

SIG-Library: Gerenciando sua biblioteca

Douglas Mateus Soares Cândido da Silva¹, Lucas da Silva Santos¹

¹Departamento de Computação e Tecnologia
Universidade Federal do Rio Grande do Norte (UFRN)
Rua Joaquim Gregório, S/N – 59.300-000 – Caicó – RN – Brasil

{douglas219.1, lucassilva01}@ufrn.edu.br,

Resumo. *Os artefatos e métodos da Engenharia de Requisitos de Software, estudados ao longo da disciplina Organização, Sistemas e Métodos, foram aplicados no desenvolvimento do SIG-Library. O SIG-Library é um Sistema de Informação Gerencial que tem como principal objetivo servir como uma ferramenta confiável que garanta a máxima eficiência na administração de uma biblioteca física de livros.*

1. Introdução

O curso de graduação Bacharelado em Sistemas Informação da Universidade Federal do Rio Grande do Norte (UFRN) tem como objetivo formar profissionais para atuar de forma competente e transformadora no desenvolvimento de sistemas e na gestão da tecnologia da informação nas organizações.

Sistemas de Informação Gerenciais são desenvolvidos para informatizar e otimizar processos manuais e ineficientes, e auxiliar na administração e tomada de decisões dos fluxos de negócios implementados por eles. Desenvolver estas soluções com a máxima qualidade possível é uma das competências exigidas pelo mercado aos profissionais da área de Tecnologia da Informação, os alunos devem adquirir esta competência ao longo do curso.

A disciplina Organização, Sistemas e Métodos, ministrada pela professora Karliane Medeiros Ovidio Vale, segundo seu plano de curso, tem como principal objetivo apresentar os conceitos fundamentais para o desenvolvimento de projetos de software. Estes conceitos englobam as tipologias de estrutura organizacional, análise estrutural, requisitos de informação, gestão de processos de negócio, metodologias, técnicas e ferramentas de mapeamento e melhoria de processos. A disciplina ensina e prepara os alunos para utilizarem os artefatos e métodos amplamente adotados na etapa Engenharia de Requisitos da Engenharia de Software.

A elicitação, análise, implementação e manutenção dos requisitos de software são o principal foco da disciplina OMS. Quando a engenharia de requisitos do sistema em desenvolvimento, desde a elicitação e análise até a documentação, é mal feita, a boa qualidade e eficiência do produto de software final é posta em risco, pois o sistema poderá não atender todas as necessidades dos clientes.

Na disciplina OMS, os alunos também são apresentados às diversas técnicas que podem ser utilizadas para projetar e modelar os softwares, auxiliando nos seus desenvolvimentos. Os estudantes aprendem a usar a UML (Linguagem de Modelagem Unificada) e todos os benefícios proporcionados por ela enquanto ferramenta para a modelagem de softwares.

O desenvolvimento de soluções de software para o ramo de negócio bibliotecário é extremamente necessário, pois é um setor que lida com o intercâmbio de materiais físicos (Livros) entre pessoas e é de vital importância manter o controle da entrada e saída dos bens e recursos da biblioteca. O sistema desenvolvido como projeto prático da disciplina pelos autores desse artigo foi o SIG-Library, este sistema de gerenciamento de bibliotecas foi projetado para adaptar-se e ser incorporado nos diversos tipos de estabelecimentos de bibliotecas, de acordo com suas regras de negócio.

2. Referencial Teórico

Para que os softwares desenvolvidos atendam às necessidades e demandas dos seus clientes, os requisitos do software devem ser cuidadosamente coletados, analisados, adaptados, mantidos e repensados. Segundo [Machado 2018], os requisitos de software expressam as características e restrições do produto de software, do ponto de vista de satisfação das necessidades do usuário e, em geral, independem da tecnologia empregada na construção da solução, sendo a parte mais crítica e propensa a erros no desenvolvimento de softwares.

Uma vez que os requisitos do software forem devidamente tratados, a etapa de modelagem do software tem início. [Lobo 2009] afirma que a modelagem de software é uma ferramenta que busca obter a representação de um software por meio de modelos abstratos de um sistema. A abstração é empregada na modelagem de software com o objetivo de buscar utilizar somente o que é necessário para se representar o software, deixando de lado o que não é relevante para se documentar. A modelagem de software não só permite que o software seja projetado com os requisitos necessários para o seu perfeito funcionamento, como também que se use uma arquitetura mais adequada para o projeto. O tempo gasto inicialmente para manter um modelo de software é insignificante se comparado ao tempo ganho com este modelo nas fases de desenvolvimento, uma vez que, quando o software é bem projetado, sua implementação é extremamente facilitada, pois sua logística de funcionamento se torna mais fluida e melhor pensada e os possíveis bugs que poderiam aparecer durante a implementação, em caso de um projeto de software mal feito, são previamente detectados e tratados.

Para a melhor qualidade possível da modelagem do software ser assegurada, faz-se necessário utilizar ferramentas e artefatos confiáveis para este processo. A UML é uma linguagem gráfica para modelagem, visualização, especificação, construção e documentação de sistemas computacionais. [Booch et al. 2006] enfatiza que a UML proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como classes escritas em determinadas linguagens de programação orientadas à objetos, esquemas de bancos de dados, componentes de softwares reutilizáveis e conta com uma grande quantidade de diagramas e artefatos para serem utilizados pelos projetistas e desenvolvedores. Os diagramas da UML são classificados em três categorias: comportamental, de interação e estrutural. Os diagramas que fazem parte da categoria comportamental são: casos de uso, atividades e máquina de estados. Já os da categoria estrutural são: classes, objetos, perfil, pacotes, estrutura composta, desenvolvimento (arquitetura de redes) e componentes. Os da categoria de interação são: sequência, comunicação, tempo e geral de interação. Essa diversidade de diagramas é útil para que o sistema em desenvolvimento possa ser visto sob a ótica de

diversos aspectos, deixando-o o mais completo possível.

Os desenvolvedores do SIG-Library utilizaram o Diagrama de Casos de Uso (Diagrama Comportamental) e o Diagrama de Classes (Diagrama Estrutural) na sua modelagem.

Segundo o site DevMedia (www.devmedia.com.br), o Diagrama de Casos de Uso documenta o que o sistema faz do ponto de vista do usuário. Em outras palavras, ele descreve as principais funcionalidades (Chamadas de Casos de Uso) do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema (Chamados de atores). Esse diagrama não se aprofunda em detalhes técnicos que explicam como o sistema faz o que faz.

Também segundo o DevMedia, um Diagrama de Classes é uma representação da estrutura e relações das classes, estas são a representação de uma entidade do mundo real no plano computacional. As classes incluem todas as características e comportamentos que uma determinada coisa possui. Pode-se afirmar de maneira mais simples que são um conjunto de objetos com as mesmas características. Segundo o site Até o Momento (www.ateomomento.com.br), O diagrama é bem versátil quanto às possibilidades de aplicação prática e o importante é entender o contexto no qual o diagrama será aplicado, para então usá-lo da melhor forma. O Diagrama de Classes pode ser utilizado para diferentes finalidades no contexto de produção de software, destacando:

1. Design (projeto de software): definir um modelo a ser seguido para um software que ainda não existe, especificando um modelo antes de efetivamente construir software executável (diminuindo o desperdício).

2. Engenharia Reversa: refletir a estrutura já existente de um software onde, ha partir do código fonte realiza-se a leitura automática das classes e suas relações e se produz o diagrama (para compreensão da estrutura do software executável).

3. Esboço de ideias: desenhar modelos estruturais para troca de ideias e alinhamento entre analistas de sistemas, por exemplo. Neste caso utiliza-se muito as paredes “rabiscáveis” das empresas mais jovens ou papel e caneta mesmo.

3. Metodologia

O SIG-Library possui funcionalidades que pretendem aumentar a qualidade do atendimento, serviços prestados e proporcionar uma melhor fiscalização nos empréstimos de livros nas bibliotecas que o adotarem. O sistema conta com uma infraestrutura capaz de gerenciar funcionários (Bibliotecários), usuários (Leitores) e livros, tornando ágil a manutenção dos bens da biblioteca. Os leitores terão que fazer seu cadastro, com os bibliotecários, no balcão principal de atendimento para utilizar todos os serviços da biblioteca. O software garante a confiabilidade na gerência de empréstimos de obras monitorando a atividade dos bibliotecários, leitores e livros; fazendo o controle de quais usuários estão irregulares com a biblioteca e estão multados por não terem cumprido com o prazo de entrega estimado ou por terem perdido o livro. Além disso, o sistema conta com muitas opções de relatórios para aumentar ainda mais o nível do controle da demanda dos serviços prestados pela biblioteca. O software guarda relatórios de todos os empréstimos realizados e permite suas consultas filtrando por datas e outros parâmetros. A implantação do sistema utiliza uma arquitetura de rede local que faz a ligação entre as máquinas

terminais, operadas pelos bibliotecários, e o servidor da biblioteca, este contém um banco de dados responsável por armazenar os dados gerados conforme o uso do sistema.

Inicialmente, os autores desenvolveram o SIG-Library com o intuito de ser um projeto final que serviu como avaliação na disciplina Programação, ministrada pelo professor Flavius da Luz e Gorgonio. A elicitação e análise dos requisitos de software se deu com o auxílio do professor e com a observação dos fluxos de processos em bibliotecas presentes na cidade Caicó, Rio Grande do Norte. Os autores e desenvolvedores decidiram adotar e implementar como modelo no documento de requisitos do SIG-Library a abordagem proposta pelo renomado autor na área de Engenharia de Software Ian Sommerville. Os requisitos de software foram divididos em requisitos funcionais e requisitos não-funcionais.

*** Requisitos funcionais:**

[RF001] Gerenciamento do acervo: O sistema deverá possuir funcionalidades de Cadastro, Consulta, Atualizações e Exclusão dos livros. * Categoria: Obrigatório / Evidente.

[RF002] Gerenciamento de dados dos usuários: O sistema deverá possuir funcionalidades de Cadastro, Consulta, Atualização e Exclusão dos usuários. * Categoria: Obrigatório / Evidente.

[RF003] Registros das movimentações: Movimentações como empréstimo, renovação de empréstimo e devolução de obras devem ser registrados por meio de relatórios. * Categoria: Obrigatório / Evidente.

[RF004] Autenticação: Para garantir a segurança, só o bibliotecário poderá acessar o sistema, isto será feito através de sua autenticação por meio de um login e senha. * Categoria: Obrigatório / Evidente.

[RF005] Relatórios do acervo: O sistema deverá gerar relatórios do acervo bibliográfico e do material emprestado. * Categoria: Obrigatório / Evidente.

[RF006] Relatórios dos usuários: O sistema deverá gerar relatórios dos usuários cadastrados e controlar quais estão em débito e serão multados. * Categoria: Obrigatório / Evidente.

[RF007] Controle da disponibilidade: O sistema deverá possuir funcionalidades que sejam capazes de controlar os empréstimos e devoluções dos exemplares dos livros com eficiência. * Categoria: Obrigatório / Evidente.

[RF008] Relatórios estatísticos: O sistema deverá gerar relatórios sobre estatísticas úteis para avaliar a demanda dos livros do acervo e avaliar os usuários mais ativos da biblioteca. * Categoria: Obrigatório / Evidente.

[RF009] Controle da atividade dos leitores através de multas: O sistema deverá aplicar penalidades aos usuários que excederem os prazos de devolução dos livros. O software implementará dois tipos de funcionalidades de multa: a multa por atraso e a multa por perda do livro. A funcionalidade de multa por atraso, que será acionada automaticamente quando um usuário extrapola o prazo de devolução de um livro, calculará um tempo de restrição no qual o usuário será impedido de realizar novos empréstimos na biblioteca. A funcionalidade de multa por perda do livro, que será acionada pelo bibliotecário quando houver a perda do livro por parte do leitor, impedirá (Por tempo

indeterminado) novos empréstimos ao usuário, até que o mesmo pague a taxa necessária estabelecida. * Categoria: Obrigatório / Evidente.

[RF010] Pagamento da multa: O sistema deverá gerenciar os pagamentos em dinheiro dos usuários multados, armazenando as informações referentes aos pagamentos. * Categoria: Obrigatório / Evidente.

[RF011] Gerenciamento de dados dos bibliotecários: O sistema deverá possuir funcionalidades de Cadastro, Consulta, Atualização e Exclusão dos bibliotecários. * Categoria: Obrigatório / Evidente.

[RF012] Gerenciamento de logs do uso do sistema: O software deverá gerar relatórios dos períodos de uso do sistema por parte dos bibliotecários e os disponibiliza-los para consultas pelo Administrador do Sistema.

*** Requisitos não-funcionais:**

[RNF001] Design: O design do programa deve ser intuitivo e com menus bem organizados em diferentes níveis. * Categoria: Evidente.

[RNF002] Armazenamento dos dados: O SIG-Library deve operar com eficiência no armazenamento e manutenção do banco de dados, de modo a garantir escalabilidade, segurança, privacidade e integridade aos dados armazenados. * Categoria: Obrigatório / Oculto.

[RNF003] Segurança: Todas as funcionalidades do sistema só poderão ser acionadas pelo bibliotecário. * Categoria: Obrigatório / Evidente.

[RNF004] Controle: O sistema não deverá realizar empréstimos caso o usuário não esteja em dias com a biblioteca (Multado). * Categoria: Obrigatório / Evidente.

[RNF005] Acervo permanente: Cada livro terá um exemplar permanente, o qual será identificado por uma tarja vermelha e não poderá ser emprestado. * Categoria: Obrigatório / Lógica de Negócio.

[RNF006] Exclusão permanente: Se um leitor ou livro for excluído permanentemente do banco de dados, ele não mais poderá ser localizado em nenhuma consulta. * Categoria: Obrigatório / Oculto.

[RNF007] O software deverá ser capaz de restaurar dados excluídos a partir do seu backup lógico. Categoria: Obrigatório / Evidente.

Para projetar e modelar o sistema, os autores optaram por utilizar o software Astah UML (astah.net) e implementaram o Diagrama de Casos de Uso e o Diagrama de Classes da UML.

Diagrama de Casos de Uso:

Os Atores identificados são:

Administrador do Sistema: Esse ator refere-se ao funcionário especialista em Tecnologia da Informação que é responsável por manter o sistema operando em perfeitas condições. Ele é o superusuário e possui credenciais (Login e Senha) que garantem o acesso as funções de configuração do sistema. Ele deve conhecer todas as variáveis necessárias para que o sistema funcione sem nenhum problema, ele deve saber fazer a manutenção da

infraestrutura de rede e servidor (Que armazena o banco de dados) e também é responsável por fazer o gerenciamento das informações dos bibliotecários que atuam na biblioteca, também definindo as credenciais de acesso para os bibliotecários poderem logar no sistema. O Administrador do Sistema também é responsável por manter e analisar os logs de uso do sistema pelos bibliotecários.

Bibliotecário: Esse ator refere-se aos funcionários balconistas da biblioteca, essas pessoas serão responsáveis por prestar os serviços que a biblioteca oferece e realizar todas as operações no software que a gerencia. Os bibliotecários são cadastrados no sistema pelo Administrador do Sistema. Os bibliotecários possuem credenciais de acesso (Login e Senha) para poder ter acesso as funcionalidades operacionais.

Leitor: Esse ator refere-se aos possíveis usuários dos serviços da biblioteca, são pessoas civis (Tratando-se de uma biblioteca pública municipal) ou alunos (Tratando-se de uma biblioteca escolar) que irão solicitar empréstimos de livros aos bibliotecários. Em caso de atraso na devolução ou na perda do livro, eles devem pagar uma taxa e podem ser multados, como forma de punição, para não poderem pegar mais livros emprestados por uma determinada quantidade de tempo.

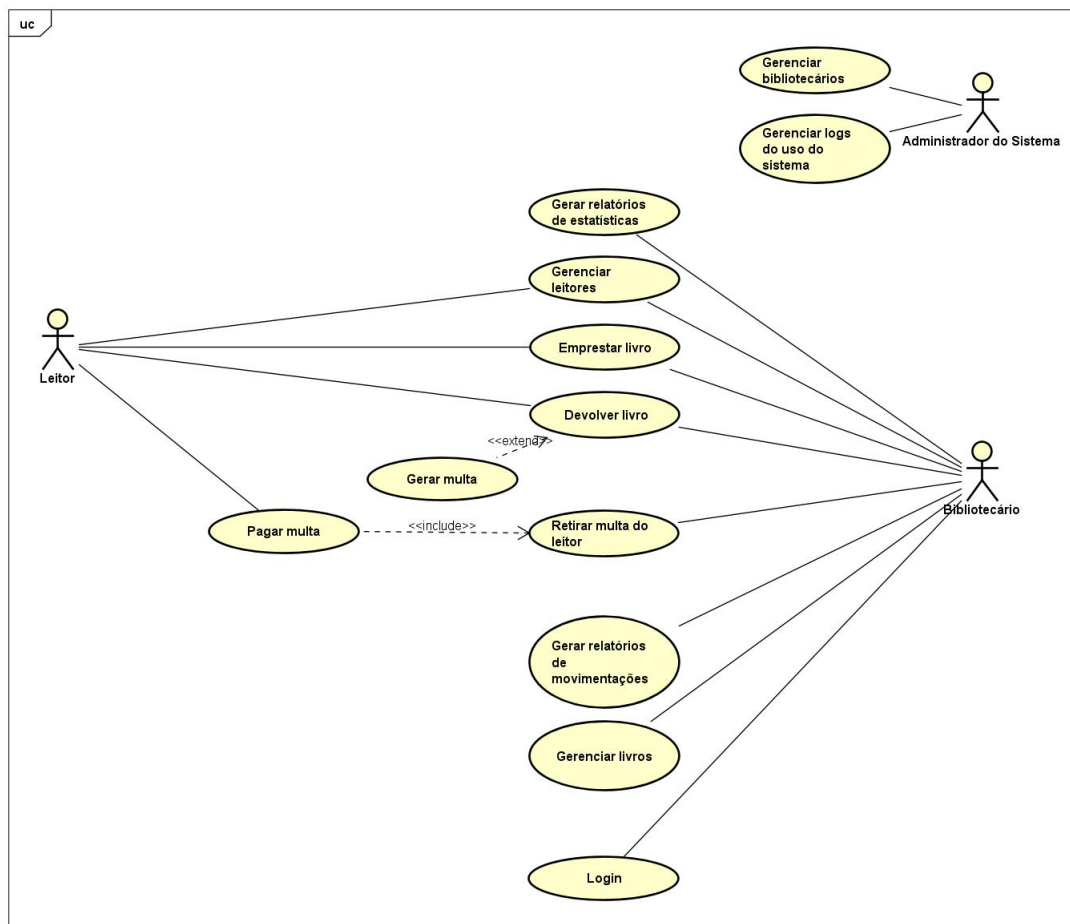


Figura 1. Diagrama de Casos de Uso
Fonte: Os próprios autores (2020)

Diagrama de Classes:

4. Conclusão

No projeto prático da disciplina Organização, Sistemas e Métodos foi desenvolvida a documentação da Engenharia de Requisitos do software SIG-Library utilizando dois tipos de diagramas UML: o Diagrama de Classes e o Diagrama de Casos de Uso. Estes dois diagramas são necessários para se ter diferentes perspectivas sobre o sistema desenvolvido. Além disso, foram desenvolvidos protótipos descartáveis para se ter uma prévia de como o sistema irá aparentar visualmente quando o seu desenvolvimento estiver concluído.

Protótipo descartável feito pelos autores:



Fonte: Os próprios autores (2020)



Fonte: Os próprios autores (2020)



Fonte: Os próprios autores (2020)



Fonte: Os próprios autores (2020)



Fonte: Os próprios autores (2020)

A documentação escrita é de extrema importância para que um sistema robusto seja desenvolvido, atendendo todas as demandas e funcionalidades que são solicitadas pelos clientes. Conforme foram estudando as técnicas empregadas na Engenharia de Requisitos e a UML, a mente dos autores do projeto se abriu; eles puderam perceber que a primeira versão do software desenvolvido na disciplina Programação continha inúmeras inconsistências e poderia ser bastante otimizado, visto que na época que cursaram tal disciplina os autores não conheciam as técnicas eficientes utilizadas pela Engenharia de Software.

Como possíveis trabalhos futuros, os autores recomendam e pretendem realizar as outras modelagens que podem ser feitas utilizando os outros diagramas que a UML oferece, em especial utilizar o Diagrama de Sequência em todos os casos de uso, pois os autores gostariam de visualizar todo o fluxo de execução do sistema.

Seria extremamente interessante implementar a funcionalidade que possibilite os leitores poderem reservar livros. Seria necessário adaptar o escopo do sistema, de modo que ele suportasse a criação de contas para os leitores e eles pudessem utiliza-lo a partir da internet, transformando o SIG-Library em um Sistema Web.

Referências

- Booch, G., Rumbaugh, J., and Jacobson, I. (2006). *UML: guia do usuário*. Elsevier Brasil.
- Lobo, E. J. (2009). *Guia prático de engenharia de software*. Universo dos Livros Editora.
- Machado, F. N. R. (2018). *Análise e Gestão de Requisitos de Software Onde nascem os sistemas*. Saraiva Educação SA.