Relatório

Kevin Jonas Teixeira Santos
Mateus Fellipe Alves Lopes

4 de setembro de 2015

Sumário

1 Introdução

Este relatório visa apresentar os resultados e conclusão obtida do primeiro trabalho pratico da disciplina Programação Orientado a objetos. Neste trabalho foi desenvolvido o conceito de análise e projeto orientados a objetos utilizando a UML.

1.1 UML- Unified Modeling Languege

O UML surgiu quando muitos usuários de métodos Orientados a Objeto tiveram problemas para encontrar satisfação completa em qualquer linguagem de modelagem. Com o objetivo de unir os métodos Booch e OMT (Object Modeling Technique), Grandy Booch e Jim Rumbaugh começaram a trabalhar na unificação dos métodos na qual resultou o UML.

A UML nos proporciona todos os ícones de desenho necessário para capturar a maioria dos conceitos ou mecanismos que considerarmos valiosos para a resolução dos problemas reais de negócios.

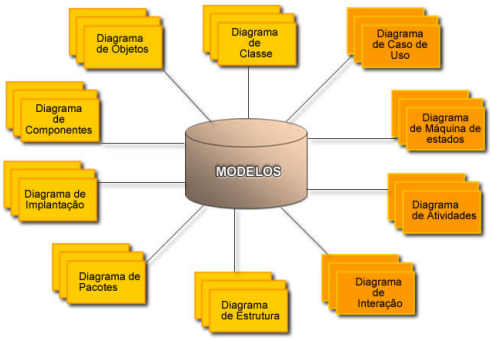

Figura 1: Diagrama de uso de caso

1.2 Vantagens de utilizar UML

Representação visual;
Legibilidade e usabilidade;
Norma;
Ferramenta de planejamento;

2 Desenvolvimento

Analise de projeto orientados objetos utilizando a UML da empresa EmTempo S.A.

2.0.1 Diagrama de Casos de Uso

O diagrama de casos de uso descreve o cenário que mostra as funcionalidades do ponto de vista do usuário do sistema. O diagrama de casos de uso é representando por atores e casos de uso que o sistema pode passar e os relacionamentos entre eles.

2.1 Análise do Sistema da empresa *EmTempo S.A.*

O sistema tem a ocorrência de dois atores, o administrador e funcionário, ambos tem pré-requisito para acessar o sistema fazendo login no sistema, neste utilizamos o método "include" onde o caso de uso base incorpora implicitamente o seu comportamento num local especificado indiretamente pelo caso de uso que é usado, e então tendo acesso as possibilidades que o sistema os proporciona. O diagrama de caso de uso e representado na figura 2.

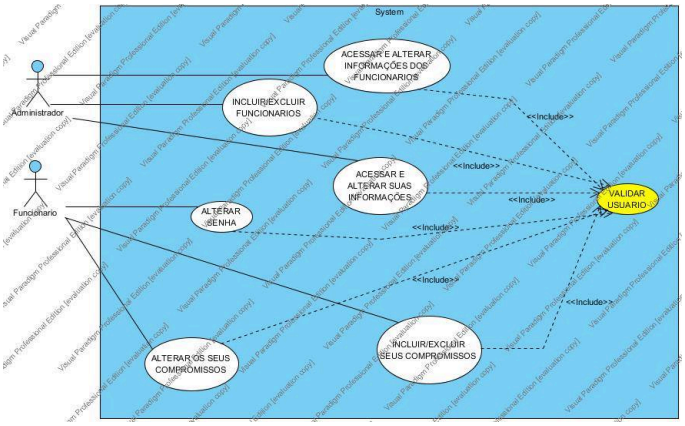

Figura 2: Diagrama de casos de uso

2.1.1 Fluxo de Eventos

É uma descrição de um processo em texto formal que descreve a sequência de ações que representam um cenário principal e cenários alternativos com o

objetivo de demonstrar o comportamento de um sistema através de interações de atores.

2.1.2 Fluxo de eventos a empresa EmTempo S.A

Para incluir um novo funcionário, funcionalidade pertencente ao administrador, é apresentado o seguinte fluxo de eventos:

Ator: Administrador

Pré-condição: O administrador deve ter feito "login" e obtido autorização do sistema.

1. O caso de uso começa quando o administrador seleciona "incluir novo funcionário".

2. O usuário fornece nome, matricula e senha do novo funcionário.

  1. Enquanto administrador não fornece uma senha valida.

  2. Sistema pede nova senha.

3. Sistema confirma adesão do novo funcionário.

Para Alterar a senha de um administrador, é apresentado o seguinte fluxo de eventos:

Ator: Administrador

Pré-condição: O administrador deve ter feito "login" e obtido autorização do sistema.

1. O caso de uso começa quando o administrador seleciona "Alterar senha".

2. O usuário fornece atual senha e nova senha

3. Enquanto administrador não fornecer senha valida.

  1. Sistema pede nova senha.

4. Sistema confirma alteração de senha.

Para incluir um novo compromisso, é apresentado o seguinte fluxo de eventos:

Ator: Funcionário

Pré-condição: O Funcionário deve ter feito "login" e obtido autorização do sistema.

1. O caso de uso começa quando o administrador seleciona "Incluir novo compromisso".

2. Usuário fornece para o sistema as características do compromisso.

  1. Sistema verifica disponibilidade desse compromisso.

  2. Enquanto não houver disponibilidade no horário sistema não registra o compromisso.

3. Sistema confirma o registro do novo compromisso.
Para incluir editar um compromisso, é apresentado o seguinte fluxo de eventos:
Ator: Funcionário
Pré-condição: O Funcionário deve ter feito "login" e obtido autorização do sistema.
1. O caso de uso começa quando o usuário seleciona "Editar compromisso".
2. O usuário fornece as novas características do compromisso.
1. Sistema verifica disponibilidade desse compromisso.
2. Enquanto não houver disponibilidade no horário sistema não registra o compromisso.
3. Sistema confirma alteração de compromisso
Para pesquisar um compromisso, é apresentado o seguinte fluxo de eventos:
Ator: Funcionário
Pré-condição: O Funcionário deve ter feito "login" e obtido autorização do sistema.
1. O caso de uso começa quando o usuário seleciona "pesquisar compromisso".
2. O usuário fornece as novas características do compromisso a ser encontrado .
1. Sistema verifica compromissos na data e horário fornecidos.
3. Sistema demonstra resultado.

2.1.3 Diagrama de sequências

Demonstra os aspectos dinâmicos de um sistema e dá ênfase à ordenação temporal das mensagens trocadas entre objetos deste sistema. Os diagramas de sequências apresentados abaixo foram baseados no fluxo de eventos descritos na questão anterior.

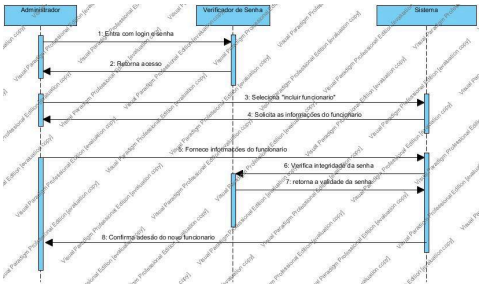

Figura 3: Diagrama de sequências

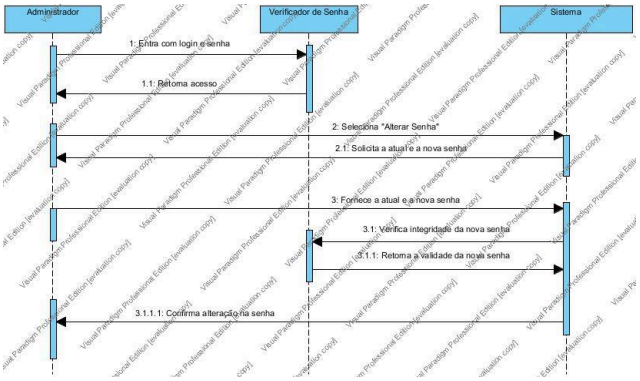

Figura 4: Diagrama de sequências

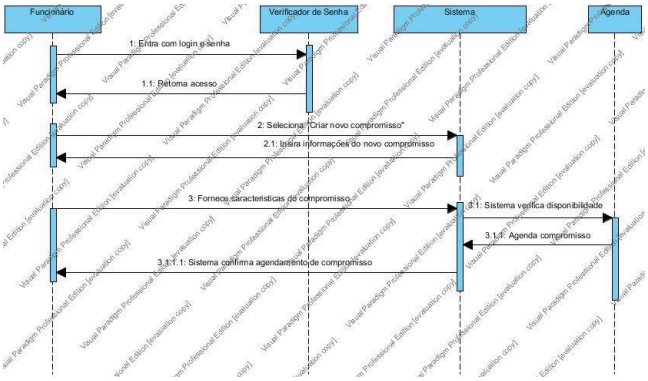

Figura 5: Diagrama de sequências

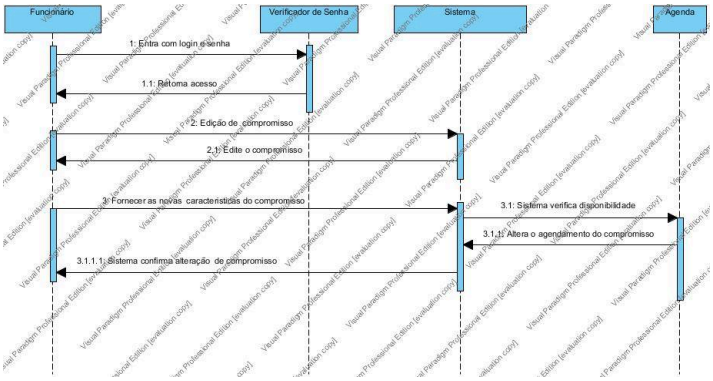

Figura 6: Diagrama de sequências

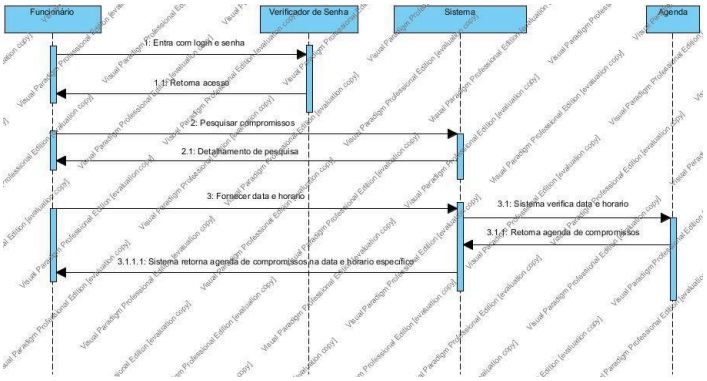

Figura 7: Diagrama de sequências

2.1.4 Diagrama de Classes

O diagrama de classes é basicamente a composição por um conjunto de classe se relacionando entre si. Uma classe representa o estado e comportamento de um Objeto implementando métodos e atributos, elas podem ser relacionadas em diversas maneira:

- Associação;
- Dependência;
- Especialização;
- Pacotes;

Inicialmente modelamos o problema e descobrimos que 5(cinco) classes seriam criadas no nosso diagrama para representarmos o sistema, foram geradas então as classes Administrador, Funcionário, Compromisso, Descrição e Usuário. A classe Usuário foi criada no intuído de que os atributos Nome, Matricula e Senha não fosse necessariamente colocadas repetidas nas classes Administrador e Funcionário, ficando assim as classes Administrador e Funcionário somente com as operações essenciais de cada uma. Foram inseridas também as associações entre Administrador e Funcionário, aonde um administrador pode incluir vários funcionários, entre Funcionário e Compromisso, em que um funcionário pode gerenciar vários compromissos, Compromisso e Descrição, uma descrição para um compromisso.

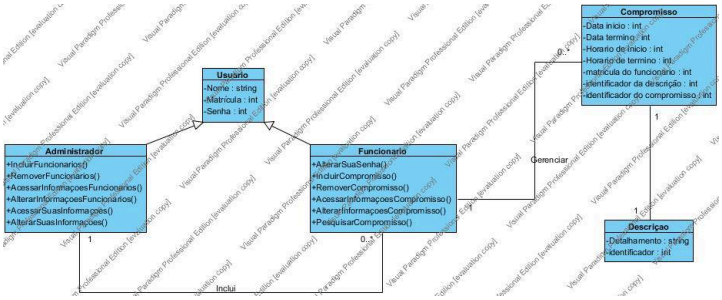

Figura 8: Diagrama de Classes

2.1.5 Diagrama de Estados

Este diagrama é usado para mostrar todos os estados possíveis que objetos de uma certa classe podem se encontra. Ele mostra os estados que um objeto pode possuir e como os eventos afetam estes estados ao passar do tempo. As transições em um diagrama de estados podem ser de dois tipos:

1. Sequencias simples: Uma linha que liga dois estados e representa o fluxo de controle entre eles.

2.Ramificações: A ramificação é representada por um losango indicando que o fluxo pode tomar caminhos alternativos mediante alguma expressão.

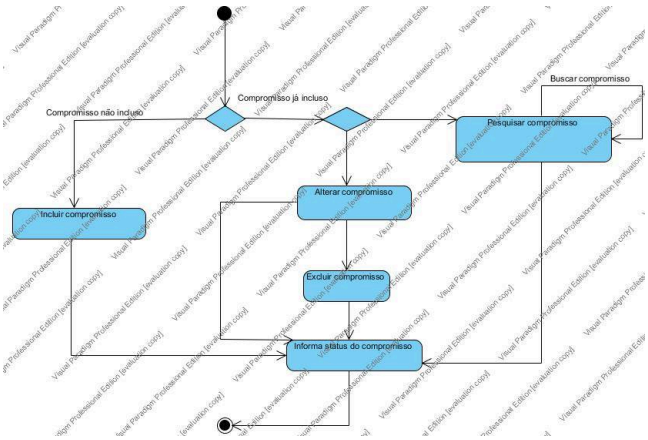

Figura 9: Diagrama de Estados

2.2 Modelagem de sistema de um Computador

- Placa Mãe:

Componente que caracteriza o computador a placa mãe tem a característica de interligar os componentes do computador bem como o processador, memória RAM e BIOS (Basic Input/Output System).

1. Processador: Item responsável pelo processamento de informações do computador.

2. Memória RAM: Memória volátil, responsável pela leitura dos conteúdos quando requeridos.
3. BIOS: Responsável pela inicialização do computador, checando os componentes, e inicializando o sistema operacional.

- Gabinete:

Estrutura que é montada os componentes e aonde estar contido a placa mãe.
1. HD: Gerencia dados de forma que as informações não serão perdidas após o computador ser desligado.
2. Placa de Rede: Dispositivo capas de conectar à rede, enviando dados e recebendo dados.
3. Fonte: Responsável por alimentar a placa mãe.
4. Unidade de Drive Óptico: Leitura e gravação de dispositivos ópticos.
5. Placa de Vídeo: Responsável pelo processamento de dados para o monitor.

- Interação Externa:

1. Monitor: Dispositivo visual que apresenta ao usuário os dados em forma de imagem.
2. Teclado: Dispositivo mecânico que possibilita a interação por meio das teclas com o sistemas.
3. Mouse: Dispositivo que interage com sistema através do espelhamento do movimento
Segue figura 10 da modelagem de um sistema de computador, organizado por subsistemas.

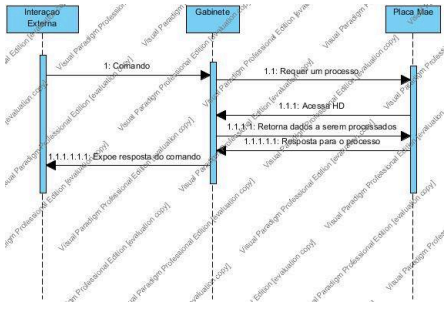

Figura 10: Diagrama de sequências

3 Conclusão

Concluímos a partir deste trabalho que a modelagem orientada objeto através UML é um meio bastante viável para a abstração do problema, gerando varias formas de gerir o problema. Tivemos dificuldades quanto a adaptação ao software Visual Paradigm, sendo este completo para o desempenho do desenvolvimento das modelagens, dificuldades sobre conceitos, consultando livro e sites, assim o trabalho nos proporcionou um cenário na qual compusemos a pesquisar conceitos a fundo de sua implementação. Ao curso este problema é de fundamental importância ao apresentar os conceitos e modelagem de problema, o que seria uma prevê introdução ao que é proposto a disciplina de engenharia de software, e nos demonstra um novo olhar na organização e demonstração de soluções de problemas.

4 Bibliografia

(a) Estivemos em discussão com a dupla de Jair Gomes e Davidson Dias do 4º período de Engenharia de Sistemas, sobre os conceitos e usos do UML e do LaTex. Assim também sanamos duvidas com Juliana Miranda do 8 periodo de engenharia de sistemas.

(b) $http://www.ufpa.br/cdesouza/teaching/cedai/APOOUMLP.pdf$

(c) $http://www.macoratti.net/11/10/uml_rev1.htm$

(d) $www.cs.pomona.edu/\ markk/cs121.f07/supp/UMLtutorial/history_of_uml.htm$

(e) $Livro: UmleC{+}{+}GuiaPraticodeDesenvolvimento, Lee, RichardC.$