

Estudo de caso no levantamento de requisitos e modelagem conceitual sobre o uso da linguagem de modelagem UML em um sistema de planejamento alimentar.

Arthur Anthony da Cunha Romão E Silva¹, Adalberto Maia da Silva¹

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

{arthuranthony, btow.m}@ufrn.edu.br,

Resumo. *Este artigo tem como objetivo abordar os principais conceitos relacionados ao desenvolvimento de software, como engenharia de software, engenharia de requisitos, levantamentos de requisitos, classificação de requisitos, e efetuar um estudo de caso sob o levantamento de requisitos e modelagem conceitual de um sistema de planejamento alimentar como atividade avaliativa da disciplina de Organização, Sistemas e Métodos ministrada pela professora Karliane durante o período suplementar remoto 2020.5*

1. Introdução

É comum a busca por planejamento alimentar, a criação de dietas com o objetivo de emagrecer, ganhar massa muscular ou simplesmente manter o peso sem riscos de excessos. Todavia, para se obter uma dieta saudável é preciso quantidades adequadas de carboidratos, proteínas e gorduras na alimentação, geralmente prescritas por um nutricionista.

Devido a chegada dos fast-foods, a industrialização, o uso de agrotóxicos e os processos de automação que se aperfeiçoaram no século presente, geraram um grande desafio com relação a saúde das pessoas. O século da obesidade, do sedentarismo, e da multitarefa, as pessoas precisam tomar medidas preventivas e corretivas em relação a saúde e bem-estar, que implicam diretamente na sua qualidade de vida e produtividade.

Sendo assim está sendo proposto um sistema web que ajude as pessoas com sua saúde para uma melhor qualidade de vida. Assim o sistema proposto será capaz de fornecer dietas a gosto de seus consumidores, dietas semanais, mensais, trimestrais ou de maneira personalizada. Possuirá um módulo de busca e filtros que garantem a integridade do consumidor, caso o mesmo possua restrições. Também permitirá o usuário gerar receitas a partir do computador ou a tradicional consulta ao nutricionista. A realização de acompanhamentos com nutricionistas por meio de dietas geradas pelo computador. A geração de dietas com a divisão na quantidade de refeições e com as restrições definidas pelo consumidor. A dieta atenderá todas as exigências, quantidade de calorias, quantidade de gordura, carboidratos e proteínas. E será gerado gráficos de progresso com período definido pelo consumidor em relação aos resultados do mesmo.

O objetivo deste trabalho é mostrar sucintamente a modelagem de um sistema de planejamento alimentar, desde o levantamento de requisitos, partindo da elicitação, análise e validação e a documentação dos requisitos, até a modelagem conceitual por meio de diagramas UML, utilizando ferramentas CASE como o Astah UML.

Os conceitos abordados durante este relatório são os conceitos de Linguagem de Modelagem Unificada (UML), o padrão utilizado nos projetos de desenvolvimento de sistemas, modelando o sistema sob diversos tipos de perspectivas com a finalidade de auxiliar o processo de desenvolvimento do software. Também é tratado de temas mais chaves, que são os requisitos funcionais, que tratam das funcionalidades do sistema, e os requisitos não funcionais que tratam das restrições do sistema.

2. Referencial Teórico

2.1. Engenharia de Software

A Engenharia de Software é o conjunto de metodologias aplicadas no desenvolvimento e manutenção de um software, visando qualidade durante o processo. Aplicando assim uma abordagem sistemática, disciplinada e quantificável durante o processo de desenvolvimento, operação e manutenção do software. [Pressman 2005]. Durante o processo é necessário que as etapas sejam efetuadas de maneira ordenada e organizada, estipulando prazos e custos, utilizando-se de todas as técnicas e modelos a disposição de maneira adequada, ou seja, a aplicação de padrões e técnicas para se desenvolver um software.

É importante definir o conceito de software, segundo [TURBAN et al. 2007], software é um programa ou conjunto de programas que permite que o hardware processe os dados, e segundo [Laudon and Laudon 2004] “O Software consiste em instruções detalhadas e pré programadas que controlam e coordenam os componentes do Hardware de um Sistema de Informação”. Após o advento da internet no ano de 1969, o mundo vem passando por uma revolução tecnológica que vem afetando cada vez mais a vida das pessoas, desde a maneira das mesmas se comunicarem, exemplo disso a criação das redes sociais, do simples acesso à informação até o surgimento de coisas mais futuristas, como a automação residencial, dentre outros.

Segundo [Pressman 2005] um processo de desenvolvimento de software possui três fases genéricas, independente do modelo de processo escolhido, sendo estes: definição, desenvolvimento e manutenção. Segue um conjunto de etapas, bem definidas e ordenadas, segundo Magela (2006), é definido as seguintes etapas: Análise de Requisitos, Análise de Negócio, Análise de Sistema, Projeto de Sistema, Implementação, Teste de Sistema, Implantação e distribuição do Software.

2.2. Engenharia de Requisitos

A Engenharia de requisitos trata de tudo que está relacionado com a extração, especificação e documentação dos requisitos de um software. Segundo [de Pádua Paula Filho 2003], é definida como um conjunto de técnicas de levantamento, documentação e análise dos requisitos dos produtos de software. Termo associado a tudo que está relacionado com a definição do escopo de um software, isto é a produção dos seus requisitos e a gerência dos mesmos, envolve técnicas, metodologias e artefatos para a investigação e validação dos requisitos e restrições do sistema e a conseqüentemente a sua respectiva documentação. A Engenharia de Requisitos é entendida segundo [Peters and Pedrycz 2001] a partir destes fatores: concepção do problema, escolha de soluções testáveis, com manutenibilidade de forma que ratifique as condutas que prezam pela qualidade do projeto.

O Levantamento de Requisitos é um subconjunto da engenharia de requisitos, que envolve a elicitacão, análise e registro dos requisitos. Elicitacão, extração dos re-

quisitos com os stakeholders. Análise, Validação dos requisitos e a definição (ambíguo, contraditório e inacabado) e por último o registro, que é documento de requisitos produzidos pelas duas fases anteriores. Segundo [Ian 2003], “os requisitos de um sistema são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais.” Também pode ser definido segundo [Magela 2006], como sendo condições extraídas dos processos e regras de negócio de uma empresa, com a finalidade de definir as funcionalidades do software.

Os requisitos são de suma importância, pois esclarecerá em conjunto do analista e o cliente o que o sistema fará, é a descrição daquilo que será automatizado do objeto de negócio na qual está sendo analisado, além disso, um bom levantamento de requisitos leva a um desempenho melhor durante a implementação e validação do sistema, diminuindo o custo de retrabalho e garantindo uma melhor qualidade de software. É importante destacar que os mesmos são divididos geralmente em três tipos, [Ian 2003] divide os requisitos de sistema de Software em três, funcionais, não funcionais e de domínio. O primeiro tipo segundo [Magela 2006] é o que o software deve realizar, como também seu comportamento com as variáveis externas do software. O segundo tipo de requisito segundo [Ian 2003], são as restrições sob as funcionalidades do sistema e os serviços. E o terceiro segundo [Ian 2003], são requisitos derivados do domínio, refletindo as características e as restrições do mesmo. Sendo classificados como funcionais ou não funcionais.

2.3. Processo de Modelagem UML (Unified Modeling Language)

O processo de modelagem de um sistema é de suma importância, pois utilizará de recursos gráficos a partir do exercício de abstração dos desenvolvedores em definir em diversos aspectos e pontos de vista, sob diversos níveis de abstração o sistema a ser desenvolvido. Desde o processo e as regras de negócio envolvidas no domínio, o comportamento de certos aspectos do sistema, a iteração dos processos entre as entidades envolvidas à estrutura do sistema e os componentes que se relacionam entre si. Segundo os autores [Booch et al. 2006], “a modelagem é uma técnica de engenharia aprovada e bem-aceita. Construímos modelos de arquitetura de casas e de grandes prédios para auxiliá-los seus usuários a visualizar qual será o produto final”.

A UML é a linguagem utilizada para a modelagem de sistemas, seu principal objetivo é definir as características que o sistema deve ter sob diversos tipos de perspectivas. A criação de uma linguagem unificada se deu por causa do surgimento de sistemas bastante complexos, percebendo-se a necessidade da criação de uma linguagem que se tornasse padrão na modelagem de sistemas. É uma linguagem criada para visualizar, especificar, construir e documentar os artefatos de um sistema de software. É utilizada em todas as etapas de desenvolvimento sob determinada perspectiva, comportamental, estrutural ou iterativa, e em determinados níveis de abstração [Booch et al. 2006].

2.4. Diagrama de Caso de Uso

O diagrama de casos é amplamente utilizado nos projetos de desenvolvimento de software, é utilizado para que se tenha uma visão geral sobre os fatores ou funções do sistema ou subsistemas e as entidades envolvidas nestas funcionalidades, no caso os atores. Segundo [Furlan 1998], pode ser definido como “um conjunto de sequências de ações que um sistema desempenha para produzir um resultado observável de valor a um ator específico”.

O conceito chave é o caso de uso, que é uma funcionalidade que o sistema deverá ter, e esta funcionalidade desencadeará um conjunto de ações promovidas pela interação com os atores. É utilizado para mostrar como um sistema funciona, o seu comportamento, mostrando fatores e suas interações com os atores, e o desencadeamento de um cenário principal e cenários alternativos. [Melo 2004]

Outro fator importante no diagrama de caso de uso são os atores, segundo [Larman 2007] um ator é uma entidade externa ao sistema que, de alguma maneira, interage na história ao caso de uso. São basicamente os agentes externos ao sistema que se utiliza de funcionalidades para interagir com o mesmo, seja para receber ou entregar algo. São identificados geralmente pelos seus papéis em relação ao sistema.

2.5. Diagrama de Classe

O diagrama de classe é o mais importante dentre outros proporcionados pela UML, ele é do tipo estrutural, utilizado para enfatizar a estrutura lógica entre os modelos e associações em um sistema, suas funcionalidades e características. Segundo [Furlan 1998], o diagrama de classes é uma estrutura lógica estática em uma superfície de duas dimensões mostrando uma coleção de elementos declarativos de modelo, como classes, tipos e seus respectivos conteúdos e relações.

O conceito chave é o de classes, é a abstração de um modelo do mundo real, um tipo de dado personalizado que possui atributos e métodos e se associa com outros modelos, representam a entidade envolvida no processo de um domínio, é a base do software, pois o mesmo é composto por classes e suas associações. É a descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica [Booch et al. 2006].

É importante destacar que as classes são as partes que formam o todo, que o software, e que as mesmas não trabalham de modo isolado, como afirma [Melo 2004], os tipos de relacionamentos entre as classes são indispensáveis, das quais são: dependências, generalizações e associações. A estrutura gráfica de uma classe em um diagrama de classes pela UML é dividido em três partições, das quais são respectivamente, o nome da classe com ou não seu esteriótipo, seus atributos, que compõe seu modificador de visibilidade, seu nome e seu tipo, pode ter outras informações, como valor predefinido ou se é um atributo calculo/derivado ou seu esteriótipo, e na última partição seus métodos, que possuem um modificador de acesso, seu nome, seus parâmetros e o tipo de valor de retorno.

É importante destacar que as classes são as partes que formam o todo, que o software, e que as mesmas não trabalham de modo isolado, como afirma [Melo 2004], os tipos de relacionamentos entre as classes são indispensáveis, das quais são: dependências, generalizações e associações. A estrutura gráfica de uma classe em um diagrama de classes pela UML é dividido em três partições, das quais são respectivamente, o nome da classe com ou não seu esteriótipo, seus atributos, que compõe seu modificador de visibilidade, seu nome e seu tipo, pode ter outras informações, como valor predefinido ou se é um atributo calculo/derivado ou seu esteriótipo, e na última partição seus métodos, que possuem um modificador de acesso, seu nome, seus parâmetros e o tipo de valor de retorno.

3. Metodologia

A modelagem do Sistema de Planejamento Alimentar foi realizada durante a disciplina Organização, Sistemas e Métodos, ministrada pela professora Karliane através do período suplementar 2020.5 remotamente. As tecnologias que foram utilizadas durante o processo de modelagem foram: Astah UML, LibreOffice Writer e Wix. O trabalho se deu pela utilização de recursos de modelagens, como tabelas, diagramas e protótipos, como auxílio e organização durante o processo da modelagem.

As etapas realizadas durante a modelagem são: Elicitação dos Requisitos; Análise dos Requisitos; Registro dos Requisitos (Tabelamento dos Requisitos Funcionais e Não Funcionais); Tabelamento dos Atores; Tabelamento Geral dos Casos de Uso; Criação do diagrama de Casos de Uso; Tabelamento específico dos Casos de Uso; Criação do diagrama de classes e Prototipagem.

Inicialmente foi abordado um estudo sobre sistemas de planejamento alimentar sob a perspectiva do avanço tecnológico, hábitos alimentares e a relação com o sedentarismo presente no século XXI. Em seguida foi realizado um estudo a cerca da linguagem de modelagem unificada (UML), com ênfase nos diagramas de casos de uso e de classes para tratar do tema em específico. Após isso foi feito o processo de levantamento de requisitos, extraíndo os requisitos funcionais e não funcionais do projeto, dando início a escrita do documento de requisitos do sistema. Foi criado a tabelas dos requisitos funcionais e não funcionais após uma escrita informal e o tabelamento dos atores e geral dos casos de uso.

Posteriormente foi desenvolvido o diagrama de casos de uso com base na tabela geral de caso de uso, com isto foi criado novas tabelas para expressar o funcionamento de determinados casos de uso, a fim de esclarecer determinados aspectos do negócio e do sistema a ser projetado. Finalmente foi desenvolvido o diagrama de classes, para obter-se uma visão estrutural do sistema a ser desenvolvido, e a criação de um protótipo com determinados aspectos que o sistema terá, sessões, templates dos planos e preços, e do módulo do consumidor.

3.1. Diagrama de Casos de Uso

A figura abaixo demonstra, através do diagrama de caso de uso, as funcionalidades, divididas em módulos do sistema de planejamento alimentar.

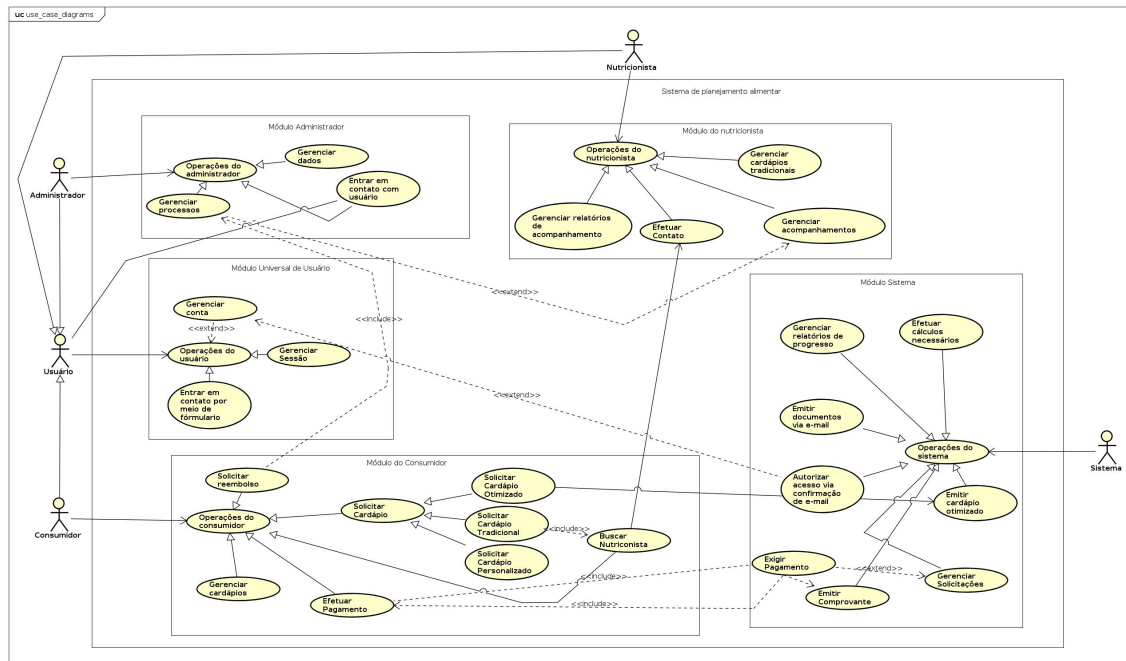


Figura 1. Diagrama de Caso de Uso do Sistema de Planejamento Alimentar
Fonte: O Próprio Autor (2020)

O sistema é dividido em três módulos principais, o módulo universal de usuário, que será utilizado por todos os usuários do sistema, do consumidor, nutricionista, administrador e do sistema. O módulo universal de usuário é o gerenciamento da conta de cada usuário do sistema, e as funcionalidade fundamentais, como realizar login ou logout incluídos no caso gerenciar sessão, também será possível efetuar um formulário, para entrar em contato com a equipe, pra dar alguma sugestão ou reclamar alguma coisa, pode ser um formulário apenas de contato simples ou um formulário de processo para solicitar reembolso, por exemplo.

O módulo do consumidor é compete aquele que comprará os serviços prestados ao sistema em relação ao planejamento alimentar em si. O consumidor poderá gerenciar os cardápios ativos e antigos (expirados), poderá efetuar pagamento de pacotes pendentes e solicitará o cardápio que bem desejar, poderá entrar em contato com algum nutricionista para mais informações, e solicitar o reembolso de algum pacote.

Já o módulo do nutricionista compete aos profissionais da nutrição que realizarão acompanhamentos com os consumidores. O nutricionista será capaz de gerenciar seus acompanhamentos, entrar em contato com consumidores, gerenciar relatórios dos acompanhamentos e emitir os cardápios propriamente ditos.

Por fim o módulo do administrador compete a equipe responsável por gerenciar o sistema, na qual um administrador poderá gerenciar os dados contidos no sistema (banco de

dados), como os dados dos consumidores, nutricionistas, alimentos, dentre outros. Ainda será capaz de autorizar processos emitidos a partir de nutricionistas ou consumidores e entrar em contato com os mesmos. Por último poderá analisar os dados do sistema, gerar relatórios e gráficos.

3.2. Diagrama de Classe

A figura abaixo demonstra, através do diagrama de classe, as entidades, suas características e funcionalidades, seus atributos e suas associações do sistema de planejamento alimentar.

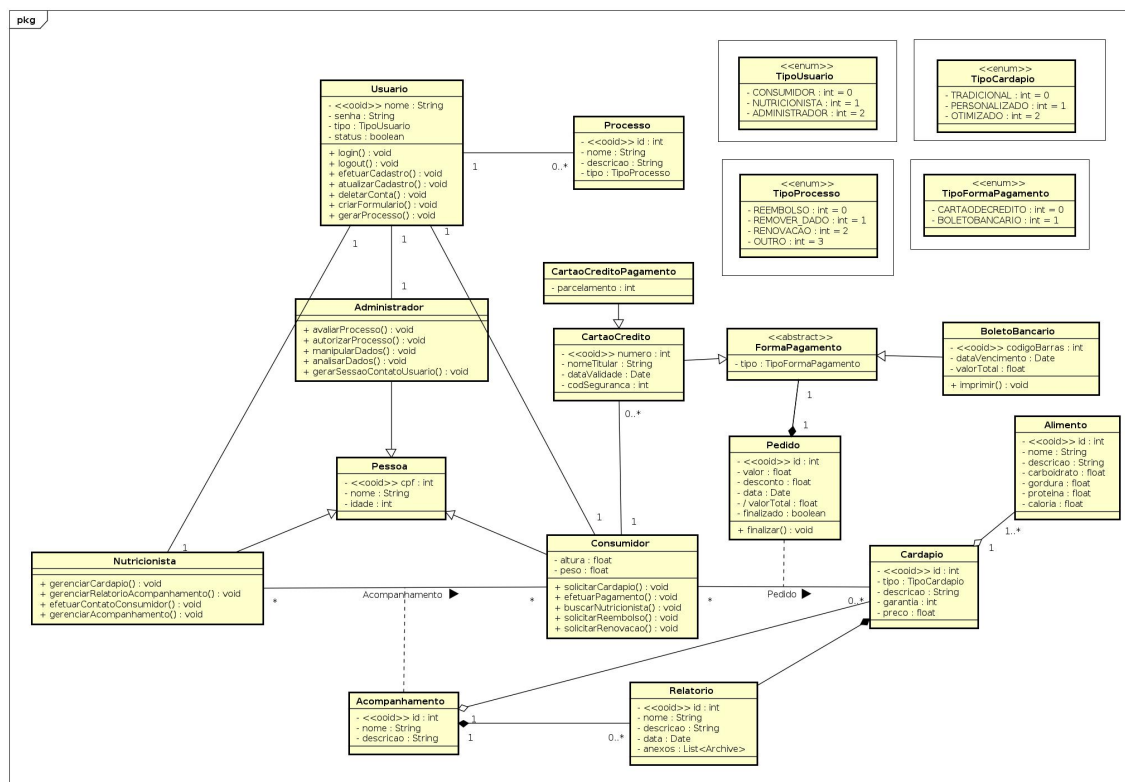


Figura 2. Diagrama de Classes do Sistema de Planejamento Alimentar

Fonte: O Próprio Autor (2020)

O diagrama de classes possui dezenove estruturas e representam a arquitetura que o sistema deverá possuir, as principais estruturas que compõe são o consumidor, nutricionista, alimento e cardápio. O consumidor em conjunto com o nutricionista geram um acompanhamento e este pode ter cardápios, tratando do pacote tradicional, o consumidor também efetuar o pedido individual de cardápios, sendo otimizado ou personalizado.

4. Conclusão

O presente trabalho apresentou um estudo de caso de um sistema de planejamento alimentar como atividade avaliativa da disciplina Organização, Sistemas e Métodos. Foi levantado conceitos importantes no setor de desenvolvimento de software, como engenharia de software, engenharia de requisitos, requisitos funcionais e não funcionais e o levantamento de requisitos.

Foi abordado o levantamento de requisitos do sistema de planejamento alimentar com a modelagem conceitual, abordando dois diagramas da linguagem de modelagem unificada (UML), o diagrama de casos de uso e o diagrama de classes. O presente trabalho servirá para possíveis melhorias durante próximos passos do desenvolvimento deste software, abordando outras perspectivas de modelagem, como o uso do diagrama de pacotes, de sequência, de atividade, bem como outros, e o desenvolvimento do software em si.

Referências

- Booch, G., Rumbaugh, J., and Jacobson, I. (2006). *UML: guia do usuário*. Elsevier Brasil.
- de Pádua Paula Filho, W. (2003). *Engenharia de software*, volume 2. LTC.
- Furlan, J. D. (1998). *Modelagem de objetos através da UML-the unified modeling language*. Makron books.
- Ian, S. (2003). *Engenharia de software. 6a. edição*, Addison-Wesley/Pearson.
- Larman, Craig, U. (2007). *padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo*.
- Laudon, K. C. and Laudon, J. P. (2004). *Sistemas de informação gerenciais*. 5ª edição.
- Magela, R. (2006). *Engenharia de software aplicada: fundamentos*. Rio de Janeiro–RJ: Alta Books.
- Melo, A. C. (2004). *Desenvolvendo aplicações com UML 2.2*. Brasport.
- Peters, J. F. and Pedrycz, W. (2001). *Engenharia de software: teoria e prática*. Rio de Janeiro: Campus, 681(519.683):2.
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave macmillan.
- TURBAN, E., RAINER Jr, R., and Potter, R. E. (2007). *Introdução a sistemas da informação*. tradução: Daniel vieira.