

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

Cláudio Novais, Henrique Carvalho, Nuno Coelho, Pedro Carvalho, Tiago Dias

Dezembro, 2009

Constituição do Grupo 20

Pedro Carvalho

40514



Tiago Dias

43204



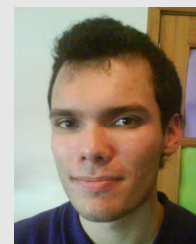
Henrique
Carvalho

49326



Cláudio Novais

49350



Nuno Coelho

49412



Projecto

GereComSaber - Gestão de Condomínios

Observações

Avaliação

Resumo

Cada vez mais a quantidade, qualidade e rapidez dos serviços prestados por uma determinada empresa são fundamentais para que se tenha sucesso nas vendas e se consiga ganhar vantagem relativamente à concorrência numa sociedade actual, cada vez mais consumista.

Pelo evoluir das tecnologias e pela crescente oferta de softwares de gestão devido à conjectura actual, torna-se necessário fazer algo inovador que corresponda na totalidade ao que o comprador deste software pretenda. Sendo assim, desenvolveu-se uma aplicação específica para gestão de condomínios capaz de controlar todas as transacções de informação de uma forma dinâmica e com o menor uso de recursos possível.

Conteúdo

Resumo	i
Conteúdo	ii
Lista de Figuras	iii
1 Introdução	1
2 Análise de Requisitos	2
3 Objectivos	4
3.1 Decisões pertinentes	5
4 Desenho do sistema	6
4.1 Modelo de domínio	6
4.2 Diagramas dos Use Cases	7
4.2.1 Diagrama do sistema	8
5 Conclusões e Trabalho Futuro	12
Bibliografia	13
Referências WWW	14

Lista de Figuras

4.1	Modelo de domínio	7
4.2	Diagrama de Sistema	7
4.3	Diagrama de gestão de Funcionários	8
4.4	Diagrama de gestão de Fornecedores	9
4.5	Diagrama de gestão de Serviços do Cliente	11
4.6	Diagrama de Consulta de Estatísticas	11

1 Introdução

No âmbito da disciplina de Desenvolvimento de Sistemas de *Software* foi-nos proposta a realização de um projecto de software relativo a uma empresa de gestão de condomínios, **GereComSaber**, que fosse capaz de assegurar os diversos trabalhos a efectuar em casa dos condóminos.

Sendo assim, a equipa responsável pela realização do projecto decidiu que o problema a resolver seria baseado software de gestão que permitisse celebrar contratos, adicionando serviços que serão prestados aos condóminos através de terceiros, bem como a gestão da facturação individual de cada cliente.

De facto, e com o surgir de novas exigências por parte da sociedade moderna, torna-se cada vez mais necessário criar um sistema capaz de automatizar todo o processo de gestão de serviços em condomínios fechados, garantindo um acesso rápido e eficaz à informação, por forma a facilitar todo o processo de prestação de serviços, além de permitir uma melhor gestão e um melhor processo de cálculo do valor a pagar por cada contrato. Deverá então, ser disponibilizada aos utilizadores deste sistema, uma plataforma simples e de fácil percepção, que permita o acesso rápido à informação, possuindo igualmente um vasto leque de funcionalidades de utilidade comprovada.

Propusemo-nos então realizar um projecto de software que permita ao utilizador aceder facilmente a esta informação, simplificando todo o processo da celebração de contratos e posterior acerto de contas.

Este relatório de acompanhamento está organizado conforme o processo de desenvolvimento de um projecto segundo o Modelo em Cascata. Assim, começa-se por abordar a Análise de Requisitos onde é feita uma primeira análise do problema proposto e selecciona-se os objectivos a cumprir. De seguida, especifica-se o Desenho do Sistema, onde é detalhado o funcionamento do *software* a criar, utilizando para isso a ferramenta de modelação **Visual Paradigm 7.1** proposta pelo docente da disciplina, mostrando as necessidades que devem ser desenvolvidas de forma a facilitar, no futuro, o trabalho dos programadores.

2 Análise de Requisitos

Ter uma boa análise de requisitos é quase obrigatório para poder esclarecer todos os objetivos a cumprir e evitar surpresas logo aquando do desenho do sistema. Assim, de forma a cumprir essa necessidade, foi feita uma pesquisa aprofundada sobre o tema deste projecto e determinaram-se os seguintes requisitos que um programa deste tipo tem em geral:

- Suporte para vários fornecedores;
- Cada fornecedor tem diferentes serviços;
- Cada serviço tem diversas ofertas;
- Pode haver fornecedores com serviços iguais;
- Calcular acréscimos e estornos;
- O estorno pode ser entregue ao cliente sobre a forma de dinheiro ou crédito;
- Cada cliente tem um histórico de compras;
- Sistema deve possibilitar vários tipos de pagamento;
- O *software* deve ter suporte para prestações e controlo de inflação e juros;
- Cliente pode subscrever várias actividades (com custos monetários individuais), de diferentes empresas ou da mesma;
- Existem comissões diferentes por cada serviço;
- Cliente pode adicionar ou retirar serviços quando quiser;
- Emissão de notificações de pagamento;
- Guardar em base de dados o valor a facturar mensalmente;
- Calcular o valor a facturar;

- Cliente pode cancelar o contrato;
- Controlo de impostos;
- Impressão de Fatura aquando do negócio e aquando dos pagamentos;
- Possibilidade de impressão de segunda via de fatura;
- Amostragem de estatísticas sobre a empresa: melhores fornecedores ao nível económico, serviços mais lucrativos, serviços mais requisitados;
- Possibilidade de descontos/promoções;
- Funcionalidades de automatização de inserção de serviços e ofertas dos fornecedores (*parsers*, por exemplo);
- Funcionalidades de exportação de dados, nomeadamente serviços pedidos pelos clientes a fornecedores, ou toda a faturação para as finanças;
- Interface adequada a diversos tipos de utilizador, funcionários, administradores, fornecedores e clientes;
- Autenticação dos utilizadores;
- Distinção entre os utilizadores, através da autenticação, restringindo os menos importantes.

3 Objectivos

Depois do grupo de trabalho ter uma ideia do que pode e deve fazer, consoante o tempo e importância de cada requisito, foi decidido que o projecto deverá seguir os objectivos seguintes:

- Desenvolver um sistema fiável e robusto que possibilite a gestão dos diferentes serviços;
- Sistema modular, desenvolvido segundo um modelo iterativo, permitindo assim actualizações fáceis;
- Controlo de utilizadores, com 2 níveis de permissões (administrador e funcionário);
- Suporte para vários fornecedores;
- Cada fornecedor suporta vários serviços;
- Cada serviço contém várias ofertas;
- Cada fornecedor tem uma comissão em concreto;
- Cliente pode efectuar o pagamento de 3 maneiras distintas (pagamento anual, semestral ou trimestral);
- Cliente pode adicionar ou retirar serviços quando quiser (sistema calcula acréscimos/estornos automaticamente);
- Consulta de estatísticas incluirá ver lucro total e lucro que determinado fornecedor dá, num determinado intervalo de tempo;
- Desenvolver um sistema que guarda a informação mensal relativa aos serviços contratados por cada cliente de modo a proceder à facturação de forma simples e automática;
- Facturação automática sem necessidade de funcionário ter de verificar a data de facturação para cada cliente;

- Cálculo de acréscimos/extornos a efectuar devido a mudanças nos serviços no período de facturação anterior.

3.1 Decisões pertinentes

Durante a realização da lista de objectivos, deparámo-nos com alguns problemas que exigiram decisões pertinentes. De seguida apresenta-se a lista dessas decisões:

- Nem o cliente nem o fornecedor interagem directamente com o sistema;
- A gestão de comissões está contida em cada fornecedor, por isso cada fornecedor tem uma comissão própria;
- Existem apenas 3 modos de pagamento: Anual, Semestral e trimestral;
- Por questões de simplificação, quando o cliente cria um contrato este será sempre aceite, isto é, o cliente não pode pedir um orçamento ou fazer uma simulação;
- Cada cliente tem um tipo de pagamento e não cada factura tem um tipo de pagamento;
- Após a primeira prestação, os pagamentos seguintes serão efectuados no início do mês correspondente;
- No caso de alteração para novos serviços, estes serão cobrados na próxima prestação. Essa prestação inclui o valor dos dias anteriores que não foram pagos e inclui o valor dos serviços até à próxima prestação;
- O administrador é o único que consegue alterar comissões, gerir funcionários e fornecedores;
- O Administrador é o único que pode ver as estatísticas;
- O Cliente quando cancela um contrato recebe logo o extorno, não tendo assim que esperar pela próxima factura para o receber, como acontece quando cancela serviços mas não o contrato;
- A facturação é diária, ou seja, se utilizou um determinado serviço durante um período, será contabilizado cada dia em termos de facturação;
- É assumido que os clientes pagam sempre as facturas no prazo e que nunca ficam a dever dinheiro à GereComSaber.

4 Desenho do sistema

Este capítulo consiste na descrição pormenorizada daquilo que foi realizado em cada uma das diferentes partes aquando da sua especificação no Visual Paradigm. Será feita uma breve análise de cada diagrama, tentando explicitar sucintamente as decisões mais importantes que foram tomadas para a realização do trabalho final.

4.1 Modelo de domínio

O Modelo de domínio é uma representação estrutural dos vários elementos que irão compor o futuro sistema permitindo aos programadores ter uma visão conceptual preliminar deste. Assim, e através deste modelo, será facilitado o trabalho de descrever o comportamento do sistema pelos *Use Cases*. Posto isto, e após a análise extensiva do enunciado do projecto, o grupo de trabalho chegou ao desenho da figura 4.1 para o seu Modelo de domínio.

Como se pode verificar pelos relacionamentos das várias entidades, é possível percorrer um caminho começando sempre pela entidade GereComSaber. Esta entidade está presente no modelo do domínio principalmente com esse objectivo, possibilitar um caminho para uma melhor compreensão do futuro sistema.

Apesar do administrador ter todos os “poderes” de gestão, neste modelo não lhe adicionamos uma relação de criação de contratos, no entanto, também poderá fazê-lo. Para além dele, o funcionário é a única identidade que pode fazer contratos, que são um relacionamento entre um cliente e várias ofertas.

Como na realidade, legalmente só se pode facturar aquando do pagamento do cliente, as facturas só são geradas quando é feito um pagamento.

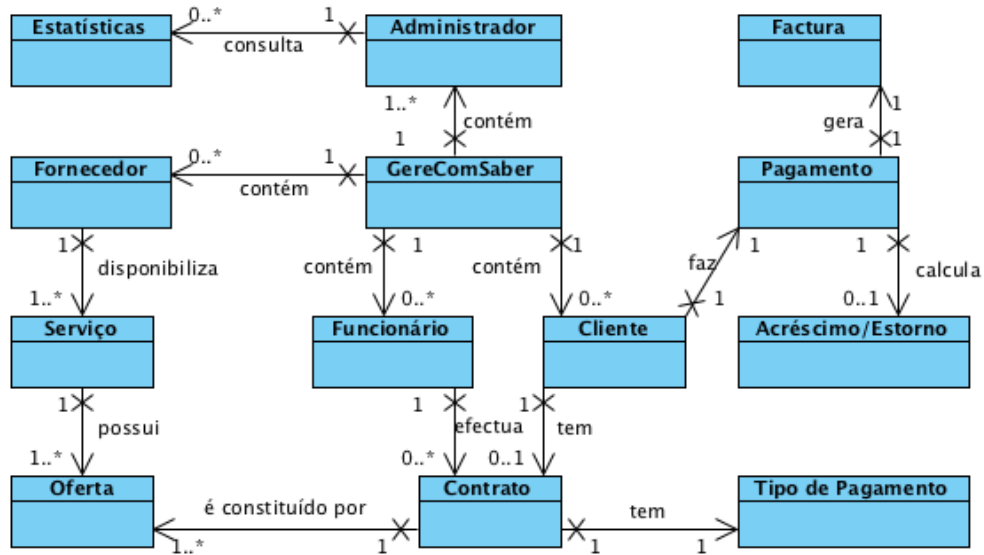


Figura 4.1: Modelo de domínio

4.2 Diagramas dos Use Cases

Tal como foi dito anteriormente, os Diagramas de *Use Cases*, servem para descrever o comportamento que os vários intervenientes têm no sistema. Tendo por base o Modelo de Domínio anteriormente especificado, chegou-se a um Diagrama de Sistema que engloba vários sub-diagramas, actores e *Use Cases*, que são explicados a seguir.

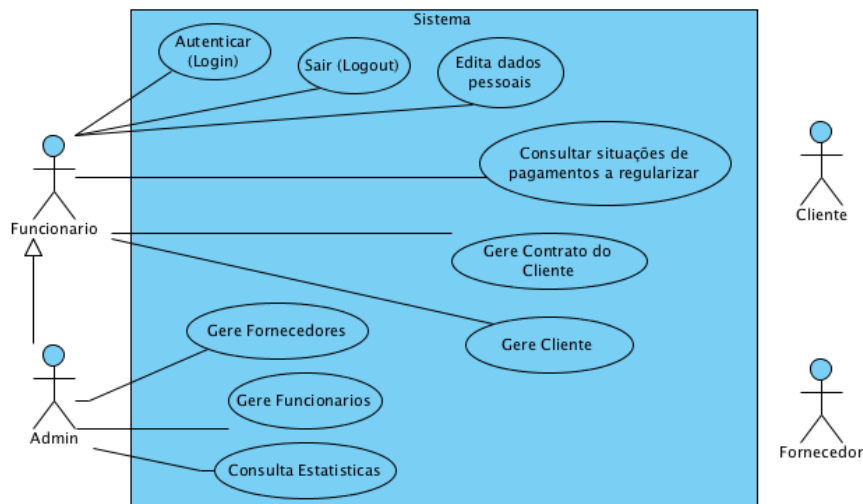


Figura 4.2: Diagrama de Sistema

4.2.1 Diagrama do sistema

Este diagrama (Figura 4.2) apresenta uma vista geral do sistema a desenvolver. Inclui todos os Use-Cases necessários para a resolução do problema proposto, desde a autenticação, gestão de Clientes/Funcionários/Fornecedores até à Gestão de Contratos do Cliente e a consulta de Estatísticas. De seguida mostramos de forma mais detalhada cada um destes componentes, tentando ser sempre o mais explícito e sucinto possíveis.

Diagrama de Gestão de Clientes e Gestão de Funcionários

Com a análise efectuada anteriormente ao problema chegou-se à conclusão de que o sistema teria de manter um registo da informação respeitante a Clientes e Funcionários. Sendo assim, foram os dois divididos nos seus respectivos sub-diagramas de Use-Cases tendo estas definidas as opções de adicionar/editar/remover os Clientes/Funcionários como se pode verificar pela figura 4.3.

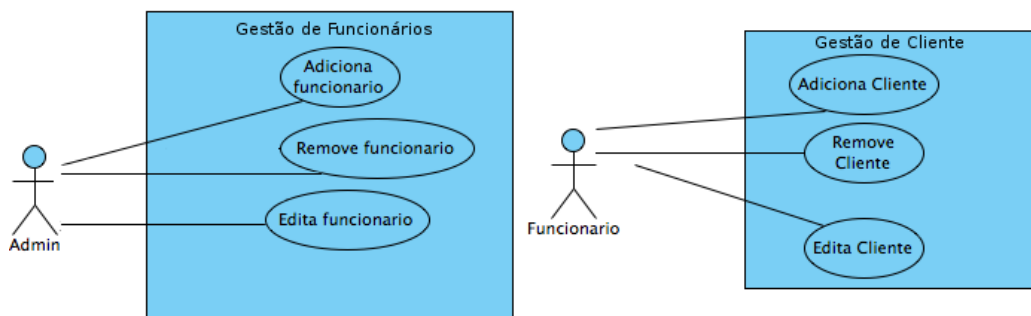


Figura 4.3: Diagrama de gestão de Funcionários

Diagrama de gestão de fornecedores

Junta-se neste diagrama (Figura 4.4) as opções básicas de adicionar/editar/remover Fornecedor com as opções de criar/editar/remover Serviços uma vez que é óbvio o seu relacionamento.

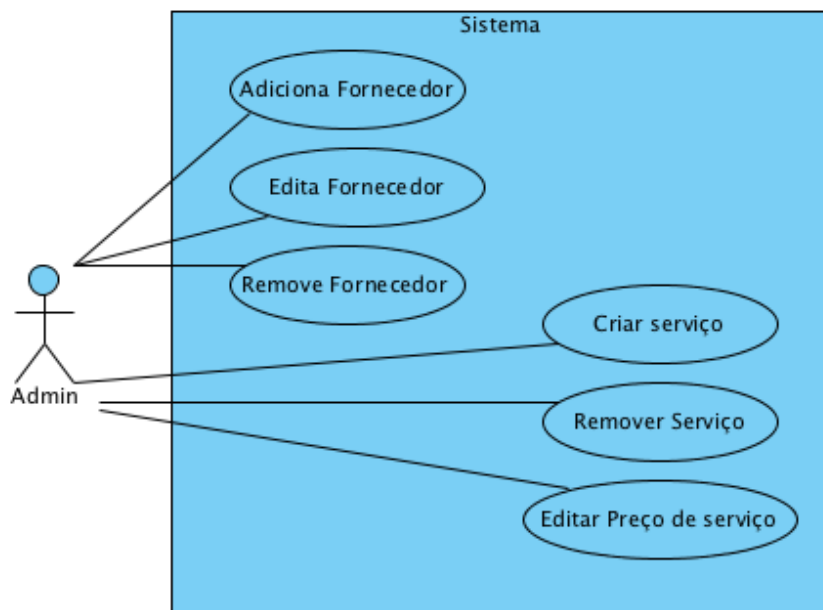


Figura 4.4: Diagrama de gestão de Fornecedores

Diagrama de gestão de serviços de cliente

Devido à complexidade deste diagrama (Figura 4.5), a explicação dos vários *Use Cases* será feita através de uma lista:

Subscreve serviço Embora este *Use Case* não seja invocado por nenhum actor, a sua existência deve-se ao facto de ser comum a outros 2 *Use Cases*: Cria Contrato e Adiciona Novo Serviço. Ele recebe os dados do serviço a subscrever e adiciona-o à ficha do cliente;

Adiciona Serviço Antes de mais, o cliente deverá existir no sistema. Depois de introduzir os dados do cliente, o funcionário escolhe os serviços a subscrever: primeiramente o tipo de serviço e depois a empresa que o fornecerá. O sistema actualiza automaticamente o acréscimo que o cliente deverá pagar;

Cria contrato Este *Use Case* começa com a introdução dos dados do cliente (que já deverá existir no sistema). Depois de validados os dados, o funcionário escolhe os serviços a subscrever: primeiramente o tipo de serviço e depois a empresa que o fornecerá. O sistema actualiza automaticamente o acréscimo que o cliente deverá pagar.

Calcula diferença factura anterior Este *Use Case* é activado quando se torna necessário calcular o valor de uma factura a pagar pelo Cliente, quer porque chegou um período de pagamento ou então porque o cliente decidiu subscrever novos serviços ou então cancelar algum dos serviços que tinha contratados. Sendo assim, e tendo por base

o valor calculado na ultima factura, é utilizado este Use-Case que recalcula o valor a cobrar.

Calcula Valor Factura Para criar uma notificação de pagamento torna-se necessário primeiramente calcular o valor a cobrar. Sendo assim e tendo como base os serviços que o cliente tem contratados no momento mais o valor dos acertos que eventualmente possam existir (créditos/extornos devido à facturação anterior), é calculado o valor que será apresentado na notificação de pagamento e que deverá ser pago pelo Cliente.

Creditar Extorno Quando a notificação de um pagamento apresenta um valor negativo, é necessário fazer um extorno ao cliente, por isso este use case tem como função a apresentação de uma mensagem ao funcionário de que terá de fazer um extorno a um determinado cliente com um determinado valor.

Notifica Pagamento Quando é necessário apresentar ao cliente uma notificação de pagamento ao cliente, quer por ele ter efectuado um contrato com a empresa, quer por se encontrar no prazo de pagamento de uma parte do serviço, ou por se encontrar no fim do contrato e ser necessário fazer ajustes de pagamentos, vai criar essa notificação com o valor a pagar.

Calcular Pagamento Mensal Neste use case , através dos serviços contratados de um determinado cliente, vai guardar ou fazer um update na base de dados do valor dos serviços para cada mês que serão facturados, desde o mês em que se encontra até ao final do contrato.

Confirma Pagamento O Funcionário terá que, após um pagamento do cliente, fazer uma confirmação manual desse mesmo pagamento. Será imprimido um talão após a confirmação sendo esta acção representada pelo use case "Imprime talão".

Emite Segunda Via de Factura O cliente pode a qualquer momento requisitar a segunda via da factura, por esse motivo o funcionário terá ao seu dispor esta simples funcionalidade. A segunda via da factura à semelhança do "Confirma Pagamento", tem como consequência a impressão de um talão.

Consulta Serviços Disponíveis Cada empresa sub-contratada pela GereComSaber contém a sua lista de serviços, este use case, descreve portanto uma forma de aceder a essa lista.

Consulta Serviços subscritos pelo Cliente Um cliente pode subscrever a um vasto número de serviços, cada cliente tem portanto um ou mais serviços a si associados. A qualquer momento o actor Funcionário poderá consultar esta lista de serviços contratados pelo cliente.

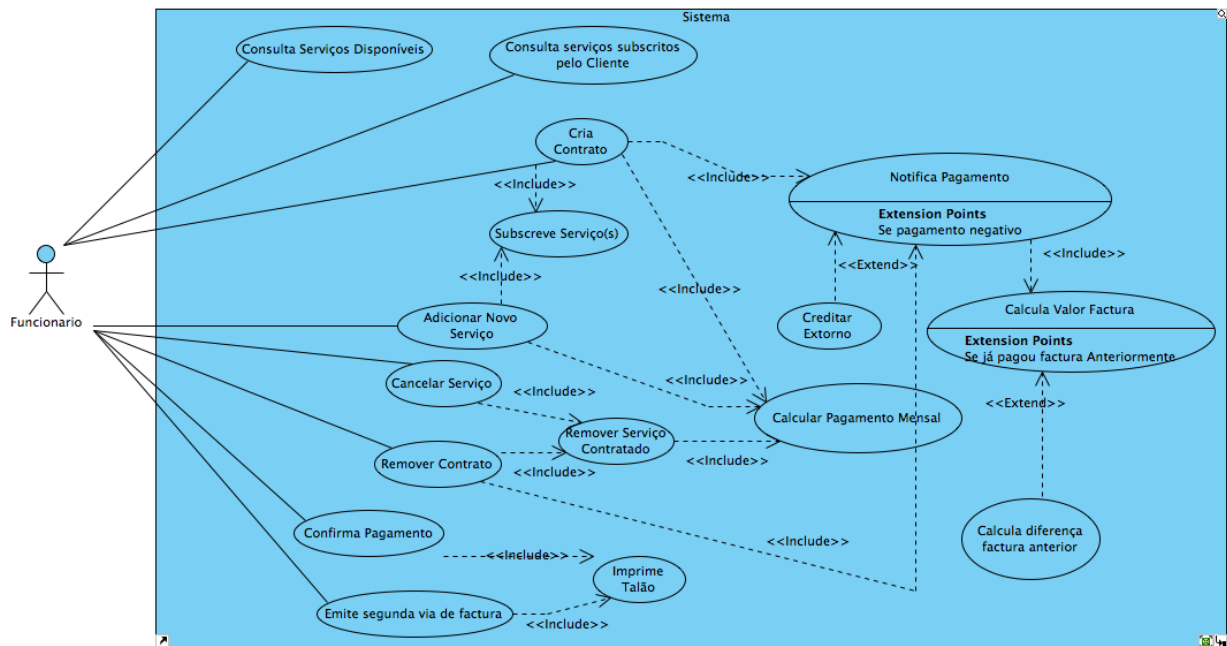


Figura 4.5: Diagrama de gestão de Serviços do Cliente

Diagrama de Consulta de Estatísticas

As estatísticas são sempre uma parte muito importante num sistema de *software* pois fornecem uma imagem clara ao seu utilizador da forma como está a correr o seu negócio. Uma vez que pode conter informações confidenciais, o seu acesso é restrito aos administradores do sistema.

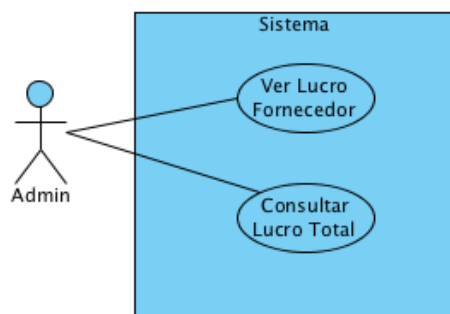


Figura 4.6: Diagrama de Consulta de Estatísticas

5 Conclusões e Trabalho Futuro

Este documento, em particular, tem como objectivo apresentar a fundamentação, análise e especificação para o projecto a realizar. Pretende-se aqui indicar e caracterizar de um ponto de vista mais geral a aplicação a ser desenvolvida, expondo as ideias e expectativas do cliente, bem como as soluções encontradas pela equipa.

Terminada a primeira fase de análise de requisitos e especificação permanece agora uma ideia semi-concreta da aplicação como um todo. Debatidas todas as funcionalidades e, após esclarecidas e contestadas todas as ambiguidades com o cliente pretende-se, por fim, avançar para a próxima fase de especificação.

Os diagramas de *use case* serão usados portanto numa próxima fase como fundamento para a concepção de diagramas de sequência, que englobam um nível de detalhe pormenorizado da interacção dos processos num espaço temporal. Os diagramas de sequência por estarem a um nível bastante próximo da programação, antecederão a fase mais prática, de implementação da aplicação.

Bibliografia

G. Booch, J. Rumbaugh, I. Jacobson. *The Unified Modeling Language User Guide*, Addison-Wesley, 1998, ISBN 0-201-57168-4

Cockburn, Alistair. *Writing Effective Use Cases*, Addison-Wesley, 2000, ISBN 978-0201702255

Referências WWW

- 01 **Condomínio fechado**(visitado em Novembro de 2009)
Site de uma empresa de condomínios fechados
- 02 **Homepage do Visual Paradigm**(visitado em Novembro de 2009)
Site do programa de modelação utilizado neste projecto
- 03 **Blog do Marco Mendes - Modelagem de Domínio**(visitado em Novembro de 2009)
Artigo do “Blog do Marco Mendes” que explica o que é o modelo de domínio,
- 04 **UML Use Case Diagrams: Tips and FAQ**(visitado em Novembro de 2009)
Página que tem as perguntas mais frequentes sobre o UML. Excelente para tirar dúvidas.