituições.

Na Figura 13 é possível visualizar a tela de gerenciar instituições por exemplo. É exibida uma listagem contendo todas as instituições cadastradas no sistema, além das opções de “editar”, “apagar” ou criar uma nova instituição.

Figura 13 - Tela do usuário "Administrador" - Gerenciar Instituições



Ao realizar quaisquer alterações nesses dados, os usuários que pertencerem às instituições alteradas terão este dado alterado.

Na Figura 14, o usuário poderá editar o nome da instituição que está cadastrada como “UNIVERSIDADE DO ESTADO MENTAL”.

Figura 14 - Tela de editar uma instituição



Após a alteração de nome, é possível visualizar na listagem atualizada mostrada na Figura 15 agora com o nome “UNIVERSIDADE ALTERADA”.

Figura 15 - Tela de listagem de instituições (atualizada)



O administrador também pode apagar uma instituição qualquer da listagem, como mostrado na após a exclusão das instituições “UFF”, “UNIVERSIDADE B” como mostrado na Figura 16.

Figura 16 - Tela com listagem de instituições atualizada



5.4. CENÁRIO DE USO 4

Neste cenário, o mesmo usuário Administrador que foi utilizado na seção 5.3, irá gerenciar os usuários cadastrados no sistema, editando o nome e e-mail deste usuário.

Na Figura 17 é possível visualizar os três usuários cadastrados no sistema, sendo dois administradores e o outro usuário comum, bem como suas instituições, área de atuação e email.

Figura 17 - Tela de Gerenciar Usuários

Nome	E-mail	Instituição	Área de Atuação	Perfil	Ações
Administrador 1	emailAdmin@email.com	UNIVERSIDADE ALTERADA	Riscos	Admin	Editar Apagar
administrador 2	teste1@email.com	UNIVERSIDADE ALTERADA	Implementação	Admin	Editar Apagar
Usuario2_Comum	teste2@email.com	UNIVERSIDADE ESTADUAL DO RIO DE JANEIRO	Requisitos	Comum	Editar Apagar

[Novo Usuário](#)

Ao clicar na opção de editar os dados do usuário comum “Usuario2_Comum”, o sistema torna todos seus dados editáveis, como é possível ser notado na Figura 18.

Figura 18 - Tela de edição dos dados de um usuário

Usuários

NOME
Usuario2_Comum

E-MAIL
Teste2@Email.Com

SENHA
.....

PERFIL
 Administrador
 Usuario Comum

INSTITUIÇÃO
UNIVERSIDADE ESTADUAL DO RIO DE JANEIRO

ÁREA DE ATUAÇÃO
Requisitos

AÇÕES
[Cancelar](#) [Salvar](#)

Após a alteração dos dados nome e email do usuário “Usuario2_Comum”, é possível visualizar na Figura 19 a listagem agora atualizada com o nome alterado para “USUARIOCOMUM_ALTERADO” e o email atualizado para “EMAILCOMUMALTERADO@email.com”.

Figura 19 - Tela de listagem de usuário atualizada

Olá administrador!

Home Meus Roteiros Roteiros Públicas Gerenciar Usuários Gerenciar Áreas de Atuação Gerenciar Instituições Encerrar Sessão

Nome	E-mail	Instituição	Área de Atuação	Perfil	Ações
Administrador 1	emailAdmin@email.com	UNIVERSIDADE ALTERADA	Riscos	Admin	Editar Apagar
administrador 2	teste1@email.com	UNIVERSIDADE ALTERADA	Implementação	Admin	Editar Apagar
USUARIOCOMUM_ALTERADO	EMAILCOMUMALTERADO@email.com	UNIVERSIDADE ESTADUAL DO RIO DE JANEIRO	Requisitos	Comum	Editar Apagar

[Novo Usuário](#)

5.5. CASOS DE TESTE

Foram descritos e executados casos de testes para as funcionalidades de Gerenciar Usuários, Gerenciar Instituição, Gerenciar Áreas de Atuação, Criar Roteiro, Criar Práticas didáticas e Criar Atividades, as principais funcionalidades desenvolvidas no protótipo. O apêndice A descreve, inicialmente, todos os casos de teste desenvolvidos e executados utilizando o protótipo desenvolvido. .

Durante a execução dos casos de teste algumas telas sofreram alteração devido à poluição visual que causavam, a exemplo da exibição da listagem dos roteiros, onde eram carregados e exibidos todos os dados de cada roteiro da listagem, incluindo Objetivos Específicos e Conteúdo Básico. Após a alteração, as listagens exibem apenas o título, e a visibilidade do roteiro, dando mais clareza na informação. Desta forma, caso o usuário deseje visualizar todos os dados de um determinado roteiro, poderá entrar e ter acesso à tudo.

Outras funcionalidades precisam de ajustes e refinamento, como a busca através de filtros ou palavras-chave por exemplo.

6. TRABALHOS RELACIONADOS

Para que este trabalho tivesse início, foi feito um breve levantamento de trabalhos voltados para o ensino superior em geral, e em engenharia de software. Muitos projetos de diferentes áreas foram encontrados, mas nenhum com a proposta semelhante à deste trabalho. Este levantamento se deu através de pesquisas em artigos acadêmicos e de bibliografias da área.

Grande parte dos trabalhos relacionados basicamente apresentam plataformas que funcionam como sistemas de apoio à comunicação entre alunos e professores. Em alguns casos oferecem suporte para que professores disponibilizem material didático, como é o caso da Plataforma Moodle de aprendizagem a distância baseada em software livre. Seu sistema de gestão do ensino e aprendizagem (conhecidos por suas siglas em inglês, LMS - *Learning Management System*, ou CMS - *Course Management System*), é um aplicativo desenvolvido para ajudar os educadores a criar cursos on-line, ou suporte on-line a cursos presenciais, de alta qualidade e com muitos tipos de recursos disponíveis (SABBATINI, 2007).

O sistema SWEnet se assemelha muito com a proposta deste trabalho, uma vez que o objetivo do projeto é fornecer um repositório acessível e expansível de materiais educativos de engenharia de software (LUTZ, 2003). No entanto, se difere do protótipo apresentado neste trabalho, pois o objetivo é que educadores e alunos tenham acesso ao material presente no SWEnet, enquanto no SACPD o objetivo é que o conteúdo presente seja acessado apenas por professores, tendo assim outro formato.

Durante a pesquisa, também foi encontrado um trabalho de grande relevância e semelhante à proposta do guia pensado inicialmente para este projeto. O “*Teaching a Project-Intensive Introduction to Software Engineering*” (TOMAYKO e JAMES, 1987) é um trabalho no qual os autores relatam suas experiência durante a aplicação de uma nova proposta de ensino. Esta proposta tem uma estrutura semelhante aos roteiros propostos neste trabalho, divididos em aulas práticas de temas específicos. No entanto, na proposta de Tomayko e James (1987) há dependência entre as aulas, por exemplo, dados da atividade da aula cinco podem depender de dados obtidos na aula quatro ou três. Enquanto que a ideia do SACPD é que os roteiros sejam independentes entre si.

7. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho propõe um sistema para apoiar o compartilhamento de práticas didáticas para o ensino de Engenharia de Software. O objetivo é estimular professores a definir e compartilhar práticas utilizadas em sala de aula visando tornar as aulas de Engenharia de Software mais dinâmicas e motivar os alunos a buscar informações e construir seu conhecimento de forma individual e também coletiva. O sistema apresentado visa auxiliar os professores na definição de seus roteiros de aula e na documentação e compartilhamento de suas práticas didáticas utilizadas em sala. Desta forma, além de conectar professores com finalidades comuns, auxilia a troca de experiências e de ideias. Por exemplo, uma prática didática que funciona muito bem com uma turma de quinze alunos pode não funcionar tão bem em uma turma de trinta alunos, e isto poderá ser documentado.

Como trabalho futuro estão o refinamento e evolução do sistema desenvolvido, como exemplo a funcionalidade de usuários comentarem e avaliarem práticas didáticas de outros usuários. O sistema também precisa da inclusão de filtros nas buscas para exibições de listagens assim como permitir que usuário incorpore práticas didáticas criadas por outros usuários nos seus próprios roteiros, funcionalidades que estiveram presentes no escopo deste trabalho mas não foram possíveis de serem concluídas.

Como trabalho futuro também fica proposta a realização de um experimento envolvendo professores de diferentes instituições para o protótipo e as ideias apresentadas neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

ASTAH: ferramenta CASE. Version 7.2. [S.l.]: ChangeVision, 2017. Disponível em: <<http://astah.net/>>. Acesso em: 20 set. 2018.

BAKER, Alex; NAVARRO, Emily Oh; VAN DER HOEK, Andre. **An experimental card game for teaching software engineering processes**. Journal of Systems and Software, v. 75, n. 1-2, p. 3-16, 2005.

BALSAMIQ: Mockups. Version 3.5.16. [S.l.]: Balsamiq Studios, LLC, 2018. Disponível em: <<https://balsamiq.com/download/>>. Acesso em: 20 mar. 2018.

BERNHEIM, Carlos Tünnerman; DE SOUZA CHAUI, Marilena. **Desafios da universidade na sociedade do conhecimento**. Unesco, Brasília, 2008, p. 44.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. Elsevier Brasil, 2006, p 6.

DBDESIGNER. Version 4. [S.l.]: DbDesigner.net, 2018. Disponível em: <<https://www.dbdesigner.net/>>. Acesso em: 18 jan. 2018.

CAMARGO, Álvaro Antônio Bueno De. KHOURI, Lourdes Halim El e GIAROLA, Paulo César. **O Uso de Sistemas Colaborativos na Gestão de Projetos: Fatores Relevantes para o Sucesso**. Trabalho de Conclusão de Curso. Fundação Instituto de Administração – FIA. 2005.

CASTANHA, Débora; DE CASTRO, Maria Bernadete. **A necessidade de refletir sobre as estratégias pedagógicas para atender à aprendizagem da Geração Y**, Revista de educação do Cogeme, v. 19, n. 36, p. 27-38, 2010.

COLLEMAN, David. **Groupware: Collaborative Strategies for corporate LANs and Intranets** – Prentice Hall – Prentice Hall PTR, 1997.

DA SILVA, Tarcila Gesteira; MÜLLER, Felipe Martins; BERNARDI, Giliane. **Panorama do ensino de engenharia de software em cursos de graduação focado em teste de software: uma proposta de aprendizagem baseada em jogos**. Novas Tecnologias na Educação, RENOTE, v. 9, n. 2, UFRGS, 2011.

ERICONE, Délcia. **Ser professor**. Edipucrs, Porto Alegre, 2001, p. 57-72.

KAMIENSKI, Carlos Alberto. **Introdução ao paradigma de Orientação a objetos** – João Pessoa, Centro Federal de Educação Tecnológica da Paraíba, 1996.

KARSNER, G. E.; POPE, S. T. **A cookbook for using the model view controller user interface paradigm in Smalltalk0e0**. Journal of Object0Orientated Program0 ming, v. 1, n. 3, p. 26-49, 1988.

LAPLANTE, Phillip A. **What every engineer should know about software engineering**. CRC Press, 2007.

LESSA, Rafael Orivaldo; LESSA JUNIOR, Edson Orivaldo. **Modelos de processos de engenharia de software**. Link para o PDF: http://xps-project.googlecode.com/svn-history/r43/trunk/outros/02_Artigo.pdf, 2009.

LUTZ, Michael et al. **The SWENET project: bridging the gap from bodies of knowledge to curriculum development**. In: *Frontiers in Education*. 2003. p. S3C-S37.

MYSQL, A. B. **MySQL reference manual**. Disponível em: <<https://dev.mysql.com/doc/refman/8.0/en/>>, 2001 Acesso em 12 jul 2018.

MANUAL, P. H. P. Php documentation group. **PHP Documentation**. Disponível em: <https://secure.php.net/manual/pt_BR/intro-whatish.php> . 2009. Acesso em 27 nov 2018.

OLIVEIRA, Carla de. **Sistemas Colaborativos: Conceito, Características e Funcionalidades**. Disponível em: <<http://imasters.com.br/artigo/4655/gerencia-de-ti/sistemas-colaborativos-conceito-caracteristicas-e-funcionalidades/>>. 2014 Acessado em 05/02/2016.

OMG, Object Management Group. **“OMG Unified Modeling Language TM (OMG UML), Superstructure – Version 2.5.1. Standard document”**. Disponível em: <<https://www.omg.org/spec/UML/2.5.1/>> . 2017. Acesso em 13 nov 2018.

PRESSMAN, Roger S. **Engenharia de Software**. 7ª ed. São Paulo: McGraw-Hill, 2011, p. 39

SABBATINI, Renato ME. **Ambiente de ensino e aprendizagem via Internet: a Plataforma Moodle**. Instituto EduMed, v. 7, 2007.

SOMMERVILLE, I. **Software Engineering** (International Computer Science Series). 5ª Edição. Reading: Addison-Wesley, 1995.

SOMMERVILLE, Ian; ARAKAKI, Reginaldo; MELNIKOFF, Selma Shin Shimizu. **Engenharia de software**. 9ª ed. São Paulo: Pearson Prentice Hall, 2011.

VELOSO, Elza Fátima Rosa; DUTRA, Joel Souza; NAKATA, Lina Eiko. **Percepção sobre carreiras inteligentes: diferenças entre as gerações y, x e baby boomers**. REGE-Revista de Gestão, v. 23, n. 2, p. 88-98, 2016.

TERADA, Routh. **Segurança de Dados**, Blucher 2ª ed. 2011

TOMAYKO, James E. **Teaching a project-intensive introduction to software engineering**. Carnegie-mellon university Pittsburgh, software engineering inst, 1987.

TIJIBOY, Ana Vilma et al. **Aprendizagem cooperativa em ambientes telemáticos. Informática na Educação: teoria & prática**. Porto Alegre. Vol. 1, n. 2 (abr. 1999), p. 19-28, 1999.

WAZLAWICK, Raul. **Engenharia de software: conceitos e práticas**. Elsevier Brasil, 2013.

APÊNDICE A – CASOS DE TESTES E RESULTADOS

1.1. Criação de um novo roteiro

Pré-condições: Usuário ter acesso às funcionalidades do sistema.

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste1@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas
2.0	Selecionar a opção “Meus Roteiros”	O sistema irá exibir uma listagem com todos os títulos, visibilidade e ações (visualizar práticas didáticas, editar, apagar) a serem tomadas em cada roteiro
3.0	Selecionar a opção “Novo Roteiro”	O sistema exibe os campos para inserção dos dados do novo roteiro: Campos = título do roteiro , objetivos específicos, conteúdo básico, palavras-chave do roteiro, visibilidade do roteiro e ações (salvar ou cancelar)
4.0	Inserir dados do novo roteiro e apertar em “Salvar” Título do roteiro: “Modelagem de Sistemas de Software e Apresentação da UML” Objetivos específicos: “Apresentar o conceito de abstração em diferentes níveis. Abordar o histórico do surgimento da linguagem UML e suas principais características.” Conteúdo Básico: “Modelagem é uma prática comum compartilhada pela maioria das disciplinas de engenharia...” Palavras-chave: “Modelagem” Visibilidade: Roteiro Público Salvar	Sistema irá armazenar os dados do novo roteiro

1.2. Cadastro de uma nova prática didática

Pré-condições:

- Usuário ter acesso às funcionalidades do sistema.
- Existir um roteiro já criado para associar-se à nova prática

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste1@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas
2.0	Selecionar a opção “Meus Roteiros”	O sistema irá exibir uma listagem com todos os títulos, visibilidade e ações (visualizar práticas didáticas,

		editar, apagar) a serem tomadas em cada roteiro criado pelo usuário
3.0	Localizar no roteiro desejado, o botão de ação de cadastrar nova prática didática	O sistema solicitará os dados (Motivação, Avaliação e Conclusão) da nova prática didática
4.0	<p>Inserir dados :</p> <p>Motivação: “Apresentar diferentes modelos com diferentes abstrações de um mesmo objeto que pode ser, por exemplo, uma casa ou um automóvel.”</p> <p>Conclusão: Destacar a importância da modelagem do sistema durante o desenvolvimento de software e os aspectos que tornam a UML diferente das linguagens de programação.</p> <p>Avaliação: A aula será considerada bem-sucedida se os alunos, durante as atividades, forem capazes de compreender o conceito de abstração e a importância da modelagem no desenvolvimento de software.</p> <p>- apertar em “Salvar”</p>	Sistema irá armazenar os dados da nova prática didática

1.3. Cadastro de uma nova atividade

Pré-condições:

- Usuário ter acesso às funcionalidades do sistema.
- Existir uma prática didática já criada para associar-se à(s) nova(s) atividade

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste1@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas
2.0	Selecionar a opção “Meus Roteiros”	O sistema irá exibir uma listagem com todos os títulos, visibilidade e ações (visualizar práticas didáticas, editar, apagar) a serem tomadas em cada roteiro criado pelo usuário
3.0	Localizar no roteiro desejado, o botão de ação de visualizar prática didática	O sistema exibe os dados (Motivação, Conclusão e Avaliação) da prática didática
4.0	Localizar na prática didática o botão de ação de cadastrar atividade	Sistema solicita a descrição, o material de apoio da atividade e o tipo da atividade
5.0	<p>Inserir os dados da atividade</p> <p>Descrição: “Apresentar exemplos de modelos de software representados por diferentes diagramas UML, previamente definidos para um mesmo domínio.</p> <p>Solicitar que os alunos, por meio da observação, descrevam quais aspectos do</p>	Sistema armazena os dados inseridos e exibe a tela com todos os dados do roteiro (título, objetivos específicos, conteúdo básico e visibilidade do roteiro), prática didática (motivação, avaliação e conclusão e dados das atividades (descrição, material de apoio e tipo de atividade)

	<p>software cada modelo representa.</p> <p>Destacar o uso das diferentes notações gráficas e pedir aos alunos para tentarem descobrir a semântica (significado) de cada notação.”</p> <p>Material de apoio: Diagrama UML.png</p> <p>Tipo da Atividade: Classe</p>	
--	---	--

1.4. Realizar download de material de apoio de uma atividade

Pré-condições:

- Usuário ter acesso às funcionalidades do sistema;
- Existir uma atividade com material de apoio anexado já cadastrada

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste1@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas
2.0	Selecionar a opção “Meus Roteiros”	O sistema irá exibir uma listagem com todos os títulos, visibilidade e ações (visualizar práticas didáticas, editar, apagar) a serem tomadas em cada roteiro criado pelo usuário
3.0	Localizar no roteiro desejado, o botão de ação de visualizar prática didática	O sistema exibe os dados (motivação, conclusão e avaliação) da prática didática
4.0	Localizar na prática didática o botão de ação de visualizar atividade	Sistema exibe a descrição, o material de apoio e o tipo de atividade
5.0	Clicar sobre o nome.extensão do arquivo para realizar download	Sistema realiza download do arquivo material de apoio

1.5. Cadastro de uma nova instituição

Pré-condições:

- Usuário ser do tipo administrador;

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste2@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas pelo administrador
2.0	Selecionar a opção “Gerenciar instituições”	O sistema irá exibir uma listagem com todos os nomes de todas as instituições cadastradas no sistema
3.0	Selecionar a opção de “Nova Instituição”	O sistema solicita o nome da nova instituição
4.0	Inserir o nome da nova instituição Nome: UNIVERSIDADE B	Sistema armazena o nome da nova instituição e exibe a tela de listagem de instituições agora atualizada

1.6. Editar uma instituição já cadastrada

Pré-condições:

- Usuário ser do tipo administrador
- Existir pelo menos uma instituição já cadastrada no sistema

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste2@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas pelo administrador
2.0	Selecionar a opção “Gerenciar instituições”	O sistema irá exibir uma listagem com o nome de todas as instituições já cadastradas no sistema
3.0	Localizar a instituição a ser editada Nome: UNIVERSIDADE B e selecionar o botão de ação “editar”	O sistema torna o nome da instituição editável e solicita a atualização
4.0	Inserir o novo nome da instituição Novo nome: UNIVERSIDADE C	Sistema armazena atualiza a tabela de nomes e exibe a tela de listagem de instituições agora atualizada

1.7. Excluir uma instituição já cadastrada

Pré-condições:

- Usuário ser do tipo administrador
- Existir pelo menos uma instituição já cadastrada no sistema

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste2@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas pelo administrador
2.0	Selecionar a opção “Gerenciar instituições”	O sistema irá exibir uma listagem com o nome de todas as instituições já cadastradas no sistema
3.0	Localizar a instituição a ser editada Nome: UNIVERSIDADE C e selecionar o botão de ação “deletar”	O Sistema realiza a ação de deletar a instituição selecionada, e atualiza a tabela de nomes e exibe a tela de listagem de nomes de instituições agora atualizada

1.8. Criar um usuário administrador

Pré-condições:

- Usuário ser do tipo administrador

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste2@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas pelo administrador
2.0	Selecionar a opção “Gerenciar usuários”	O sistema irá exibir uma listagem com o nome, email, instituição, área de atuação, senha e tipo de todos os usuários cadastradas no sistema
3.0	Selecionar a opção de “Novo usuário”	O Sistema solicita os dados (nome,email, área de atuação, instituição e tipo do novo usuário

4.0	Inserir os dados solicitados e salva Nome: Usuário1_Comum Email: email2@email.com Instituição: UNIVERSIDADE C Senha: “123456” Tipo: Usuário Comum	O sistema armazena os dados, atualiza a base de usuário e retorna a listagem de usuário agora atualizada
-----	---	--

1.9. Editar os dados de um usuário

Pré-condições:

- Usuário ser do tipo administrador

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste2@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas pelo administrador
2.0	Selecionar a opção “Gerenciar usuários”	O sistema irá exibir uma listagem com o nome, email, insituição, área de atuação, senha e tipo de todos os usuários cadastradas no sistema
3.0	Localizar o usuário desejado e clicar no botão de ação “Editar” Nome: Usuário1_Comum	O Sistema torna os dados do usuário editáveis e solicita os novos dados
4.0	Inserir os novos dados solicitados e salvar Novo Nome: Usuario1_ComumAlterado Novo Email: emailalterado@email.com	O sistema armazena os dados, atualiza a base de usuário e retorna a listagem de usuário agora atualizada

1.10. Excluir um usuário

Pré-condições:

- Usuário ser do tipo administrador

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste2@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas pelo administrador
2.0	Selecionar a opção “Gerenciar usuários”	O sistema irá exibir uma listagem com o nome, email, insituição, área de atuação, senha e tipo de todos os usuários cadastradas no sistema
3.0	Localizar o usuário desejado e clicar no botão de ação “Excluir” Nome: Usuario1_ComumAlterado	O Sistema excluir o usuário selecionado atualiza a base de usuário e retorna a listagem de usuário agora atualizada

1.11. Cadastro de uma nova área de atuação

Pré-condições:

- Usuário ser do tipo administrador;

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste2@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas pelo administrador
2.0	Selecionar a opção “Gerenciar áreas de atuação”	O sistema irá exibir uma listagem com o nome de todas as áreas de atuação já cadastradas no sistema
3.0	Selecionar a opção de “Nova área de atuação”	O sistema solicita o nome da nova área de atuação
4.0	Inserir o nome da nova área de atuação Nome: Requisitos de Sistema	Sistema armazena o nome da nova área de atuação e exibe a tela de listagem agora atualizada

1.12. Editar uma área de atuação já cadastrada

Pré-condições:

- Usuário ser do tipo administrador
- Existir pelo menos uma área de atuação já cadastrada no sistema

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste2@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas pelo administrador
2.0	Selecionar a opção “Gerenciar Áreas de atuação”	O sistema irá exibir uma listagem com o nome de todas as Áreas de atuação já cadastradas no sistema
3.0	Localizar a Área de atuação a ser editada e selecionar o botão de ação “editar” Nome: Requisitos de Sistema	O sistema torna o nome da Área de atuação editável e solicita a atualização
4.0	Inserir o novo nome da Área de atuação Novo Nome: Requisitos de Sistema alterado	Sistema armazena atualiza a tabela de nomes e exibe a tela de listagem de Áreas de atuação agora atualizada

1.13. Excluir uma área de atuação já cadastrada

Pré-condições:

- Usuário ser do tipo administrador
- Existir pelo menos uma área de atuação já cadastrada no sistema

Passo	Ação	Resultado Esperado
1.0	Entrar com login = “teste2@email.com” e senha = “123456”.	Sistema exibe tela com todas as funcionalidades possíveis de serem acessadas pelo administrador
2.0	Selecionar a opção “Gerenciar áreas de atuação”	O sistema irá exibir uma listagem com o nome de todas as áreas de atuação já cadastradas no sistema
3.0	Localizar a área de atuação a ser deletada e selecionar o botão de ação “deletar” Nome: Requisitos de Sistema alterado	O Sistema realiza a ação de deletar a área de atuação selecionada, e atualiza a tabela de nomes e exibe a tela de listagem de insituições agora atualizada