

RELATÓRIO DE ESTÁGIO

ÓSCAR MATOS MENDES

Abril/2008

RELATÓRIO FINAL PARA A OBTENÇÃO DO GRAU DE LICENCIADO
EM ENGENHARIA INFORMÁTICA

GESTÃO DE IMÓVEIS

ARRENDAMENTO-SE QUARTOS ON-LINE



ÓSCAR MATOS MENDES N.º. 6491

RELATÓRIO DE PROJECTO

Relatório Submetido para a Obtenção do Grau de Licenciado em
Eng. Informática

Orientador

Prof. Milton Pinto

Prof. Rui Pereira

Departamento de Informática

Escola Superior de Tecnologia e Gestão

Instituto Politécnico da Guarda

Abril de 2008

GESTÃO DE IMÓVEIS

ARRENDAMENTO ON-LINE

ÓSCAR MATOS MENDES



INSTITUTO POLITÉCNICO DA GUARDA

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO DA GUARDA

Orientador: Prof. Milton Pinto
Prof. Rui Pereira

Abril de 2008

Resumo

Este projecto consiste no desenvolvimento de uma aplicação informática que se baseia-se num Sistema de Gestão de Rendas com determinadas especificações. Pretende-se efectuar a informatização do processo de arrendamento dos quartos de vários imóveis.

Esta aplicação encontra-se dividida em três partes: o público em geral, que pode visualizar os quartos que estão disponíveis para arrendar e pedir informação sobre um determinado quarto, o administrador que faz toda gestão da informação necessária para o suporte da aplicação, e por fim temos os arrendatários que podem consultar os seus pagamentos e alterar os seus dados pessoais.

Cada imóvel possui serviços, uns obrigatórios que à partida estão imputados, como é o caso da luz e água que têm que estar associados a cada quarto desse imóvel, e os não obrigatórios que podem ser adquiridos pelo arrendatário como por exemplo a internet e a TV cabo entre outros.

Cada imóvel tem serviços que no final do mês são divididos pelo número de quartos que o imóvel possui arrendado. Cada imóvel suporta um serviço de limpeza que se efectua, semanalmente, mensalmente e anualmente e que é associado a uma ou mais empregadas de limpeza. Este tipo de serviço é suportado pelo proprietário do Imóvel.

Nesta aplicação, está disponível um serviço de SMS que permite ao administrador contactar os arrendatários.

Para que o processo fique completo podemos processar o recibo das contas pagas por cada arrendatário como também manter um histórico dos pagamentos.

Palavras-chave

Programação Web, ASP.Net, Visual Basic.

Abstract

This project consists on the development of an informatics' application that is based in an Administration of Incomes System with certain specifications. It's intended to computerize the process of leasing of the rooms of several immobile.

This application is divided in three parts: the public in general, that can visualize the rooms that are available to lease and to ask information on a certain room; the administrator who makes every administration of the necessary information for the support of the application; and finally we have the tenants that can consult their payments and to alter their personal data.

Each immobile possesses services, some obligatory that are imputed on departure, as it is the case of the light and water that have to be associated to each room of that immobile, and the non obligatory that can be acquired by the tenant, as for instance the internet and the TV cable among others.

Each immobile has services that are divided in the end of the month by the number of rooms leased. Each immobile supports a cleaning service that is done weekly, monthly and annually and it is associated to one or more cleaning maids. This service type is supported by the owner of the immobile.

In this application, it is available a service of SMS that allows to the administrator to contact the tenants.

To complete the process we can process the receipt of the paid bills for each tenant, as well to maintain a report of the payments.

Keywords

Web Programming, ASP.Net, Visual Basic.

Agradecimentos

Antes de iniciar o meu relatório, não podia de deixar de agradecer a todas as pessoas que me apoiaram ao longo desta etapa da minha vida de estudante e que me incentivaram a lutar por aquilo em que sempre acreditei.

Em primeiro lugar queria agradecer a Deus pelos pais maravilhosos com que me abençoou. A eles devo a minha vida e a minha educação. É graças aos meus pais e aos sacrifícios por eles feitos que consegui atingir esta meta da minha vida.

Agradeço ao meu irmão, Orlando Mendes, pelo apoio que me deu para ultrapassar todos os obstáculos á medida que estes iam surgindo.

À minha namorada com que pude contar nos bons e maus momentos, pela sua paciência nesta fase difícil.

Não posso, de maneira alguma, esquecer aos meus amigos que me acompanharam ao longo deste tempo de estudante.

Também àqueles que encontrei pela primeira vez na cidade da Guarda, a cidade dos 5 F's, que foram a minha segunda família. Com eles pude contar sempre que precisei e, assim guardarei com saudade os momentos que passámos juntos.

Um muito obrigado a todos os professores que me acompanharam ao longo do curso, em especial, aos meus dois orientadores Eng.º Milton Pinto e Eng.º Rui Pereira, por me orientarem, pela disponibilidade e pela paciência que tiveram para me atender e esclarecer qualquer duvida.

Por fim, não podia também de deixar de agradecer ao Instituto Politécnico da Guarda, à Escola Superior de Tecnologia e Gestão da Guarda, e a todos os funcionários desta instituição.

A todos, o meu muito e sincero Obrigado

Oscar Mendes

Índice Geral

RESUMO	I
ABSTRACT	II
AGRADECIMENTOS	III
ÍNDICE DE FIGURAS	VI
ÍNDICE DE TABELAS	VII
LISTA DE ACRÓNIMOS E ABREVIATURAS	IX
1. INTRODUÇÃO	10
1.1. OBJECTIVOS	10
1.2. ESTRUTURA DO RELATÓRIO.....	11
2. ANÁLISE DO SISTEMA	12
2.1. DIAGRAMA DE CASOS DE USO	12
2.1.1. <i>Descrição dos Actores</i>	15
2.1.2. <i>Descrição dos casos de uso</i>	17
2.2. DIAGRAMA DE SEQUÊNCIA	28
2.2.1. <i>Diagrama de Sequência: Registrar Arrendatário</i>	28
2.2.2. <i>Diagrama de Sequência: Efectuar Login</i>	29
2.2.3. <i>Diagrama de Sequência: Recuperar Password</i>	30
2.2.4. <i>Diagrama de Sequência: Alterar Password</i>	31
2.2.5. <i>Diagrama de Sequência: Registrar Administradores</i>	32
2.2.6. <i>Diagrama de Sequência: Inserir Empregada de Limpeza</i>	33
2.2.7. <i>Diagrama de Sequência: Inserir Imóveis</i>	33
2.2.8. <i>Diagrama de Sequência: Pagamentos dos Arrendatários</i>	34
2.2.9. <i>Diagrama de Sequência: Processa Recibo</i>	35
2.2.10. <i>Diagrama de Sequência: Pagamento das Empregadas</i>	36
2.3. DIAGRAMA DE CLASSES.....	37
2.4. DIAGRAMA ENTIDADE-RELACIONAMENTO (ER)	39
2.5. DICIONÁRIO DE DADOS	42
3. FERRAMENTAS E TECNOLOGIAS UTILIZADAS	70
3.1. SOFTWARE	70
3.1.1. <i>Sybase Power Designer 12.0</i>	70
3.1.2. <i>Microsoft Visual Studio 2005</i>	72
3.1.3. <i>Microsoft MySQL</i>	75

3.2.	LINGUAGENS DE PROGRAMAÇÃO.....	77
3.2.1.	<i>ASP.Net 2.0 com VB.Net</i>	77
4.	IMPLEMENTAÇÃO.....	78
4.1.	BASE DE DADOS.....	78
4.2.	CÓDIGO MYSQL.....	80
4.3.	SERVIÇOS DISPONÍVEIS	82
4.4.	LIGAÇÃO À BASE DE DADOS	83
4.5.	SEGURANÇA.....	84
4.6.	CONTROLOS	87
5.	CONCLUSÃO.....	89
5.1.	CONCLUSÃO.....	89
5.2.	PERSPECTIVAS FUTURAS.....	90
	BIBLIOGRAFIA.....	91
	GLOSSÁRIO.....	93

Conteúdo Protegido

Índice de Figuras

Figura 1 - Diagrama de Casos de Uso	13
Figura 2 - Actores do Gestão de Imóveis	15
Figura 3 - Diagrama de Sequência: Registrar Arrendatário	28
Figura 4 - Diagrama de Sequência: Efectuar Login	29
Figura 5 - Diagrama de Sequência: Recuperar Password	30
Figura 6 - Diagrama de Sequência: Alterar Password.....	31
Figura 7 - Diagrama de Sequência: Registrar Administradores.....	32
Figura 8 - Diagrama de Sequência: Inserir Empregada de Limpeza	33
Figura 9 - Diagrama de Sequência: Inserir Imóveis.....	33
Figura 10 - Diagrama de Sequência: Pagamentos dos Arrendatários.....	34
Figura 11 - Diagrama de Sequência: Processa Recibo.....	35
Figura 12 - Diagrama de Sequência: Pagamento das Empregadas.....	36
Figura 13 - Diagrama de Classes	38
Figura 14 - Diagrama de Entidade-Relacionamento.....	40
Figura 15 – Tipo de Relacionamento.....	41
Figura 16 – Estrutura de Base de Dados	79

Índice de Tabelas

Tabela 1: Lista de casos de uso por actor	16
Tabela 2: Descrição do Caso de Uso – Registrar Arrendatário	17
Tabela 3: Descrição do Caso de Uso – Alterar Registo	18
Tabela 4: Descrição do Caso de Uso – Recuperar <i>Password</i>	19
Tabela 5: Descrição do Caso de Uso – Ajuda:.....	20
Tabela 6: Descrição do Caso de Uso – Contactos	20
Tabela 7: Descrição do Caso de Uso – Alterar Password.....	21
Tabela 8: Descrição do Caso de Uso – Consultar Pagamentos.....	22
Tabela 9: Descrição do Caso de Uso – Criar Administrador	23
Tabela 10: Descrição do Caso de Uso – Inserir Imobiliárias	24
Tabela 11: Descrição do Caso de Uso – Efectuar Pagamentos.....	25
Tabela 12: Descrição do Caso de Uso – Pagamento dos Arrendatários.....	26
Tabela 13: Descrição do Caso de Uso – Processar Recibo	27
Tabela 14: Entidade – Arrendatários	43
Tabela 15: Entidade – Imóvel	45
Tabela 16: Entidade – Imobiliária	47
Tabela 17: Entidade – Despesas Imóvel	48
Tabela 18: Entidade – Despesas.....	49
Tabela 19: Entidade – Emp_Limpeza.....	50
Tabela 20: Entidade – Contrato_Limpeza	51
Tabela 21: Entidade – Dias_limpeza	52
Tabela 22: Entidade – Pagamento_Limpeza.....	53
Tabela 23: Entidade – Tarefas_efectuadas_limpeza.....	54
Tabela 24: Entidade – Tarefas_limpeza	55
Tabela 25: Entidade – Funcoes_limpeza	56
Tabela 26: Entidade – Servico Imovel	57
Tabela 27: Entidade – Servicos.....	58
Tabela 28: Entidade – Valor_apagar	59
Tabela 29: Entidade – Ocupacao Servicos	60
Tabela 30: Entidade – Pagamento_Serviços.....	61
Tabela 31: Entidade – Pagamento	62

Tabela 32: Entidade – Ocupacao	63
Tabela 33: Entidade – Quarto.....	65
Tabela 34: Entidade – Características.....	67
Tabela 35: Entidade – qtimovel	68
Tabela 36: Entidade – qtquarto	69

Conteúdo Protegido

Lista de Acrónimos e Abreviaturas

ASP	<i>Active Server Pages</i>
BD	<i>Base de Dados</i>
ER	<i>Entidade – Relacionamento</i>
ODBC	<i>Open DataBase Connectivity</i>
HTML	<i>HyperText Markup Language</i>
UML	<i>Unified Modeling Language</i>
VS	<i>Visual Studio</i>
XML	<i>Extensible Markup Language</i>
URL	<i>Uniform Resource Locator</i>

Conteúdo Protegido

Capítulo 1

1. Introdução

Nos últimos anos tem-se verificado um crescimento galopante da Internet como ferramenta essencial à comunicação e partilha de informação. A utilização da Internet entrou na vida quotidiana de uma forma fácil, para obter a mais variada informação, no intuito de assegurar um serviço mais rápido, completo e transparente para os seus utilizadores, permitindo um feedback imediato.

Comprar, vender e alugar, acções que outrora implicavam a deslocação das pessoas, hoje, com os avanços tecnológicos torna-se desnecessário, pois a grande parte dos serviços encontram-se disponíveis a partir de casa, com tendência a aumentar cada vez mais. Foi com este intuito que surgiu o projecto, **Gestão de Imóveis**, que permite consultar e arrendar quartos on-line, contribuindo desta forma para o desenvolvimento deste tipo de sistemas.

O serviço de “**Gestão de Imóveis**”, tem como finalidade dar a conhecer os mais variados tipos de imóveis e/ou quartos de um determinado imóvel que estão por alugar, como também gerir as rendas e despesas desses mesmos imóveis, de forma a assegurar um serviço rápido, simples e de fácil compreensão.

1.1. Objectivos

O objectivo do projecto de **Gestão de Imóveis** é desenvolver uma aplicação para a Internet, que permita efectuar uma gestão informatizada de um conjunto de imóveis.

Cada imóvel possui vários quartos, que têm uma determinada renda que pode diferir de quarto para quarto.

As despesas de um determinado imóvel serão divididas pelo número de quartos que estão arrendados nesse momento. No imóvel para além das despesas normais, tais como água e luz, existem também outro tipo de despesas que podem ser ou não divididas pelos arrendatários, tais como: internet, telefone, TV cabo, entre outros, desde

que estes usufruam destes serviços. No fim das contas pagas, poderemos processar o recibo de cada mês.

Em cada imóvel existe uma empregada de limpeza, cujas horas têm que ser registadas assim como todas as despesas de material.

Esta aplicação terá as seguintes fases essenciais:

- Análise estruturada do problema em causa;
- Criação de uma Base de Dados;
- Implementação de uma aplicação para a Internet;
- Teste da aplicação;

1.2. Estrutura do Relatório

Este relatório encontra-se organizado em cinco capítulos.

No primeiro capítulo é feita uma introdução ao tema do trabalho e à forma como o relatório se encontra organizado, no segundo capítulo, é feita toda a análise e modelação necessária para a elaboração da aplicação, no capítulo seguinte é feita uma descrição das ferramentas e tecnologias utilizadas, no quarto capítulo é feita a descrição da estrutura interna da aplicação e da sua interligação com a base de dados; no último capítulo encontram-se as conclusões retiradas após a realização do projecto e perspectivas para trabalho futuro.

Esta estrutura complementa-se ainda com a Bibliografia, Glossário e Anexos. Os Anexos contêm:

- **Manual de Utilizador** – Encontra-se em suporte digital e contém uma descrição de cada interface da aplicação.
- **Stored Procedures** – Contém o código utilizado nos Stored Procedures.

Capítulo 2

2. Análise do Sistema

Para o desenvolvimento do sistema de informação é preciso realizar a análise do sistema em questão. Nesta análise, pretende-se ver o funcionamento de um sistema sob o ponto de vista de dados desse mesmo sistema, possibilitando uma visão geral do mesmo isto é, especificar o que se requer que o sistema ou aplicação faça.

Para o tratamento destes dados foi usada a linguagem UML (*Unified Modeling Language*). Esta linguagem é utilizada para especificação, visualização e documentação de sistemas de software. O **PowerDesigner 12.0**, é uma ferramenta Case de modelação de sistemas orientados a objectos, foi o programa utilizado para a linguagem de modelação.

A análise pretende relacionar, documentar as operações do sistema, para que seja mais explícita a análise. Nesta análise foram elaborados os seguintes diagramas:

- Diagramas de Casos de Uso;
- Diagrama de Classes;
- Diagramas de Sequencia;
- Diagrama Entidade-Relacionamento (ER);

2.1. Diagrama de Casos de Uso

O diagrama de casos de uso descreve a funcionalidade proposta para o nosso sistema. Um caso de uso representa uma interacção entre um utilizador e o sistema. Cada caso de uso tem uma descrição a qual descreve a funcionalidade que irá ser construída no sistema proposto. Um caso de uso pode "incluir" outra funcionalidade de caso de uso ou "estender" outro caso de uso com seu próprio comportamento. Casos de uso são relacionados a actores.

Casos de uso especificam o comportamento do sistema ou parte dele e descrevem a funcionalidade do sistema desempenhada pelos actores. Os casos de uso são representados em forma de elipse.

A figura seguinte apresenta o diagrama de caso de uso para a **Gestão de Imóveis**. O diagrama só apresenta o login para dois casos de uso (Registar Utilizador, Alterar Dados de Registo), mas é obrigatório para todos os casos de uso à excepção do Consultar Contactos, Visualizar Ajuda e Recuperar Password.



Figura 1 - Diagrama de Casos de Uso

Os Diagramas de Casos de Uso descrevem as relações e as dependências entre um grupo de Casos de Uso e os Actores que participam no processo.

É importante reparar que os Diagramas de Casos de Uso não são adequados para representar o desenho e também não podem descrever os detalhes internos de um sistema. Os Diagramas de Casos de Uso pretendem facilitar a comunicação com os utilizadores futuros do sistema e com o cliente, e são especialmente úteis para determinar as funcionalidades necessárias que o sistema deverá ter. Os Diagramas de Casos de Uso indicam o que o sistema deverá fazer mas não devem especificar como isto deverá ser feito.

Os Casos de Uso são descrições das interações típicas entre os utilizadores de um sistema e o sistema propriamente dito. Representam a interface externa do sistema e especificam um dado tipo de requisitos sobre o que o sistema tem.

Os Casos de Uso também poderão ter relações com outros Casos de Uso.

- <<include>>, indica que um Caso de Uso toma lugar dentro de outro Caso de Uso. (Neste Caso, são todos excepto o Consultar Contactos, Visualizar Ajuda e Recuperar Password não tem o incluir, como já foi referido anteriormente)
- <<extends>>, indica que, em certas situações ou numa dada altura, um Caso de Uso será estendido por outro.
- Generalization, indica que um Caso de Uso herda as características do “Super”-Caso de Uso e poderá implementar novamente algumas delas ou adicionar novas de uma forma semelhante à da herança de classes.

As descrições dos Casos de Uso são narrativas textuais do Caso de Uso. Normalmente tomam a forma de uma nota ou de um documento que esteja associado de alguma forma ao Caso de Uso e explicam os processos ou actividades que tomam lugar no Caso de Uso.

Adaptado de: <http://docs.kde.org/stable/pt/kdesdk/umbrello/uml-elements.html>

2.1.1. Descrição dos Actores

Um actor é uma entidade externa que interage com o sistema. A figura seguinte representa os actores intervenientes no sistema.



Figura 2 - Actores do Gestão de Imóveis

No sistema temos como intervenientes os seguintes actores. Seguidamente será apresentado uma descrição de cada actor.

Público Geral – Este actor não necessita estar registado podendo aceder a algumas páginas do site. As páginas que pode visualizar, são as que estão disponíveis para quem não está registado na Gestão de Imóveis.

Arrendatários – O actor necessita de estar registado, para poder usufruir da aplicação, ou seja, após o registo este actor pode arrendar um quarto, que se encontre livre.

Administrador – O actor administrador pode efectuar todas as operações disponíveis na aplicação. É ele que controla toda a envolvente do site.

Destes três actores apresentados o actor principal é o Administrador porque é o fundamental utilizador do sistema.

Cada actor tem vários casos de uso, tal como mostra a tabela seguinte.

Tabela 1: Lista de casos de uso por actor

ACTORES	CASO DE USO
Público Geral	<ol style="list-style-type: none"> 1. Registrar Arrendatário; 2. Recuperar <i>Pass</i>; 3. Alterar Registo; 4. Ajuda; 5. Contactos;
Arrendatário	<ol style="list-style-type: none"> 6. Alterar <i>Password</i>; 7. Alterar dados Pessoais; 8. Consultar Pagamentos; 9. Consultar Serviços;
Administrador	<ol style="list-style-type: none"> 10. Cria Administradores; 11. Cria Arrendatários; 12. Inserir Imobiliária; 13. Inserir Imóveis; 14. Inserir Quartos; 15. Inserir Características; 16. Inserir Serviços nos Imóveis; 17. Inserir Características nos Imóveis; 18. Efectuar Pagamentos do Arrendatários; 19. Inserir Empregada de Limpeza; 20. Efectuar pagamento Empregada; 21. Registrar Recebimento; 22. Gerir Ocupação; 23. Gerir Serviços; 24. Gerir Tarefas efectuadas; 25. Inserir Fotos dos Quartos; 26. Processa Recibo;

Serão apenas descritos os mais relevantes.

2.1.2. Descrição dos casos de uso

Nesta secção será descrita uma descrição estruturada segundo um conjunto de passos numerados, de alguns dos casos de uso referidos no diagrama de casos de uso anterior.

Caso de Uso: Registrar Arrendatário

Descrição: Este caso de uso permite registar um novo arrendatário. Qualquer utilizador se pode registar no sistema, mas tem que aguardar a validação do registo que é feita pelo administrador do sistema.

Tabela 2: Descrição do Caso de Uso – Registrar Arrendatário

<p>Sequência de Passos</p>	<ol style="list-style-type: none"> 1. O arrendatário selecciona a opção <u>registo</u> e são mostrados os campos necessários a preencher para a criação de um novo arrendatário: <ul style="list-style-type: none"> • Dados do arrendatário, onde terá de preencher a seguinte informação: Nome, Apelido, Morada, E-mail, Data de Nascimento, Telefone, Estado Civil, Nº de Contribuinte, <i>Login</i>, <i>Password</i>, Data de Entrada, Data de Saída, Nome do Pai, Nome da Mãe, Morada dos Pais, Telefone dos Pais 2. O sistema guarda os dados do arrendatário ao seleccionar o botão <u>inserir</u>
<p>Caminhos Alternativos</p>	<p>Em qualquer um dos passos (1-2) Selecciona opção <u>Cancelar</u>.</p>
<p>Pós-Condição</p>	<p>O arrendatário visualiza mensagem “É necessário aguardar pela validação da conta. Em caso de dúvidas contacte o administrador do sistema.”</p>

Relativamente aos caso de uso inserir **empregada de limpeza**, o processo de sequências de passos é praticamente o mesmo, mudando só as tabelas e os campos.

Neste caso, tem que existir um administrador registado para se poder registar uma empregada de limpeza.

Caso de Uso: Alterar Registo

Descrição: Neste caso de uso, tanto o administrador com os arrendatários que já efectuaram os registo e a conta já lhes foi validada, podem alterar os seus dados pessoais.

Tabela 3: Descrição do Caso de Uso – Alterar Registo

Pré-Condição	Login com Sucesso
Sequência de Passos	<ol style="list-style-type: none"> 1. O utilizador escolhe a opção <u>Alterar Registo</u>; 2. São mostradas as alterações possíveis a preencher para a alteração do arrendatário: <ul style="list-style-type: none"> • Introduce as alterações (Nome, ...) 3. O arrendatário confirma as alterações feitas no registo de arrendatário seleccionando a opção <u>Alterar</u>.
Caminhos Alternativos	Em qualquer um dos passos (2-3) Selecciona opção <u>Cancelar</u> .
Pós-Condição	O arrendatário visualiza mensagem "Conta Alterada com sucesso!"

Caso de Uso: Recuperar *Password*

Descrição: Neste caso de uso, tanto o administrador com os arrendatários que já efectuaram os registo e a conta já lhes foi validada, podem recuperar a sua *password*, para isso basta colocar um email valido e aguardar que lhe seja enviado um *email* com a nova *password* para o endereço indicado. A *password* nova é gerada aleatoriamente pelo sistema.

Tabela 4: Descrição do Caso de Uso – Recuperar *Password*

Pré-Condição	O arrendatário tem que estar registado, só assim pode recuperar a <i>password</i> com sucesso.
Sequência de Passos	<ol style="list-style-type: none"> 1. O utilizador escolhe a opção <u>Recuperar Pass</u>; 2. O arrendatário coloca o <i>e-mail</i> que está associado à conta e selecciona a opção <u>enviar</u>. 3. O sistema envia <i>e-mail</i> ao arrendatário com uma nova <i>password</i>.
Caminhos Alternativos	Em qualquer um dos passos (2-3) Selecciona opção <u>Cancelar</u> .
Pós-Condição	4. O arrendatário visualiza mensagem "Uma nova <i>password</i> foi enviada com sucesso para o endereço"

Caso de Uso: Ajuda

Descrição: Este caso de uso serve para todos os utilizadores que necessitem de consultar a ajuda.

Tabela 5: Descrição do Caso de Uso – Ajuda:

Sequência de Passos	<ol style="list-style-type: none"> 1. O utilizador escolhe a opção <u>Ajuda</u> no menu; 2. Lê a ajuda_ 3. O utilizador termina ao selecciona o botão <u>voltar</u>.
----------------------------	---

Caso de Uso: Contactos;

Descrição: Este caso de uso serve para todos os utilizadores que necessitem de aceder à opção Contactos. Aqui podemos enviar um e-mail ao administrador do sistema.

Tabela 6: Descrição do Caso de Uso – Contactos

Sequência de Passos	<ol style="list-style-type: none"> 1. O utilizador escolhe a opção <u>Contactos</u> no menu; 2. São mostrados os campos a preencher pelo utilizador: <ul style="list-style-type: none"> • Terá que preencher a seguinte informação: Nome, Email, Mensagem 3. O utilizador termina ao seleccionar o botão <u>enviar</u>.
Caminhos Alternativos	Em qualquer um dos passos (2-3) Selecciona opção <u>Cancelar</u> .
Pós-Condição	4. O utilizador visualiza mensagem "A mensagem foi enviada com sucesso!"

Caso de Uso: Alterar *Password*

Descrição: Neste caso de uso, tanto o administrador com os arrendatários que já efectuaram os registo e a conta já lhes foi validada, podem alterar a sua password. Para isso devem colocar a password antiga e digitalizar a nova e a sua confirmação.

Tabela 7: Descrição do Caso de Uso – Alterar Password

Pré-Condição	Login com Sucesso
Sequência de Passos	<ol style="list-style-type: none"> 1. O arrendatário/administrador selecciona a opção <u>alterar password</u> no menu; 2. São mostrados os campos a preencher pelo arrendatário/administrador; <ul style="list-style-type: none"> • Terá que preencher a seguinte informação: Password actual, Nova Password, Confirmar Password 3. O utilizador termina ao seleccionar o botão <u>alterar Password</u>.
Caminhos Alternativos	Em qualquer um dos passos (2-3) Selecciona opção <u>Cancelar</u> .
Pós-Condição	O utilizador visualiza mensagem “Uma nova password foi enviada com sucesso para o endereço!”

Caso de Uso: Consultar Pagamentos

Descrição: Neste caso de uso, os arrendatários que já efectuaram os registo e a conta já lhes foi validada, tem acesso aos pagamentos. Neste caso o arrendatário tem que possuir já um quarto atribuído e assim já pode consultar se tem alguma conta por pagar ou se já tem as contas liquidadas.

Tabela 8: Descrição do Caso de Uso – Consultar Pagamentos

Pré-Condição	Login com Sucesso
Sequência de Passos	<ol style="list-style-type: none"> 1. O arrendatário selecciona a opção <u>Quarto</u> no menu; 2. São mostrados dois Botões: <u>Que tipos de Serviços Possui</u> e <u>Contas</u>; 3. Ao Seleccionar o Botão <u>Que tipos de Serviço Possui</u> <ul style="list-style-type: none"> • É apresentada uma lista dos serviços que Possui 4. Ao Seleccionar o Botão <u>Contas</u> São mostrados dois Botões: <u>Contas Pagas</u> e <u>Contas em Atraso</u> 5. Ao Seleccionar o botão <u>Contas Pagas</u> <ul style="list-style-type: none"> • Visualiza as contas que já foram pagas; 6. Ao Seleccionar o botão <u>Contas em Atraso</u> <ul style="list-style-type: none"> • Visualiza as contas que ainda não foram Pagas;
Caminhos Alternativos	Em qualquer um dos passos (3,4 e 5) Selecciona opção <u>Voltar</u> .

Caso de Uso: Criar Administradores

Descrição: Neste caso de uso, tem que existir um administrador registador no sistema. Só os administradores é que podem criar novos administradores para o sistema.

Tabela 9: Descrição do Caso de Uso – Criar Administrador

Pré-Condição	Login com Sucesso
Sequência de Passos	<ol style="list-style-type: none"> 1. O administrador entra com os seu dados de login e <i>password</i> 2. Selecciona a opção <u>nova conta</u> no menu são mostrados os campos necessários a preencher para a criação de um novo administrador: <ul style="list-style-type: none"> • Dados do administrador, onde terá de preencher a seguinte informação: Nome, Apelido, Morada, E-mail, Data de Nascimento, Telefone, Estado Civil, N° de Contribuinte, <i>Login</i>, <i>Password</i>, Data de Entrada, Data de Saída, Nome do Pai, Nome da Mãe, Morada dos Pais, Telefone dos Pais, 3. Activa a opção <u>Administrador</u> 4. O sistema guarda os dados administrador ao clicar no botão <u>inserir</u>
Caminhos Alternativos	Em qualquer um dos passos (2-3) Selecciona opção <u>Cancelar</u> .
Pós-Condição	O Administrador visualiza mensagem "Conta criada com Sucesso!"

Caso de Uso: Inserir Imobiliárias

Descrição: Este caso de uso, permite inserir uma nova imobiliária no sistema.

Tabela 10: Descrição do Caso de Uso – Inserir Imobiliárias

Pré-Condição	Login com Sucesso
Sequência de Passos	<ol style="list-style-type: none"> 1. Selecciona a opção <u>imobiliária</u> no menu e seguidamente na opção <u>Inserir Imobiliárias</u> 2. Selecciona o botão <u>Inserir</u> são mostrados os campos necessários a preencher para a criação de uma novo Imobiliária: <ul style="list-style-type: none"> • Dados da Imobiliária, onde terá de preencher a seguinte informação: Nome, Morada, Telefone, Número da conta. 4. O sistema guarda os dados administrador ao clicar no botão inserir
Caminhos Alternativos	No passo (2) Selecciona opção <u>Cancelar</u> .
Pós-Condição	6. O Administrador visualiza mensagem "Imobiliária criada com Sucesso!"

Nota:

Relativamente aos caso de uso **inserir quarto** e **inserir Imóveis**, o processo de sequências de passos é praticamente o mesmo, mudando só as tabelas e os campos.

Caso de Uso: Efectuar Pagamentos Empregada de Limpeza

Descrição: Este caso de uso, permite efectuar o pagamento às empregadas de limpeza. Para isso, a empregada de limpeza tem que estar atribuída a um imóvel, e já tenha feito alguma limpeza nesse imóvel. Aqui o sistema vai calcular o pagamento da empregada automaticamente. O sistema possui o valor que a empregada ganha por hora, e determina as horas que a empregada de limpeza fez numa determinada limpeza e calcula o valor.

Tabela 11: Descrição do Caso de Uso – Efectuar Pagamentos

Pré-Condição	Login com Sucesso
Sequência de Passos	<ol style="list-style-type: none"> 1. Selecciona a opção <u>Pagamentos</u> no menu e seguidamente na opção <u>Pagamentos Empregadas</u> 2. É exibido uma tabela, com o nome das empregadas. <ul style="list-style-type: none"> • Selecionamos a <u>empregada</u> que pretendemos. 3. É exibido uma tabela com a informação relativamente às limpezas efectuadas pela empregada. <ul style="list-style-type: none"> • Selecionamos que <u>tarefa</u> quer efectuar o pagamento. 4. O sistema guarda os cálculos totais ao clicar no botão <u>Pagar</u>
Caminhos Alternativos	<ol style="list-style-type: none"> 1.a. Consultar/ver pagamentos já efectuados 1.b. O Sistema mostra <p>Em qualquer um dos passos permite seccionar a opção <u>Cancelar</u>.</p>
Pós-Condição	O Administrador visualiza mensagem "Pagamento Efectuado com Sucesso!"

Caso de Uso: Pagamentos dos Arrendatários

Descrição: Neste caso de uso, permite ao administrador efectuar o pagamento dos arrendatários. Para isso o arrendatário tem que possuir um quarto, e ter serviços associados a esse quarto. Assim, o administrador vai escolher qual o imóvel que quer efectuar o pagamento e seguidamente escolhe o quarto. É apresentada uma lista dos serviços que possui num determinado mês. Seguidamente o sistema verifica quantas pessoas estão no imóvel e divide os serviços fixos pelo número de pessoas que o imóvel possui. A aplicação permite ainda alterar o número de pessoas, caso o administrador entenda que é necessário. Em seguida é apresentado o valor dos serviços automaticamente.

Tabela 12: Descrição do Caso de Uso – Pagamento dos Arrendatários

Pré-Condição	Login com Sucesso
Sequência de Passos	<ol style="list-style-type: none"> 1. Selecciona a opção <u>Pagamentos</u> no menu e seguidamente na opção <u>Pagamentos dos Arrendatário</u>. 2. Selecciona o botão <u>Pagar</u>. 3. É exibido uma tabela, com o nome dos proprietários dos Imóveis. <ul style="list-style-type: none"> • Selecionamos o <u>imóvel</u> que pretendemos. 4. É exibido uma tabela com a informação relativamente aos quartos do imóvel seleccionado. <ul style="list-style-type: none"> • Selecionamos o <u>quarto</u> pretendido. 5. É exibido duas caixas de selecção: Uma relativamente ao <u>ano</u> e outra relativamente ao <u>mês</u>. 6. Seleccionar o Botão <u>OK</u> 7. É exibido uma tabela com as contas para pagar. <ul style="list-style-type: none"> • Selecionamos a <u>conta</u> pretendida 8. É apresentado o valor dos serviços dividido pelo número de quarto que o imóvel possui arrendados. <ul style="list-style-type: none"> • É exibido uma caixa que permite alterar o número de pessoas do imóvel. 8. O sistema guarda os dos pagamentos ao clicar no botão <u>Pagar</u>
Caminhos Alternativos	Em qualquer um dos passos permite seleccionar opção <u>Cancelar</u> .

Pós-Condição	O Administrador visualiza mensagem "Pagamento Efectuado com Sucesso!"
---------------------	---

Caso de Uso: Processar Recibo

Descrição: Neste caso de uso, permite em qualquer o administrador processar o recibo de um pagamento que já foi pago.

Tabela 13: Descrição do Caso de Uso – Processar Recibo

Pré-Condição	Login com Sucesso
Sequência de Passos	<ol style="list-style-type: none"> 1. Selecciona a opção <u>Processar Recibo</u> no menu 2. Selecciona o botão <u>Pagar</u>. 3. É exibido três caixas de selecção: Uma relativamente ás <u>imobiliárias</u>, outra aos <u>arrendatários</u> e o <u>quarto</u>. 4. É exibido uma tabela com informação. <ol style="list-style-type: none"> a) Seleccionamos <u>Recibo</u> 5. O sistema imprime o Recibo ao clicar no botão <u>Imprimir</u>
Caminhos Alternativos	No passo (4) Selecciona opção <u>Cancelar</u> .

2.2. Diagrama de Sequência

O diagrama de sequência representa a informação de uma forma simples e lógica, descreve a maneira como os grupos de objectos colaboram em algum comportamento ao longo do tempo, registando comportamentos de um único caso de uso e exibe os objectos e as mensagens passadas entre esses objectos no caso de uso. Nesta divisão será exposta alguns diagramas de sequência dos casos de uso mais importante

2.2.1. Diagrama de Sequência: Registrar Arrendatário

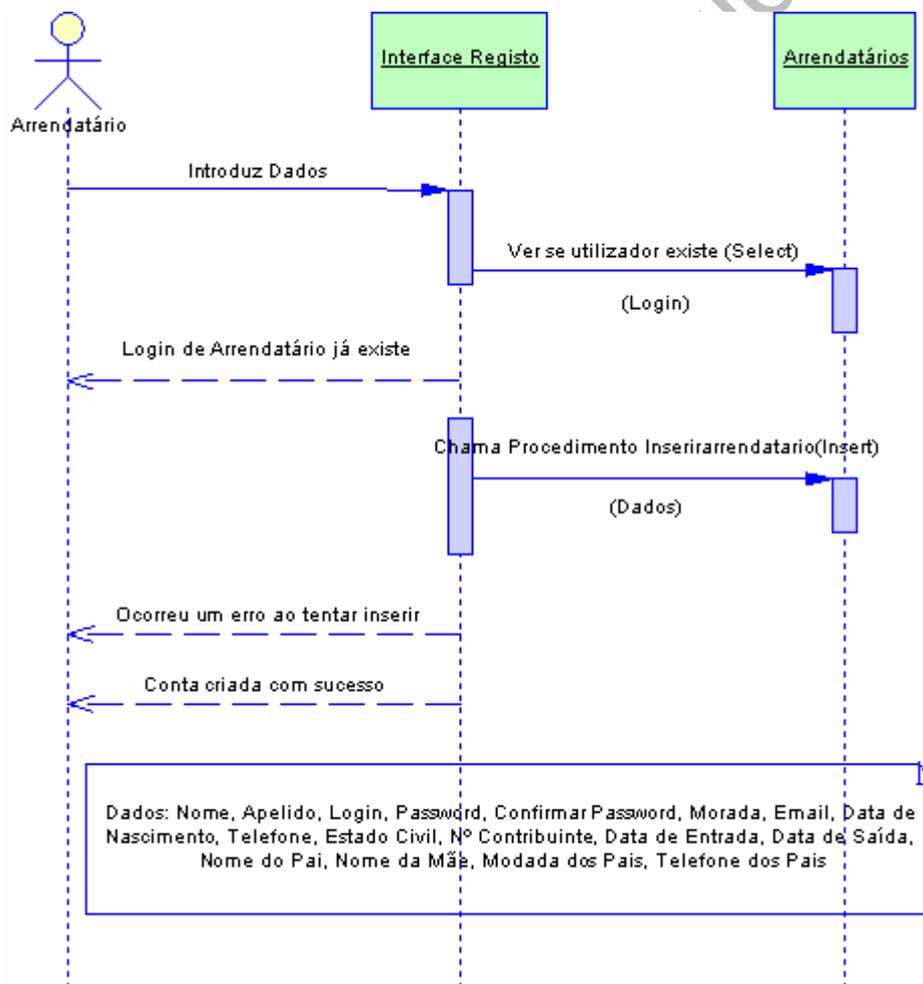


Figura 3 - Diagrama de Sequência: Registrar Arrendatário

2.2.2. Diagrama de Sequência: Efectuar Login

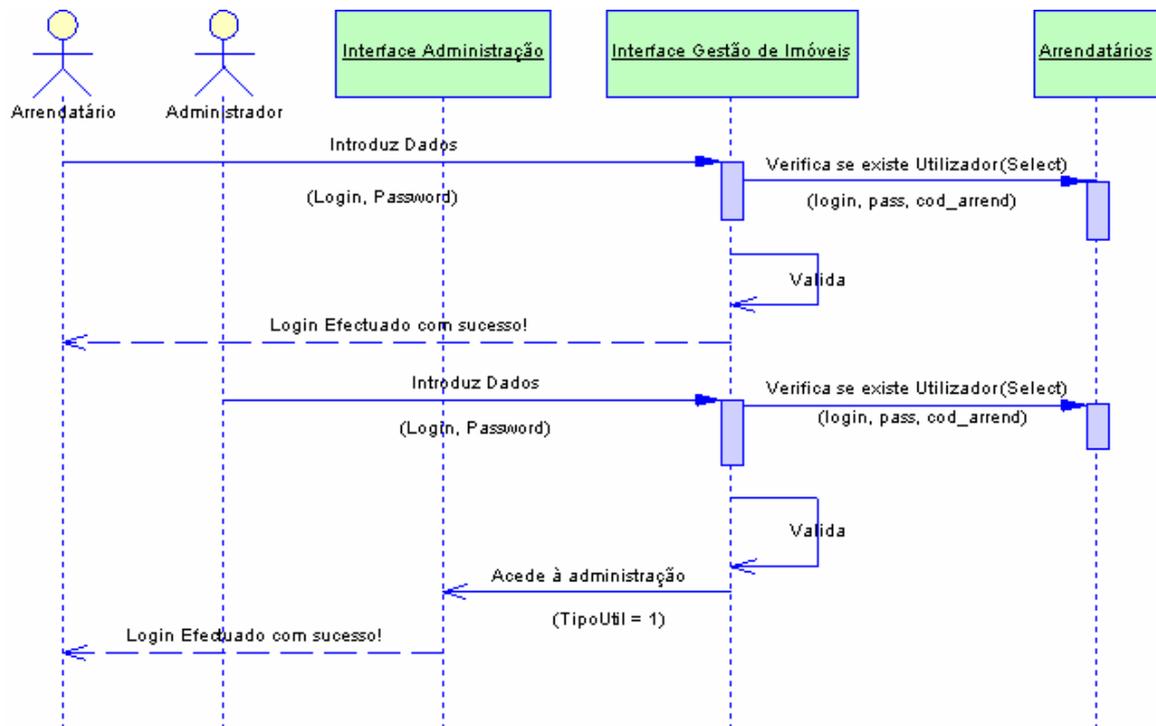


Figura 4 - Diagrama de Sequência: Efectuar Login

2.2.3. Diagrama de Sequência: Recuperar Password

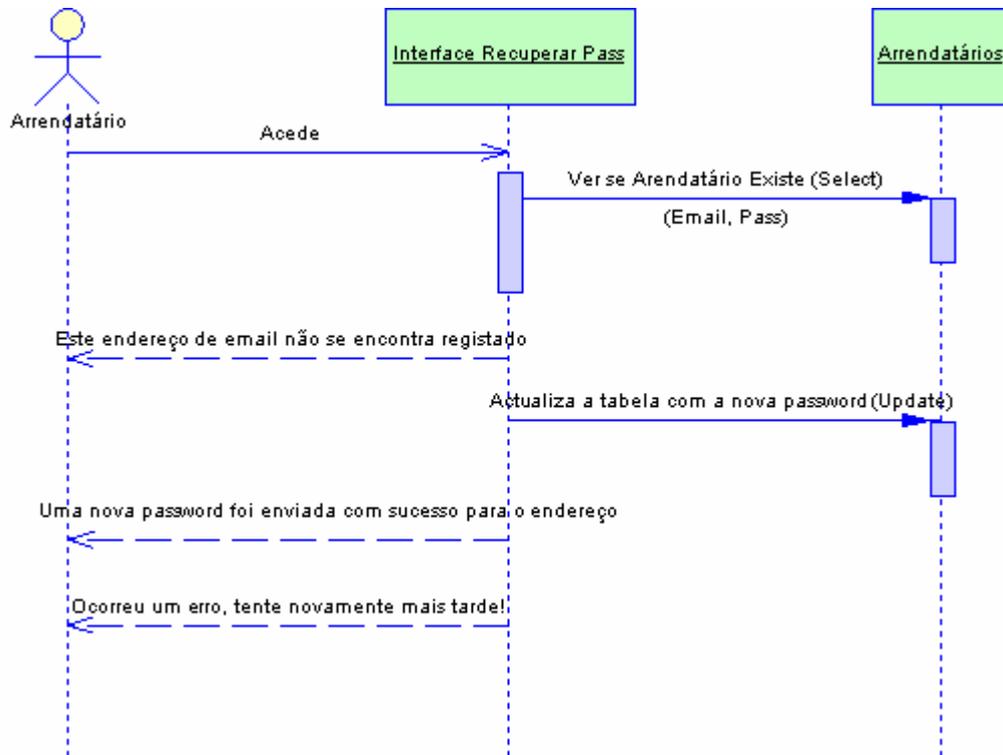


Figura 5 - Diagrama de Sequência: Recuperar Password

2.2.4. Diagrama de Sequência: Alterar Password

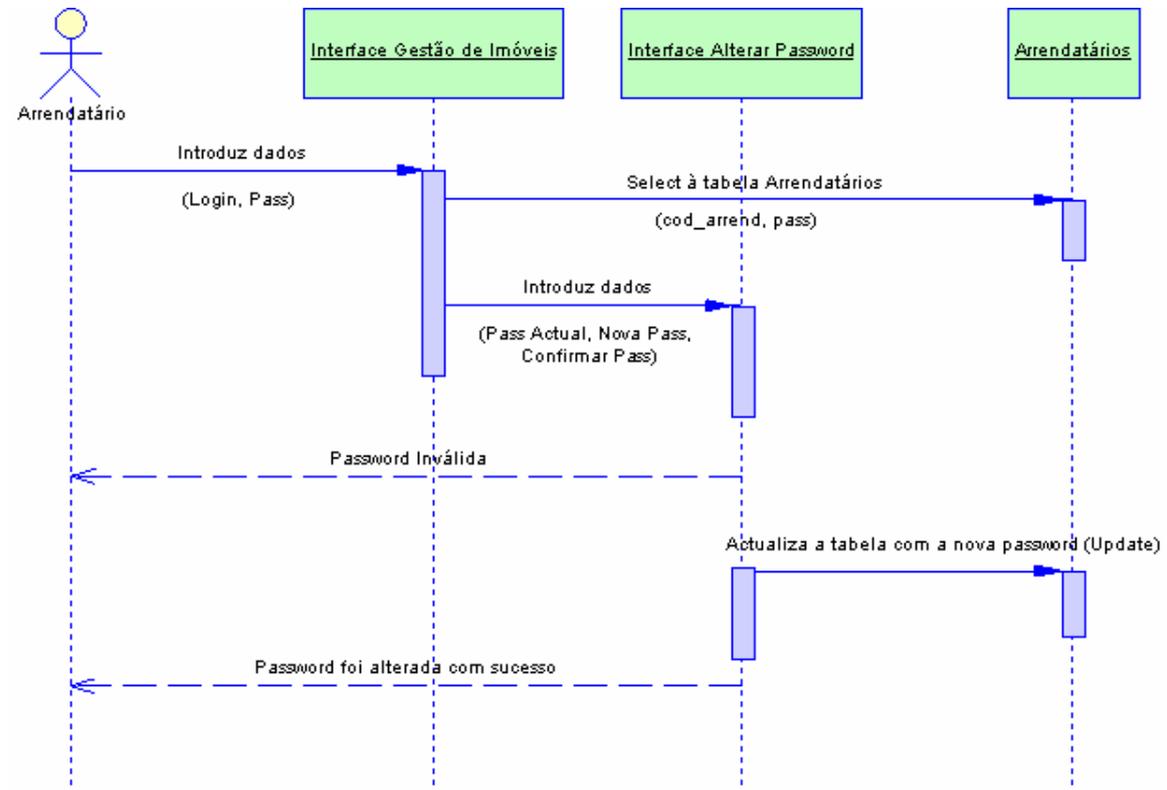


Figura 6 - Diagrama de Sequência: Alterar Password

2.2.5. Diagrama de Sequência: Registrar Administradores

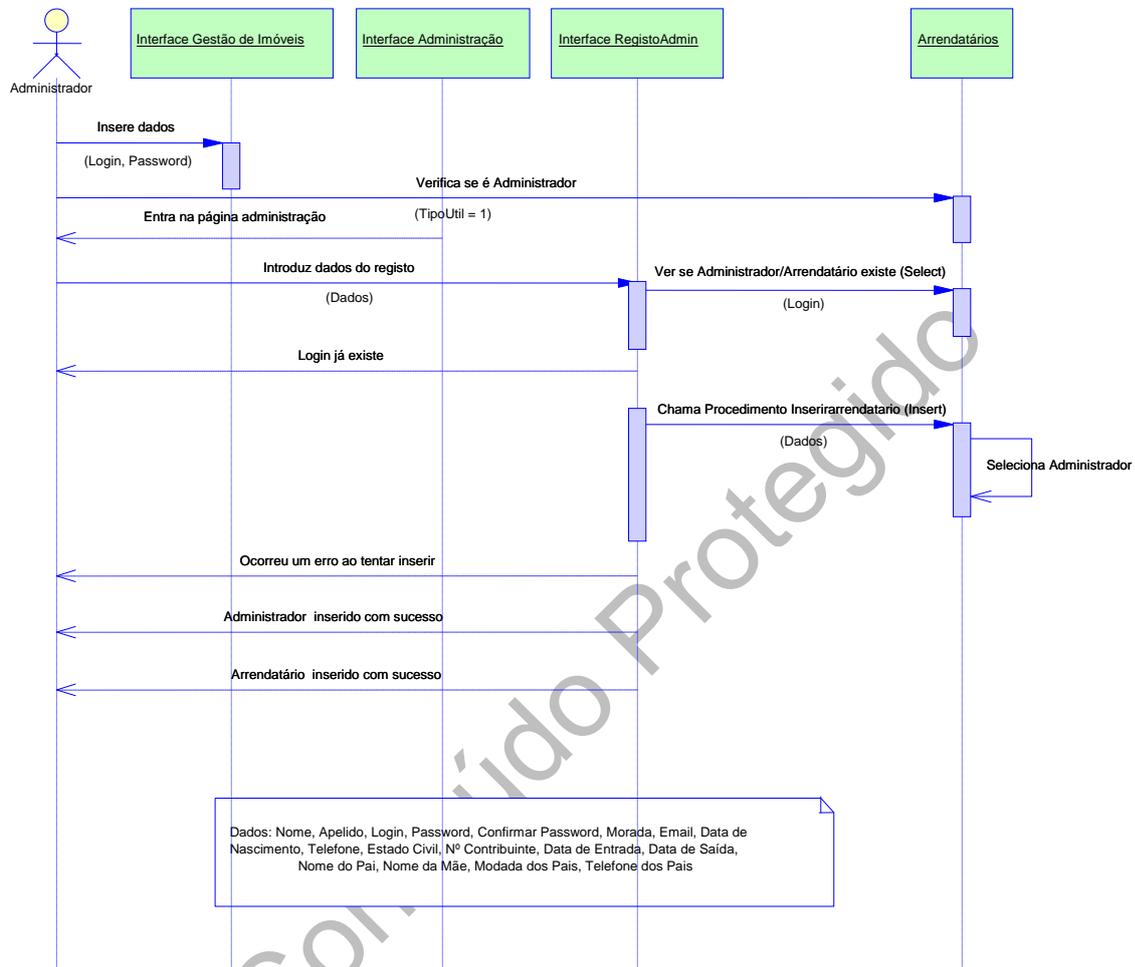


Figura 7 - Diagrama de Sequência: Registrar Administradores

2.2.6. Diagrama de Sequência: Inserir Empregada de Limpeza

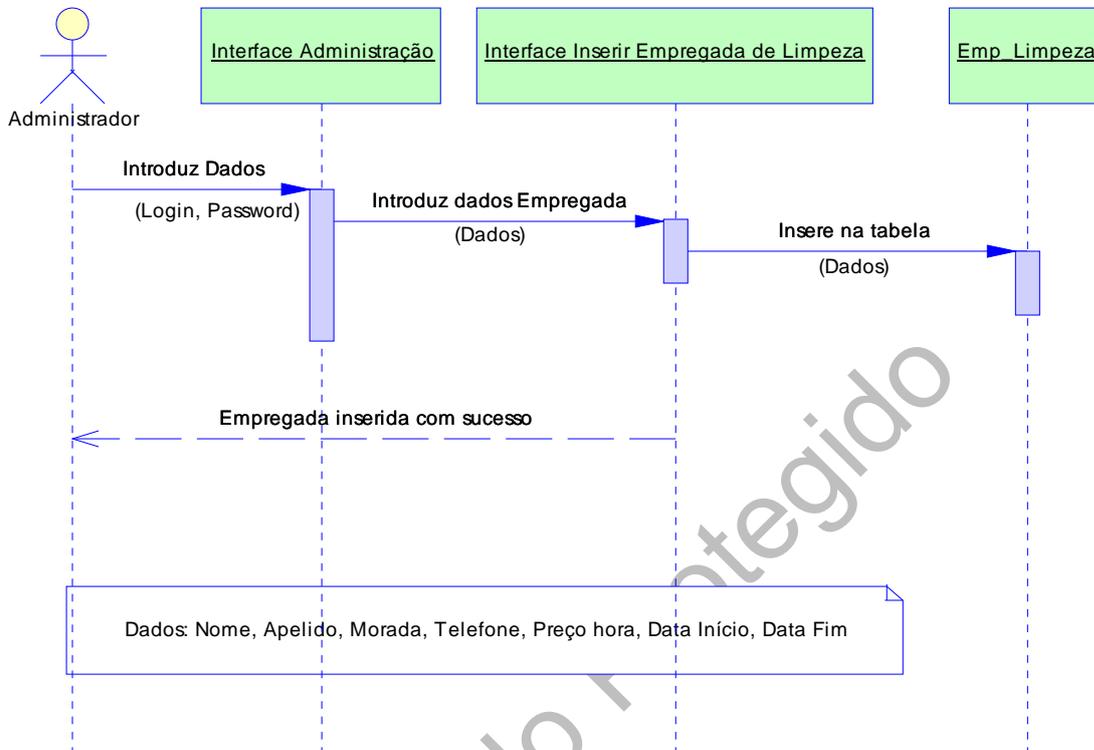


Figura 8 - Diagrama de Sequência: Inserir Empregada de Limpeza

2.2.7. Diagrama de Sequência: Inserir Imóveis

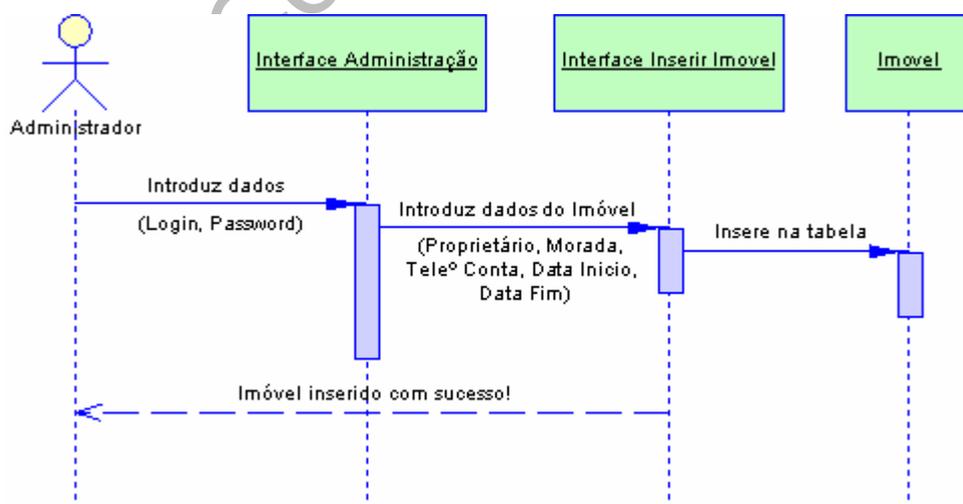


Figura 9 - Diagrama de Sequência: Inserir Imóveis

2.2.8. Diagrama de Sequência: Pagamentos dos Arrendatários

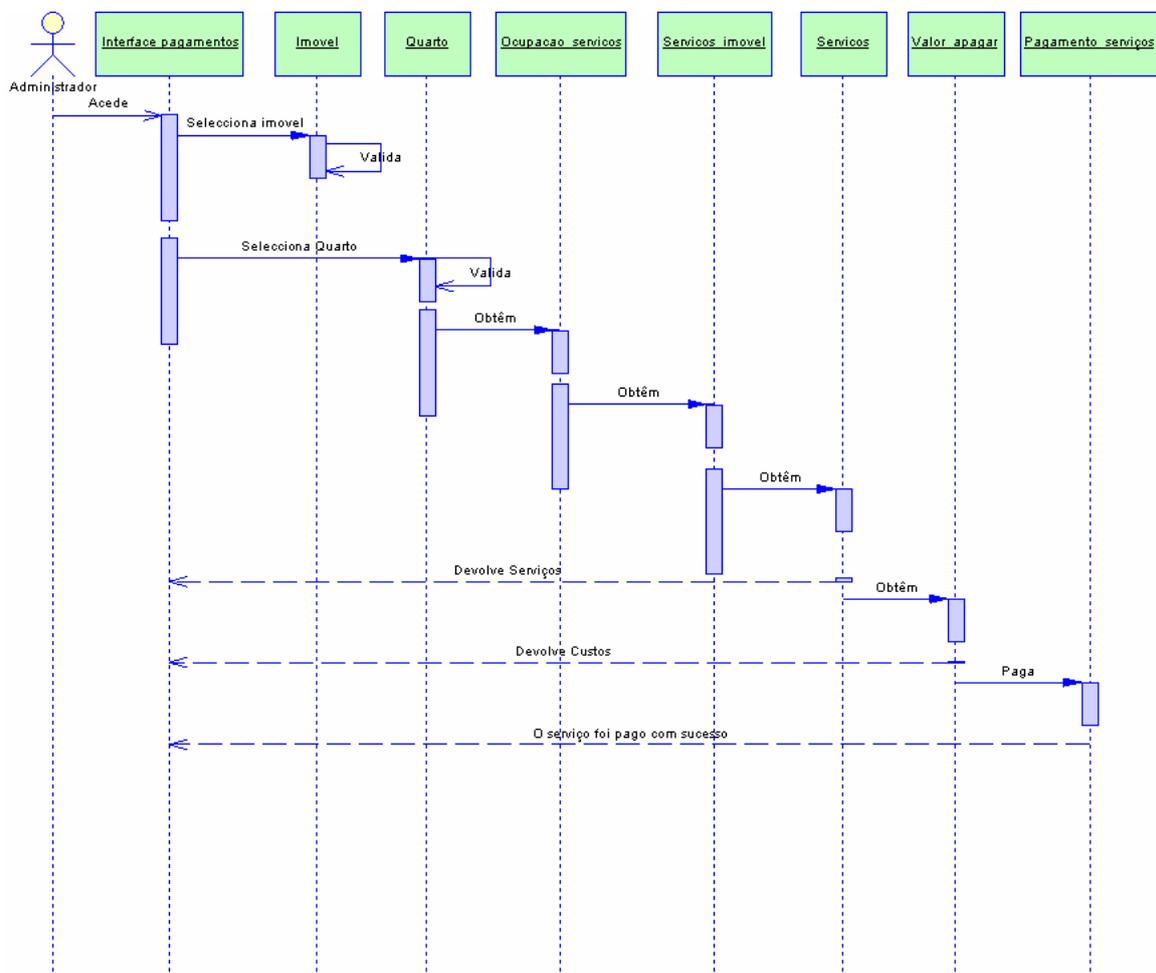


Figura 10 - Diagrama de Sequência: Pagamentos dos Arrendatários

2.2.9. Diagrama de Sequência: Processa Recibo

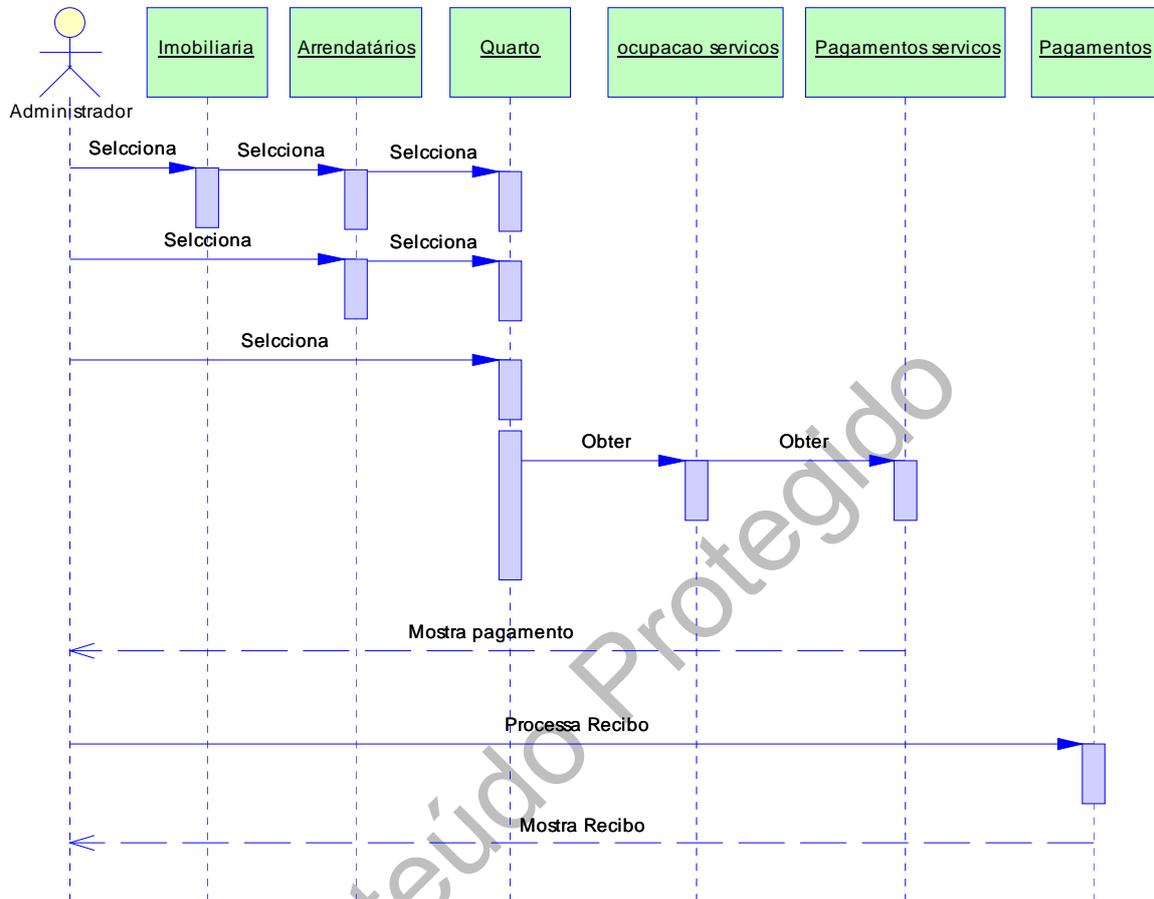


Figura 11 - Diagrama de Sequência: Processa Recibo

2.2.10. Diagrama de Sequência: Pagamento das Empregadas

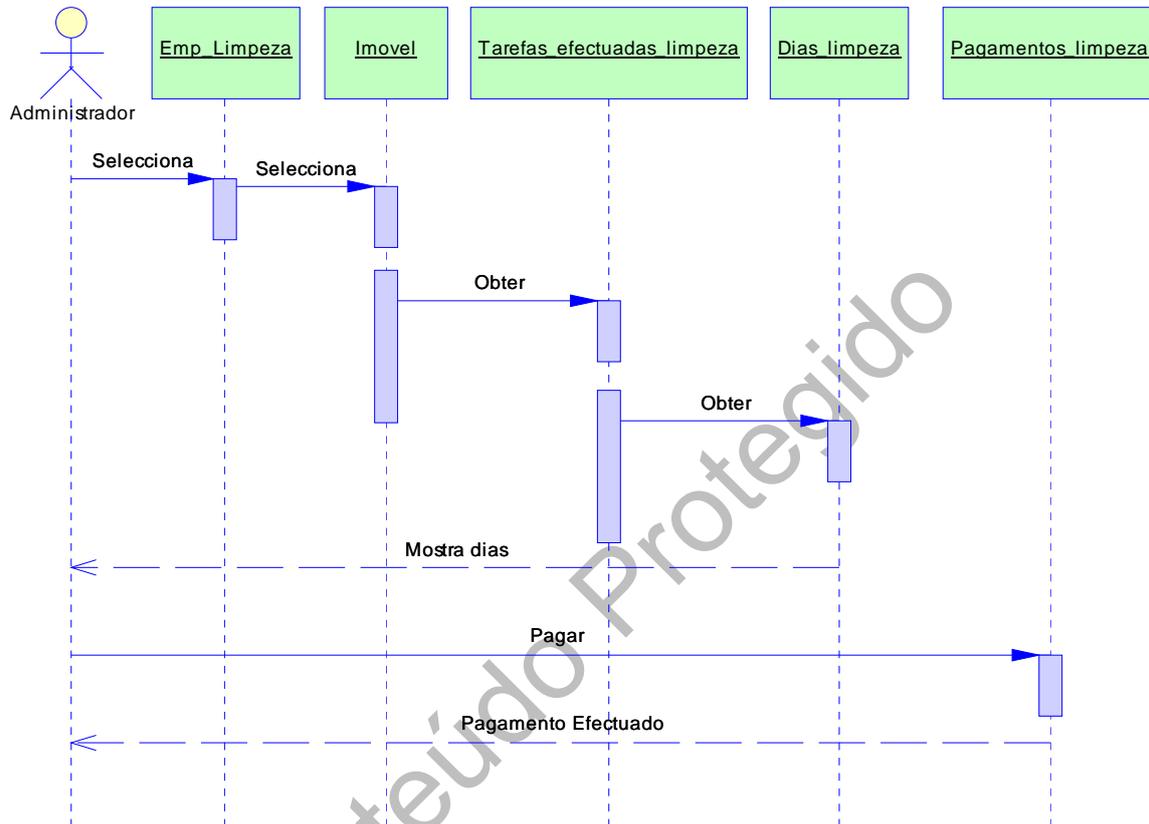


Figura 12 - Diagrama de Sequência: Pagamento das Empregadas

2.3. Diagrama de Classes

Este diagrama é utilizado para modelar a estrutura do nosso sistema, em particular as entidades existentes, as suas estruturas internas e relações entre si. Uma das finalidades do diagrama de classes é organizar e classificar os conceitos do vocabulário do utilizador em objectos, classes, atributos, operações e relações.

Conteúdo Protegido

Conteúdo Protegido

Figura 13 - Diagrama de Classes

2.4. Diagrama Entidade-Relacionamento (ER)

O Diagrama entidade-relacionamento é um modelo que descreve os dados de um sistema com alto nível. Sua maior aplicação é para visualizar o relacionamento entre tabelas de uma base de dados, no qual as relações são construídas através da associação de um ou mais atributos destas tabelas.

Conteúdo Protegido

Conteúdo Protegido

Figura 14 - Diagrama de Entidade-Relacionamento

Tipo de relacionamento de 1 para N:



Figura 15 – Tipo de Relacionamento

Conteúdo Protegido

2.5. Dicionário de Dados

O dicionário de dados consiste numa lista organizada de todos os elementos de dados que são pertinentes para o sistema. O dicionário de dados descreve entradas, saídas, composição de depósitos de dados e alguns cálculos intermédios, para que possuamos ter um modelo completo.

O dicionário de dados baseia-se num ponto de referência de todos os elementos envolvidos na medida em que permite associar um significado a cada termo utilizado. Nesta divisão será descrita todas as entidades existentes na base de dados com a descrição dos seus atributos

Conteúdo Protegido

Arrendatários: Contém Informação acerca dos arrendatários registados. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 14: Entidade – Arrendatários

Entidade: Arrendatário			
Atributos	Tipo	Descrição	Domínio
Cod_arrend	int	Código atribuído ao Arrendatário (chave primária)	1...999
Nome	varchar	Nome do Arrendatário	[100]
Apelido	varchar	Apelido do Arrendatário	[100]
Morada	varchar	Morada do Arrendatário	[100]
Email	varchar	Endereço de <i>E-mail</i> do Arrendatário	[100]
Data_Nascimento	Date	Data de Nascimento do Arrendatário	dd-mm-aaaa
Telefone	varchar	Telefone do Arrendatário	[50]
Estado_Civil	varchar	Estado Civil do Arrendatário	[100]
Num_Contribuinte	varchar	Nº de Contribuinte do Arrendatário	[25]
Login	varchar	Username/Login do Arrendatário	[25]
Password	varchar	<i>Password</i> do Arrendatário	[100]
Data_Entrada	Date	Data de entrada do Arrendatário	dd-mm-aaaa
Data_Saida	Date	Data de Saída do Arrendatário	dd-mm-aaaa
Nome_Pai	varchar	Nome do Pai do Arrendatário	[100]
Nome_Mae	varchar	Nome do Mãe do Arrendatário	[100]
Morada_Pais	varchar	Morada dos Pais do Arrendatário	[100]
Telefone_Pais	varchar	Telefone dos Pais do Arrendatário	[100]
TipoUtil	Bool	Tipo de Arrendatário (1 Administrador; 0 Arrendatário)	[1,0]
Activo	Bool	Se arrendatário está (1 bloqueado; 0 desbloqueado).	[1,0]

Operações		
Nome	Descrição	Acção
Criar	Operação que permite criar um novo arrendatário ou um administrador	<p>O código Arrendatário/Administrador é calculado através do código + 1.</p> <p>Pede para introduzir:</p> <ul style="list-style-type: none"> • Nome completo • Apelido • Morada • E-mail • Data de Nascimento • Telefone (Telemóvel) • Estado Civil • Nº de Contribuinte • Login • Password • Data Entrada • Nome do Pai • Nome da Mãe • Morada dos Pais • Telefone do Pais <p>Valida os campos obrigatórios/ valores válidos e guarda</p>
Alterar	Operação que permite alterar os dados de um Arrendatário/Administrador	<p>Pode alterar:</p> <ul style="list-style-type: none"> • Nome • Apelido • Morada • E-mail • Data de Nascimento • Telefone (Telemóvel) • Estado Civil • Nº de Contribuinte • Nome do Pai • Nome da Mãe • Morada dos Pais • Telefone do Pais <p>Confirma e guarda</p>
Consultar	Operação que permite a consulta dos dados de um determinado arrendatário/administrador	Pode consultar todos os dados de um determinado arrendatário/Administrador

Imóveis: Contem informação sobre os Imóveis disponíveis. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 15: Entidade – Imóvel

Entidade: Imovel			
Atributos	Tipo	Descrição	Domínio
Cod_Imovel	int	Código atribuído ao Imóvel (chave primária)	1...9999
cod_Imobiliaria	int	Código atribuído à Imobiliária (chave estrangeira)	1...9999
Morada	varchar	Morada do Imóvel	[100]
Telefone	varchar	Telefone do Proprietário	[100]
Num_Conta	varchar	Nº da Conta do Proprietário	[100]
Proprietario	varchar	Nome do Proprietário	[100]
Data_Inicio	Date	Data de inicio de aquisição	dd-mm-aaaa
Data_Fim	Date	Data de fim	dd-mm-aaaa
Data_saida	Date	Data de saída do arrendatário do quarto	dd-mm-aaaa
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar um novo Imóvel	<p>O código do Imóvel é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Nome do Proprietário • Morada do Imóvel • Telefone do Proprietário • NIB do Proprietário do Imóvel • Data de Inicio de aquisição do Imóvel <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	

Alterar	Operação que permite alterar os dados de um Imóvel	Pode alterar: <ul style="list-style-type: none">• Nome do Proprietário• Morada do Imóvel• Telefone do Proprietário• NIB do Proprietário do Imóvel Confirma e guarda
Remover	Operação que permite remover os dados de um Imóvel	Permite ao administrador remover um imóvel, caso não tenha nenhum quarto associado.
Consultar	Operação que permite a consulta dos dados de um determinado Imóvel	Pode consultar todos os dados de um determinado Imóvel

Conteúdo Protegido

Imobiliária: Contem informação sobre as Imobiliárias. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 16: Entidade – Imobiliária

Entidade: Imobiliária			
Atributos	Tipo	Descrição	Domínio
Cod_Imobiliaria	int	Código atribuído á imobiliário (chave primária)	1...9999
Nome	varchar	Nome da Imobiliária	[100]
Morada	varchar	Morada da Imobiliária	[200]
Telefone	varchar	Telefone da Imobiliária	[20]
Num_Conta	varchar	Nº de conta da Imobiliária	[50]
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar uma nova imobiliária	<p>O código da imobiliária é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Nome da Imobiliária • Morada da Imobiliária • Telefone da Imobiliária • NIB da Imobiliária <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Alterar	Operação que permite alterar os dados de uma imobiliária	<p>Pode alterar:</p> <ul style="list-style-type: none"> • Nome da Imobiliária • Morada da Imobiliária • Telefone da Imobiliária • NIB da Imobiliária <p>Confirma e guarda</p>	
Remover	Operação que permite remover os dados de uma imobiliária	Permite ao administrador remover uma imobiliária, caso não tenha nenhum imóvel associado.	
Consultar	Operação que permite a consulta dos dados de uma determinada imobiliária	Pode consultar todos os dados de uma determinada Imobiliária	

Despesas Imóvel: Contem informação sobre as despesas de um determinado imóvel.

A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 17: Entidade – Despesas Imóvel

Entidade: Despesas Imovel			
Atributos	Tipo	Descrição	Domínio
Cod_Despesa	int	Código atribuído às despesas do Imóvel (chave primária)	1...9999
cod_Imobiliaria	int	Código atribuído à Imobiliária (chave estrangeira)	1...9999
Cod_Imovel	int	Código atribuído ao Imóvel (chave estrangeira)	1...9999
Cod_Despesas	int	Código atribuído às despesas (chave estrangeira)	1...9999
Data	Date	Data das despesas efectuadas	dd-mm-aaaa
Data_Pagamento	Date	Data em que foi efectuado o pagamento	dd-mm-aaaa
Valor	Float	Valor das despesas	[8,2]
Operações			
Nome	Descrição	Acção	
Consultar	Operação que permite a consulta dos dados de uma determinada despesa	Pode consultar os dados de uma determinada despesa de um determinado imóvel. <ul style="list-style-type: none"> • Data • Data de pagamento de uma despesa • Valor de uma determinada despesa 	

Despesas: Contem informação sobre o tipo de despesas de um determinado imóvel. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 18: Entidade – Despesas

Entidade: Despesas			
Atributos	Tipo	Descrição	Domínio
Cod_Despesas	int	Código atribuído às despesas (chave estrangeira)	1...9999
Descrição	varchar	Descrição sobre uma determinada despesa	[100]
TipoDespesa	smallint	Tipo de Despesa	[20]
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar uma nova despesa	<p>O código da despesa é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Tipo de despesa • Descrição de uma despesa <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Alterar	Operação que permite alterar os dados de uma despesa	<p>Pode alterar:</p> <ul style="list-style-type: none"> • Tipo de despesa • Descrição de uma despesa <p>Confirma e guarda</p>	
Remover	Operação que permite remover os dados de uma despesa	Permite ao administrador remover uma despesa, caso não tenha nenhum imóvel associado.	
Consultar	Operação que permite a consulta dos dados de uma determinada despesa	Pode consultar todos os dados de uma determinada despesa	

Emp_Limpeza: Contém Informação geral acerca das empregadas de limpeza. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 19: Entidade – Emp_Limpeza

Entidade: Emp_Limpeza			
Atributos	Tipo	Descrição	Domínio
Cod_Emp	int	Código atribuído á empregada de limpeza (chave primária)	1...9999
Nome	int	Nome da Empregada de limpeza	[100]
Apelido	int	Apelido da empregada de limpeza	[100]
Morada	varchar	Morada da empregada de limpeza	[100]
Telefone	varchar	Telefone da empregada de limpeza	[100]
Preco Hora	varchar	Preço à hora da empregada de limpeza	[100]
Data Inicio	date	Data de início de contrato	dd-mm-aaaa
Data Fim	date	Data de fim de contrato	dd-mm-aaaa
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar uma nova empregada de limpeza	<p>O código do Emp_limpeza é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Nome da empregada de limpeza • Apelido da empregada de limpeza • Morada da empregada de limpeza • Telefone da empregada • Preço por hora da empregada de limpeza • Data do Inicio de contrato • Data de fim de contrato <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Alterar	Operação que permite alterar os dados de uma empregada de limpeza	<p>Pode alterar:</p> <ul style="list-style-type: none"> • Nome da empregada de limpeza • Apelido da empregada de limpeza • Morada da empregada de limpeza • Telefone da empregada • Preço por hora da empregada de limpeza <p>Confirma e guarda</p>	

Remover	Operação que permite remover os dados de uma empregada de limpeza	Permite ao administrador remover uma empregada de despesa, caso não tenha nenhum imóvel associado.
Consultar	Operação que permite a consulta dos dados de uma determinada empregada de limpeza	Pode consultar todos os dados de uma determinada empregada de limpeza

Contrato_Limpeza: Contém Informação geral acerca dos contratos de limpeza. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 20: Entidade – Contrato_Limpeza

Entidade: Contrato_Limpeza			
Atributos	Tipo	Descrição	Domínio
cod_contrato_limpez as	int	Código atribuído ao contrato de limpeza (chave primária)	1...9999
cod_Imobiliaria	int	Código atribuído à Imobiliária (chave estrangeira)	1...9999
Cod_Imovel	int	Código atribuído ao Imóvel (chave estrangeira)	1...9999
Cod_Emp	int	Código atribuído à empregada de limpeza (chave estrangeira)	1...9999
Data Inicio	date	Data de início de contrato	dd-mm-aaaa
Data Fim	date	Data de fim de contrato	dd-mm-aaaa
Operações			
Nome	Descrição	Acção	
Consultar	Operação que permite a consulta dos dados de um contrato de Limpeza	Pode consultar os dados de um determinado contrato de limpeza de um determinado imóvel. <ul style="list-style-type: none"> Data de inicio de contrato com um determinado imóvel Data de fim de contrato com um determinado imóvel (1 Ano) Valor de uma determinada despesa	

Dias_limpeza: Contém Informação geral acerca dos dias da limpeza. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 21: Entidade – Dias_limpeza

Entidade: Dias_limpeza			
Atributos	Tipo	Descrição	Domínio
cod_dias_limpeza	int	Código atribuído aos dias de limpeza (chave primária)	1...9999
cod_pagamento_limpeza	int	Código atribuído ao pagamento da limpeza (chave estrangeira)	1...9999
cod_contrato_limpezas	int	Código atribuído ao contrato de limpeza (chave estrangeira)	1...9999
Despesas_Material	float	Despesas de material.	[8,2]
hora_inicio	time	Hora de inicio da limpeza.	[00:00]
hora_fim	time	Hora de fim de limpeza.	[00:00]
Data	date	Data da limpeza.	dd-mm-aaaa
Pago	boll	Se o pagamento foi efectuado ou nao	[0,1]
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar um novo dia de Limpeza	<p>O código do dias_limpeza é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Data em que a limpeza foi realizada • Hora do inicio da limpeza • Hora do fim da limpeza • Despesa de material, caso existam <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Calcular	Operação que permite calcular o número de horas de limpeza	<p>Calcula o número de horas que uma empregada de limpeza demorou a efectuar a limpeza a um determinado imóvel.</p> <ul style="list-style-type: none"> • Hora de inicio • Hora de fim • Total = horas fim-horas inicio * Preço Hora da empregada de Limpeza 	

Pagamento_Limpeza: Contém Informação geral acerca do pagamento da limpeza. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 22: Entidade – Pagamento_Limpeza

Entidade: Pagamento_limpeza			
Atributos	Tipo	Descrição	Domínio
cod_pagamento_limpeza	int	Código atribuído ao pagamento da limpeza (chave primária)	1...9999
Cod_dias_limpeza	int	Código atribuído aos dias de Limpeza (chave estrangeira)	1...9999
data	date	Data do pagamento da limpeza.	dd-mm-aaaa
valor_total	varchar	Valor total das do pagamento às empregadas de limpeza	[20]
num_horas	float	Número de horas efectuado pela empregada de Limpeza	[20]
valor_despesas	float	Valor das despesas do Imóvel	[8,1]
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar um novo pagamento de limpeza	<p>O código do pagamento_limpeza é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Data = data do Sistema • Valor total, é a multiplicação do valor do número de horas efectuadas pelo preço da empregada por hora. O sistema atribui este valor automaticamente. • Num_ horas, é o número de horas efectuado pela empregada de limpeza. Este número é calculado pelo sistema automaticamente • Valor das despesas, é introduzido pelo manualmente <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Consultar	Operação que permite a consultar dos dados de pagamento de uma empregada de limpeza	<p>Pode consultar todos os dados de pagamento de uma empregada de limpeza.</p>	

Tarefas_efectuadas_limpeza: Contém Informação geral acerca das tarefas efectuadas por uma determinada empregada de Limpeza A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 23: Entidade – Tarefas_efectuadas_limpeza

Entidade: Tarefas_efectuadas_limpeza			
Atributos	Tipo	Descrição	Domínio
Cod_tarefas_efectuadas_limpeza	int	Código atribuído às tarefas efectuadas das limpezas (chave primária)	1...9999
Cod_dias_limpeza	int	Código atribuído aos dias de Limpeza (chave estrangeira)	1...9999
Cod_tarefas_limpeza	int	Código atribuído às tarefas de Limpeza (chave estrangeira)	1...9999
efectuadas	boll	Se uma determinada tarefa foi efectuada (0 por efectuar; 1 Efectuada)	[0,1]
Operações			
Nome	Descrição	Acção	
Consultar	Operação que permite a consultar dos dados se uma empregada realizou uma determinada tarefa de limpeza	Pode consultar os dados de uma determinada tarefa de limpeza, se foi efectuada ou não	

Tarefas_Limpeza: Contém Informação geral acerca das tarefas de Limpeza A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 24: Entidade – Tarefas_limpeza

Entidade: Tarefas_limpeza			
Atributos	Tipo	Descrição	Domínio
Cod_tarefas_limpeza	int	Código atribuído às tarefas efectuadas das limpezas (chave primária)	1...9999
Cod_dias_limpeza	int	Código atribuído aos dias de Limpeza (chave estrangeira)	1...9999
Cod_contracto_limpeza	int	Código atribuído ao contracto de Limpeza (chave estrangeira)	1...9999
Tarefas_limpeza	varchar	Tipo de tarefa	[100]
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar uma nova tarefa de limpeza	<p>O código da tarefa de limpeza é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> Tarefa de Limpeza <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Consultar	Operação que permite a consulta dos dados de uma determinada tarefa de limpeza	Pode consultar todos os dados de uma determinada tarefa de limpeza	

Funcoes_limpeza: Contém Informação geral acerca das funções de Limpeza. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 25: Entidade – Funcoes_limpeza

Entidade: Funcoes_limpeza			
Atributos	Tipo	Descrição	Domínio
Cod_funcoes_limpeza	int	Código atribuído às funções de Limpeza (chave primária)	1...9999
Descricao	varchar	Descrição da função de Limpeza	[100]
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar uma nova função de limpeza	<p>O código da função de limpeza é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Descrição de Limpeza <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Consultar	Operação que permite a consulta dos dados de uma determinada função de limpeza	Pode consultar todos os dados de uma determinada tarefa de limpeza	
Alterar	Operação que permite alterar dos dados de uma determinada função de limpeza	<p>Pode alterar:</p> <ul style="list-style-type: none"> • Descrição da limpeza <p>Confirma e guarda</p>	

Serviços Imóvel: Contém informação sobre os serviços que contem um determinado Imóvel. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 26: Entidade – Serviço Imovel

Entidade: Serviço Imovel			
Atributos	Tipo	Descrição	Domínio
Cod_servicos_imovel	int	Código atribuído ao tipo de ficheiro (chave primária)	1...9999
cod_Imobiliaria	int	Código atribuído à Imobiliária (chave estrangeira)	1...9999
Cod_Imovel	int	Código atribuído ao Imóvel (chave estrangeira)	1...9999
Cod_Servicos	int	Código atribuído ao Tipo de Serviço (chave estrangeira)	1...9999
valor_servico_fixo	float	Valor fixo do tipo de serviço.	[8,2]
Operações			
Nome	Descrição		Acção
Consultar	Operação que permite a consultar dos dados se de um serviço de imovel		Pode consultar os dados de um determinado imóvel. Se possui serviço, e se possui que serviços tem.

Serviços: Contém informação sobre os serviços. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 27: Entidade – Serviços

Entidade: Serviços			
Atributos	Tipo	Descrição	Domínio
Cod_Serviços	int	Código atribuído ao serviço (chave primária)	1...9999
descricao	varchar	Descrição do Serviço.	[100]
opcional	boll	Se o serviço é opcional ou não	[0,1]
fixo	boll	Se o serviço tem um valor fixo ou não	[0,1]
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar um novo serviço	<p>O código do serviço é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Descrição do serviço • Opcional, se é um serviço opcional ou não. Selecciona a caixa de texto: SIM; NÃO • Fixo, se é um serviço que tem um valor fixo ao não. Selecciona a caixa de texto: SIM; NÃO <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Alterar	Operação que permite alterar um serviço	<p>Pode alterar:</p> <ul style="list-style-type: none"> • Descrição do serviço • Opcional, se é um serviço opcional ou não. Selecciona a caixa de texto: SIM; NÃO • Fixo, se é um serviço que tem um valor fixo ao não. Selecciona a caixa de texto: SIM; NÃO <p>Confirma e guarda</p>	
Remover	Operação que permite remover um serviço	Permite ao administrador remover um serviço, caso não tenha nenhum imóvel associado.	
Consultar	Operação que permite a consultar dos dados de um serviço	Pode consultar todos os serviços.	

Valor_apagar: Contém informação sobre o valor a pagar de um determinado serviço de imóvel. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 28: Entidade – Valor_apagar

Entidade: Valor_apagar			
Atributos	Tipo	Descrição	Domínio
cod_valor_apagar	int	Código atribuído ao valor a pagar (chave primária)	1...9999
Cod_servicos_imovel	int	Código atribuído aos serviços do imóvel (chave estrangeira)	1...9999
valor	float	Valor que tem a pagar.	[8,2]
mes	int	Mês do pagamento.	1...9999
ano	int	Ano do pagamento.	1...9999
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar valor para um determinado serviço.	<p>O código do valor_apagar é calculado através do código + 1.</p> <p>Pede para introduzir:</p> <ul style="list-style-type: none"> • Custo, é o valor mensal de um determinado serviço • Mês, selecciona na caixa de texto o mês do valor do custo do serviço. • Ano, selecciona na caixa de texto o ano do valor do custo do serviço. <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Alterar	Operação que permite o custo de um serviço	<p>Pode alterar:</p> <ul style="list-style-type: none"> • Custo, é o valor mensal de um determinado serviço • Mês, selecciona na caixa de texto o mês do valor do custo do serviço. • Ano, selecciona na caixa de texto o ano do valor do custo do serviço. <p>Confirma e guarda</p>	
Consultar	Operação que permite a consultar os custos de um serviço.	Pode consultar os custos anteriores.	

Ocupacao Servicos: Contém Informação sobre a ocupação de um serviço. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 29: Entidade – Ocupacao Servicos

Entidade: Ocupacao Servicos			
Atributos	Tipo	Descrição	Domínio
Cod_ocupacao_servicos	int	Código atribuído á ocupação dos serviços (chave primaria)	1...9999
Cod_servicos_imovel	int	Código atribuído aos serviços do imóvel (chave estrangeira)	1...9999
Cod_ocupacao	int	Código atribuído à ocupação (chave estrangeira)	1...9999
Valor_fixo	float	Valor fixo.	[8,2]
mes	int	Mês	1...9999
ano	int	Ano	1...9999
Pago	boll	Se foi pago ou não (1 pago; 0 por pagar)	[0;1]
ocupado	boll	Se o serviço pertence ao imóvel ou não (1 ocupado; 0 desocupado)	[0;1]
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar uma ocupação de serviços.	<p>O código da ocupação serviço é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Valor_fixo, o sistema atribui este valor automaticamente. • Mês, o sistema devolve o mês automaticamente. • Ano, o sistema devolve o ano automaticamente. <p>O sistema atribui '0' caso o administrador queria abandonar um serviço, ficando logo desocupado. Ao pagar um serviço este fica logo com o campo pago a 1. Valida os campos obrigatórios/ valores válidos e guarda</p>	
Consultar	Operação que permite a consultar dos dados de ocupação de serviços.	Pode consultar todas as ocupações de serviços.	

Pagamento_Serviços: Contém Informação sobre os pagamentos de serviços. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 30: Entidade – Pagamento_Serviços

Entidade: Pagamento_Serviços			
Atributos	Tipo	Descrição	Domínio
cod_pagamento_servico	int	Código atribuído ao pagamento do serviço (chave primária)	1...9999
Cod_ocupacao_servicos	int	Código atribuído á ocupação dos serviços (chave estrangeira)	1...9999
Valor_apagar	float	Valor que tem a pagar.	[8,2]
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar valor para um determinado serviço.	<p>O código do valor_apagar é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Valor_apagar, é o valor total do pagamento <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Consultar	Operação que permite a consultar os pagamentos de serviços.	Pode consultar os pagamentos de serviços pagos.	

Pagamentos: Contém Informação sobre os pagamentos. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 31: Entidade – Pagamento

Entidade: Pagamento			
Atributos	Tipo	Descrição	Domínio
cod_pagamento	int	Código atribuído ao pagamento (chave primária)	1...9999
Cod_ocupacao_servicos	int	Código atribuído á ocupação dos serviços (chave estrangeira)	1...9999
Data_emissao	date	Valor que tem a pagar.	dd-mm-aaaa
mes	int	Mês do pagamento	[1,12]
ano	int	Ano do Pagamento	1...9999
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar valor para um determinado serviço.	<p>O código do valor_apagar é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Data de emissão = data do sistema é atribuído automaticamente pelo sistema • Mês é atribuído automaticamente pelo sistema • Ano é atribuído automaticamente pelo sistema <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Consultar	Operação que permite a consultar os pagamentos.	Pode consultar os pagamentos.	
Imprimir	Operação que imprime os dados relativos a um pagamento		

Ocupacao: Contém Informação sobre a ocupação de um quarto. A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 32: Entidade – Ocupacao

Entidade: Ocupacao			
Atributos	Tipo	Descrição	Domínio
Cod_ocupacao	int	Código atribuído à ocupação (chave primária)	1...9999
Cod_arred	int	Código atribuído ao arrendatário (chave estrangeira)	1...9999
Cod_Quarto	int	Código atribuído ao Quarto (chave estrangeira)	1...9999
data_inicio	date	Data em que o arrendatário entra para o quarto.	dd-mm-aaaa
data_fim	date	Data em que foi efectuado o ultimo pagamento	dd-mm-aaaa
renda	float	Renda do quarto ocupado	[8,2]
Data_saida	date	Data em que o arrendatário deixa o quarto.	dd-mm-aaaa
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar valor para um determinado serviço.	<p>O código de ocupação é calculado através do código + 1.</p> <p>Pede para introduzir:</p> <ul style="list-style-type: none"> Data inicio, quando o quarto é ocupado por um arrendatário, neste caso é a data do sistema, que pode ser alterada ou não. Renda, é o valor do quarto <p>Data de Saída e a data de Fim é inicializada a nulo</p> <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Consultar	Operação que permite a consultar os pagamentos.	Pode consultar os pagamentos.	

Actualizar	Operação que permite a actualizar os dados	Data de Saída = Data do Sistema
------------	--	---------------------------------

Conteúdo Protegido

Quarto: Contém Informação sobre os quartos A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 33: Entidade – Quarto

Entidade: Quarto			
Atributos	Tipo	Descrição	Domínio
Cod_Quarto	int	Código atribuído ao Quarto (chave primária)	1...9999
Cod_Imovel	int	Código atribuído ao Imóvel (chave estrangeira)	1...9999
Renda	float	Renda do quarto.	[8,2]
Disponibilidade	boll	Disponibilidades dos quartos.	[0,1]
Descricao	varchar	Contem uma breve descrição do quarto	[20]
Imagem	varchar	Caminho da Imagem	[300]
Descricaocaract	varchar	Contém a descrição com as características do quarto	[500]
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar valor para um determinado serviço.	<p>O código do quarto é calculado através do código + 1.</p> <p>Pede para introduzir:</p> <ul style="list-style-type: none"> Renda Descrição, breve descrição do quarto <p>Cada quarto pode ter uma imagem associada, com uma descrição mais pormenorizada do quarto. Valida os campos obrigatórios/ valores válidos e guarda</p>	
Alterar	Operação que permite o custo de um serviço	<p>Pode alterar:</p> <ul style="list-style-type: none"> Descrição Descricaocaract Pode alterar a imagem do quarto <p>Confirma e guarda</p>	
Consultar	Operação que permite a consultar os quartos.	Pode consultar os custos quartos, que estão ocupado e desocupados.	

Actualizar	Operação que permite a actualizar os dados.	Quando há uma desocupação a disponibilidade=0
------------	---	---

Conteúdo Protegido

Características: Contém Informação sobre as características A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 34: Entidade – Características

Entidade: Características			
Atributos	Tipo	Descrição	Domínio
cod_caract	int	Código atribuído à característica do quarto (chave primária)	1...9999
Descrição	varchar	Descrição de cada característica.	[20]
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar uma nova imobiliária	<p>O código da característica é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> • Descrição da característica <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Consultar	Operação que permite a consulta dos dados de uma característica	Pode consultar todos os dados das características	

qtimovel: Contém Informação sobre as quantidades de características de um imóvel A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 35: Entidade – qtimovel

Entidade: qtimovel			
Atributos	Tipo	Descrição	Domínio
Cod_qti	Int	Código atribuído à quantidade de imóveis (chave primária)	1...9999
Cod_Imovel	Int	Código atribuído ao Imóvel (chave estrangeira)	1...9999
quantidade	int	Quantidade de imóveis.	1...9999
Operações			
Nome	Descrição	Acção	
Criar	Operação que permite criar uma nova imobiliária	<p>O código da qtimovel é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> Quantidade, é o número de uma determinada característica de um imóvel <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Consultar	Operação que permite a consulta dos dados de uma características	Pode consultar todos os dados das características	

qtquarto: Contém Informação sobre as quantidades de características de um quarto A descrição desta entidade pode ser observada na seguinte tabela:

Tabela 36: Entidade – qtquarto

Entidade: qtquarto			
Atributos	Tipo	Descrição	Domínio
Cod_qtq	Int	Código atribuído à quantidade de quartos (chave primária)	1...9999
Cod_Quarto	Int	Código atribuído ao Quarto (chave estrangeira)	1...9999
quantidade	int	Quantidade de quartos.	1...9999
Operações			
Nome	Descrição	Ação	
Criar	Operação que permite criar uma nova imobiliária	<p>O código da qtquarto é calculado através do código + 1. Pede para introduzir:</p> <ul style="list-style-type: none"> Quantidade, é o número de uma determinada característica de um quarto <p>Valida os campos obrigatórios/ valores válidos e guarda</p>	
Consultar	Operação que permite a consulta dos dados de uma características	Pode consultar todos os dados das características	

Capítulo 3

3. Ferramentas e Tecnologias Utilizadas

Para o desenvolvimento do “Gestão de Imóveis” foi imprescindível a utilização de várias ferramentas e tecnologias que serão apresentadas seguidamente.

3.1. Software

Assim, para do desenvolvimento desta aplicação foram usadas as seguintes ferramentas:

- Sybase Power Designer 12;
- Microsoft Visual Studio 2005;
- Microsoft MySQL Sever 5.

3.1.1. Sybase Power Designer 12.0

“O *PowerDesigner*, e a funcionalidade acrescida de meta *Data Web Repository*, permitiram-nos conquistar produtividade e eficiência, ao mesmo tempo que nos proporcionaram uma enorme segurança, respondendo exactamente às nossas necessidades”¹.

O *PowerDesigner* foi construído com base em uma arquitectura flexível, expansível, permitindo virtualmente o suporte para qualquer linguagem de objecto.

Assim, a ferramenta que permitiu a análise á linguagem UML (*Unified Modeling Language*) do projecto foi o *PowerDesigner* 12. Esta ferramenta, possui técnicas de modelagem de padrões de mercado: a modelação de aplicação através da UML (*Unified Modeling Language*), modelagem de processos de negócios e modelagem de dados.

¹ Isabel Ribeiro

O *PowerDesigner* 12 facilita de algum modo, as descrições das aplicações corporativas e os sistemas de informação, através da integração da modelação de dados e dos processos.

O *PowerDesigner* adapta-se aos *frameworks* da arquitectura corporativa. Possui uma nova interface bastante melhorada, que permite aos utilizadores pouco familiarizados com estas novas áreas, possam projectar e modelar processos de negócio de uma forma simples, fácil de usar e muito gráfico.

Esta ferramenta é muito “poderosa” ao nível de suporte para perfis UML, assim os utilizadores podem utilizar o diagrama UML para documentar melhor as estruturas físicas do código que será implementado pelo sistema.

O UML Profile permite técnicas de personalização no padrão UML com a utilização de símbolos personalizados, baseados em estereótipos. O suporte para perfis UML foi reaplicado a todos os modelos, dando as mesmas vantagens à modelação de dados e de negócios - gráficos expressivos para incrementar a análise e o design completo do sistema.

O que nos permite ter uma maior comunicação entre os membros da equipa técnica, de tecnologias de informação de negócios é a tecnologia "*Link and Synch*". Esta tecnologia garante que as mudanças feitas em qualquer visão, negócios objecto ou modelos de dados possam ser comunicadas facilmente às outras visões sem interromper o trabalho em andamento em qualquer modelo.

Esta ferramenta suporta as seguintes técnicas de modelação:

- **Modelação de Objectos:** O *PowerDesigner* gera e reverte a engenharia do código e dos arquivos de implementação em ambientes conhecidos como Java™ J2EE (JSP, Servlet e EJB), NET (C#, VB.NET, ASP.NET e outros), *Web Services* (UDDI e WSDL), XML, C++, *PowerBuilder®* e *Visual Basic®* 6.
- **Modelação Integrada:** As técnicas "*Link and Synch*" e o mapeamento do tempo de design Objecto Relacional do *PowerDesigner* permitem transformações interactivas entre modelos

- **Modelação de Aplicações:** Suporta todos os diagramas de UML e permite o mapeamento avançado de objecto relacional para a gestão de implementações.
- **Modelação de Processos de Negócio:** Os utilizadores pouco familiarizados com estas novas áreas, conseguem de uma forma simples, projectar e modelar processos de negócio
- **Modelação de Dados:** Suporta modelos de dados físicos, lógicos e conceptuais
- **Arquivo Corporativo:** Permite fácil visualização e partilha de modelos e de outras informações

O *PowerDesigner* é uma ferramenta muito útil para a tarefa de modelação de dados, principalmente se o **SGBD** (Sistema Gestor de Base de Dados) utilizado for o **MySQL**, e demonstra estar em um estágio de desenvolvimento estável e avançado, em que os seus recursos funcionam perfeitamente e conta com uma comunidade de desenvolvimento grande e activa que torna a ferramenta “altamente” funcional e produtiva.

Adaptado de: [http://www.lh.com.br/downloads/Folder_SybasePowerDesigner9.5_\(Portugues\).pdf](http://www.lh.com.br/downloads/Folder_SybasePowerDesigner9.5_(Portugues).pdf)

3.1.2. **Microsoft Visual Studio 2005**

O **Microsoft Visual Studio 2005**, foi a ferramenta utilizada para o desenvolvimento da “**Gestão de Imóveis**”.

Esta ferramenta, possibilita o desenvolvimento utilizando várias linguagens de programação, com por exemplo: **Visual C#, Visual C++, Visual Basic e Visual J#**, usando todas elas o mesmo ambiente de desenvolvimento integrado (*IDE*), permitindo assim que troquem ferramentas, o que facilita na criação de soluções de linguagem mistas. Estas linguagens usufruem da funcionalidade do **.NET Framework**, que facilita acesso às tecnologias que simplificam o desenvolvimento de aplicações da **Web ASP e XML Web Services**.

Assim, esta ferramenta permite aos programadores obter melhores resultados e em menos tempo, aumentando assim a produtividade e a qualidade durante o processo de desenvolvimento.

No visual Studio.Net 2005, já não é necessário instalar o IIS. Já possui um servidor interno que é executado automaticamente sempre que compilamos a aplicação.

Esta ferramenta quando clicarmos num novo projecto é nos confortado com novas opções tais com:

- **File-based:** Contém todos os ficheiros do Website estruturados em pastas. Este tipo, usa um servidor web para Asp.Net interno do Visual Studio 2005
- **FTP-based:** É usado quando pretendemos usar FTP para trabalhar com ficheiros de um *WebSite* local ou remoto. Mais frequentemente usado quando precisamos trabalhar com um *WebSite* que está alojado num servidor remoto.
- **Local HTTP-Based:** É usado quando trabalhamos com o IIS da máquina local. Neste caso o *WebSite* é configurado no directório raiz do IIS, ou na pasta virtual que foi determinada.
- **Remote HTTP-Based:** Também usado para gerir *WebSites* em servidores remotos

Para além destas funcionalidades, foi criada a compilação dinâmica:

- **Dynamic Web Site Compilation:** Para nos garantir uma manutenção de código, existe a compilação dinâmica dos *Websites*, onde todos os ficheiros de code-behind do Asp.Net 2.0 usam uma nova funcionalidade de linguagem chamada Partial Class, que permite que todo código seja compilado dinamicamente.

Assim, cada página é compilada quando requisitado pelo utilizador, e quando o código for alterado, a compilação será feita novamente quando o utilizador requisitar novamente a página e assim não é concebido um assembly (DLL) de saída.

- **ViewState: Control and Data State Separation:** No Asp.Net 2.0, usufrui de controlos que têm a capacidade de armazenar o estado do controlo separado dos dados, isto não acontecia em versões anteriores, porque armazenavam o estado dos dados dos controlos juntamente com aparência, quando a propriedade *EnableViewState* era seleccionada com *FALSE*, o controlo perdia a aparência, além dos dados. No *ASP.NET 2.0* quando essa propriedade está seleccionada como *FALSE* os controlos perdem somente os dados, mantendo sua aparência.

Funcionalidades Gerais

- Já não é necessário ter ficheiros de projectos, basta apontar uma pasta e abrir os ficheiros;
- Não é necessário possuir uma única dll por aplicação, pode-se modificar uma única página, compilar e enviar;
- O Código HTML agora é preservado, não sendo mais modificado. (*HTML Source Preservation*)
- *Intellisense* sempre disponível (*Intellisense Everywhere*)
- Pode-se navegar pelas *tags* HTML (*HTML TAG NAVIGATOR*)
- Já trás duas aplicações completas de exemplos.

Adaptado de: <http://www.microsoft.com/brasil/msdn/Tecnologias/aspnet/Whidbey.aspx>

Conteúdo Protegido

3.1.3. *Microsoft MySQL*

Para a implantação da base de dados nesta aplicação foi usado *EMS SQL Manager 2005 for MySQL*, por ser um base de dados *Open Source* e porque possui estabilidade de alta performance, uma grande fiabilidade e é fácil de usar. Actualmente é usado em mais de 6 milhões de instalações em todos os continentes, que vão desde instalações em grandes corporações a específicas aplicações. Além disso, o MySQL tornou-se a escolha de uma nova geração de aplicações, que utilizam o modelo LAMP (*Linux, Apache, MySQL, PHP*).

- **Custo:** o custo total do MySQL é limitado ao custo do suporte técnico, e está disponível para download e é uso livre.
- **Suporte:** existe uma grande comunidade contribuindo na Internet
- **Compatibilidade:** MySQL pode-se conectar facilmente com a maioria das aplicações *SQL-compliance* com ODBC.

Características da linguagem Mysql:

- Portabilidade (suporta praticamente qualquer plataforma actual)
- Compatibilidade (existem drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, Python, Perl, PHP e Ruby)
- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de hardware;
- Facilidade de uso;
- É um Software Livre;
- Suporte a vários tipos de tabelas (como MyISAM, InnoDB e Maria), cada um específico para um fim;

- Faltam alguns recursos quando comparados como outras base de dados, como o PostgreSQL.

Adptado de :<http://pt.wikipedia.org/wiki/MySQL>

Conteúdo Protegido

3.2. Linguagens de Programação

3.2.1. ASP.Net 2.0 com VB.Net

O *ASP.NET* é a derradeira geração do *ASP*, e tem a finalidade de poder ser usada para criar aplicações *WEB* dinâmicas, através de um conjunto de classes e controlos que simplificam a vida dos programadores.

A forma de compilarmos a aplicação no *ASP.NET* é uma das mudanças mais valorizadas em relação às versões anteriores. Através do *Visual Studio .NET* durante a compilação da versão 1.x do *ASP.NET*, era gerado um ficheiro do tipo *DLL* dentro do directório `\bin` na raiz da aplicação, que correspondia a todo o código fonte da aplicação em questão.

Nas versões anteriores do *ASP.NET 2.0* tínhamos o conceito de *Code-Behind* que veio para tornar o *ASP.NET* muito mais elegante que o *ASP* clássico. Essa funcionalidade consiste em termos dois ficheiros para cada página do site, ou seja, uma página *ASPX* que contém somente códigos *HTML* e uma página **.cs* (*C#*) ou **.vb* (*VB.NET*) (dependendo da linguagem escolhida para a aplicação), que é o *CodeBehind*. As páginas de *CodeBehind* herdam directamente da classe base *System.Web.UI.Page* e o arquivo *ASPX*, indica qual o arquivo de *CodeBehind* que corresponde ao código e, em tempo de execução, o *ASP.NET* faz essa herança automaticamente.

Foi usada a plataforma *Framework 2.0* para a elaboração da aplicação, por ser a plataforma que simplifica o desenvolvimento de aplicações para a Internet e tem como principais objectivos:

- Fornecer uma programação orientada a objectos e consistente, onde o código é guardado e executado localmente, mas, ao mesmo tempo possível de distribuir pela Internet ou de ser executado remotamente;
- Fornecer ambiente de código executável que minimize a distribuição física de *software*;
- Fornecer execução de código que elimine problemas de performance de *scripts* ou ambientes interpretados;

Adaptado de: http://www.microsoft.com/brasil/msdn/Tecnologias/aspnet/Pre_Compilacao_ASPNET.msp

Capítulo 4

4. Implementação

4.1. Base de Dados

Foi através do diagrama de classe que foi gerado o script de criação de tabelas e respectivas relações entre elas. Deste modo, foi alcançado o total de vinte e três tabelas, que estão representadas com os respectivos atributos, chaves primárias e chaves estrangeiras, e com as suas respectivas ligações, e com podemos visualizar na seguinte estrutura de base de dados seguinte.

Conteúdo Protegido

Figura 16 – Estrutura de Base de Dados

Conteúdo Protegido

4.2. Código MySQL

Foi elaborado um procedimento na base de dados para melhor desempenho da aplicação. Um procedimento (*Stored procedures*), é um conjunto de comandos SQL que podem ser armazenados no servidor. Uma vez que isto tenha sido feito, os clientes não precisam de reenviar os comandos individuais mas podem fazer referência aos procedimentos.

Este procedimento pode fornecer um aumento no desempenho já que não é enviada tanta informação servidor e o cliente. Contém um lado negativo que é o facto de aumentar o “peso” no servidor da Base de Dados, já que a maior parte do trabalho é feita no servidor e menor parte é feita do lado do cliente.

Existem situações em que devemos utilizar procedimentos:

- Quando várias aplicações clientes são escritas em diferentes linguagens ou funcionam em diferentes plataformas, mas precisam realizar as mesmas operações de base de dados.
- Quando a segurança é prioritária. Temos os casos dos bancos, utilizam todos eles, procedimentos para todas as operações comuns, porque fornece um ambiente consistente e seguro, e assim asseguram que cada operação é sempre registada.

Os procedimentos possuem três características muito importantes: Segurança, Desempenho e facilita a gestão da base de dados.

- **Segurança** – A segurança dos dados também é mais forte, pois podemos definir diferentes tipos de atributos para cada utilizador.
- **Desempenho** – Como os procedimentos são analisados e armazenados na memória do servidor da base de dados, depois de terem sido executados a primeira vez, a sua execução posterior é mais rápida.

- **Facilita a Gestão da Base de Dados** – Quando se pretender alterar uma tarefa, basta só alterar o procedimento correspondente a essa tarefa, melhorando assim a eficácia da base de dados.

O procedimento que foi elaborado nesta aplicação, foi um procedimento para a inserção dos arrendatários na base de dados, chamado de ***ProInserirarrendatario()***, que tem os seguintes parâmetros de entrada:

- Nome
- Apelido
- Morada
- Email
- Data_nascimento
- Telefone
- Estado_Civil
- Num_contribuinte
- Login
- Pass
- Data_entrada
- Data_saida
- Nome_pai
- Nome_mae
- Morada_pais
- Telefone_pais
- TipoUtil

Este procedimento é usado para inserir um novo registo de arrendatários, na tabela arrendatários.

4.3. Serviços disponíveis

Nesta aplicação foi implementado um serviço de SMS, para que o administrador possa contactar os arrendatários, sempre que ache necessário. Também se encontra disponível este serviço para os arrendatários, para que entre em contacto com o administrador.

Através do serviços disponibilizado pela <http://www.voipcheap.com> foi possível implementar este serviço SMS. Eles disponibilizam um código HTML que é implementado na página.

```
https://myaccount.VoipCheap.com/clx/sendsms.php?username=xxxxxxxxx&password=xxxxxxxxx&from=xxxxxxxxx&to=xxxxxxxxx&text=xxxxxxxxx
```

Para isso temos que nos registar no voipcheap para termos um username e uma password que vai ser colocada no código disponibilizado.

Nesta aplicação também se encontra implementado um serviço de correio electrónico para que o publico em geral ou até mesmo os arrendatários possam entrar em contacto com o administrador do sítio.

Este serviço funciona por SMTP, e para isso foi configurado no Web.config o mailSttings.

```
<mailSettings>
  <smtp from="geral@gestimovel.pt">
    <network host="smtp.gmail.com" port="25" userName="Admi"
password="Admin"/>
  </smtp>
</mailSettings>
```

4.4. Ligação à Base de Dados

Ao nível da programação da ligação à base de dados foi necessário fazer alguns importes para a página *Web.config*, para que sejam validados todas as páginas do projecto. Esse importe foram colocado no Namespaces para se por fazer a ligação á Base de Dados.

Neste caso tiveram que se fazer dois importe: Um por ODBC e o outro por *MySQLClient*. A ligação ao ODBC, é necessária para a página que não utiliza procedimentos, a ligação *MySQL* é usada para as páginas que chamam o procedimento, neste caso, quando registamos arrendatários e administradores.

```
<add namespace="System.data.Odbc" />  
<add namespace="MySQL.Data.MySqlClient" />
```

Depois dos importes foi adicionado também no *Web.config*, uma *connectionStrig* que faz a ligação á Base de Dados, também aqui por duas formas, como podemos verificar no código seguinte.

```
<connectionStrings>  
<add name="ConnectionString" connectionString="Server=localhost;User  
id=root;Password=mendes;database=gestimov" />  
  
<add name="gestimov" connectionString="DRIVER={MySQL ODBC 3.51  
Driver};SERVER=localhost;PORT=3306;DATABASE=gestimov;USER=root;PASSWORD  
=mendes;OPTION=3"  
providerName="System.Data.Odbc" />  
</connectionStrings>
```

Para abrir a ligação á base de dados pode-se fazer de dois modos como podemos verificar seguidamente:

```
Dim con2 As New
MySQLConnection(ConfigurationManager.ConnectionStrings("ConnectionStrin
g").ConnectionString)
Dim cmdSelect = New MySqlCommand(sql, con2)
con2.Open()

-----

Dim conn As New
OdbcConnection(ConfigurationManager.ConnectionStrings("gestimov").Conne
ctionString)
conn.Open()
Dim cmd As New OdbcCommand(sql, conn)
```

4.5. Segurança

A segurança hoje em dia na criação de páginas para a internet é um factor bastante importante, para isso é necessário fazer uma abordagem segura a nível de acesso para a Gestão de Imóveis. Nesta aplicação, está embutida um nível de segurança com alguma credibilidade, ao acesso aos dados e à autenticação dos arrendatários.

O *ASP.NET* possui inúmeras funcionalidades bastante abrangentes, com vários conceitos e técnicas, é muito simples interagir com elas, configurando ou até criando seus próprios recursos. Para isso, usamos o *Web.config*.

O *Web.config* é uma orientação para determinada aplicação *ASP.NET* que pode ser configurada. A extensão *.config* é apenas uma nomenclatura, este é apenas um arquivo XML que pode ser alterado em qualquer editor de textos.

Assim quando criamos um projecto novo está logo configurado na página *Web.config* na secção *namespace*, alguns dos *namespace* já embutidos como se verifica no quadro seguinte:

```

<namespaces>
  <clear />
  <add namespace="System" />
  <add namespace="System.Collections" />
  <add namespace="System.Collections.Specialized" />
  <add namespace="System.Configuration" />
  <add namespace="System.Text" />
  <add namespace="System.Text.RegularExpressions" />
  <add namespace="System.Web" />
  <add namespace="System.Web.Caching" />
  <add namespace="System.Web.SessionState" />
  <add namespace="System.Web.Security" />
  <add namespace="System.Web.Profile" />
  <add namespace="System.Web.UI" />
  <add namespace="System.Web.UI.WebControls" />
  <add namespace="System.Web.UI.WebControls.WebParts" />
  <add namespace="System.Web.UI.HtmlControls" />
</namespaces>

```

O *System.Web.Security* é uma classe que é utilizada para implementar segurança em aplicações Web ASP.NET.

Nesta aplicação está embutido em sistema de autenticação para o arrendatário registado e para os administradores. Assim quando se regista um arrendatário ou administrador, este vai inserir uma *password* que vai ficar na base de dados contudo vai ficar encriptada com mostra os código seguinte:

```

.Add(NewSqlParameter( "?APass" ,
FormsAuthentication.HashPasswordForStoringInConfigFile(pass1.Text ,
"SHA1" )))

```

Depois, quando o arrendatário ou o administrador faz o login na página, vai chamar a função *VerificarLogin* e ver se o arrendatário ou o administrador existe. Caso exista vai comparar a password que é introduzida no login com a password que se encontra guardada na base de dados. O código seguinte ilustra com compara a password:

```

DimPassEncriptadaAsString=
FormsAuthentication.HashPasswordForStoringInConfigFile(Pass, "SHA1")

```

Depois de se autenticar, se for o caso de um arrendatário, este só fica possibilitado de aceder a alguma página que lhe é permitida. No caso de se autenticar um administrador este fica possibilitado a aceder a todo o site, á excepção dos conteúdos dos arrendatários.

Caso seja um utilizador normal, este só pode aceder às páginas que estão disponíveis para o público em geral. Assim para isto não acontecer, é usado um código que vai validar que tipo de utilizador é ou seja, verifica o tipo de permissão que esse utilizador tem. Foi criada uma função chamada ValidarPerfil que devolve um valor booleano (0 ou 1), que nos indica se é um administrador (1) ou se é um arrendatário (0).

No *Page_Load* foi colocado o seguinte código, só nas pastas da área de administração.

```
Dim pag As String = "Admin"
Dim permissoes As String =
System.Configuration.ConfigurationManager.AppSettings(pag)
    Dim CGP As New Classperfil
    If Not CGP.ValidarPerfil(permissoes,
Session("tipo_util").ToString)
        Then
            Response.Redirect("~/NoAccess.aspx")
        End If
```

4.6. Controlos

O **Visual Studio.net 2005** apresenta-nos com alguns tipos controlos.

Este tipos de controlo são os mais comuns, **Standard, Data, Validation, Navigation, Logins, WebParts, Html e Crystal Reports**.



Nesta aplicação foram utilizados alguns controlos: **Standard, Data, Validation**.

Os controlos **Standard** usados.

- **Label**
- **TextBox;**
- **Button;**
- **LinkButton;**
- **ImageButton**
- **HyperLink;**
- **DropDownList;**
- **CheckBox;**
- **Image**
- **MultiView;**
- **Panel;**
- **View.**

Os controlos **Data** usados:

- **GridView** - Permite exibir os dados.
- **SqlDataSource** - O controlo *SqlDataSource* permite aceder a dados localizados numa base de dados.

Os controlos **Validation** usados:

- **RequiredFieldValidator**
- **RangeValidator**
- **RegularExpressionValidator**
- **CompareValidator**
- **CustomValidator**

Conteúdo Protegido

Capítulo 5

5. Conclusão

5.1. Conclusão

Com a realização deste projecto “Gestão de Imóveis”, pôs-se em prática alguns dos conhecimentos adquiridos ao longo da minha vida estudantil, sendo um factor importante no aprofundamento e avaliação dos meus conhecimentos.

Esta aplicação intitulada por “**Gestão de Imóveis**” tem como finalidade disponibilizar aos utilizadores de uma forma interactiva e clara um conjunto de informação necessárias, com meios de segurança necessários para uma utilização correcta. Visto funcionar na *Web*, tem que possuir alguma credibilidade para que os utilizadores se sintam mais seguros e convictos de uma boa solução para os seus problemas.

A aplicação proposta é uma aplicação construída numa tecnologia Web “**ASP.NET2.0**”, bastando para isso o utilizador aceder a um browser de qualquer computador e desfrutar de todas as funcionalidades que a aplicação apresenta. Esta aplicação possui um interface de fácil manuseamento e bastante dinâmico.

Para a elaboração desta aplicação foi utilizado a ferramenta **Microsoft Visual Studio 2005** por ser uma ferramenta muito poderosa que possui muitas funcionalidades. Como foi referido anteriormente a base de dados utilizada foi **MySQL**.

Finalizado este projecto, apercebi-me que minha formação ficou mais enriquecida e bastante mais fortalecida e que ao inicio tive algumas dificuldades que não passavam de pequenos contratemplos. Cheguei ao final de projecto com grande entusiasmo e verifiquei que é uma área bastante interessante.

Comparando o resultado final com os objectivos propostos, penso que estes foram alcançados de forma bastante satisfatórios, dando espaço à implementação de novas funcionalidades tais como: Serviços de SMS, Envio de E-mails, adicionar fotografias a cada quarto.

5.2. *Perspectivas Futuras*

Como perspectivas futuras, poderá adaptar esta aplicação a arrendar imóveis completos em vez de só arrendar por quartos.

Poder-se-á adaptar também a base de dados para o suporte a uma galeria de fotos dos quartos para que o utilizador tenha mais informação visual sobre o quarto.

Elaborar um contrato automaticamente, a partir da reunião de dados já existentes na aplicação de forma a facilitar todo o processo de arrendamento.

Conteúdo Protegido

Bibliografia

Publicações:

SILVA, Alberto e Videira – UML, Metodologias e Ferramentas CASE, Edições Centro Atlântico, Abril 2001;

VIEIRA, João – Programação com ASP.NET; Volume I, FCA – Editora de Informática Lda., Outubro 2002;

VIEIRA, João – Programação com ASP.NET; Volume II, FCA – Editora de Informática Lda., Julho 2003;

NUNES, Mauro; O'Neill, Henrique – Fundamental de UML, 4ª Edição Actualizada e Aumentada, FCA – Editora de Informática Lda, Dezembro 2004;

ABREU, Luís – ASP.NET 2.0, Curso Completo, FCA – Editora de Informática Lda., Lisboa Janeiro 2006;

Conteúdo Protegido

Sites Visitados:

ASP.NET <http://www.asp.net>, última consulta Abril 2008;

Microsoft, <http://www.microsoft.com/>, última consulta Abril 2008.

MSDN asp.net; <http://www.msdn.microsoft.com/asp.net>, última consulta Abril 2008;

The Code Project; <http://www.codeproject.com>, última consulta Abril 2008;

Masters, <http://imasters.uol.com.br/>, última consulta Abril 2008.

Sybase, <http://www.sybase.pt/>, última consulta Abril 2008

Conteúdo Protegido

Glossário

E-Mail – Sistema que permite um utilizador trocar mensagens com outro através de uma rede de comunicação.

HTML (*Hipertext Markup Language*) – é uma linguagem de marcação utilizada para produzir páginas na Web.

Internet – Colecção de redes que usam o conjunto de protocolos TCP/IP desenhado para o uso do público em geral.

Login — O *login* é o nome que o utilizador utiliza para aceder ao servidor de uma rede.

Password – Característica de segurança usada para validação de entrada numa rede.

UML (*Unified Modeling Language*) – é uma linguagem para especificação, documentação, visualização e desenvolvimento de sistemas orientados a objectos.

Conteúdo Protegido

Anexos

Conteúdo protegido

Anexo 1 - Manual de Utilizador

O Manual do Utilizador encontra-se em suporte CD. Este manual exemplifica o funcionamento da imobiliária “Gestão de Imóveis”. Aqui o utilizador pode visualizar e ter uma maior noção do funcionamento da aplicação.

Conteúdo Protegido

Anexo 2 - Stored Procedures

Stored Procedures “Inserirarrendatario”

```
CREATE PROCEDURE `Inserirarrendatario`(IN Anome VARCHAR(100), IN Aapelido
VARCHAR(100), IN Amorada VARCHAR(200), IN Aemail VARCHAR(100), IN
Adata_nascimento DATE, IN Atelefone VARCHAR(50), IN Aestado_civil
VARCHAR(100), IN ANum_contribuinte VARCHAR(100), IN ALogin VARCHAR(25), IN
APass VARCHAR(100), IN AData_entrada DATE, IN ANome_pai VARCHAR(100), IN
ANome_mae VARCHAR(100), IN AMorada_pais VARCHAR(200), IN ATelefone_pais
VARCHAR(100), IN ATipoUtil BOOLEAN, IN AActivo BOOLEAN)
    NOT DETERMINISTIC
    SQL SECURITY DEFINER
    COMMENT ''
BEGIN
INSERT INTO arrendatarios (Nome, apelido, morada, email, data_nascimento,
telefone, estado_civil, Num_contribuinte, Login, Pass, Data_entrada, Nome_pai,
Nome_mae, Morada_pais, Telefone_pais, TipoUtil,Activo)
VALUES (Anome, Aapelido, Amorada, Aemail, Adata_nascimento, Atelefone,
Aestado_civil, ANum_contribuinte, ALogin, APass, AData_entrada, ANome_pai,
ANome_mae, AMorada_pais, ATelefone_pais, ATipoUtil, AActivo);
END;
```