

Modelagem de um sistema de gerenciamento de atendimentos efetuados pelo núcleo pedagógico da Universidade Federal de Itajubá - Campus Itabira

Modeling of an attendance management system performed by the pedagogical sector of the Federal University of Itajubá - Itabira Campus

Modelado de un sistema de gestión de atendimientos efectuados por el núcleo pedagógico de la Universidad Federal de Itajubá - Campus Itabira

Caio Scalon Borges de Oliveira

ORCID: <https://orcid.org/0000-0002-7432-5612>

Universidade Federal de Itajubá, Brasil

E-mail: caioscalon@unifei.edu.br

Rossana de Paula Junqueira Almeida

Universidade Federal de Itajubá, Brasil

E-mail: rossana.junqueira@unifei.edu.br

Recebido: 07/11/2018 | Revisado: 03/12/2018 | Aceito: 18/12/2018 | Publicado: 18/12/2018

Resumo

A utilização dos *softwares* é de grande importância atualmente, pois a partir deles é possível desenvolver novas tecnologias, facilitar o trabalho em diversas áreas e, conseqüentemente, trazer praticidade nas atividades realizadas no dia a dia. Nesta situação, o artigo descreve a modelagem de um sistema de auxílio ao núcleo pedagógico da Universidade Federal de Itajubá – *campus* Itabira, com o objetivo de auxiliar na organização dos documentos dos atendimentos realizados pelo mesmo, pois atualmente o trabalho é todo feito manualmente, o que dificulta o processamento dos atendimentos já feitos. A primeira etapa se deu a partir da entrevista, buscando coletar as informações do que o cliente desejava adicionar no *software* a ser desenvolvido. Com base nisso, fez-se o levantamento de requisitos e sua categorização, para que não falte nada importante nas fases seguintes do projeto. Estes requisitos foram utilizados no próximo ciclo, no qual foi necessário descrever as funcionalidades do sistema projetado, deixando claro ao cliente como o produto solicitado irá funcionar. Para ilustrar esta etapa, foram criados diagramas de casos de uso, apresentando todos os agentes externos que irão se comunicar com o sistema internamente e, para cada caso, foi criado um fluxo explicativo de seu funcionamento. Por fim, visando organizar as ações a serem feitas durante este projeto, o diagrama de atividades foi utilizado e, também, houve a prototipagem das

interfaces do sistema. Ao final deste artigo, temos a descrição da documentação necessária para que o sistema proposto possa ser implementado.

Palavras-chave: Sistema; Gerenciamento; Requisitos; Diagramas.

Abstract

The use of softwares has a great importance nowadays, it is possible to develop new technologies from them, to facilitate the tasks in several areas and, consequently, to bring practicality in the activities performed every day. In this situation, the article describes the modelling of a help system to the pedagogical sector of the Federal University of Itajubá - Itabira campus, with the purpose of assisting in the organization of the attendance documents created by this sector, because, currently, all the work is performed manually, which makes it difficult to process the treatments already done. The first step was based on the client's interview, seeking to collect the informations that the client wanted to add in the software to be developed. Based on these data, the requirements were researched and categorized so that nothing important was missing in the next phases of the project. The gathered requirements were used in the next cycle, in which it was necessary to describe the functionalities of the system that will be designed, making clear to the customer how the product requested by him will work. To illustrate this step, use case diagrams have been created, presenting all external agents that will communicate with the system internally and, for each use case, an explanatory flow was created in order to show its operation. Finally, in order to organize the actions to be done during this project, the activity diagram was used and, also, the prototyping of the system interfaces was done. At the end of this article, we have the description of the necessary documentation so that the proposed system can be implemented.

Keywords: System; Management; Requirements; Diagrams.

Resumen

La utilización de los softwares es de gran importancia actualmente, pues a partir de ellos es posible desarrollar nuevas tecnologías, facilitar el trabajo en diversas áreas y, consecuentemente, traer practicidad en las actividades realizadas en el día a día. En esta situación, el artículo describe el modelado de un sistema de auxilio al núcleo pedagógico de la Universidad Federal de Itajubá - campus Itabira, con el objetivo de auxiliar en la organización de los documentos de las atenciones realizadas por el mismo, pues actualmente el trabajo es todo hecho manualmente, que dificulta el procesamiento de las llamadas ya realizadas. La primera etapa se dio a partir de la entrevista, buscando recoger las informaciones de lo que el

cliente deseaba agregar en el software a ser desarrollado. Con base en ello, se hizo el levantamiento de requisitos y su categorización, para que no falte nada importante en las fases siguientes del proyecto. Estos requisitos se utilizaron en el siguiente ciclo, en el que fue necesario describir las funcionalidades del sistema proyectado, dejando claro al cliente cómo el producto solicitado funcionará. Para ilustrar esta etapa, se crearon diagramas de casos de uso, presentando todos los agentes externos que se comunicarán con el sistema internamente y, para cada caso, se creó un flujo explicativo de su funcionamiento. Por último, para organizar las acciones a realizar durante este proyecto, el diagrama de actividades se utilizó y, también, hubo el prototipado de las interfaces del sistema. Al final de este artículo, tenemos la descripción de la documentación necesaria para que el sistema propuesto pueda ser implementado.

Palabras clave: Sistema; gestión; requisitos; Diagramas.

1. Introdução

A partir do desenvolvimento e avanço da sociedade, a quantidade de informações que são geradas e transmitidas globalmente alcança um patamar no qual é necessário com que haja uma plena organização e métodos de sistematização destes dados. Assim, surge a demanda destes meios, geralmente sendo aplicados softwares específicos para cada situação.

A Engenharia de Software busca trazer praticidade para o desenvolvimento destes softwares, com técnicas que auxiliarão desde sua etapa inicial até a finalização do mesmo, de maneira eficiente e fácil visualização dos passos.

Os softwares são avaliados como produtos dentro da área de Engenharia de Software e, para tratá-los adequadamente, é necessário incorporar o máximo de informações possível e realizar testes. Isso faz com que a qualidade do produto final seja satisfatória, tentando alcançar os objetivos do projeto.

Para dar início à modelagem do sistema, a comunicação com o cliente é imprescindível, pois assim é possível determinar os requisitos necessários ao software. O levantamento de requisitos, portanto, se trata da primeira etapa a ser documentada, de maneira que fiquem claros os desejos do cliente e, ao mesmo tempo, ser transparente para ambos os lados (cliente e desenvolvedor).

Os requisitos podem ser definidos em duas categorias: os funcionais, os quais tratam o comportamento do sistema em função das ações tratadas pelo usuário durante o uso do mesmo; e não-funcionais, que quantificam os aspectos deste comportamento. Há diferentes técnicas para realizar o levantamento destes requisitos, sendo a mais comum a entrevista,

utilizada para dar andamento a este artigo. Ao fim desta etapa, o documento deve possuir uma lista de requisitos com seus respectivos identificadores, descrições e prioridade.

Após a etapa de levantamento de requisitos, é possível dar prosseguimento ao processo de modelagem do sistema. Neste momento, são desenvolvidos os diagramas de casos de uso, nos quais são descritas as funcionalidades que o sistema terá, bem como os fluxos de ações dele e como os usuários se comportam perante a estes fluxos. O próximo diagrama a ser feito é o de classes, apresentando os objetos que estruturarão o sistema em desenvolvimento, cada um com seus atributos, funções e relacionamentos.

Por fim, são criados os diagramas de atividades, os quais apresentam as ações que o usuário efetuará durante seu acesso ao sistema. Este tipo de diagrama permite uma visualização mais clara do software, visto que descreve o passo a passo de forma simples e organizada.

Com base nas etapas descritas e a partir da aplicação da área de Engenharia de Software, este artigo foca na descrição da modelagem de um sistema de gerenciamento de atendimentos a fim de atender às necessidades dos funcionários do setor pedagógico da Universidade Federal de Itajubá - *campus* Itabira. O trabalho realizado por estes funcionários, atualmente, é totalmente manual e não apresenta qualquer forma de catalogação, o que atrapalha o andamento dos atendimentos e a busca daqueles que já foram feitos. Portanto, por meio da modelagem, será possível desenvolver um *software* com mais detalhes e clareza para sua implementação.

2. A engenharia e o desenvolvimento de *software*

Há uma diversidade de definições para explicar o que é a Engenharia de Software, mas é possível abstrair seu conceito a partir da citação apresentada por Pressman (2011, p. 39), proposta por Fritz Bauer em uma conferência sobre este tema:

[A Engenharia de Software é] o estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um *software* que seja confiável e que funcione eficientemente em máquinas reais.

Porém, Pressman (2011) acredita que esta definição é apenas uma base à verdadeira dimensão na qual se abrange este conteúdo, fortemente ligado à qualidade do artefato, seus prazos e a satisfação do cliente. Ele defende que a Engenharia de Software se trata de uma tecnologia em camadas.

Assim que o cliente faz o pedido de um sistema, um processo é iniciado e ele possui várias etapas, sendo elas o levantamento de requisitos, a análise dos mesmos, o design do

produto, sua implementação, os testes realizados nele e, por fim, a entrega do produto final caso tudo esteja acordado com o cliente.

Sommerville (2011) acredita que a etapa de avaliação dos requisitos é crítica para o processo de *software*, visto que erros nesta etapa certamente irão gerar problemas no futuro. Além disso, há dois níveis de detalhe dos requisitos: em alto nível - para os clientes e usuários -, e com mais especificações - para os desenvolvedores.

O levantamento de requisitos envolve uma abordagem inicial do cliente para que todas as suas necessidades em relação ao sistema que deseja sejam listadas em forma de requisitos. Para isso, há diversas maneiras de realizar este levantamento, sendo a mais comum a entrevista, na qual consiste em encontros entre o cliente e os desenvolvedores que irão planejar e implementar o sistema, com uma série de perguntas sobre o sistema desejado.

Durante a investigação dos requisitos, os desenvolvedores precisam ter conhecimento sobre os requisitos que farão parte do projeto. De acordo com Paula Filho (2003), estes requisitos podem apresentar características funcionais, que representarão o comportamento do programa a partir das ações de seu usuário; ou características não-funcionais, que quantificarão os aspectos do comportamento. Além dessas características, os requisitos podem ser classificados em três tipos: explícitos, normativos e implícitos. Também é necessário priorizar os requisitos, dividindo-os em uma lista clara dos mais prioritários para os menos. Assim, é possível classificá-los em três categorias: essenciais, desejáveis e opcionais.

Devido ao fato dos projetos de *software* serem um pouco instáveis, já que o produto é pensado em função da percepção e dos desejos do usuário, pode-se mudar alguns requisitos ou surgir novos. Portanto, a necessidade de sempre verificar as possibilidades é constante, permitindo a repriorização e revisão dos requisitos.

Antes da implementação do sistema, é importante documentá-lo da maneira mais clara possível, com descrições detalhadas e atualizadas, para que seja possível controlá-lo de forma mais prática e com uma quantidade reduzida de falhas futuras. A modelagem, portanto, é bastante útil não apenas para a elaboração de um projeto de *software*, mas para o escopo de desenvolvimento do mesmo, principalmente quando se trata de algo mais complexo. Para isso, grande parte das empresas utilizam uma linguagem padrão de modelagem, conhecida como Linguagem de Modelagem Unificada ou UML (em inglês, Unified Modeling Language).

Para todo sistema, é necessário que haja uma série de descrições de como cada parte dele funciona e como elas vão ser interconectadas. Assim, sua implementação se torna mais fácil no ponto de vista dos programadores, pois eles já terão uma ideia formada sobre o

sistema a ser criado. Porém, não se pode confundir o funcionamento do sistema com sua organização interna, algo que não será descrito nesta fase de desenvolvimento. O diagrama de casos de uso é o diagrama que descreverá esta etapa.

Neste diagrama, estão presentes três elementos básicos de modelagem: atores, casos de uso e comunicação. O primeiro elemento é apresentado pelos atores (representados por um boneco – *stick man* -, possuindo uma breve identificação sobre seu papel) são os agentes externos ao sistema, ou seja, estão fora da fronteira do sistema em si, porém são entidades que interagem com o mesmo, entrando com dados e ‘pedidos’, ou recebendo algo do sistema.

Após ser feito o diagrama e identificado o limite do sistema, é necessário elaborar os fluxos de casos de uso, ou seja, uma descrição de toda e qualquer ação que poderá ser executada por cada um deles, tanto a sequência de ações a partir da suposição que nada dê errado durante a execução do sistema, também chamado de fluxo principal, quanto as subfunções do fluxo principal ou as exceções do mesmo (erros ou mau funcionamento), chamados de fluxos alternativos.

O último diagrama a ser utilizado na etapa de projeto e documentação do mesmo é o diagrama de atividades. Ele permite descrever as operações associadas a um objeto e as interações deste com outros. Além disso, ele apresenta uma sequência de ações de determinada atividade.

Pressman (2001, p. 737) diz que este diagrama ‘mostra o comportamento dinâmico de um sistema ou parte de um sistema através do fluxo de controle entre ações que o sistema executa’. Isso significa que a partir das ações que o *software* seguirá durante seu funcionamento, definindo-se o início e o fim de cada um dos fluxos, sendo possível referenciar outros fluxos previamente modelados, reduzindo o tamanho do diagrama.

Por fim, um ponto importante no desenvolvimento de *software* é sua visualização de maneira clara e objetiva, proporcionando tanto aos desenvolvedores quanto aos clientes uma perspectiva sobre os conceitos e funcionalidades do sistema em questão. Dessa maneira, ambos os lados podem opinar sobre ele e propor modificações ou apenas ter uma noção de como esse sistema será. Há alguns tipos de protótipos que podem ser utilizados, baseando-se nos prazos e orçamento para a confecção dos mesmos, sendo divididos em: *wireframes* (rascunhos), visuais e interativos.

3. Metodologia

Para auxiliar no desenvolvimento do projeto conforme seu avanço, o mesmo foi dividido nas seguintes etapas:

- a) Levantamento dos requisitos a partir de uma abordagem direta com os funcionários do setor pedagógico, bem como a separação destes requisitos de acordo com sua prioridade e necessidade.
- b) Agrupamento de todas as funcionalidades que executavam atividades relacionadas umas às outras em casos de uso, delimitando-se os atores do sistema e traçando os relacionamentos entre si ou com os casos de uso.
- c) Montagem da descrição da sequência de ações tomadas pelo sistema durante sua utilização, definidas em diagramas de atividades.
- d) Desenvolvimento dos *layouts* das interfaces, visando a padronização e facilidade de utilização.
- e) Avaliação da satisfação (qualidade) dos funcionários perante aos requisitos levantados.

4. Resultados e discussão

Pela extensa quantidade de funcionalidades, para exemplificar o resultado do projeto, foi selecionada uma delas - “Gerenciar Aluno” - e disposta em sua modelagem, representadas pelos diagramas e interface. Esta função envolve o cadastro, alteração e busca de dados de alunos.

Na etapa de levantamento de requisitos são definidas as funções que o sistema deve possuir. Os requisitos estão separados em funcionais (RF) e não-funcionais (RNF). Como se trata de um número considerável de requisitos funcionais e não-funcionais, serão descritos apenas alguns deles. Os Requisitos Funcionais: [RF05], [RF06] e [RF07] e o Requisito Não-Funcional: [RNF02].

Identificação:	[RF05] Cadastrar alunos no sistema		
Casos de uso relacionados:	[UC03]		
Descrição:	Permite o cadastro de novos alunos no sistema que estão em busca de atendimento.		
Prioridade:	<input type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável

Identificação:	[RF06] Buscar registros de atendimento por aluno
----------------	--

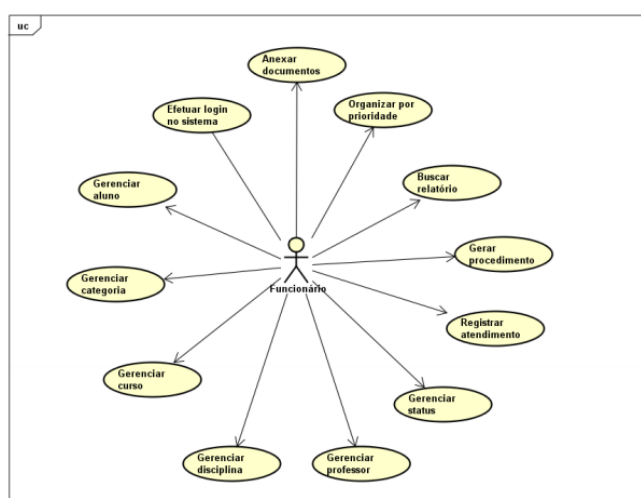
Casos de uso relacionados:	[UC03]		
Descrição:	Permite a busca de registros relacionados a um aluno através de qualquer informação referente a ele.		
Prioridade:	<input type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável

Identificação:	[RF07] Alterar os dados de alunos		
Casos de uso relacionados:	[UC03]		
Descrição:	Permite a alteração dos dados de um determinado aluno que esteja cadastrado no sistema. Não há possibilidade de exclusão de alunos previamente cadastrados.		
Prioridade:	<input type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável

Identificação:	[RNF02] Interface do sistema		
Casos de uso relacionados:	Todos.		
Descrição:	Interface Gráfica do Usuário (GUI) que permite a comunicação entre usuário e sistema de maneira clara e sem grandes dificuldades. A usabilidade intuitiva é importante para sua implementação.		
Prioridade:	<input type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável

A próxima etapa se trata dos diagramas de casos de uso e das descrições de cada caso, expondo os fluxos de funcionamento para cada situação. Há dois tipos de diagramas descritos, podendo um deles ser visto na Figura 1.

Figura 1 – Diagrama de casos de uso do funcionário



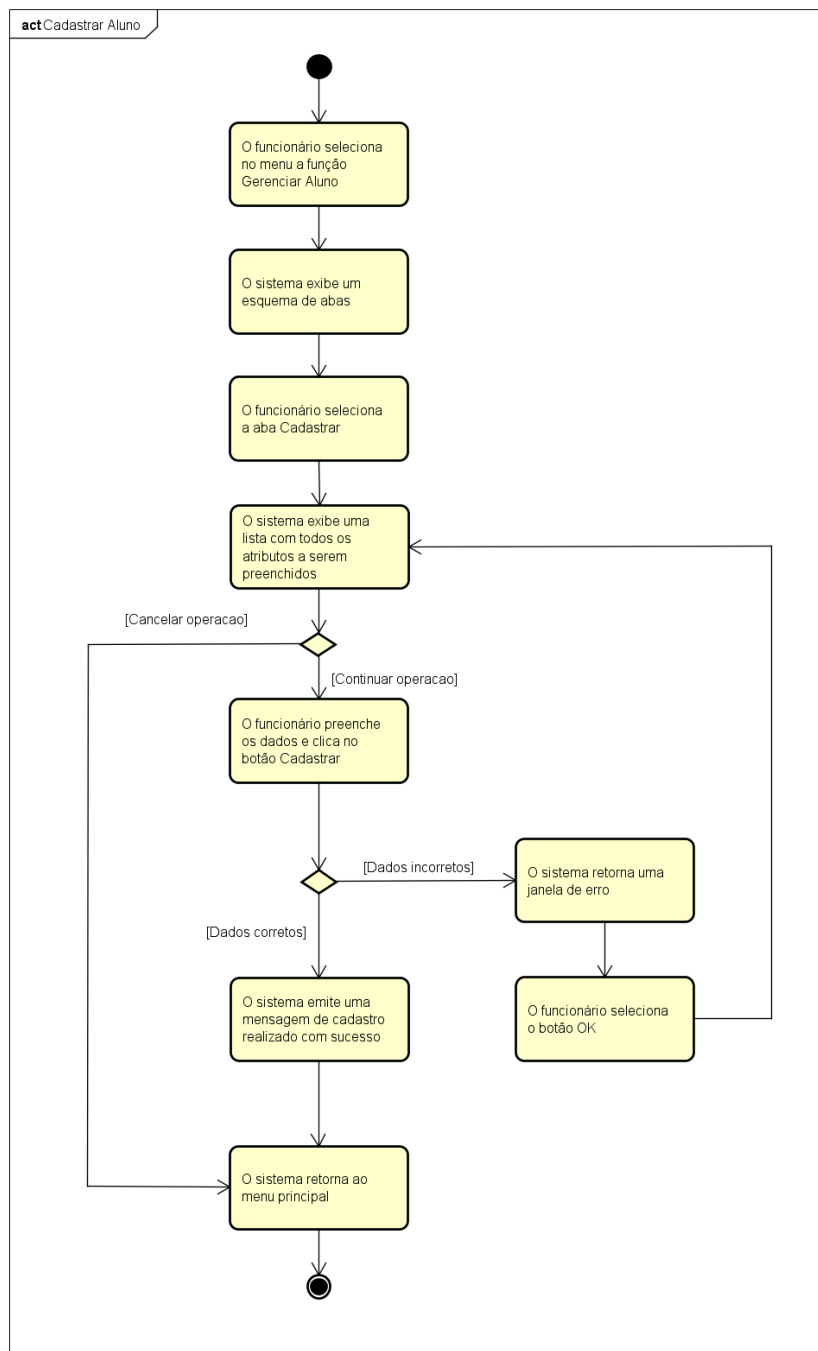
Fonte: Elaborado pelo autor

Identificador:	[UC03] Gerenciar aluno		
Descrição:	Responsável por realizar o cadastro de alunos que ainda não constam no sistema e que estão em busca de atendimento. Permite, também, a alteração dos dados daqueles já cadastrados, porém não permite a exclusão. Além disso, é possível realizar uma busca dos alunos que estão no sistema.		
Atores:	Funcionário do sistema.		
Prioridade:	<input type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável
Pré-condições:	O sistema deve estar disponível; o ator deve estar conectado ao sistema.		
Pós-condições:	O sistema libera a página de registro de atendimento ao aluno.		
Fluxo Principal			
<ol style="list-style-type: none"> 1. O funcionário seleciona no menu a função <i>Gerenciar Aluno</i>. 2. O sistema exibe um esquema de abas: <i>Cadastrar</i>, <i>Buscar</i> e <i>Alterar</i>. 3. O funcionário seleciona a aba “Cadastrar”. 4. O sistema exibe a listagem de todos os atributos necessários para o cadastro de um novo aluno. 5. O funcionário preenche os dados e seleciona o botão “Cadastrar”. 6. O sistema verifica os dados inseridos e, caso sejam válidos, emite uma mensagem de cadastro realizado com sucesso. 			

Cada funcionalidade implementada no sistema desenvolvido segue uma série de passos e pode ser descrita pelo diagrama de atividades. Este tipo de diagrama foi escolhido visto que permite uma fácil visualização do funcionamento do sistema por parte dos funcionários do setor que irão utilizá-lo, trazendo uma abstração maior das ações que o sistema proporciona. Os diagramas apresentados nas Figura 2, 3 e 4 representam os fluxos de gerenciamento de dados de um aluno.

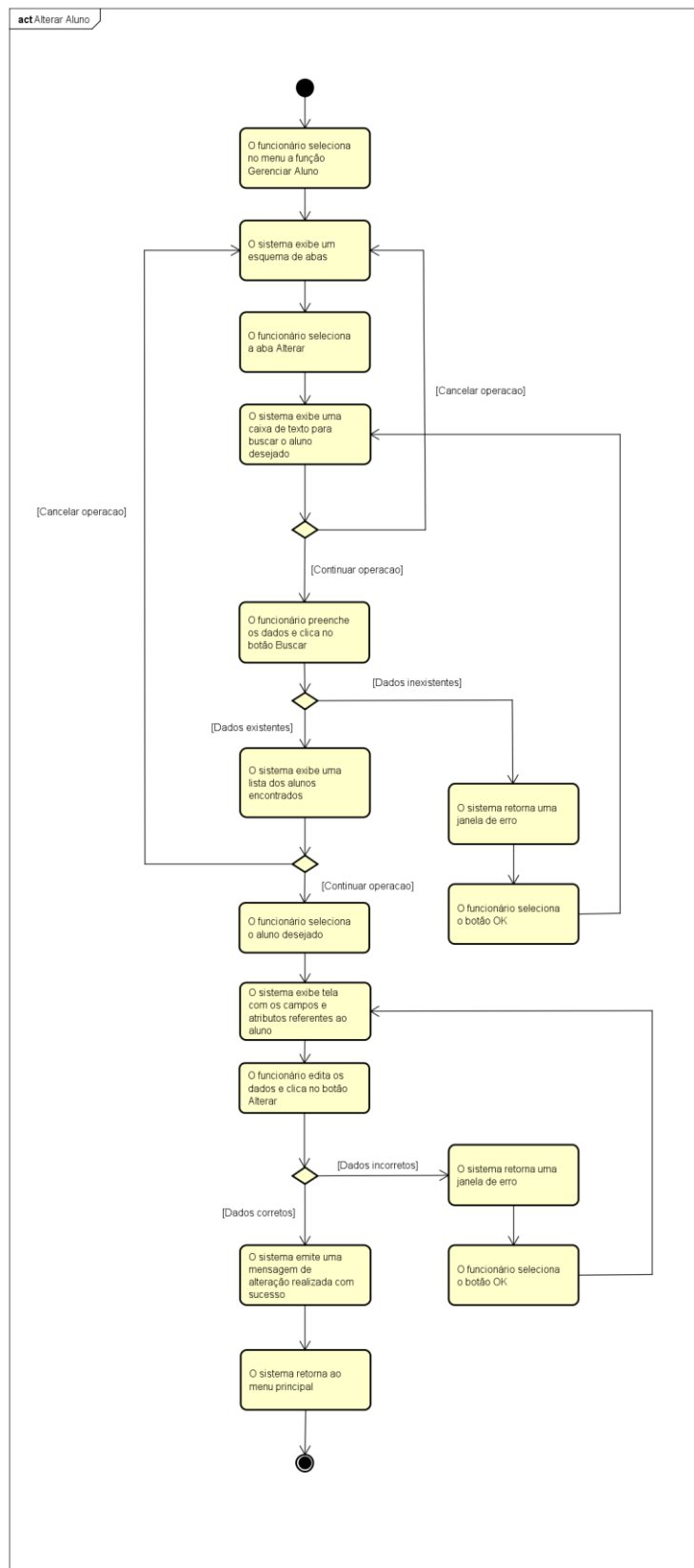
Por fim, são desenvolvidos os protótipos de interface, ilustrando como o sistema pode ser apresentado aos funcionários. Os protótipos de interface para o gerenciamento de alunos podem ser observados nas Figuras 5, 6 e 7.

Figura 2 – Diagrama de atividades para Cadastrar Aluno



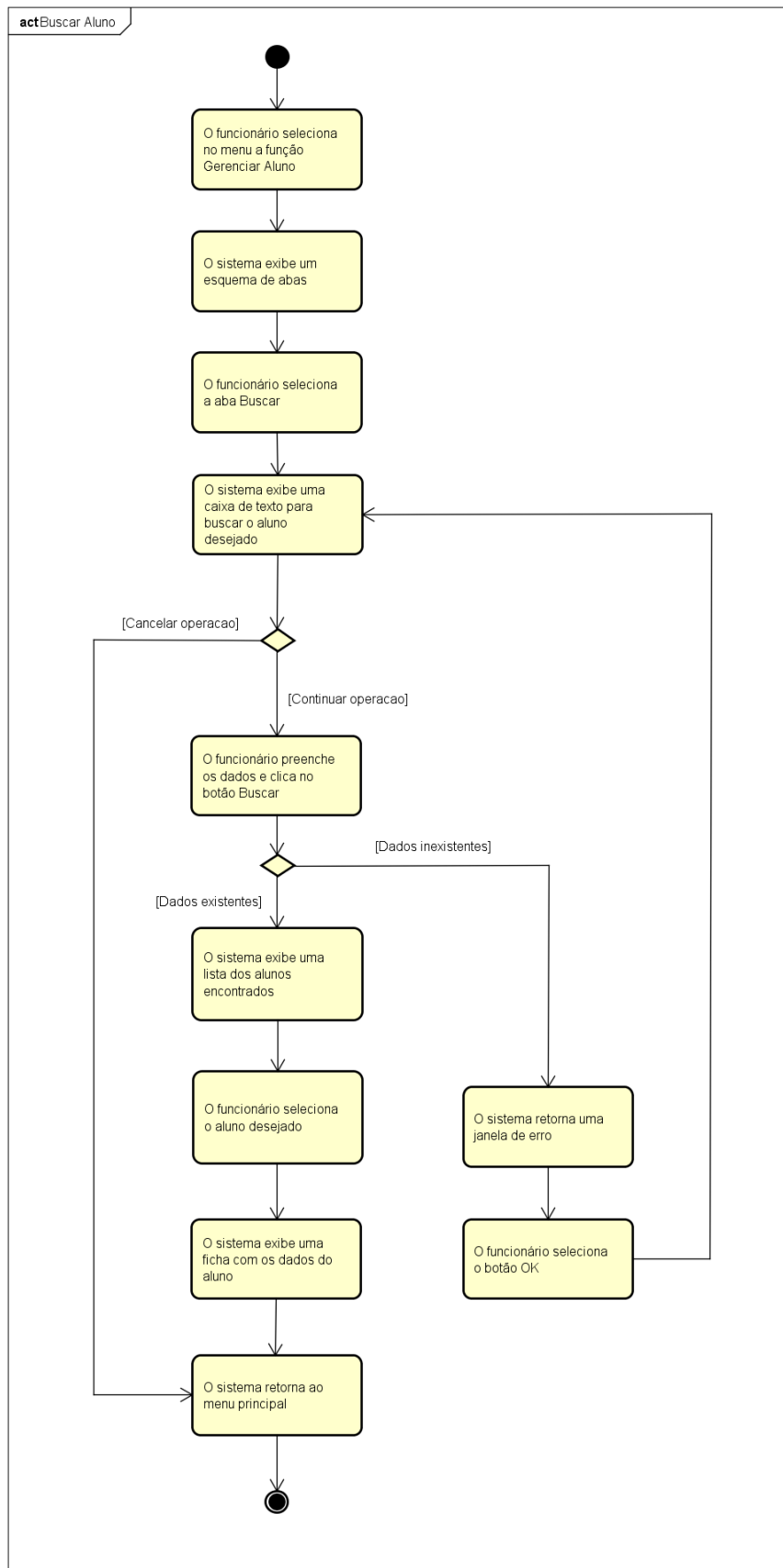
Fonte: Elaborado pelo autor

Figura 3 – Diagrama de atividades para Alterar Aluno



Fonte: Elaborado pelo autor

Figura 4 – Diagrama de atividades para Buscar Aluno



Fonte: Elaborado pelo autor

Figura 5 – Protótipo de interface para Cadastrar Aluno

SAANP - Sistema de Auxílio ao Aluno do Núcleo Pedagógico - UNIFEI

Gerenciar Aluno - Cadastrar

· Preencha os campos

Nome: Matrícula/RA:

Identidade/RG: CPF: Título de Eleitor

Data de nascimento: Estado civil: Sexo: Masculino Feminino

Endereço: Telefone:

Fonte: Elaborado pelo autor

Figura 6 – Protótipo de interface para Alterar Aluno

SAANP - Sistema de Auxílio ao Aluno do Núcleo Pedagógico - UNIFEI

Gerenciar Aluno - Alterar

· Modifique os campos desejados

Nome: Matrícula/RA:

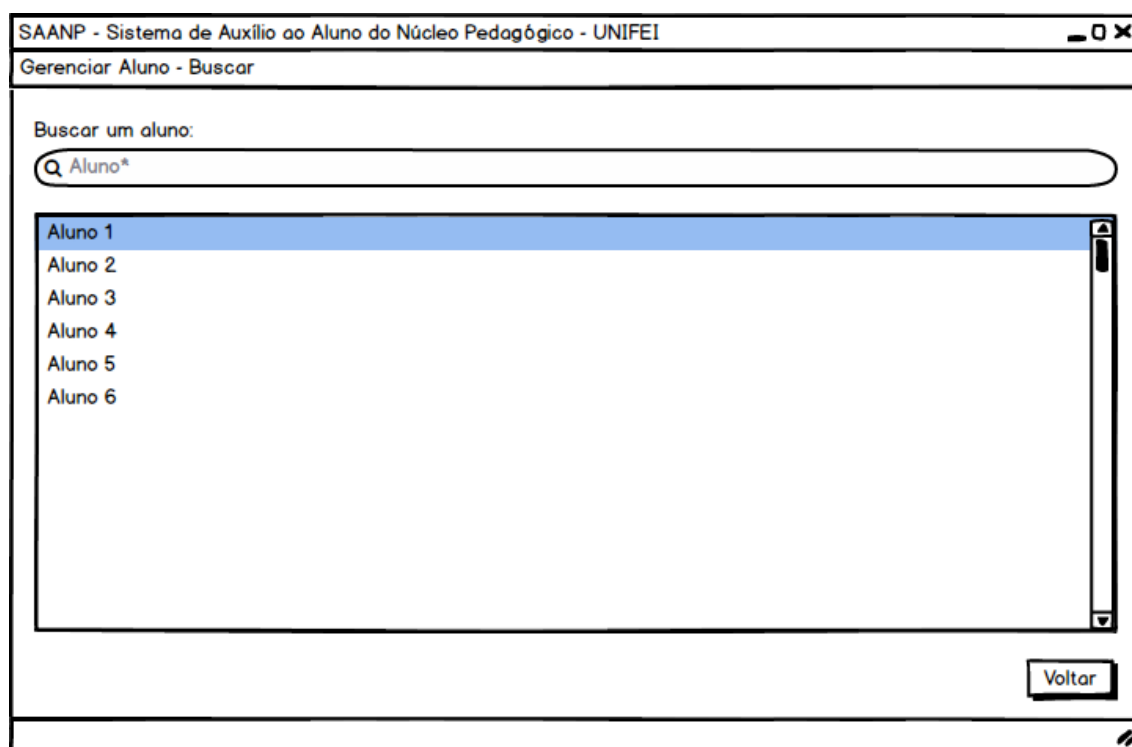
Identidade/RG: CPF: Título de Eleitor

Data de nascimento: Estado civil: Sexo: Masculino Feminino

Endereço: Telefone:

Fonte: Elaborado pelo autor

Figura 7 – Protótipo de interface para Buscar Aluno



Fonte: Elaborado pelo autor

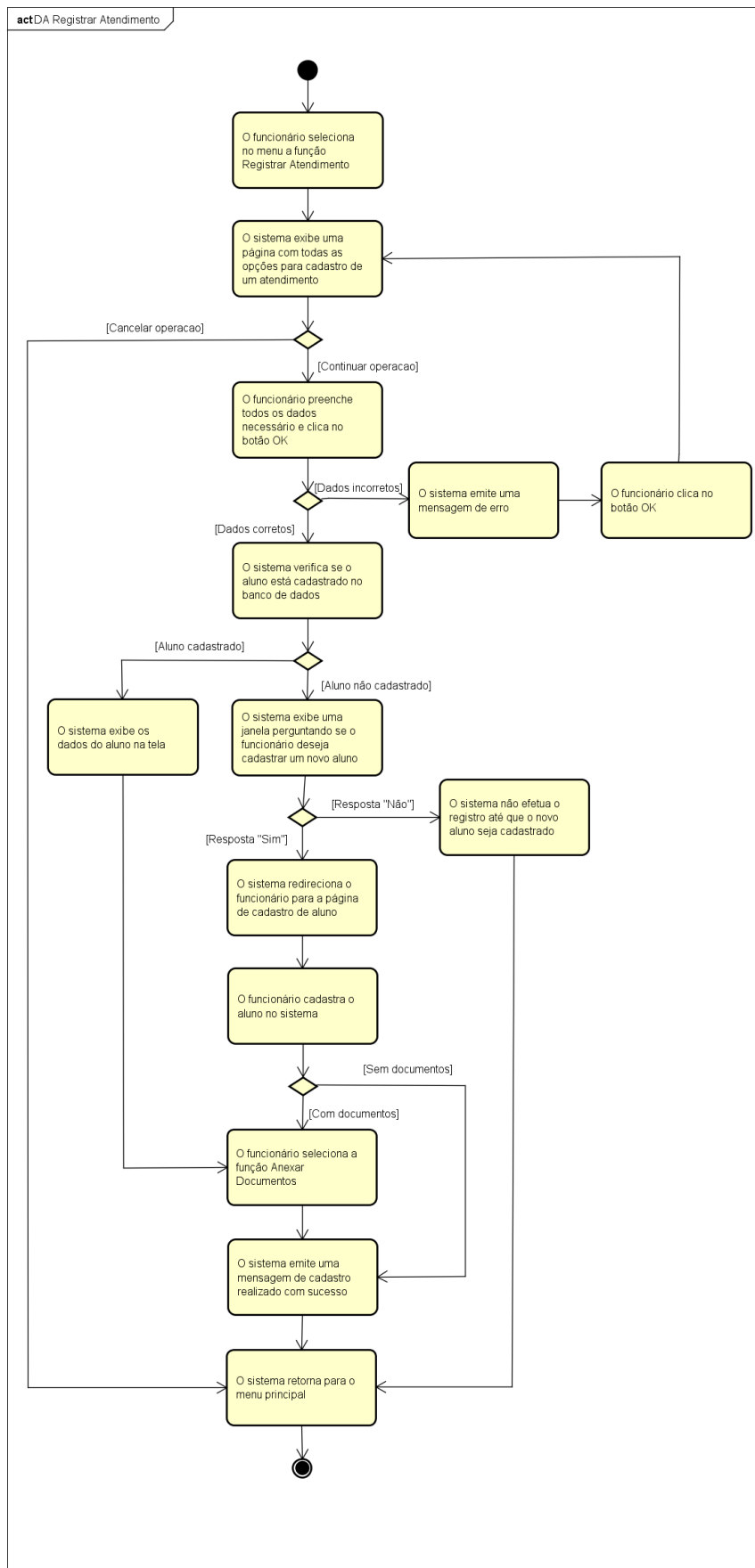
No setor pedagógico, são realizados os atendimentos dos alunos. Cada atendimento é registrado com suas respectivas informações e se relaciona a um dos alunos. O mesmo atendimento pode vir a ter continuidade com o passar do tempo, gerando procedimentos cada vez que ele retorna ao setor. O processo realizado para os registros de atendimentos segue o mesmo padrão do gerenciamento de alunos, como pode ser observado nas tabelas abaixo e nas Figuras 8 e 9.

A primeira tabela [RF20] representa o requisito de gerenciamento de atendimento, descrevendo seus detalhes de forma clara e objetiva. Já a segunda [UC09], trata do caso de uso que gerencia estes atendimentos, possuindo um fluxo principal e dois outros fluxos alternativos, apresentando as possibilidades de ação dentro do sistema.

Identificação:	[RF20] Registrar atendimentos no sistema		
Casos de uso relacionados:	[UC09]		
Descrição:	Permite o registro de novos atendimentos referentes a um aluno previamente cadastrado no sistema.		
Prioridade:	<input type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável

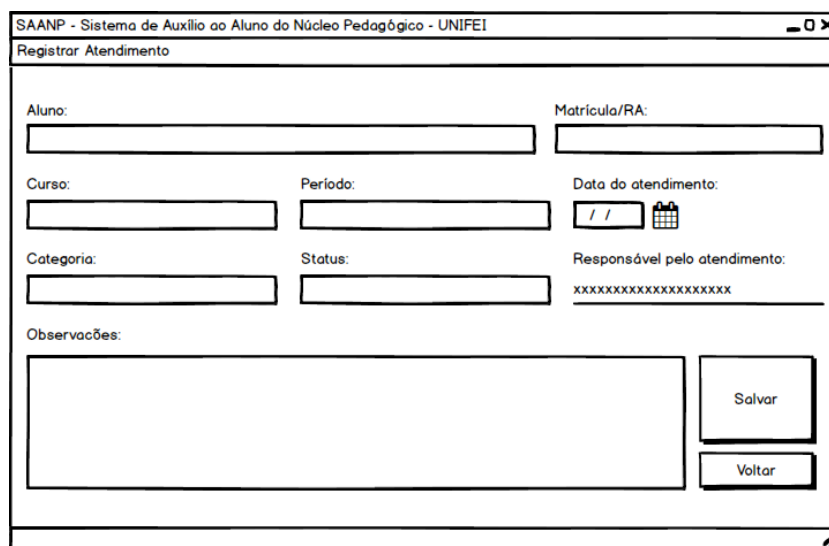
Identificador:	[UC09] Registrar atendimento		
Descrição:	Cadastra novos registros de atendimentos de alunos, permitindo futuros registros de procedimentos de um mesmo atendimento. Não é permitida a sua exclusão uma vez cadastrados no sistema. No seu cadastro é possível relacionar uma categoria, status, professor e curso.		
Atores:	Funcionário do sistema.		
Prioridade:	<input type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável
Pré-condições:	O sistema deve estar disponível; o ator deve estar conectado ao sistema.		
Pós-condições:	O sistema retorna ao menu principal ou à tela anterior.		
Fluxo Principal			
<ol style="list-style-type: none"> 1. O funcionário seleciona no menu a função <i>Registrar Atendimento</i>. 2. O sistema exibe uma página com todas as opções para cadastro de um atendimento. 3. O funcionário preenche todos os dados necessários e aperta o botão “OK”. 4. O sistema verifica se o aluno já está cadastrado. (+F.A.1) 5. O sistema verifica os dados inseridos, emitindo uma mensagem de cadastro realizado com sucesso caso sejam válidos. (+F.A.2) 			
Fluxo Alternativo 1			
<ol style="list-style-type: none"> 1. Se o aluno não estiver cadastrado no sistema, o sistema emite uma pergunta se o funcionário deseja cadastrar novo aluno. 2. Se o funcionário responde <i>Sim</i>, ele é redirecionado para a página de cadastro de aluno. 3. Se o funcionário responde <i>Não</i>, o sistema não altera a página, não efetuando o registro até que o funcionário cadastre o aluno ou edite os dados. 			
Fluxo Alternativo 2			
<ol style="list-style-type: none"> 1. O sistema verifica os dados inseridos, emitindo uma mensagem de dados incompletos caso tenha algum campo obrigatório não preenchido e redireciona o funcionário lhe apresentando o campo com preenchimento inválido. 			

Figura 8 – Diagrama de atividades para Registrar Atendimento



Fonte: Elaborado pelo autor


Figura 9 – Protótipo de interface para Registrar Atendimento



SAANP - Sistema de Auxílio ao Aluno do Núcleo Pedagógico - UNIFEI

Registrar Atendimento

Aluno: Matrícula/RA:

Curso: Período: Data do atendimento: 

Categoria: Status: Responsável pelo atendimento:

Observações:

Salvar

Voltar

Fonte: Elaborado pelo autor

5. Considerações finais

A partir da abordagem aos clientes do setor pedagógico, pôde-se definir e organizar as funcionalidades essenciais, importantes e opcionais do sistema posteriormente desenvolvido, possibilitando a montagem das tabelas de requisitos funcionais e não funcionais do mesmo. Além disso, com os requisitos definidos, desenvolveu-se os casos de uso para cada um ou um grupo deles, a fim de descrever o fluxo principal e o fluxo secundário (opcional) das operações do sistema de maneira que o entendimento de como o sistema funciona mais detalhadamente. Com as tabelas, modelou-se os diagramas de caso de uso para cada agente externo. Após essas etapas serem feitas, seguiu-se no desenvolvimento dos diagramas de atividades do projeto e o protótipo de sua interface, auxiliando na visão do sistema tanto na parte de funcionalidades durante o uso do mesmo, quanto a interação do usuário com o sistema a partir de uma tela clara e fácil de manipular.

Com toda a documentação descrita no desenvolvimento deste projeto, é possível refinar alguns aspectos desejados e dar início à implementação do sistema, aplicando os requisitos e fluxos previamente projetados.

Referências

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 1. ed. São Paulo:

Pearson,2006.

GUEDES, G. T. A. **UML 2: Uma abordagem prática**. 2. ed. São Paulo: Novatec, 2011.

HEUSER, C. A. **Projeto de Banco de Dados**. 4. ed. São Paulo: Sagra Luzzatto, 2001.

PAULA FILHO, Wilson de Pádua. **Engenharia de Software: Fundamentos, métodos e padrões**. 2.ed. Rio de Janeiro: Ltc, 2003.

PRESSMAN, Roger S. **Engenharia de Software: Uma abordagem profissional**. 7. ed. Porto Alegre: Bookman, 2011.

SOMMERVILLE, Ian. **Engenharia de Software** . 9. ed. São Paulo: Pearson, 2011.