

Universidade do Minho
Conselho de Cursos de Engenharia
Licenciatura em Engenharia Informática

Desenvolvimento de Sistemas de Software

Ano Lectivo de 2009/10



escola de engenharia



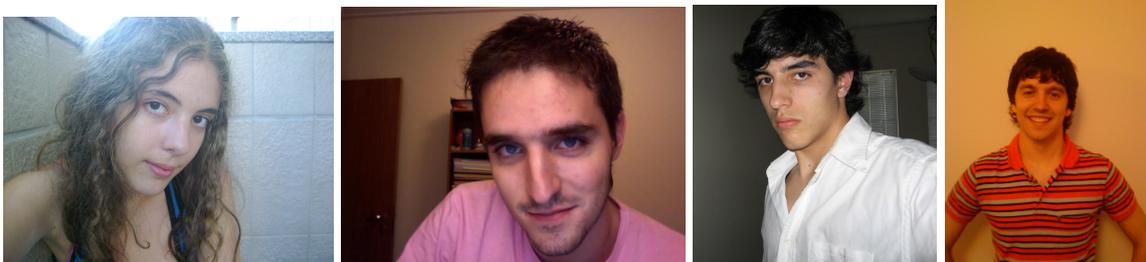
departamento de
informática

GereComSaber

Ana Duarte, André Guedes, Eduardo Lopes, Jorge Pinheiro

Dezembro, 2009

Data de Recepção	
Responsável	
Avaliação	
Observações	



GereComSaber - Serviços a Condomínios

Grupo 27

Ana Filipa Sampaio Calçada Duarte nº51806

André Filipe Macedo Guedes nº47127

Eduardo Luís Silva Lopes nº51753

Jorge Miguel Oliveira Pinheiro nº51780

Dezembro, 2009

Conteúdo

Conteúdo	i
1 Descrição	1
2 Introdução	2
3 Modelação do Problema	3
3.1 Modelo de Domínio	3
3.2 Diagrama de Use-Cases	5
4 Conclusão	7
Bibliografia	8

1 Descrição

Com base num problema real, pretende-se modelar e organizar todos os requisitos de um sistema de gestão de condomínios, recorrendo à ferramenta *Visual Paradigm* e à linguagem UML.

A partir de um conjunto de dados fornecidos, que incluem os vários modos de gestão pretendidos e o funcionamento da empresa *GereComSaber*, tenciona-se antever os possíveis cenários de utilização, preparando o programa para o suporte de vários fornecedores (com diferentes ofertas), a gestão dos serviços contratados e possíveis alterações, e o cálculo de todos os fluxos financeiros, incluindo o saldo da empresa.

Após algumas tentativas falhadas e identificação final de todos os dados necessários à resolução do problema, chegámos a um modelo de domínio e diagrama de *use-cases* adequados aos requisitos do software.

Palavras-chave: Modelação; UML; *Use-Cases*; modelo de domínio; *Visual Paradigm*; desenvolvimento de software

2 Introdução

O trabalho que de seguida é explanado, vem de encontro aos objectivos propostos na disciplina, Desenvolvimento de Sistemas de Software (DSS), do 1 ° semestre do 3 ° ano do curso de Engenharia Informática da Universidade do Minho.

DSS é uma unidade curricular que pretende preparar os alunos para o desenvolvimento de *software* segundo os princípios da *Engenharia de Software*, tendo sempre como pontos guia a qualidade, a eficiência e sobretudo a modelação do paradigma da programação orientada ao objecto.

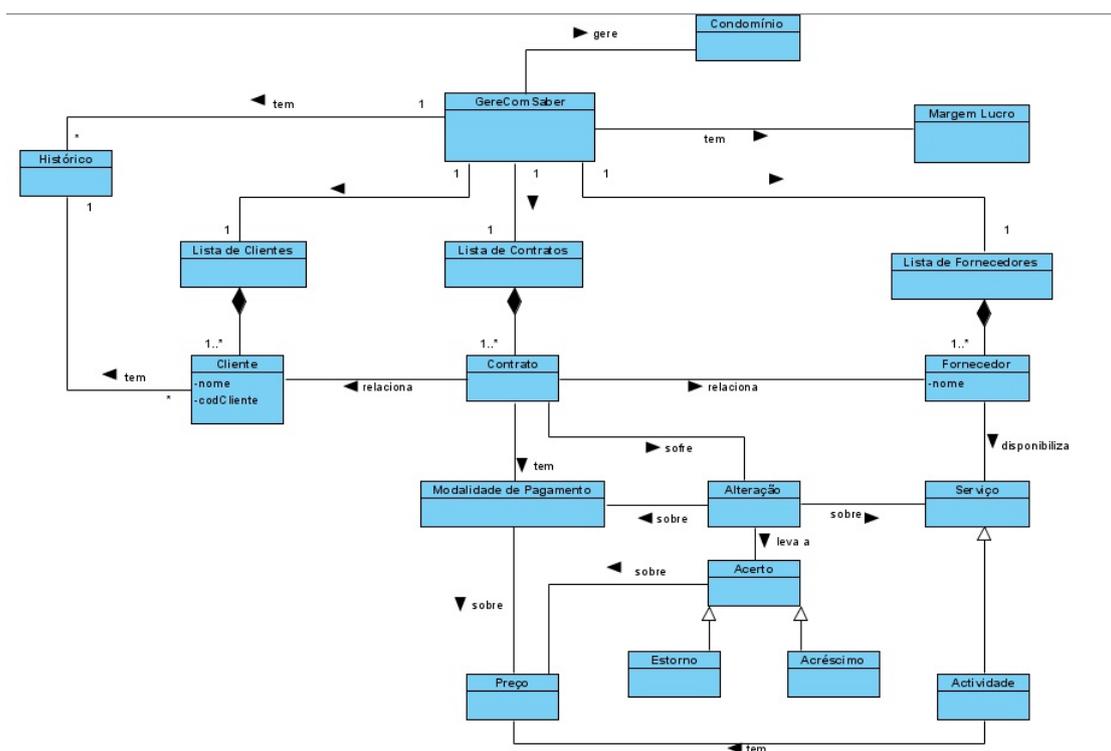
Este projecto foi desenvolvido com o objectivo de servir uma agência mediadora, que possui uma carteira de fornecedores que prestam serviços a condomínios.

Depois de uma breve descrição do problema, são apresentados o modelo de domínio e o diagrama de *use-cases*, bem como a justificação das opções tomadas para a resolução do problema proposto.

3 Modelação do Problema

3.1 Modelo de Domínio

Para a construção do modelo de domínio, foi feita a identificação prévia de todos os termos utilizados, bem como as relações existentes entre eles. O modelo construído, que poderá sofrer alterações numa fase posterior, encontra-se no organograma seguinte.



Este modelo gira principalmente em torno do termo *GereComSaber*, pois esta será a peça mais importante do nosso futuro sistema.

No nosso domínio, *GereComSaber* gere o condomínio, possuindo várias listas, cada uma com características específicas. Entre elas, encontram-se a lista de clientes, de con-

tratos e de fornecedores.

A primeira contém toda a informação de todos os clientes residentes no condomínio (nome, código, etc). A segunda guarda os dados de todos os contratos que relacionam clientes com fornecedores e seus serviços, bem como e preços e modos de pagamento. Estes contratos podem sofrer alterações, o que poderá levar a pequenos acertos monetários com o cliente, dando origem a acréscimos ou estornos no preço contratado. A última diz respeito aos fornecedores, contendo toda a informação relativa às actividades dos seus serviços.

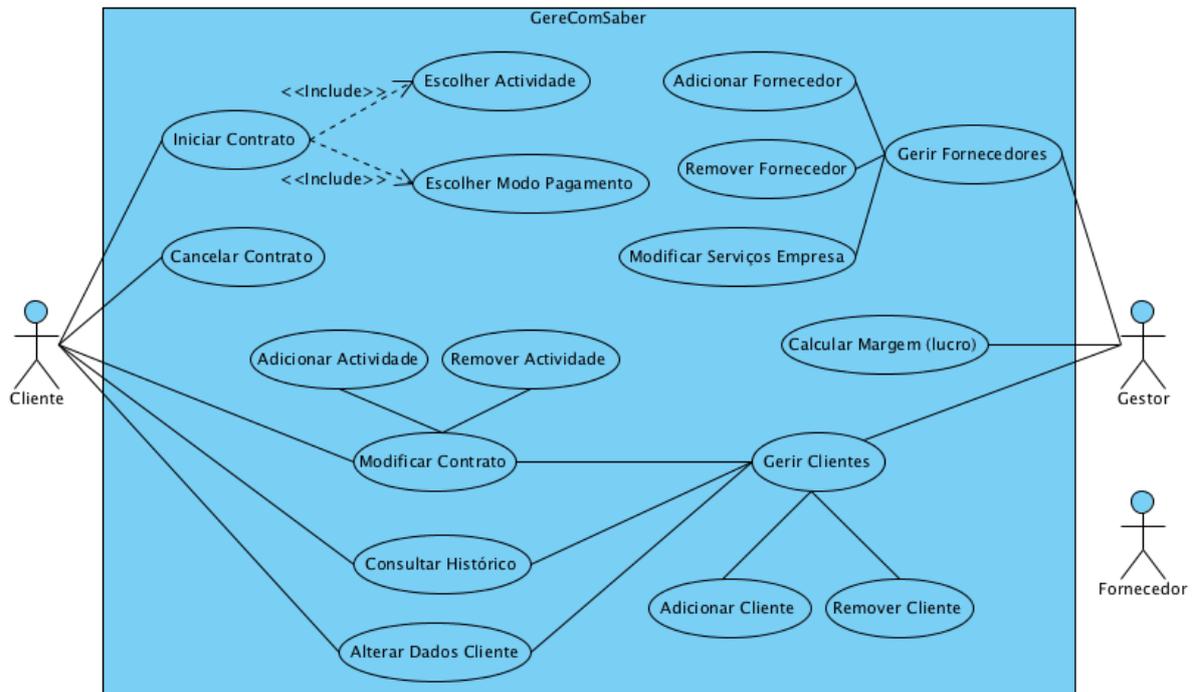
Para além das listas descritas, a GereComSaber também está directamente relacionada com um Histórico para cada cliente, ou seja, existe um ficheiro por cada sócio onde são registadas todas as alterações até ao momento.

Foi criado também um termo designado "Margem Lucro", que permite à GereComSaber calcular a qualquer momento o seu saldo de negócio.

Assim, em traços largos, será este o modelo a seguir nas próximas fases.

3.2 Diagrama de Use-Cases

O diagrama de *use-case* é usado para mostrar como o sistema se comporta nas várias situações possíveis ao longo da sua utilização. É como que uma modelação do diálogo entre os actores e o sistema.



Para descrever e definir os requisitos funcionais deste sistema foram utilizados três actores e vários *use-cases*. Um destes actores não interage directamente com o sistema (Fornecedor). Visto que todo o sistema gira à volta da empresa mediadora entre os clientes e os serviços prestados pelos fornecedores, foi necessário mencionar a importância deste actor no diagrama. Para além deste actor existem outros 2 (cliente, gestor), que comunicam com o sistema através dos *use-cases*.

O actor Cliente tem ao seu dispor uma variedade de acções que pode executar no sistema:

- Iniciar contrato - o cliente é obrigado a escolher uma actividade e um método de pagamento, dando assim inicio a um contrato (único que o pode fazer);
- Cancelar contrato - o cliente cessa um contrato actualmente em vigor (único que o pode fazer);
- Modificar contrato - o cliente pode adicionar ou remover actividades contratadas;

- Consultar histórico - o cliente pode consultar, apenas, o seu histórico de contratos e de modificações;
- Alterar dados cliente - o cliente pode alterar a sua ficha de dados pessoais;

Ao actor Gestor é também possível realizar um conjunto de operações no sistema:

- Gerir fornecedores - o gestor pode adicionar ou remover fornecedores e também modificar os serviços oferecidas por um determinado fornecedor (único que o pode fazer). Quando o gestor efectua alterações nos fornecedores, essas alterações vão-se repercutir na lista de actividades disponíveis para os clientes;
- Calcular margem de lucro - o gestor pode consultar o lucro/prejuízo actual da empresa (único que o pode fazer);
- Gerir clientes - o gestor pode adicionar ou remover clientes, alterar os seus dados pessoais bem como modificar os seus contratos e consultar os seus históricos;

Numa primeira análise, todos os *use-cases* aqui apresentados correspondem às interações mais básicas que os actores, referidos anteriormente, podem ter com o sistema. Numa fase posterior, este diagrama poderá ser aperfeiçoado para mostrar mais detalhadamente todos os pormenores.

4 Conclusão

A Engenharia de Software assenta em vários conceitos chave, dos quais o levantamento de requisitos é um dos mais importantes. Neste contexto, os diagramas de *use-cases* e os modelos de domínio representam uma informação muito importante para a implementação do sistema em desenvolvimento.

O desenvolvimento de sistemas de software de grande porte é suportado por métodos de análise e projecto que modelam esse sistema de modo a fornecer para todo o grupo envolvido (cliente, analista, programador, etc) uma compreensão única do projecto.

Os organogramas que foram apresentados anteriormente, além de mostrarem as várias ações que é possível a um actor executar no programa, mostram também um conjunto de conceitos fundamentais à análise do problema.

Bibliografia

- (1) PRESSMAN, Roger S., *Software Engineering*, McGraw-HILL, 1987.