

MODELAGEM VISUAL DE UM SOFTWARE PARA O GERENCIAMENTO DAS COMUNICAÇÕES EM GESTÃO DE PROJETOS

VISUAL MODELING OF SOFTWARE FOR COMMUNICATION MANAGEMENT IN PROJECT MANAGEMENT

Ricardo Rall¹ Arilson José de Oliveira Junior²

RESUMO

Durante o ciclo de desenvolvimento de um software, a depender da sua complexidade algorítmica e/ou quantidade de funções, é necessária a utilização de modelos de processos de software, que especificam etapas nas quais são determinadas atividades que devem ser realizadas até a entrega do software. A modelagem visual é uma ferramenta que representa através de notações aspectos estruturais e possíveis funções de um programa computacional, e que pode ser utilizada em determinados modelos de processos de software. Muitos profissionais que atuam no desenvolvimento de softwares e até mesmo empresas que desenvolvem e implantam programas computacionais, não aplicam a modelagem visual durante seus processos de desenvolvimento, por não possuírem conhecimento dos padrões e processos necessário para elaboração dos diagramas, ou por não possuírem um modelo de processos de software aplicado. Desta forma, este trabalho teve como objetivo representar as notações de quatro diagramas da Linguagem de Modelagem Unificada (UML – *Unified Modeling Language*) e a importância da modelagem de softwares em paradigmas de desenvolvimento. Para a elaboração da modelagem visual foram utilizados os diagramas de Casos de Uso, Atividades, Classes e Implantação da UML 2.0. Como resultado, obteve-se um diagrama de Casos de Uso seguido de quatro descrições estruturais, três diagramas de Atividades, um diagrama de Classes e um diagrama de Implantação. A partir dos resultados, conclui-se que a modelagem de software, por meio dos diagramas da UML, é de fácil aprendizado e compreensão por uma equipe previamente capacitada em desenvolvimento de softwares. E que a modelagem visual apresenta grande importância nos modelos de processos de software, já que por ela é possível entender as funcionalidades que um software possui ou deve possuir.

Palavras-chave: Modelagem visual. Modelagem de software. UML.

¹ Professor da Faculdade de Tecnologia de Botucatu. Av. José Ítalo Bacchi, s/n – Jd. Aeroporto – Botucatu/SP – CEP: 18606-851. E-mail: rrall@fatecbt.edu.br

² Faculdade de Ciências Agrárias de Botucatu – UNESP E-mail arilsonjr@outlook.com

ABSTRACT

During the development cycle of software, depending on algorithmic complexity and/or number of functions, it is necessary to use software model processes which specify the steps in which activities are determined in order to be performed until software delivery.

Visual modeling is a tool which represents through noting of structural features and possible functions a computer program which can be used in certain software process models. Many professionals working with software development or even in companies that develop and implement computer programs do not apply visual modeling during development processes due to lack of knowledge on necessary patterns and processes for elaborating diagrams as well as lacking ability with applied software model processes. Thus, this paper aimed to represent the notations of four diagrams of Unified Modeling Language (UML) and the importance of software modeling in development paradigms. For the development of the visual modeling it was used *Use Cases*, *Activities*, *Classes* and *Deployment* of UML 2.0 diagrams. As a result, it was obtained *Use Cases* diagram followed by four structural descriptions, three *Activities* diagrams, one *Classes* diagram and one diagram of *Deployment*. Results showed that software modeling, using UML diagrams, is of easy learning and understanding by a previously trained team in software development. Results also showed that visual modeling is of great importance on software model processes for it is possible to understand the features that software has or should have.

Keywords: Visual Modeling. Software Modeling. UML.

¹ Professor da Faculdade de Tecnologia de Botucatu. Av. José Ítalo Bacchi, s/n – Jd. Aeroporto – Botucatu/SP – CEP: 18606-851. E-mail: rrall@fatecbt.edu.br

² Faculdade de Ciências Agrárias de Botucatu – UNESP E-mail arilsonjr@outlook.com

1 INTRODUÇÃO

Durante o ciclo de desenvolvimento de um programa computacional, a depender da sua complexidade algorítmica e/ou quantidade de funções, é necessária a utilização de um paradigma de desenvolvimento. Os paradigmas de desenvolvimento, hoje também conhecidos como modelos de processos de software, especificam etapas nas quais são determinadas atividades que devem ser realizadas até a entrega do software (SOMMERVILLE, 2011).

A modelagem visual é uma ferramenta que pode ser aplicada em determinados modelos de processos de software. Segundo Nunes e O’Neill (2000), consideramos a modelagem visual como a representação de aspectos estruturais e das possíveis funções de um programa computacional, através de símbolos padronizados com significados específicos.

Muitos profissionais que atuam no desenvolvimento de softwares e até mesmo empresas que desenvolvem e implantam constantemente programas computacionais, não aplicam a modelagem visual durante seus processos de desenvolvimento. O motivo da não utilização desta ferramenta, a princípio, advém da falta de conhecimento dos padrões e processos necessários para a elaboração dos diagramas e dos benefícios que a modelagem proporciona, ou por não possuírem um modelo de processos de software aplicado.

Atualmente a modelagem visual (também chamada de “Modelagem de software”), juntamente com os modelos de processos de software, ganhou destaque nas fábricas de softwares. A importância da aplicação de métodos para o desenvolvimento de programas computacionais se dá pelo fato de que os métodos e a modelagem definem, a partir de notações, um meio de comunicação uniforme entre a equipe de desenvolvimento e produtos de trabalho padronizados (HIRAMA, 2011). Para a aplicação da modelagem visual, existe uma linguagem de modelagem composta por um conjunto de notações que auxiliam e orientam os profissionais a desenvolverem diagramas comportamentais e estruturais referentes a um determinado software. Esta linguagem é conhecida como Linguagem de Modelagem Unificada (UML – *Unified Modeling Language*), que possui na sua segunda edição nove diagramas: Casos de Uso, Classes, Objetos, Sequências, Colaborações, Gráficos de Estados, Atividades, Componentes e Implantação (BOOCK; RUMBAUGH; JACOBSON, 1999; NUNES; O’NEILL, 2000).

Como parte dos modelos de processos de software e da modelagem visual, são necessários um conjunto de requisitos do programa a ser desenvolvido para que os diagramas possam ser elaborados. Assim, foi utilizado como requisitos de software os processos do gerenciamento das comunicações em gestão de projetos. O gerenciamento de projetos, de

acordo com o Project Management Institute (2012), é o emprego de ferramentas, técnicas, conhecimentos e habilidades durante o ciclo de vida de um projeto. A gerência de projetos é composta por nove áreas de conhecimento, dentre elas o gerenciamento das comunicações, que tem por objetivo garantir que todas as partes envolvidas num projeto mantenham uma comunicação eficiente, buscando assegurar ao gerente do projeto de que as informações estão sendo geradas, coletadas, distribuídas, armazenadas, recuperadas e organizadas (PROJECT MANAGEMENT INSTITUTE, 2008; GREENCE; STELLMAN, 2010).

Este projeto demonstrou sua importância por aplicar a modelagem visual em um conjunto de requisitos reais de um determinado segmento. Elaborando, através da Linguagem de Modelagem Unificada (UML), os diagramas de Casos de Uso, Atividades, Classes e Implantação, de um possível software para o gerenciamento das comunicações em gestão de projetos. O objetivo deste trabalho foi desenvolver e demonstrar a representação das notações de quatro diagramas da Linguagem de Modelagem Unificada (UML) e a importância da modelagem de softwares em seu ciclo de desenvolvimento.

2 MATERIAL E MÉTODOS

Para o desenvolvimento dos diagramas de Casos de Uso, Atividades, Classes e Implantação, foi utilizado o software Microsoft Visio Professional 2010 Versão: 14.0.6123 (32-bit).

Os requisitos utilizados como base para a modelagem visual foram coletados seguindo especificações do Project Management Institute (2008), do Guia do Conhecimento em Gerenciamento de Projetos (PMBOK – *Project Management Body of Knowledge*), em sua quarta edição. O PMBOK não descreve os processos das áreas de conhecimento em forma de requisitos de software, mas sim em grupos de processos que possuem entradas, ferramentas e técnicas e saídas. A partir desses grupos de processos, foram projetadas as funcionalidades do software.

Conforme especificações descritas em Boock, Rumbaugh e Jacobson (1999), a modelagem visual, por meio da UML versão 2.0, foi realizada utilizando quatro diagramas: Diagrama de Casos de Uso (representação dos requisitos do software por meio de casos de uso, atores e relacionamentos), Diagrama de Atividades (modelagem de aspectos dinâmicos do sistema, envolvendo as etapas sequenciais, concorrentes ou não, de um processo computacional), Diagrama de Classes (representação da estrutura dos dados e seus relacionamentos através de classes, atributos e métodos) e Diagrama de Implantação

3.1.1 Descrição Estruturada

A descrição estruturada especifica a pré-condição, descrição e pós-condição dos principais Casos de Uso, tendo como objetivo:

- Especificar as condições necessárias para que um determinado Caso de Uso seja executado;
- Descrever as ações necessárias para a realização de um Caso de Uso, detalhando quando e como o mesmo se inicia;
- Especificar a saída de um determinado Caso de Uso.

Foram criados a descrição estrutura dos seguintes Casos de Uso: Compor Mensagem, Criar Estratégia de Gerenciamento, Criar Plano de Comunicação, Gerar Smart Report e Visualizar Notificações.

O Quadro 1 ilustra o Caso de Uso **Compor Mensagem**.

Quadro 1 - Compor Mensagem

Compor Mensagem (cenário principal)	
Pré-condição	-
Descrição	<ol style="list-style-type: none"> 1) <i>Include: Inserir Stakeholder(s).</i> 2) O Caso de Uso começa quando o Stakeholder seleciona a opção Compor no painel principal do sistema. 3) O Sistema então abre uma tela para que os detalhes da nova comunicação sejam registrados. 4) Todos os campos devem ser preenchidos para que a nova mensagem seja enviada. 5) Após o preenchimento dos campos, a opção “Enviar” envia a comunicação ao seu destinatário.
Pós-condição	Comunicação realizada.

O Quadro 2 ilustra o Caso de Uso **Criar Estratégia de Gerenciamento**

Quadro 2 - Criar Estratégia de Gerenciamento

Criar Estratégia de Gerenciamento (cenário principal)	
Pré-condição	-
Descrição	<ol style="list-style-type: none"> 1) Include: Inserir Stakeholder(s). 2) Include: Configurar Programa. 3) O Caso de Uso começa quando o Gerente de Projetos acessa o painel Estratégia de Gerenciamento. 4) O Gerente de Projetos deve preencher todos os campos da tabela. 5) Após o preenchimento dos campos, a opção “Definir Estratégia” cria a Estratégia de Gerenciamento.
Pós-condição	Estratégia de Gerenciamento definida.

No Quadro 3, observa-se o Caso de Uso **Criar Plano de Comunicação**.

Quadro 3 - Criar Plano de Comunicação

Criar Plano de Comunicação (cenário principal)	
Pré-condição	-
Descrição	<ol style="list-style-type: none"> 1) Include: Inserir Stakeholder(s). 2) Include: Configurações. 3) O Caso de Uso começa quando o Gerente de Projetos acessa o painel Plano de Comunicação. 4) O Gerente de Projetos deve preencher todos os campos da tabela. 5) Após o preenchimento dos campos, a opção “Definir Plano” cria o Plano de Comunicação.
Pós-condição	Plano de Comunicação definido.

O Quadro 4 exibe o Caso de Uso **Gerar Smart Report**.

Quadro 4 - Gerar Smart Report

Gerar Smart Report (cenário principal)	
Pré-condição	-
Descrição	<ol style="list-style-type: none"> 1) O Caso de Uso começa quando o Usuário seleciona a opção “<i>Smart Report</i>”. 2) O sistema então abre uma tela com opções de Relatórios, dentre os quais podem ser gerados: Notificações por período, Comprometimento da equipe e Análise de Stakeholder(s). 3) Após a escolha do modelo, a opção “Gerar“ gera o Relatório.
Pós-condição	<i>Smart Report</i> gerado.

No Quadro 5, observa-se o Caso de Uso **Visualizar Notificações**.

Quadro 5 - Visualizar Notificações

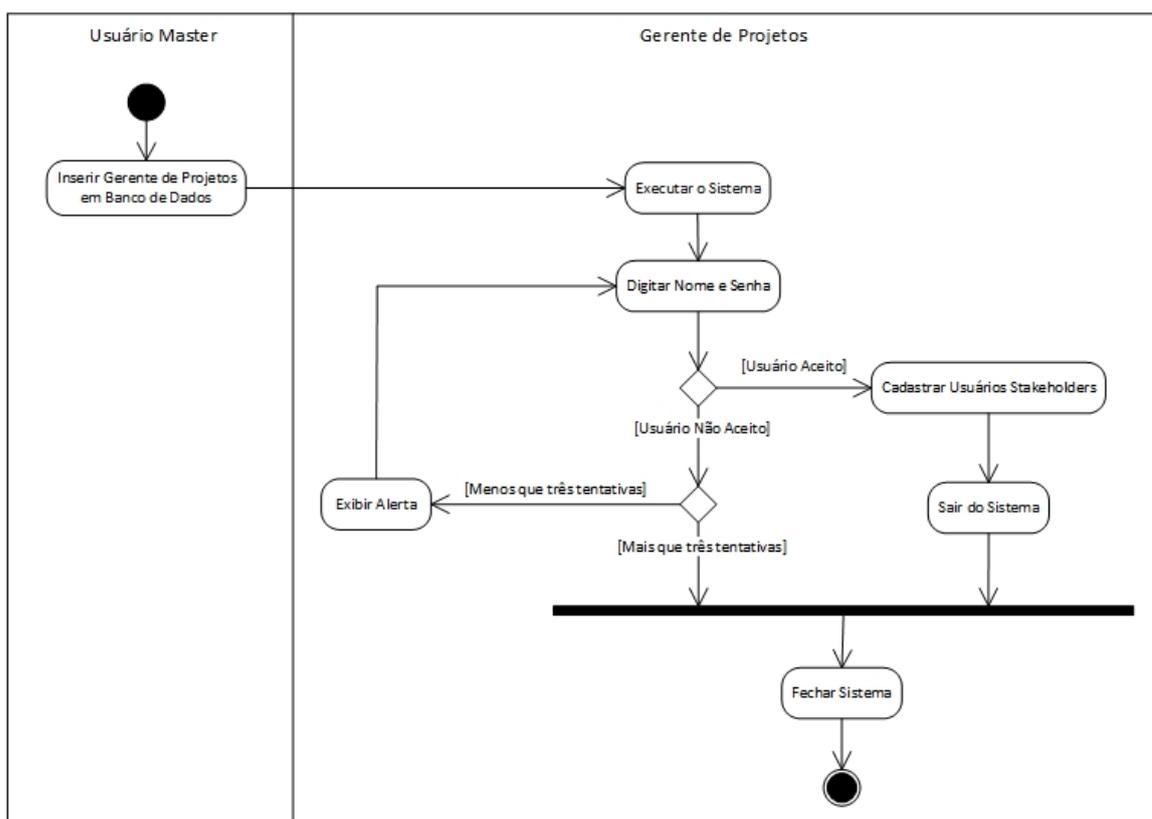
Visualizar Notificações (cenário principal)	
Pré-condição	-
Descrição	<ol style="list-style-type: none"> 1) O Caso de Uso começa quando o <i>Stakeholder</i> acessa o painel Notificações. 2) O Sistema então abre uma tela detalhando todas as comunicações feitas para este usuário. 3) A tela de notificações pode ser alterada para outros visuais através da opção Filtro: <ol style="list-style-type: none"> a) <i>Extend</i>: Filtro. 4) Para visualizar toda a mensagem enviada, o usuário deve clicar em cima do resumo da mensagem. 5) Após a abertura da mensagem completa, a opção “Confirmar Leitura“ notifica ao remetente de que esta mensagem foi lida pelo seu destinatário. <ol style="list-style-type: none"> a) <i>Extend</i>: Confirmar Leitura.
Pós-condição	Visualização de notificações.

3.2 Diagrama de Atividades

3.2.1 Adição de usuário

A Figura 2 apresenta um diagrama que demonstra o fluxo de cadastro do usuário Gerente de Projetos por meio de um usuário Master e, posteriormente, o cadastro dos usuários *Stakeholder* pelo Gerente de Projetos.

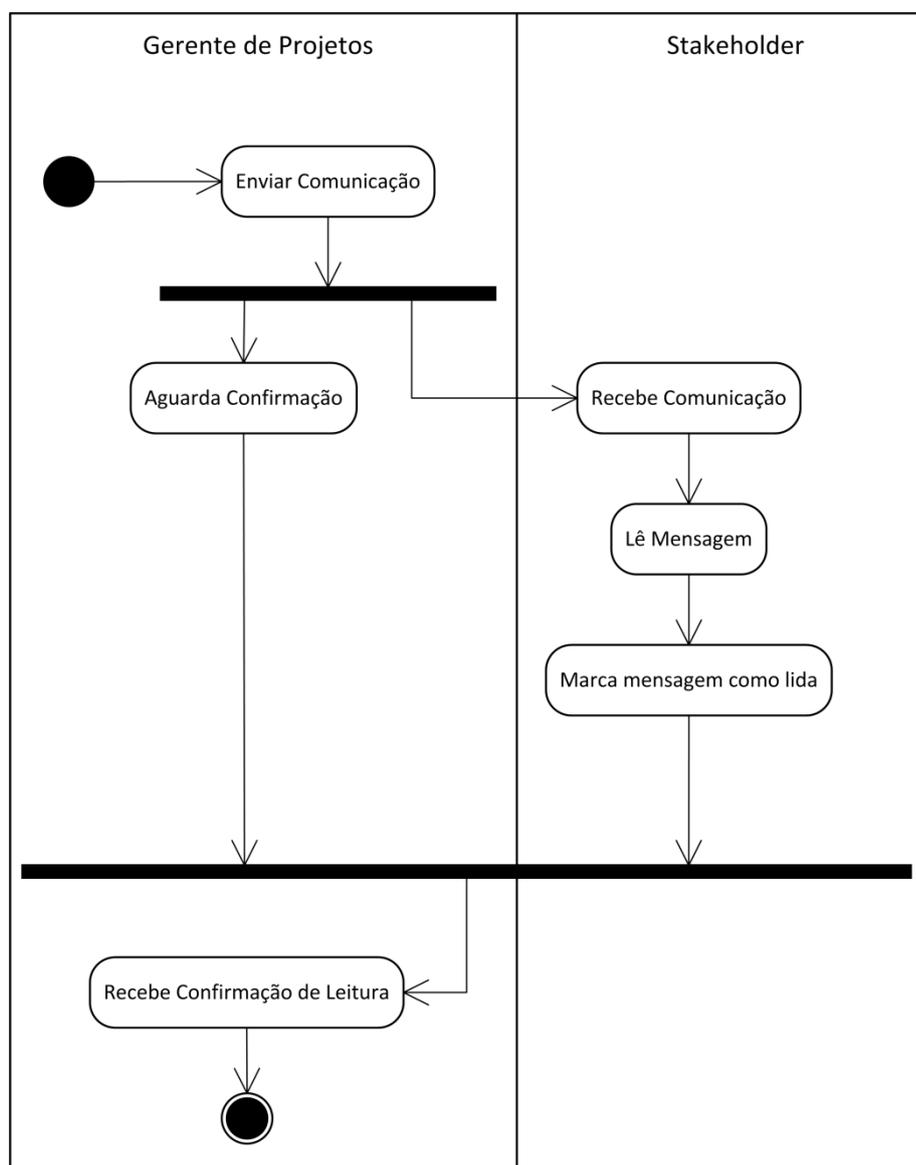
Figura 2 – Diagrama de Atividades Adição de Usuário



3.2.2 Enviar Comunicação (Gerente de Projetos para *Stakeholders*)

A Figura 3 mostra um diagrama que demonstra o fluxo de comunicação entre o Gerente de Projetos e os *Stakeholders*.

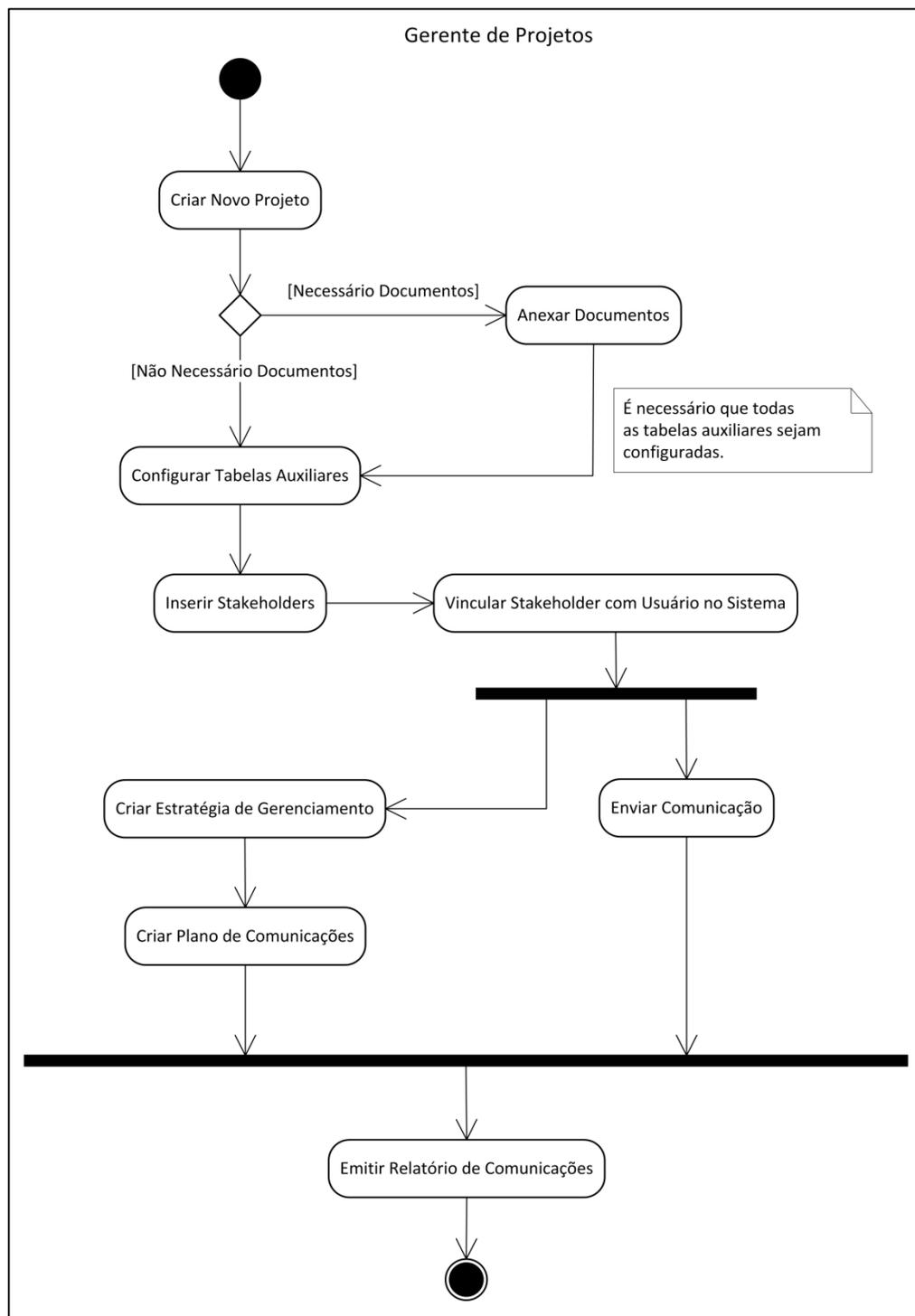
Figura 3 - Diagrama de Atividades *Enviar Comunicação (Gerente de Projetos para Stakeholders)*



3.2.3 Fluxos para o gerenciamento das comunicações

A Figura 4 demonstra os fluxos necessários para realizar o gerenciamento das comunicações do(s) projeto(s).

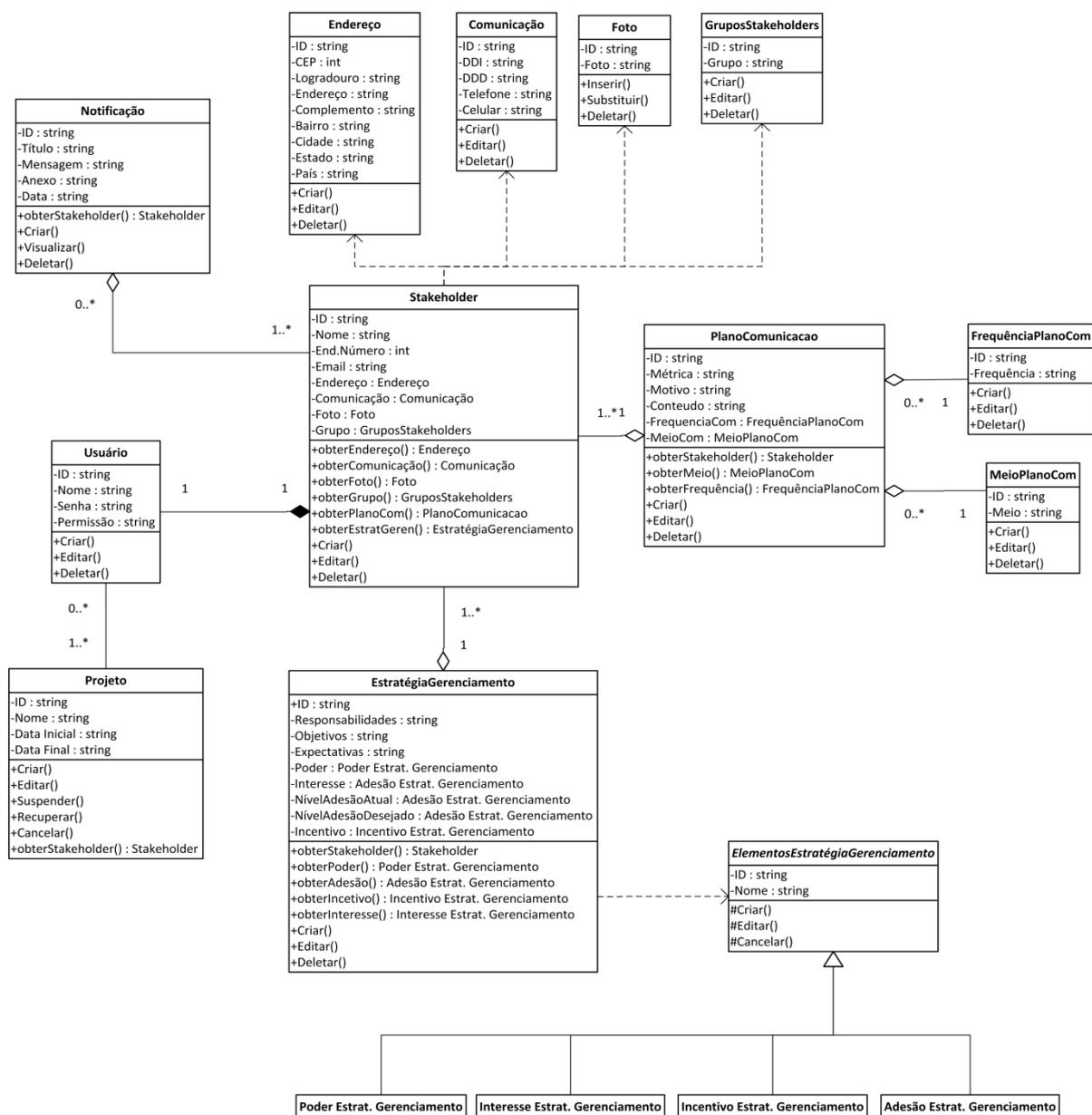
Figura 4 - Diagrama de Atividades Fluxos para o gerenciamento das comunicações



3.3 Diagrama de Classes

A Figura 5 ilustra o Diagrama de Classes do software.

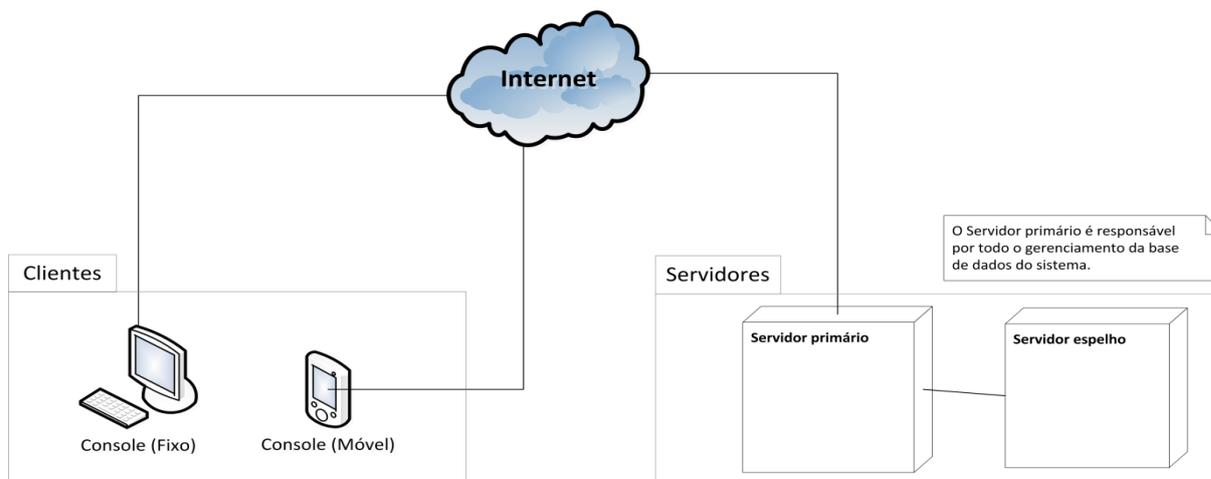
Figura 5 - Diagrama de Classes



3.4 Diagrama de Implantação

O Diagrama de Implantação, conforme ilustra a Figura 6, mostra como as partes físicas do sistema irão se comunicar e por quais meios. Neste diagrama, são apresentados dois pacotes (Clientes e Servidores), compostos pelos consoles (Fixo e Móvel) responsáveis pela interface do usuário e os servidores (Primário e Espelho) responsáveis pelo armazenamento e gerenciamento dos dados. Entre eles, o meio de comunicação (Internet) representa o caminho do tráfego dos dados.

Figura 6 - Diagrama de Implantação



4 CONCLUSÃO

A partir da elaboração dos diagramas da Linguagem de Modelagem Unificada (UML), cumprindo-se parte da modelagem visual de um programa computacional para o gerenciamento das comunicações em gestão de projetos, pode-se concluir que as notações estabelecidas pela UML são de fácil aprendizado e compreensão para uma equipe previamente capacitada em desenvolvimento de softwares. Também se pode observar que através do esforço aplicado na elaboração da modelagem, a ferramenta de modelagem visual apresentou grande importância nos modelos de processos de software, já que por ela foi possível entender as funcionalidades que um software possui ou deve possuir. Além de eliminar, durante a construção dos diagramas, possíveis inconsistências ou funcionalidades desnecessárias que seriam criadas pela equipe de desenvolvimento caso não houvesse tais especificações por meio da modelagem visual.

REFERÊNCIAS

BOOCK, G.; RUMBAUGH, J.; JACOBSON, I. **UML, guia do usuário**. Rio de Janeiro: Editora Campus Ltda., 1999.

GREENCE, J.; STELLMAN, A. **Use a Cabeça!** 2. ed. Rio de Janeiro: Starlin Alta Con. Com. Ltda., 2010.

HIRAMA, K. **Engenharia de Software: Qualidade e Produtividade com Tecnologia**. Rio de Janeiro: Elsevier Editora Ltda, 2011.

NUNES, M.; O'NEILL, H. **Fundamental de UML**. 7. ed. Lisboa: Editora Fca, 2000.

PROJECT MANAGEMENT INSTITUTE. (Estados Unidos da América). **Um guia do conhecimento em gerenciamento de projetos**. 4. ed. Newtown Square: Project Management Institute, 2008.

SOMMERVILLE, I. **Software Engineering**. 9. ed. [s.l]: Pearson Education Inc, 2011.