

Análise e Desenvolvimento de um Sistema para Gestão de Associações: uma aplicação prática da Engenharia de Software

Raquel S. Oliveira¹, Rafaela C. Silva¹, Frank W. Cardoso²,
Marcelo F. Terenciani²

¹Acadêmicos do Curso de Bacharelado em Engenharia de Software
Campus Paranavaí – Instituto Federal do Paraná (IFPR)

²Docentes do Eixo de Informação e Comunicação
Campus Paranavaí – Instituto Federal do Paraná (IFPR)

{rafaelacs.rc, raquel.dossantos1965}@gmail.com
{frank.willian, marcelo.terenciani}@ifpr.edu.br

Abstract. *Associations, as institutions, often face managerial problems that could be easily solved by means of computer systems. Therefore, this article aims to present the analysis and development of the first version of an association management system. Therefore, a case study and a survey of related works were carried out in order to understand how the solutions related to this project are. Furthermore, the Scrum methodology combined with the pair programming of the Extreme Programming methodology - XP was used. The developed features were made available in a test environment, where they could be tested and analyzed during requirements extraction. In addition, it is intended to deploy the solution in a real environment, where it will be possible to better assess its behavior and, consequently, improve it continuously.*

Key-words: *analyze, development, association management.*

Resumo. *As associações, enquanto instituições, frequentemente deparam-se com problemas gerenciais que poderiam ser facilmente resolvidos por meio de sistemas computacionais. Sendo assim, esse artigo tem como objetivo apresentar a análise e o desenvolvimento da primeira versão de um sistema para gestão de associações. Para tanto, foi realizado um estudo de caso e o levantamento de trabalhos correlatos de modo compreender como estão as soluções correlatas a este projeto. Além disso, foi utilizado a metodologia Scrum combinada com a programação em pares da metodologia Programação Extrema - XP. As funcionalidades desenvolvidas foram disponibilizadas em um ambiente de testes, onde puderam ser testadas e analisadas durante a extração de requisitos. Além disso, pretende-se implantar a solução em um ambiente real, onde será possível avaliar melhor seu comportamento e, consequentemente melhorá-la continuamente.*

Palavras-chave: *análise, desenvolvimento, gestão de associação.*

1. Introdução

De acordo com o [IBGE – Instituto Brasileiro de Geografia e Estatística 2016] existem aproximadamente 230 mil fundações privadas e associações sem fins lucrativos, sendo

que 18.055 estão no Estado do Paraná. Essas entidades oferecem uma grande variedade de serviços à sociedade. As associações, foco deste trabalho, agregam uma série de modelos de organização que possuem objetivos e finalidades diferentes entre si, mas que se unem sob essa nomenclatura por possuírem características básicas semelhantes [SEBRAE – Serviço Brasileiro de Apoio às Micro e Pequenas Empresas 2013].

Essas características são: (i) reunião de duas ou mais pessoas para a realização de objetivos comuns; (ii) patrimônio constituído pela contribuição dos associados, por doações, por subvenções, etc.; (iii) os fins podem ser alterados pelos associados; (iv) os associados tomam decisões livremente; (v) são entidades do direito privado e não público [SEBRAE – Serviço Brasileiro de Apoio às Micro e Pequenas Empresas 2013].

As associações, enquanto instituições, frequentemente deparam-se com problemas gerenciais que poderiam ser facilmente resolvidos por meio de sistemas computacionais, entre eles destacam-se a gestão dos membros ativos e inativos, controle das mensalidades, publicidade das decisões, comunicação entre os associados, entre outras.

Nesse sentido, nos últimos anos observa-se uma preocupação cada vez maior acerca da informatização das atividades executadas por instituições públicas e privadas. Essa informatização busca promover maior celeridade nos processos, otimização de recursos humanos e financeiros, redução de custos, entre outros. Por isso, estas instituições procuram empresas especializadas para desenvolver sistemas que auxiliem na informatização de suas atividades.

O processo de desenvolvimento de um sistema inicia-se pela análise e projeto do sistema. Parece haver desorientação em como planejar e conduzir o processo de desenvolvimento de *software*. Os desenvolvedores concordam que não utilizam um processo adequado, mas não tem tempo e recursos financeiros para fazê-lo [Wazlawick 2019]. Essa falta de controle do processo de desenvolvimento acarreta em problemas quando por ventura acontecem mudanças na equipe de desenvolvimento, mudanças nos requisitos do *software*, controle de versões, atualizações de tecnologia, com isso os projetos acabam estourando o cronograma e orçamento, produtos de *software* não atendem aos requisitos ou são de baixa qualidade, e realizar manutenção não é uma tarefa simples nestes casos.

Considerando o ambiente de desenvolvimento de *softwares* dentro de uma instituição de ensino, sempre nos depararemos com rotatividade de estudantes, além de problemas de comunicação, validação e verificação de *software*. Sem falar nos aspectos técnicos, pois quando um estudante de primeiro ano ingressa em um projeto, geralmente, ele ainda não tem experiência suficiente para produzir *softwares*.

Neste sentido, é necessário realizar ações sistemáticas para evitar esses problemas, utilizando métodos e técnicas da engenharia de requisitos. Com isso espera-se que os problemas sejam reduzidos e até mesmo eliminados. Além dessas ações sistemáticas faz-se necessário a realização de treinamentos em linguagens de programação para melhorar os códigos produzidos.

Portanto este trabalho tem como objetivo apresentar os resultados do processo de desenvolvimento da primeira versão de um sistema para gestão de associações. Permitindo que outras instituições possam usufruir do sistema desenvolvido, considerando que é um software genérico que poderá ser adaptado para qualquer associação.

Este artigo está organizado da seguinte forma. A Seção 2 oferece uma apresentação da situação do mercado e das soluções semelhantes existentes. A fundamentação teórica sobre processo de desenvolvimento de *software* está na Seção 3. A Seção 4 relata os materiais e métodos adotados nesta pesquisa. A Seção 5 está destinado a apresentar o escopo e a arquitetura do sistema. A Seção 6 traz a organização dos requisitos funcionais e os principais serviços fornecidos pelo produto. Finalmente, conclusões e trabalhos futuros são apresentados na Seção 7.

2. Trabalhos Relacionados

Durante esta seção serão apresentados alguns sistemas que possuem uma proposta semelhante à aplicação descrita neste artigo.

Para essa pesquisa foram usadas as palavras-chave: associação, gestão e desenvolvimento como filtro para escolher um grupo específico de sistemas. Posteriormente deste grupo foram selecionados os sistemas com base nos requisitos funcionais em comum com gerenciamento de associações, como por exemplo, comunicação entre os associados, gestão de eventos, geração de boletos, cadastro e consulta de associados, entre outros. Os sistemas escolhidos foram: Associatec [SINCTEC SISTEMAS 2021], Call Tecnologia [Call Tecnologia 2021] e Clube Associados [CORE 2021].

O sistema Associatec é um sistema que atende novas associações, sindicatos e federações. Algumas das funcionalidades desse sistema são: gestão e a comunicação com os associados; emissão de cobrança das suas contribuições; gestão de eventos realizados pela associação e área privada para seus associados.

Já o sistema Call Tecnologia, também atende associações, federações, fundações e sindicatos. Esse sistema possui gerenciamento de associados, geração de boletos e todas contribuições praticadas pela entidade, bem como espaço para comunicação entre os associados. Além disso, o sistema conta com um embasamento jurídico financeiro.

Por fim, o sistema Clube Associados é voltado para associações, sendo suas principais funcionalidades: cadastro e consulta de sócios; gestão financeira completa; mural de informações sobre a associação; site institucional e área do associado.

Na busca realizada foram encontradas empresas que oferecem soluções para o domínio da gestão de associações, entretanto não foram encontrados trabalhos acadêmicos que abordassem aspectos da análise e desenvolvimento do sistema em si. Além disso, nenhum dos produtos ofertados pelas empresas são *open source*, assim sendo, destaca-se a relevância da aplicação desenvolvida neste trabalho para a área de gestão de associações.

3. Processo de Desenvolvimento de *Software*

O desenvolvimento de *software* como atividade profissional busca implementar programas que atendem à um propósito específico do negócio. Geralmente, para a criação dos programas é feita por equipes e durante toda a sua vida é necessário a realização de manutenções [Sommerville 2019].

Desenvolver programas não é uma tarefa simples, é necessário a utilização de técnicas e execução de atividades sistemáticas durante todo o processo de implementação do *software* em todas as suas formas e em todos os seus campos de aplicação, deve

passar pelos processos de engenharia, levando assim ao termo engenharia de *software* [Pressman and Maxim 2016].

A engenharia de *software* tem por objetivo apoiar o desenvolvimento profissional de *software*, ela inclui técnicas que apoiam especificação, projeto, implementação e evolução de programas [Sommerville 2019]. A engenharia de *software* foca em todos os aspectos da produção de sistemas, desde os estágios iniciais da especificação do sistema até sua manutenção, quando o sistema já está sendo usado [Sommerville 2019].

Em outras palavras, a engenharia de *software* foca no processo de desenvolvimento de *software*. Um processo de engenharia de *software* é formado por um conjunto de atividades interdependentes, com responsáveis e entradas e saídas definidas [Wazlawick 2019]. Em outras palavras, é um conjunto de regras que definem como um conjunto de atividades devem ser executadas.

Existem diversos modelos de processo de desenvolvimento de *software*, dentre os mais conhecidos estão os modelos Cascata, Prototipação, Incremental, Espiral [Schach 2009], e os métodos ágeis Scrum e XP [Sbrocco and Macedo 2012]. Os modelos possuem algumas atividades genéricas, são elas: especificação de *software*, projeto e implementação de *software*, validação e evolução [Sommerville 2019], conforme pode-se observar na Figura 1.



Figura 1. Atividades genéricas de um processo de desenvolvimento de *software*.

Fonte: elaborada pelos autores, baseado em [Sommerville 2019].

Em resumo, a engenharia de *software* se preocupa com todo o processo de desenvolvimento de *software*, com a realização da especificação do sistema junto ao cliente, com a gerência do processo, com a manutenção e, também com técnicas de implementação de *software*.

4. Materiais e Métodos

Esta seção apresenta os materiais e o método de desenvolvimento deste trabalho.

4.1. Materiais

A Tabela 1 traz a descrição dos materiais utilizados durante a análise, projeto e desenvolvimento, além disso cada um dos materiais utilizados possui sua descrição e para o que foi utilizado no âmbito deste projeto.

Tabela 1. Materiais utilizados durante a análise, projeto e desenvolvimento.

Ferramenta	Descrição
Trello	Plataforma colaborativa gratuita para o gerenciamento de projetos. Esta ferramenta organiza projetos em quadros (<i>boards</i>), em que são inseridas listas de tarefas a serem seguidas individualmente ou em equipe. Cada lista recebe cartões (<i>cards</i>) com descrições, prazos determinados e objetivos a serem concluídos.
Astah	Software para modelagem UML (<i>Unified Modeling Language</i> – Linguagem de Modelagem Unificada). Utilizado para desenvolver o Diagrama de Caso de Uso e o Diagrama de Classes presentes neste artigo.
Google Documentos	Pacote de aplicativos da Google. Esta ferramenta permite a criação, edição e visualização de documentos de textos e compartilhá-los com amigos e contatos profissionais. Esta plataforma foi utilizada para criar documentos colaborativos nos quais cada integrante da equipe pudesse editar o mesmo documento concomitantemente.
Google Meet	Plataforma da Google que permite profissionais fazerem reuniões de forma <i>on-line</i> . Esta ferramenta foi utilizada para fazer reuniões entre a equipe de desenvolvimento.
WhatsApp	Aplicativo gratuito que oferece um serviço de mensagens e chamadas simples, seguro e confiável para celulares em todo o mundo. Este aplicativo foi utilizado como meio de comunicação rápida entre a equipe de desenvolvimento.
Robo 3T	Sistema gerenciador de banco de dados não relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados. Utilizado para executar consultas, administrar o banco, criar e manter a base de dados através de um ambiente integrado.
Node	Plataforma de execução da linguagem JavaScript. Utilizado para criar aplicações sem depender do navegador (<i>browser</i>).
React	Biblioteca de código aberto para interfaces gráficas (GUI) que tem como objetivo tornar a experiência do usuário com a interface mais eficiente. Ela pode ser categorizada como o “V” no padrão MVC (<i>Model-View-Controller</i>). Utilizada para criar interfaces do usuário (UI).
VisualStudio Code	Editor de código-fonte desenvolvido pela Microsoft. Ele inclui suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe, complementação inteligente de código, <i>snippets</i> e refatoração de código. Utilizado para codificação do sistema.
Github	Gerenciador de repositório de <i>software</i> baseado em Git. Utilizado para manter os repositórios, bem como controlar o versionamento do código do sistema.

4.2. Método de Análise e Desenvolvimento de Projeto

A metodologia de desenvolvimento deste trabalho, acompanhou as atividades do processo de desenvolvimento de *software*. Portanto, iniciou-se pela especificação do sistema em questão. Nessa atividade, foram realizadas entrevistas *on-line* com um membro de uma associação da cidade de Maringá/PR. Essas entrevistas serviram para levantar os requisitos do sistema, isto é, as funcionalidades e restrições do *software*.

Além disso, em busca de compreender como estão as soluções correlatas a este projeto, foi realizado o levantamento de trabalhos correlatos e/ou soluções proprietárias por meio de pesquisas no Google utilizando as palavras-chave: associação, gestão e desenvolvimento. Mediante esta pesquisa e considerando os requisitos levantados por meio das entrevistas, foram encontrados três sistemas desenvolvidos por empresas que oferecem soluções para o domínio de gestão de associações. Entretanto, o que difere este trabalho dos outros é que nenhum dos produtos ofertados por essas empresas são *open source*.

Em seguida foi elaborada a documentação de requisitos identificados nas entrevistas. Além disso, foram criados o Diagrama de Caso de Uso (DCU) e o Diagrama de Classes (DC), da Linguagem de Modelagem Unificada (UML) [OMG 2021] do sistema em questão.

Durante a implementação do sistema, adotou-se o Scrum [Schwaber and Sutherland 2020] para reduzir os riscos de insucesso, entregando valor mais rápido e lidando com as inevitáveis mudanças de escopo, transformando-as em vantagens e melhorando a produtividade da equipe.

Em conjunto com o Scrum utilizou-se a programação em pares da metodologia Programação Extrema (*Extreme Programming - XP*). Na programação em par enquanto um programador implementa o código o outro revisa o que está sendo digitado, apontando problemas e pensando na solução como um todo. Após um determinado tempo invertem os papéis.

5. Abordagem Proposta

Esta seção apresenta o escopo geral do sistema, exemplifica a arquitetura de *software* utilizada, assim como, detalha os requisitos funcionais (RF) do sistema desenvolvido.

5.1. Escopo

As associações, enquanto instituições, frequentemente deparam-se com problemas gerenciais que poderiam ser facilmente resolvidos por meio de sistemas computacionais. Desse modo, o sistema desenvolvido tem como principal objetivo realizar o gerenciamento de associações. Sendo assim, esse sistema permite o cadastro de usuário, a comunicação entre os membros da associação, o gerenciamento de eventos e classificados, e a emissão de relatórios de pagamentos de mensalidades.

Vale destacar que o sistema em questão está organizado em duas áreas: a área institucional e a área privada/administrativa. Sendo que a área institucional servirá para que usuários ainda não associados visualizem informações gerais sobre a associação, tais como: notícias, eventos que já aconteceram ou que ainda irão acontecer, fotos e vídeos da associação, informações sobre classificados, entrar em contato com a associação, bem como o que fazer para associar-se.

Já a área privada/administrativa servirá para que os administradores além de gerenciar suas informações pessoais possam gerir as informações sobre a associação que serão apresentadas na área institucional.

Ademais, os associados também terão acesso a sua área privada/administrativa para que possam gerenciar suas informações pessoais. Deste modo, esse sistema trará mais agilidade e comodidade para associações que ainda realizam essas atividades manualmente.

5.2. Arquitetura

O foco deste trabalho é desenvolver um sistema de gestão de associações de modo que possa amenizar alguns problemas gerenciais comuns enfrentados por associações, tais como: a gestão dos membros ativos e inativos, controle das mensalidades, publicidade das

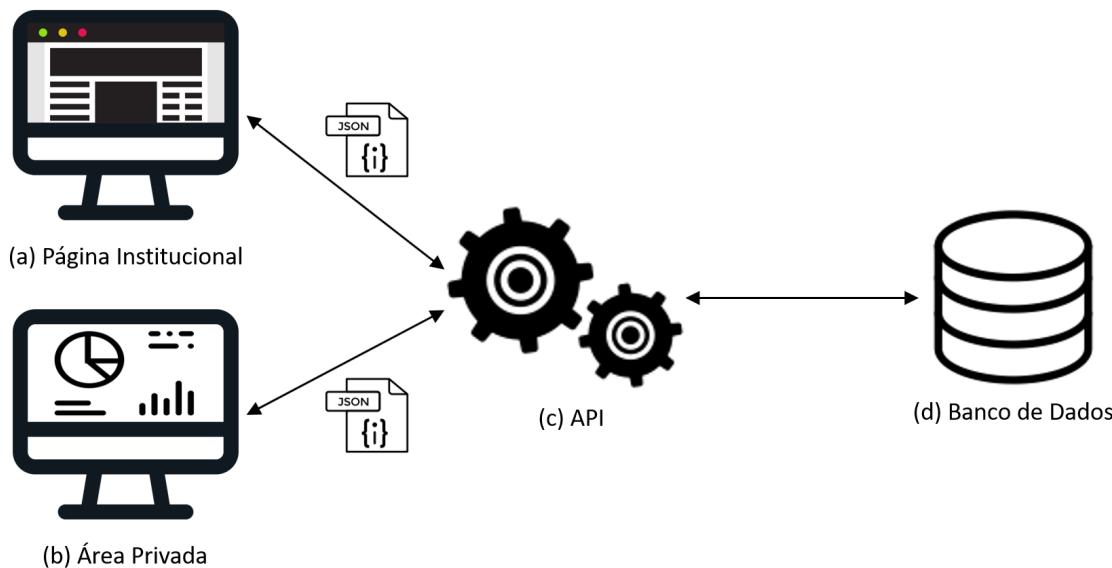


Figura 2. Arquitetura geral da plataforma proposta.

Fonte: elaborada pelos autores.

decisões, comunicação entre os associados, entre outras. Para solucionar esse problema, o projeto foi organizado de acordo com a arquitetura geral detalhada na Figura 2.

O sistema proposto possui uma página institucional pública (Figura 2a), logo não é necessário estar autenticado para acessar. Nessa página é possível entrar em contato com a associação, assim como visualizar as informações gerais da mesma. Além disso é possível autocadastrar-se para acessar a área privada (Figura 2b). Na área privada, o usuário (autenticado) pode visualizar suas informações e questões acerca da associação e alterar seu perfil.

A área privada possui níveis de acesso, assim sendo, além das funcionalidades supracitadas, um usuário com privilégios de administrador pode alterar as informações da página institucional, gerenciar eventos, fotos e vídeos, classificados e usuários, além de acompanhar as solicitações de novos associados. Nessa área também são oferecidos relatórios gerenciais para os administradores terem maior controle sobre mensalidades pagas e pendentes. Já um usuário com privilégios de associado poderá visualizar informações sobre o pagamento de suas mensalidades.

A página institucional e a área privada compartilham as informações por meio de uma Interface de Programação de Aplicação (API, acrônimo da expressão em inglês *Application Programming Interface*) (Figura 2c) com banco de dados MongoDB. A transferência dos dados entre os módulos é feita utilizando *JavaScript Object Notation* (JSON), tornando-os completamente independentes. Portanto, as manutenções e evoluções da plataforma não afetarão o projeto como um todo.

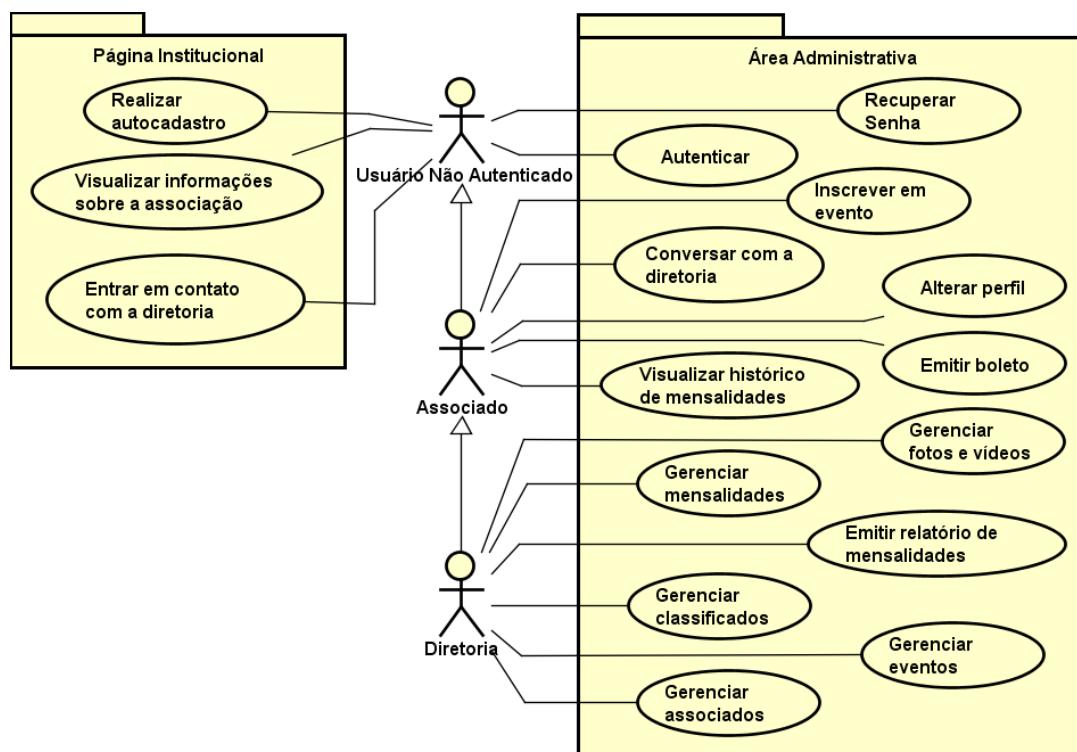
5.3. Requisitos

A Tabela 2 detalha os requisitos funcionais das áreas citadas na Figura 2.

Tabela 2. Requisitos funcionais.

Identificação	Descrição
RF01	O sistema deve permitir a visualização das informações gerais da associação para os usuários não autenticados.
RF02	O sistema deve permitir que novos usuários realizem autocadastro.
RF03	O sistema deve permitir que usuários não autenticados enviem mensagem de contato.
RF04	O sistema deve permitir que usuários recuperem sua senha informando o e-mail cadastrado.
RF05	O sistema deve permitir que usuários autenticados visualizem informações sobre suas mensalidades. Inclusive baixar os boletos.
RF06	O sistema deve permitir que os usuários autenticados alterem suas informações cadastrais.
RF07	O sistema deve permitir que os usuários autenticados alterem sua senha.
RF08	O sistema deve permitir que os associados tenham acesso a lista de membros da associação.
RF09	O sistema deve permitir que somente administradores alterem dados de outro associado.
RF10	O sistema deve permitir a troca de mensagens entre administradores e associados.
RF11	O sistema deve permitir que usuários administradores gerenciem usuários cadastrados.
RF12	O sistema deve permitir que usuários administradores gerenciem informações sobre as mensalidades dos associados.
RF13	O sistema deve permitir que usuários administradores modifiquem informações estáticas da página institucional.
RF14	O sistema deve permitir que usuários administradores gerencie informações sobre os eventos.
RF15	O sistema deve permitir que usuários administradores gerenciem informações dos classificados.
RF16	O sistema deve permitir que administradores gerenciem informações sobre notícias publicadas.
RF17	O sistema deve permitir que usuários administradores tenham acesso ao relatório de mensalidades pagas e pendentes dos associados.

A Figura 3 traz o Diagrama de Casos de Uso. Esse diagrama possibilita uma melhor compreensão de como os requisitos estão relacionados e, quais autores podem realizar as operações disponíveis no sistema.


Figura 3. Diagrama de casos de uso da plataforma proposta.

Já a Figura 4 apresenta o diagrama de classe, que segundo Bezerra [Bezerra 2015], tem o objetivo de definir a estrutura das classes utilizadas pelo sistema, determinando as características e ações que cada classe tem, além de estabelecer como se relacionam e trocam informações entre si.

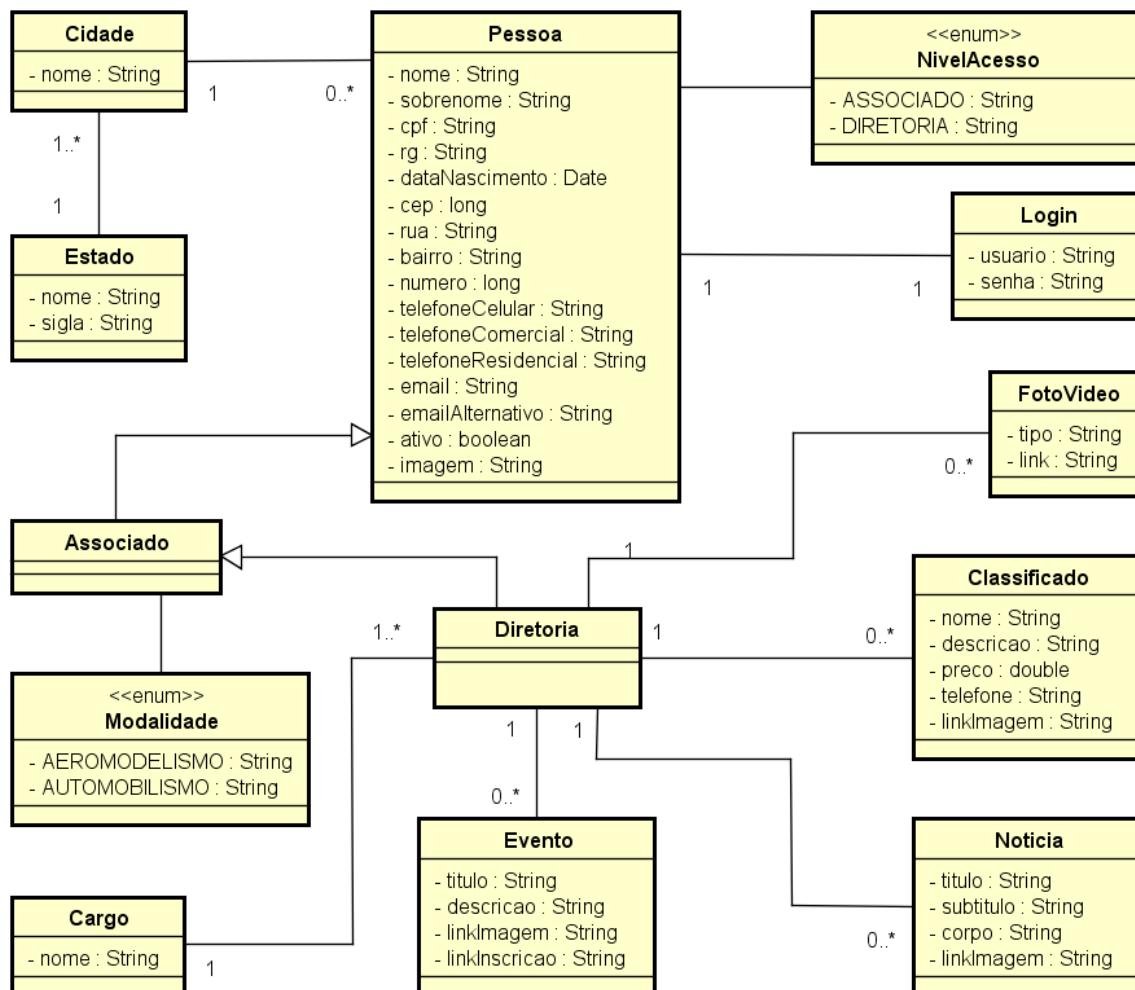


Figura 4. Diagrama de classes da solução proposta.

Fonte: elaborado pelos autores.

Para viabilizar a implementação de todas as funcionalidades oferecidas, foram elencadas treze classes que representam o modelo domínio. Essas classes são utilizadas para realizar o mapeamento com o banco de dados. Neste momento, optou-se por não especificar os métodos para facilitar a visualização das informações.

A classe *Diretoria* herda de *Associado* que por sua vez estende o comportamento da classe *Pessoa*. Uma *Pessoa* está relacionada a uma *Cidade*, que está relacionada com *Estado*. Vale destacar *Pessoa* tem um atributo (*enum*) para os níveis de acesso (*NívelAcesso*). Um objeto da classe *Pessoa* também possui um *Login*. Para o escopo deste projeto, a entidade *Associado* também tem um *enum* para representar as modalidades de atuação (*Modalidade*). A classe *Diretoria* está relacionada a um *Cargo*.

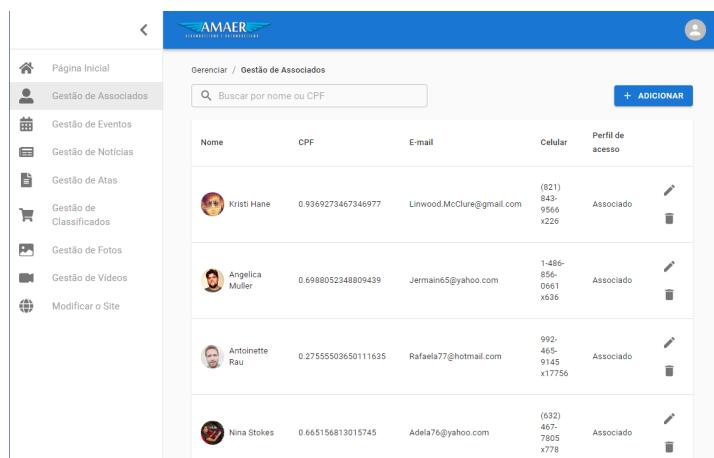
O relacionamento entre as classes *Evento*, *Classificado*, *FotoVideo* e *Noticia* com a classe *Diretoria* é utilizado para registrar qual usuário criou a informação. Além disso,

essa operação pode ser realizada somente por um usuário com perfil de diretoria. A classe *FotoVideo* é utilizada para armazenar os dados publicados na galeria de fotos e na galeria de vídeos, não tendo relação com a área de notícias.

6. Apresentação da Solução Implementada

Esta seção se inicia com a apresentação da interface da principal funcionalidade do sistema desenvolvido, no caso, a gestão de associações. A Figuras 5, 6 e 7 apresentam os screenshots referentes ao processo de gestão de associados.

Ao acessar o sistema, é exibida uma área para que o associado possa realizar a sua autenticação. Após realizar a autenticação, a página inicial é exibida com informações úteis. No menu de navegação lateral há a opção “Gestão de Associados” (Figura 5). Ao selecionar essa opção, um usuário com privilégios de administrador pode visualizar os associados cadastrados, podendo inclusive cadastrar um novo.



Nome	CPF	E-mail	Celular	Perfil de acesso
Krilli Hane	0.9369273467346977	Linwood.McClure@gmail.com	(82) 843-9566 x226	Associado
Angelica Muller	0.6988052348809439	Jermaine65@yahoo.com	1-486-886-0661 x636	Associado
Antoinette Rau	0.27555503650111635	Rafaela77@hotmail.com	992-465-9145 x17756	Associado
Nina Stokes	0.665156813015745	Adela76@yahoo.com	(632) 467-7805 x778	Associado

Figura 5. Área de gestão de associados.

Dentro da área de criação de associado, o administrador deve preencher os campos com seus dados cadastrais (Figura 6). Ao salvar, as informações são salvas no banco de dados e uma mensagem de confirmação é exibida.

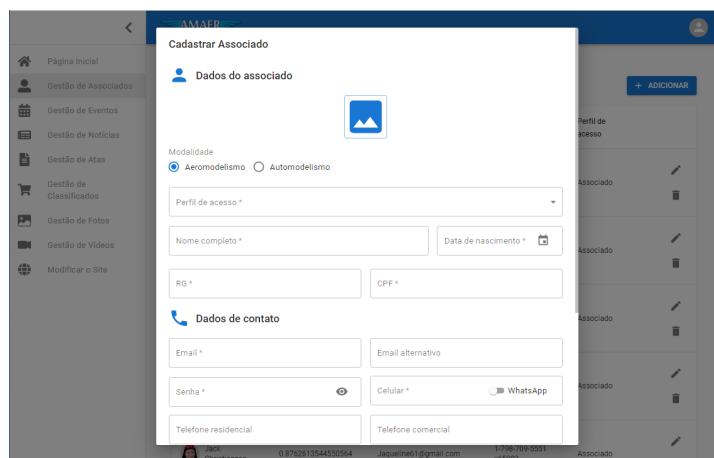


Figura 6. Formulário de cadastro de associado.

O administrador pode excluir ou editar os dados cadastrais dos associados ou, como mostrado na (Figura 7) o próprio associado pode realizar as alterações na sua área de edição de perfil, chamada de “Minha Conta”.

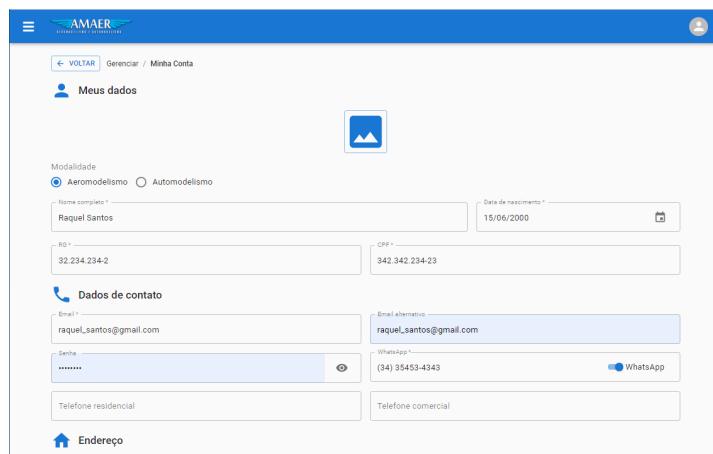


Figura 7. Área individual para alteração dos dados cadastrais.

Nesta primeira versão, o sistema realiza o cadastro, a listagem, a edição e exclusão de associados. Essas funcionalidades foram disponibilizadas em um ambiente de testes, nos quais puderam ser testadas e analisadas durante a extração de requisitos.

Durante a execução dos testes foi possível coletar *feedbacks* com o colaborador da associação de Maringá/PR, de modo a melhorar a solução continuamente. Ademais, pretende-se implantar a solução em um ambiente real, onde será possível avaliar melhor seu comportamento.

7. Considerações Finais

Tendo em vista que as associações, enquanto instituições, frequentemente deparam-se com problemas gerenciais tais como: a gestão dos membros ativos e inativos, controle das mensalidades, publicidade das decisões e comunicação entre associados, e considerando que nos últimos anos observa-se uma preocupação cada vez maior acerca da informatização das atividades executadas por instituições públicas e privadas. Este artigo teve por objetivo apresentar o desenvolvimento da primeira versão de um sistema para gestão de associações.

Nesta primeira versão, o sistema realiza o cadastro, a listagem, a edição e exclusão de usuários. Além disso, essas funcionalidades desenvolvidas foram disponibilizadas em um ambiente de teste, porém pretende-se implantar a solução em um ambiente real, onde será possível avaliar melhor seu comportamento e, consequentemente, melhorá-la continuamente.

Para trabalhos futuros pretende-se implementar: a página de gerenciamento de eventos, classificados, fotos e vídeos, notícias, atas, página institucional, além de um *chat* para troca de mensagens entre administradores e associados. Espera-se que o desenvolvimento desse sistema possa vir a ajudar as diversas associações em suas atividades.

Referências

- Bezerra, E. (2015). *Princípios de Análise e Projeto de Sistemas com UML*. Elsevier Rio de Janeiro.
- Call Tecnologia (2021). Call Tecnologia. Disponível em: <https://www.call.inf.br/>. Último acesso: 30-08-2021.
- CORE (2021). Clubes Associados Software – Sistema para gestão de clube e associação. Disponível em: <https://clubesassociados.com.br/>. Último acesso: 30-08-2021.
- IBGE – Instituto Brasileiro de Geografia e Estatística (2016). As Fundações Privadas e Associações sem Fins Lucrativos no Brasil. Disponível em: <https://bit.ly/3bZpd93>. Último acesso em 27-07-2021.
- OMG (2021). Unified Modeling Language - UML. Disponível em: <https://www.uml.org/>. Último acesso: 14-05-2021.
- Pressman, R. and Maxim, B. (2016). *Engenharia de Software*. McGraw Hill Brasil, Porto Alegre, 8 edition.
- Sbrocco, J. H. T. C. and Macedo, P. C. d. (2012). *Metodologias ágeis: engenharia de software sob medida*. Editora Érica, São Paulo, 1 edition.
- Schach, S. R. (2009). *Engenharia de Software: Os Paradigmas Clássico e Orientado a Objetos*. AMGH, Porto Alegre, 9 edition.
- Schwaber, K. and Sutherland, J. (2020). *O Guia do Scrum – O guia definitivo para o Scrum: As regras do jogo*. ScrumGuides.org.
- SEBRAE – Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (2013). Conheça os tipos de associações existentes no Brasil. Disponível em: <https://bit.ly/3BRK1dr>. Último acesso em 27-07-2021.
- SINCTEC SISTEMAS (2021). Associatec: Sistema de Gestão para Associações, Sindicatos e Federações. Disponível em: <https://www.associatec.com.br>. Último acesso: 30-08-2021.
- Sommerville, I. (2019). *Engenharia de Software*. Pearson Education – BR, São Paulo, 10 edition.
- Wazlawick, R. (2019). *Engenharia de software: conceitos e práticas*. Elsevier Editora Ltda., Rio de Janeiro, 2 edition.