



Licenciatura em Informática de Gestão

Relatório Final do Projecto de Fim de Curso (PFC)

Office Facility Manager

Pedro Machado e Cunha Sousa e Costa (nº 20063152)

Pedro Jorge de Almeida Crespo (nº 20063674)

3N1

2008/2009

Professor Doutor Alexandre Pereira

Lisboa, 31 de Julho de 2009

Universidade Lusófona de Humanidades e Tecnologias

Licenciatura em Informática de Gestão

Relatório Final do Projecto de Fim de Curso (PFC)

Office Facility Manager

Pedro Machado e Cunha Sousa e Costa (nº 20063152)

Pedro Jorge de Almeida Crespo (nº 20063674)

3N1

2008/2009

Professor Doutor Alexandre Pereira

Lisboa, 31 de Julho de 2009

Abstract

*In this report it is defined the development and the use of the application **Office Facility Manager (OFM)**. This application was designed to help the management of office facilities, such as the aspects of an office daily basis management: schedules and reserve company resources(room and equipments) and facilitate entrances of visitors. The application it is categorized mainly to be an interface easy to access from any computer within a simple Internet connection (or local network connection), since it is designed as a web interface. This application can be use by a single organisation, for internal use, or shared in a multi-tenant system through a SaaS (software as a Service) subscription.*

Agradecimentos

Um projecto na área de informática é um projecto muito especial, pois a tarefa gigantesca que o profissional de informática leva a cabo nunca é um acto isolado; depende essencialmente da informação que outros possuem e da sua habilidade em trabalhar em equipa, chegando a dominar dois universos e linguagens diferentes.

Por tal, a lista de agradecimentos é inevitável. Queremos agradecer a todos os intervenientes neste processo e àqueles que nos motivaram e acreditaram nas nossas qualificações profissionais:

À direcção da licenciatura: *Professor Doutor Manuel da Costa Leite;*

Ao professor responsável pelo acompanhamento da cadeira de projecto:
Professor Doutor Alexandre Pereira;

Às esposas: *Raquel Machado e Cunha e Maria Natália João Silva;*

À filha recém-chegada: *Jéssica Cunha, pelas noites que deixou o pai dormir*

Índice Geral

Abstract	i
Agradecimentos	ii
1 Introdução	7
1.1 Organização do documento.....	7
2 Perfil do projecto	9
2.1 Identificação do projecto.....	9
2.2 Objectivo do projecto.....	9
2.3 Características Gerais	9
3 Análise funcional	11
3.1 Gestão de utilizadores e colaboradores	11
3.2 Controlo de cartões	12
3.3 Controlo de visitantes.....	13
3.4 Controlo de Recursos (salas / equipamentos).....	15
3.5 Recepção de encomendas e correspondência.....	16
4 Conclusão	18
5 Bibliografia	20
Anexo A - Especificação de requisitos do sistema	23
Anexo B - Descrição de desenho do sistema	25
Anexo C - Testes	43
Anexo D - Plano de instalação	45
Anexo E - Manual de utilizador	46
Anexo F - Manual técnico	49

Índice de Figuras

Figura 1: OFM – Use Case – Super Admin.....	25
Figura 2: OFM – Use Case – Admin	26
Figura 3: OFM – Use Case – User.....	26
Figura 4: OFM – Use Case – Porteiro	27
DAL - Colaborador	28
DAL – Utilizador.....	29
DAL - Role.....	30
DAL – Condominio.....	30
DAL - Empresa	31
DAL - Recurso	31
DAL - Reserva.....	32
DAL - Visitante.....	33
UI – Página Master.....	34
UI – Página Inicial	34
UI - Condomínios	35
UI - Empresas	36
UI - Utilizadores.....	37
UI - Visitantes.....	39
UI - Reservas	40
UI - Recursos	41
Modelo de Dados	42
Figura 5: OFM - Área de acesso.	46
Figura 6: OFM - Senha inválida.....	47
Figura 7: OFM – Funções por áreas reservadas.	48

Glossário

ASP

É um acrónimo para *Active Server Pages*, uma estrutura de *scripts* para gerar páginas dinâmicas na *web*.

ASP.NET

É a plataforma da *Microsoft* para o desenvolvimento de aplicações *web* e é o sucessor da tecnologia ASP.

HTML

A sigla HTML deriva da expressão inglesa *HyperText Markup Language*. Trata-se de uma linguagem utilizada para produzir páginas na *Internet*.

OFM

Office Facility Manager.

Scripts

Scripts são arquivos contendo linguagem interpretada; são normalmente chamados de arquivos de *script* ou *scripts* em vez de executáveis.

Os *Scripts* precisam de *softwares* interpretadores porque são blocos de códigos não compilados, que são interpretados no tempo de execução.

SGBD

Sistema de Gestão de Base de Dados.

Software

Aplicação de computador.

SQL

É um acrónimo para *Structured Query Language*, ou em português Linguagem de Consulta Estruturada. É uma linguagem de pesquisa declarativa para base de dados relacionais. Muitas das características originais do SQL foram inspiradas na álgebra relacional.

ULHT

Universidade Lusófona de Humanidades e Tecnologias.

UML

É um acrónimo para *Unified Modelling Language*. É uma linguagem de modelação e de especificação usada na engenharia da programação.

VB

É um acrónimo para *Visual Basic*. O *Visual Basic* é uma linguagem de programação da autoria da empresa *Microsoft* e que faz parte integrante do *Microsoft Visual Studio*.

1 Introdução

Este documento descreve as diversas fases e os conceitos fundamentais da aplicação Office Facility Manager (OFM), assim como características técnicas e de utilização. Este capítulo introdutório tem por objectivo descrever a organização do documento.

1.1 Organização do documento

Este relatório está dividido em cinco capítulos e sete anexos. Cada capítulo descreve um tópico diferente. Foram também utilizados anexos referentes a documentação técnica que normalmente são entregues em separado, mas dado o presente relatório ser do âmbito académico e por uma questão de simplicidade, essa mesma documentação foi anexada. Os seguintes parágrafos oferecem uma descrição sucinta de cada capítulo:

- Primeiro capítulo – Introdução. Este capítulo descreve a organização do documento.
- Segundo capítulo – Perfil do projecto. Este capítulo descreve características, objectivos e módulos inerentes ao projecto.
- Terceiro capítulo – Análise funcional. Este capítulo descreve a topologia e funcionalidades do sistema.
- Quarto capítulo – Conclusão. Este capítulo apresenta a conclusão ao projecto.
- Quinto capítulo – Bibliografia. Este capítulo descreve documentação consultada durante o desenvolvimento da aplicação e da escrita do relatório final.
- Anexo A – Especificação de requisitos do sistema. Este anexo descreve os requisitos relacionados com a fase de levantamento de conhecimento.

- Anexo B – Descrição de desenho do sistema. Este anexo descreve a análise do modelo através da tecnologia UML.
- Anexo C – Descrição do modelo de dados. Este anexo descreve o desenho da arquitectura de base de dados e modelo de dados, necessários à implementação da aplicação.
- Anexo D – Testes. Este capítulo descreve os testes efectuados à aplicação.
- Anexo E – Plano de instalação. Este anexo descreve os procedimentos necessários para instalar e a aplicação.
- Anexo F – Manual de utilizador. Este anexo descreve a aplicação do ponto de vista do utilizador.
- Anexo G – Manual técnico. Este anexo contém a descrição das classes e métodos.

2 Perfil do projecto

2.1 Identificação do projecto

O projecto é identificado como «Office Facility Manager».

2.2 Objectivo do projecto

É objectivo deste projecto obter um sistema que permita facilitar a gestão de instalações de uma organização, especificamente a recepção, sob os seguintes aspectos:

- Registo de visitas com sistema de notificações;
 - Reserva de salas;
 - Reserva de equipamentos.
-

2.3 Características Gerais

Solução preparada para correr sobre redes com grande dispersão geográfica em termos de utilizadores finais e localização dos servidores de base de dados. Para o efeito, pretende-se que esta solução seja Web-Based e possa ser implementada como Software As A Service (SaaS).

Versão multiposto de operação e com suporte para múltiplos sites não integrados, ou seja, possibilidade de instalar outras instâncias da mesma aplicação de forma separada (mesmo servidor ou servidores distintos).

Multiposto e multiutilizador com permissões. Deverá estar previsto módulo de administração de utilizadores.

A interface dos utilizadores dispõe de todos os módulos da aplicação no ecrã, embora em tabs (ou janelas) distintas. Os módulos terão controlo de acesso

conforme as permissões dos utilizadores. Não se encontra previsto controlo a nível dos campos.

Módulos previstos: Gestão de Instituições (i.e. Condomínios e Empresas), Gestão de Utilizadores e Colaboradores, Controlo de Visitantes, Controlo de Recursos da Empresa (Salas e Equipamentos).

Por Utilizadores da aplicação consideramos os seguintes perfis:

- Super Administrador – Responsável pela gestão do sistema.
- Administrador – Responsável por sistema de uma conta/instituição.
- Utilizador – Funcionário de uma instituição que poderá receber visitas.
- Porteiro – Funcionário que se encontra na recepção.

3 Análise funcional

A Análise Funcional foi elaborada no início do projecto, após levantamento de requisitos para um sistema de gestão de instalações. A definição dos requisitos originou a elaboração de um primeiro documento, o qual se encontra replicado no presente tópico.

O desenvolvimento do projecto foi realizado de forma modular e interactiva. Cada módulo responde às funcionalidades base da aplicação. Durante o processo de desenvolvimento interactivo procedeu-se à redefinição de requisitos e consequentemente à reformulação desses mesmos módulos.

Foi decidido preservar a Análise Funcional inicial e identificar os módulos retirados/modificados, como exercício académico de Gestão de Mudança. Espelhamos assim, uma situação real num Mundo Real.

3.1 Gestão de utilizadores e colaboradores

Este módulo permite a gestão de Utilizadores (staff técnico) e Colaboradores (end-users). Entendemos pelos primeiros pessoas que podem utilizar a aplicação e/ou administrá-la. Pelos segundos pessoas que pertençam ou não aos quadros da Entidade e que beneficiem dos serviços da mesma.

Cada utilizador deve ter indicado:

- Número único (ID).
- Login
- Password
- Política de segurança (por módulo / ecrã).
- Estado (activo/desactivo).

Cada colaborador deve ter indicado:

- Número único (ID).
- Nome.
- Funcao.
- Telefone.
- Email.
- N° Identificação da Empresa

Alteração de Requisitos: os dois perfis iniciais foram desdobrados em mais dois:

- Utilizador ⇔ Super Administrador e Administrador;
- Colaborador ⇔ Utilizador e Porteiro.

3.2 Controlo de cartões

Este módulo servirá para o controle de empréstimo de cartões transmissíveis (com e sem chip). A emissão e impressão e destes cartões não faz parte do âmbito desta aplicação. Cada cartão deverá:

- Ter um número único.
- Ter data e hora de validade.
- Status: activo, inactivo, expirado, perdido ou cancelado.
- Tipo de cartão: acesso e/ou porta-moedas.
- Cartão exclusivo (e.g. reservado empresa de limpezas).
- Modalidade (e.g. validade diária, provisória, permanente).

O módulo deve guardar histórico por “Número Único” do cartão:

- Data e hora da requisição.
- Data e hora de validade (ver Modalidade).
- Data e hora da devolução do cartão.

- Empresa do visitante.
- Nome do visitante.
- Pessoa responsável (da Entidade).
- Departamento da pessoa responsável (da Entidade).

Relatórios entre datas:

- Histórico de cartão
- Por departamento
- Por utilizador

Relatórios/listagem de cartões:

- Por tipo.
- Por status.

Alteração de Requisitos: após análise mais aprofundada ao Mercado Português, constatou-se que a gestão de cartões implicaria operacionalizar e testar equipamentos. Como trabalho académico e sem verbas associadas, este módulo foi retirado, mas poderá ser incluso num projecto de iniciativa empresarial.

3.3 Controlo de visitantes

Este módulo servirá para o controle de visitantes (para visitantes pontuais, se ficarem mais que um dia, é habito receberem um cartão sem chip). Serão impressos, no momento, cartões intransmissíveis em papel que serão colocados em bolsas plásticas.

- Ter um número único.
- Ter data e hora de validade.
- Status: activo, inactivo, expirado, perdido ou cancelado.

O módulo deve registar:

- Data e hora da requisição.
- Data e hora da devolução do cartão.
- Empresa do visitante.
- Nome do visitante.
- N° Documento Identificação; Validade e Local de Emissão
- Pessoa responsável (da Entidade).
- Departamento da pessoa responsável (da Entidade).
- Número de acompanhantes.
- Observação (apenas no sistema, não impresso).

Relatórios entre datas:

- Histórico de cartão
- Por departamento
- Por utilizador

Relatórios/listagem de cartões:

- Por status.

Alteração de Requisitos: acrescentou-se a funcionalidade do próprio Utilizador reservar a sala e equipamentos, bem como notificar o Porteiro que irá decorrer uma reunião em determinado período e sala. Desta forma, o Porteiro ao recepcionar o visitante terá uma informação prévia que a reunião está prevista e possa assim encaminhar de melhor forma o visitante para a sala reservada.

3.4 Controlo de Recursos (salas / equipamentos)

Este módulo permite efectuar a gestão das reservas de recursos da empresa (salas, equipamentos). Monitorização do estado dos recursos, reservas e desmarcação de recursos. Os recursos têm de ser configuráveis, i.e., as suas características devem identificadas:

- Número identificador.
- Tipo de Recurso
- Quantidade de pessoas que pode acomodar
- Localização (edifício, piso).

O módulo deve registar:

- Recurso a reservar
- Nome e departamento da pessoa que reserva
- Nome e departamento da pessoa para quem a sala é reservada
- Data / Hora de reserva
- Data / Hora de início e fim da reserva
- Numero de pessoas
- Observações (equipamento adicional a prever; não impresso).

Devem ser previstas as seguintes funcionalidades:

- Ecrã de marcação/desmarcação de recursos (meias horas).
- Ecrã de salas por datas. Reservadas e por reservar.
- Envio de emails sempre que se marca/altera uma sala (vide agenda Outlook).
- Impressão de folha A4 para colocar na entrada da sala.

Relatórios entre datas

- Total reservas por sala
- Reservas por departamento
- Reservas por utilizador
- Estatística - % ocupação

3.5 Recepção de encomendas e correspondência

Este módulo serve para o registo e controle da correspondência e encomendas entregues.

- Toda a correspondência é registrada e distribuída.
- Toda a encomenda é registrada. Encomendas de maior dimensão podem ser encaminhadas para outra área. Encomendas de pequeno porte ficam na recepção. Deve ser emitido um e-mail ao “receptor” da encomenda, pois esta não será distribuída, mas sim recolhida pelo destinatário.

A aplicação deve registar:

- Número de identificação.
- Tipo: correspondência/encomenda.
- Nome e departamento da pessoa para quem se destina a correspondência.
- Nome e empresa de quem enviou.
- Nome e empresa de quem entregou.
- Se vai para a outra divisão.
- Data hora de chegada.
- Data hora de entrega.
- Quem recebeu.
- Numero de “pacotes”.
- Notas gerais (observações).

Devem ser previstas as seguintes funcionalidades:

- Ecrã de registo de recebimento; ecrã entrega de encomenda.
- A aplicação deve gerar (em papel autocolante) um número / código de barras para se poder efectuar o tracking da mesma.

Relatórios entre datas:

- Por departamento.
- Por utilizador.
- Quantidades.
- Encomendas por levantar.

Alteração de Requisitos: após análise de mercado concluiu-se que o registo de correspondência/encomendas seria de tal forma elevado que implicava um grande investimento de tempo e por isso de Recursos Humanos. Poderá ser um requisito pontual de algumas instituições, mas à priori não será uma funcionalidade reconhecida como essencial.

4 Conclusão

O projecto Office Facilites foi realizado por um grupo constituído por 2 (dois) alunos do curso da *Licenciatura de Informática de Gestão*, fruto da boa relação entre estes.

A ideia de desenvolver esta aplicação nasceu da observação do Mundo Real, onde um dos elementos de grupo, teve por uma das principais competências profissionais a função de *Business Development Manager*.

Nas suas visitas a clientes observou a necessidade de haver uma aplicação no Mercado Português que pudesse responder às necessidades básicas da recepção de uma organização. O registo de visitantes é normalmente efectuado em papel; a gestão de espaços para reuniões não se encontra reservado; a correspondência e encomendas têm pouco acompanhamento.

Tendo em consideração que nas médias e grandes empresas, e entidades públicas, a gestão da recepção é feita por empresas de segurança, rapidamente se constatou a necessidade de extrapolar para um sistema dito *multi-tenant*.

O desenvolvimento da aplicação foi realizado da forma mais profissional possível, dentro do contexto educacional, *i.e.*, as primeiras fases de levantamento de conhecimento, reuniões e análise de requisitos, tiveram sempre o acompanhamento pedagógico e não foram feitas de forma isolada. A fase posterior de desenvolvimento da aplicação foi feita de forma individual, dado o âmbito e avaliação da cadeira de projecto.

Embora o objectivo primário do projecto seja a demonstração e aplicação de conhecimentos dos alunos, o projecto encontra-se desenvolvido e testado com

cuidado, para que possa funcionar plenamente de acordo com o especificado, num contexto de uso profissional.

A aplicação aparenta ter um conjunto exíguo de funcionalidades, mas estas foram analisadas, pensadas e revistas inúmeras vezes para que o resultado final, quer a nível de utilização quer a nível de resultados, fosse o mais simples e directo possível. Assim, as funcionalidades desenvolvidas são garantidamente usadas na sua plenitude. É de notar que também houve um trabalho cuidadoso na interface com o utilizador, nas cores, na disposição dos elementos e nas descrições, para uma utilização amigável.

Como pontos de melhoria à aplicação registam-se aqui algumas ideias:

- Gestão de equipamentos de controlo de acesso (e.g. torniquetes);
- Gestão de equipamentos e cartões de acesso (e.g. cartões de proximidade);
- Gestão de assiduidade;
- Gestão de encomendas e/ou correspondência.

Como resultado final, tem-se uma aplicação que à primeira vista parece muito simples. No entanto, é na simplicidade que está a beleza, e por detrás da simplicidade esconde-se um trabalho complexo.

5 Bibliografia

A enciclopédia livre, «Wikipédia», <http://pt.wikipedia.org>

Alexander, Ian F.; Stevens, Richard, «*Writing Better Requirements*», Addison-Wesley, 2002

Damas, Luís, «*SQL – Structure Query Language*», FCA, Setembro 1999

Dicionário de Inglês *on-line*, «*Dictionary.com*», <http://dictionary.reference.com>

Dicionário de Português *on-line*, «*Priberam Informática*», <http://www.priberam.pt>

Gunderloy, Mike, «*Developing and Implementing Web Applications with Visual Basic® .NET and Visual Studio® .NET*», QUE *Second printing with corrections*, November 2003

Nunes, Luís Miguel, «*Regras para elaboração de relatórios técnicos e científicos*», Universidade do Algarve, Abril 2001

Nunes, Mauro; O'Neill, Henrique, «*Fundamental de UML*», FCA, Setembro 2001

Anexos

Os seguintes anexos dizem respeito a documentação técnica e normalmente são entregues em separado, mas dado o presente relatório ser do âmbito académico e por uma questão de simplicidade, essa mesma documentação foi anexada. Note-se que pelo facto desta documentação ser apresentada em anexo, o seu valor não deve ser diminuído.

Segue-se um índice de anexos na página seguinte, de forma a facilitar a sua consulta.

Índice de Anexos

Anexo A - Especificação de requisitos do sistema	23
Anexo B - Descrição de desenho do sistema	25
B.1 Diagramas de caso de uso.....	25
B.2 Diagramas de classes.....	27
B.3 Diagramas de Base e Dados	41
Anexo C - Testes.....	43
C.1 Formulário de Login	43
C.2 Funções de Super Administrador	43
C.3 Funções de Administrador.....	43
C.4 Funções de Utilizador	43
C.5 Funções de Porteiro	44
Anexo D - Plano de instalação	45
D.1 Conteúdo	45
D.2 Instalação da base de dados	45
D.3 Instalação da aplicação.....	45
Anexo E - Manual de utilizador	46
E.1 Acesso à aplicação.....	46
E.2 Área de acesso.....	46
E.3 Área reservada.....	48
Anexo F - Manual técnico	49

Anexo A - Especificação de requisitos do sistema

Com base na análise efectuada e tendo em consideração os seguintes factores:

- Tempo de desenvolvimento.
- Custo das aplicações.
- Consistência e Robustez da Base de Dados.
- Facilidade de implementação de requisitos futuros.

Considera-se que as ferramentas mais adequadas para esta aplicação e projecto, serão as seguintes:

Ambiente de desenvolvimento:

- MS Visual Studio .NET 2008
- MS ASP.NET (Páginas Web dinâmicas).

Requisitos mínimos, máquina cliente (ver hardware recomendado pela Microsoft):

- MS Windows XP/Vista
- MS NET Framework 3.5
- MDAC 2.8 ou superior
- MSXML 6.0
- Conexão à Internet (ou Intranet)

Requisitos mínimos, servidor aplicacional (ver hardware recomendado pela Microsoft):

- MS Windows 2003/2008
- MS SQL Server 2008 ou versão Express

- MS IIS 6.0 ou superior

Não se encontram especificados dispositivos adicionais de hardware, nomeadamente impressora de etiquetas ou cartões assim como interfaces para integrar equipamentos periféricos específicos.

Anexo B - Descrição de desenho do sistema

A seguir apresentam-se os seguintes diagramas:

- Diagramas de Uso (*use case*)
- Diagramas de Classes
- Diagramas de Base de Dados

B.1 Diagramas de caso de uso

Os diagramas de caso de uso apresentados a seguir, apenas como exemplo, reflectem a estrutura de “Acesso às Operações” e a estrutura de “Acesso à Aplicação” dos diversos utilizadores.

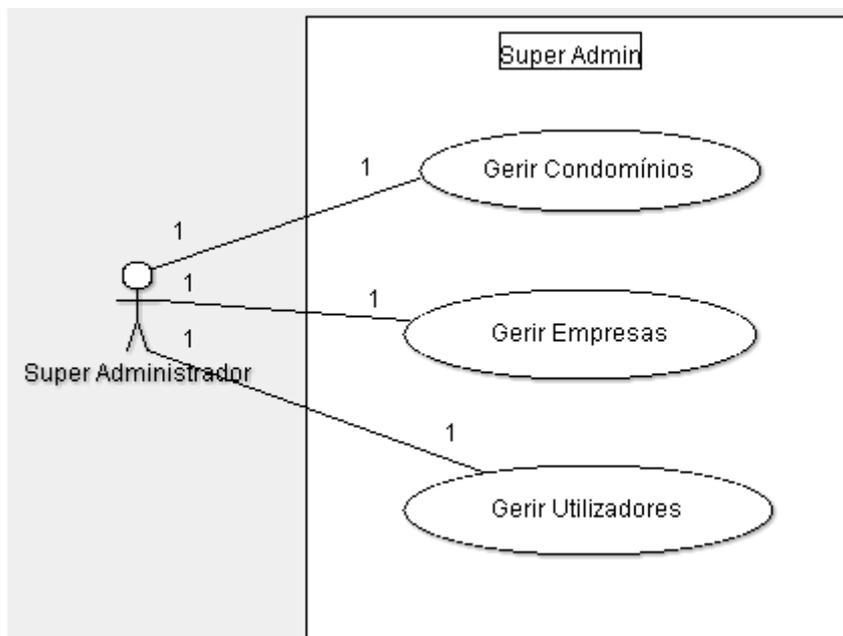


Figura 1: OFM – Use Case – Super Admin

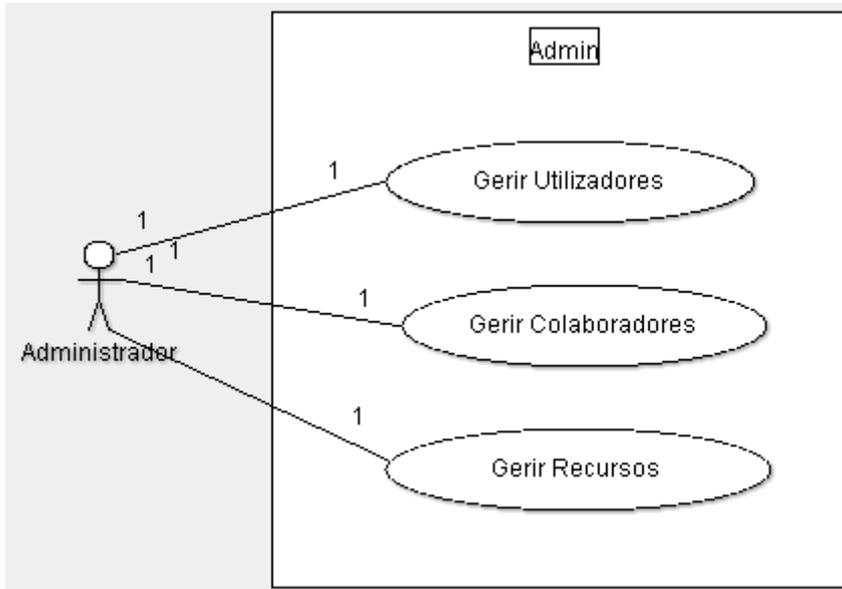


Figura 2: OFM – Use Case – Admin

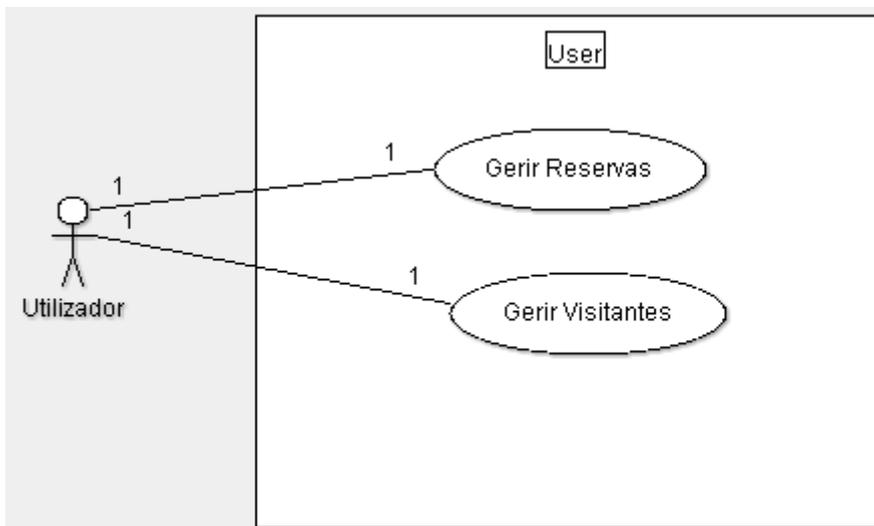


Figura 3: OFM – Use Case – User

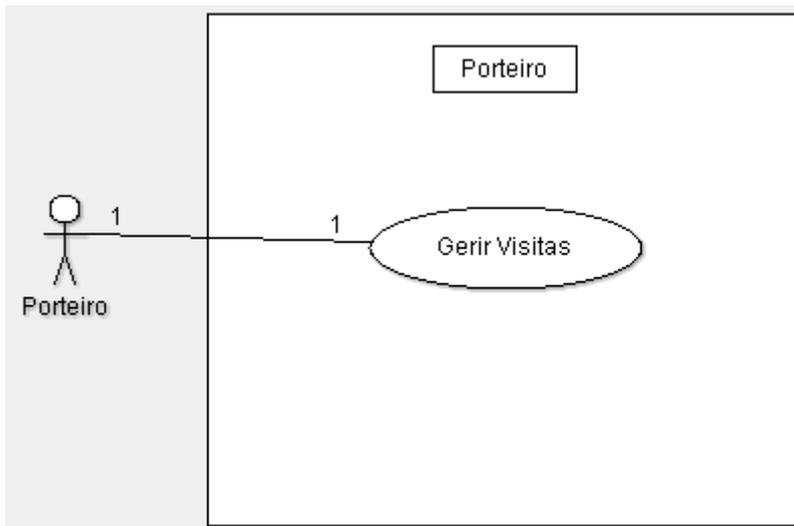


Figura 4: OFM – Use Case – Porteiro

B.2 Diagramas de classes

Esta secção é produzida pela aplicação *Microsoft Visual Studio 2008* e é composta em duas partes principais:

- DAL: *Data Access Layer*. Onde estão as classes que suportam o acesso à base de dados;

Colaborador
Class

Fields

- _CodigoColaborador As String
- _ColaboradorID As Integer
- _DataModificacao As Date
- _Email As String
- _EmpresaID As Integer
- _EmpresaNome As String
- _Funcao As String
- _Nome As String
- _Telefone As String
- _Username As String
- _UtilizadorID As Integer

Properties

- CodigoColaborador As String
- ColaboradorID As Integer
- DataModificacao As Date
- Email As String
- EmpresaID As Integer
- Funcao As String
- Nome As String
- Telefone As String
- Username As String
- UtilizadorID As Integer

Methods

- Actualizar()
- Apagar() (+ 1 overload)
- Inicializar()
- ListarTodos() As List(Of Colaborador)
- New() (+ 1 overload)

DAL - Colaborador

Utilizador
Class

Fields

- _ColaboradorID As Integer
- _CondominioID As Integer
- _DataModificacao As Date
- _EmpresaID As Integer
- _EmpresaNome As String
- _Password As String
- _RoleID As Integer
- _RoleNome As String
- _Username As String
- _UtilizadorID As Integer

Properties

- ColaboradorID As Integer
- CondominioID As Integer
- DataModificacao As Date
- EmpresaID As Integer
- EmpresaNome As String
- Password As String
- RoleID As Integer
- RoleNome As String
- Username As String
- UtilizadorID As Integer

Methods

- Actualizar()
- Apagar() (+ 1 overload)
- Inicializar()
- ListarPorteirosSemCondominio() As List(Of Utilizador)
- ListarRoles() As String()
- ListarTodos() As List(Of Utilizador) (+ 1 overload)
- ListarTodosSemColaborador() As List(Of Utilizador)
- New() (+ 2 overloads)
- ValidarAutenticacao() As Boolean

DAL – Utilizador

Role
Class

- Fields
 - _Nome As String
 - _RoleID As Integer
- Properties
 - Nome As String
 - RoleID As Integer
- Methods
 - ListarTodos() As List(Of Role)
 - ListarTodosLight() As List(Of Role)

DAL - Role

Condominio
Class

- Fields
 - _CondominioID As Integer
 - _DataModificacao As Date
 - _Morada As String
 - _Nome As String
 - _PorteiroID As Integer
 - _PorteiroUsername As String
- Properties
 - CondominioID As Integer
 - DataModificacao As Date
 - Morada As String
 - Nome As String
 - PorteiroID As Integer
 - PorteiroUsername As String
- Methods
 - Atualizar()
 - Apagar() (+ 1 overload)
 - Inicializar()
 - ListarTodas() As List(Of Condominio)
 - New() (+ 1 overload)

DAL – Condominio

Empresa
Class

Fields

- _CondominioID As Integer
- _CondominioNome As String
- _DataModificacao As Date
- _EmpresaID As Integer
- _Nome As String

Properties

- CondominioID As Integer
- CondominioNome As String
- DataModificacao As Date
- EmpresaID As Integer
- Nome As String

Methods

- Atualizar()
- Apagar() (+ 1 overload)
- Inicializar()
- ListarTodas() As List(Of Empresa)
- New() (+ 2 overloads)

DAL - Empresa

Recurso
Class

Fields

- _CodigoRecurso As String
- _DataModificacao As Date
- _EmpresaID As Integer
- _Nome As String
- _RecursoID As Integer

Properties

- CodigoRecurso As String
- DataModificacao As Date
- EmpresaID As Integer
- Nome As String
- RecursoID As Integer

Methods

- Atualizar()
- Apagar() (+ 1 overload)
- Inicializar()
- ListarTodos() As List(Of Recurso)
- New() (+ 1 overload)

DAL - Recurso

Reserva
Class

Fields

- _CanEdit As Boolean
- _ColaboradorID As Integer
- _ColaboradorNome As String
- _DataFim As Date
- _DataInicio As Date
- _DataModificacao As Date
- _RecursoID As Integer
- _RecursoNome As String
- _ReservaID As Integer

Properties

- CanEdit As Boolean
- ColaboradorID As Integer
- ColaboradorNome As String
- DataFim As Date
- DataInicio As Date
- DataModificacao As Date
- RecursoID As Integer
- RecursoNome As String
- ReservaID As Integer

Methods

- Atualizar()
- Apagar() (+ 1 overload)
- Inicializar()
- ListarTodos() As List(Of Reserva)
- New() (+ 1 overload)

DAL - Reserva

Visitante
Class

Fields

- _CanEdit As Boolean
- _ColaboradorID As Integer
- _ColaboradorNome As String
- _DataFim As Date
- _DataInicio As Date
- _DataModificacao As Date
- _EmpresaNome As String
- _Nome As String
- _VisitanteID As Integer

Properties

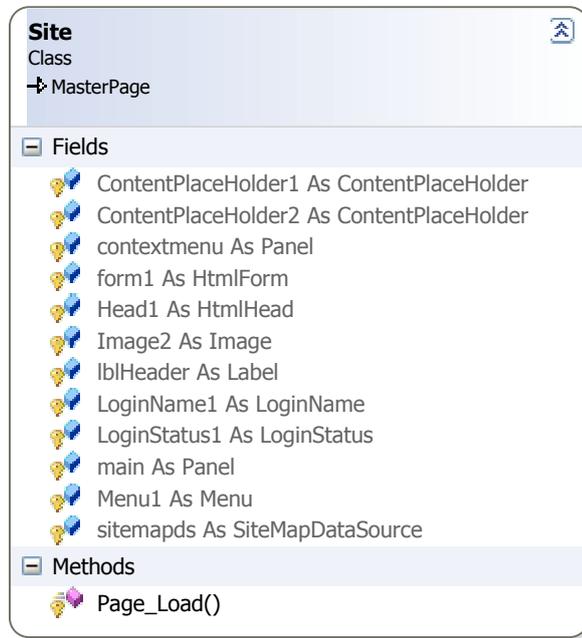
- CanEdit As Boolean
- ColaboradorID As Integer
- ColaboradorNome As String
- DataFim As Date
- DataInicio As Date
- DataModificacao As Date
- EmpresaNome As String
- Nome As String
- VisitanteID As Integer

Methods

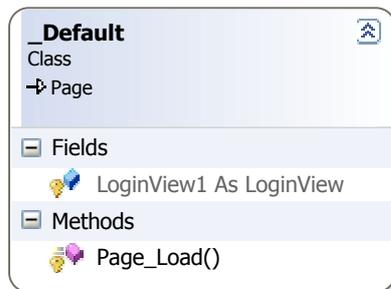
- Atualizar()
- Apagar() (+ 1 overload)
- ListarTodosPorCondominio() As List(Of Visitante)
- ListarTodosPorVisitante() As List(Of Visitante)
- New() (+ 1 overload)

DAL - Visitante

- UI: *User Interface*. Onde estão os componentes que suportam a interface com o utilizador.



UI – Página Master



UI – Página Inicial

Condominios 

Class
 ↳ Page

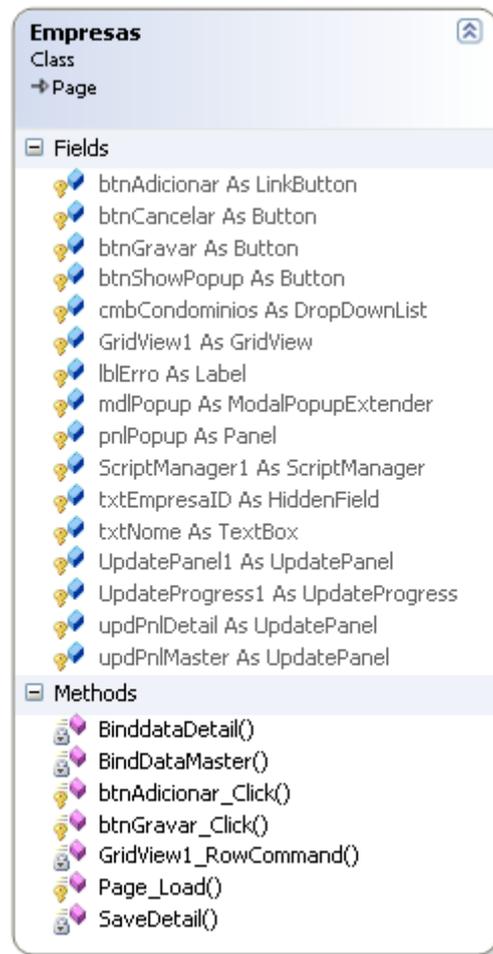
Fields

-  btnAdicionar As LinkButton
-  btnCancelar As Button
-  btnGravar As Button
-  btnShowPopup As Button
-  cmbPorteiro As DropDownList
-  GridView1 As GridView
-  lblErro As Label
-  mdlPopup As ModalPopupExtender
-  pnlPopup As Panel
-  ScriptManager1 As ScriptManager
-  txtCondominioID As HiddenField
-  txtMorada As TextBox
-  txtNome As TextBox
-  UpdatePanel1 As UpdatePanel
-  UpdateProgress1 As UpdateProgress
-  updPnlDetail As UpdatePanel
-  updPnlMaster As UpdatePanel

Methods

-  BinddataDetail()
-  BindDataMaster()
-  btnAdicionar_Click()
-  btnGravar_Click()
-  GridView1_RowCommand()
-  Page_Load()
-  SaveDetail()

UI - Condominios



UI - Empresas

Utilizadores 

Class
 ↳ Page

Fields

-  btnAdicionar As LinkButton
-  btnCancelar As Button
-  btnGravar As Button
-  btnShowPopup As Button
-  cmbEmpresas As DropDownList
-  cmbRoles As DropDownList
-  GridView1 As GridView
-  lblErro As Label
-  mdlPopup As ModalPopupExtender
-  pnlPopup As Panel
-  ScriptManager1 As ScriptManager
-  txtPassword As TextBox
-  txtUsername As TextBox
-  txtUtilizadorID As HiddenField
-  UpdatePanel1 As UpdatePanel
-  UpdateProgress1 As UpdateProgress
-  updPnlDetail As UpdatePanel
-  updPnlMaster As UpdatePanel

Methods

-  BinddataDetail()
-  BindDataMaster()
-  btnAdicionar_Click()
-  btnGravar_Click()
-  GridView1_RowCommand()
-  Page_Load()
-  SaveDetail()

UI - Utilizadores

Colaboradores 

Class
 ↳ Page

Fields

-  btnAdicionar As LinkButton
-  btnCancelar As Button
-  btnGravar As Button
-  btnShowPopup As Button
-  cmbUsername As DropDownList
-  GridView1 As GridView
-  lblErro As Label
-  mdlPopup As ModalPopupExtender
-  pnlPopup As Panel
-  ScriptManager1 As ScriptManager
-  txtCodigoColaborador As TextBox
-  txtColaboradorID As HiddenField
-  txtEmail As TextBox
-  txtFuncao As TextBox
-  txtNome As TextBox
-  txtTelefone As TextBox
-  UpdatePanel1 As UpdatePanel
-  UpdateProgress1 As UpdateProgress
-  updPnlDetail As UpdatePanel
-  updPnlMaster As UpdatePanel

Methods

-  BinddataDetail()
-  BindDataMaster()
-  btnAdicionar_Click()
-  btnGravar_Click()
-  GridView1_RowCommand()
-  Page_Load()
-  SaveDetail()

UI - Colaboradores

Visitantes 

Class
 ↳ Page

Fields

-  btnAdicionar As LinkButton
-  btnCancelar As Button
-  btnGravar As Button
-  btnShowPopup As Button
-  Calendar1 As Calendar
-  GridView1 As GridView
-  lblErro As Label
-  mdlPopup As ModalPopupExtender
-  pnlPopup As Panel
-  ScriptManager1 As ScriptManager
-  txtDataFim As TextBox
-  txtDataInicio As TextBox
-  txtNome As TextBox
-  txtVisitanteID As HiddenField
-  UpdatePanel1 As UpdatePanel
-  UpdateProgress1 As UpdateProgress
-  updPnlDetail As UpdatePanel
-  updPnlMaster As UpdatePanel

Methods

-  BinddataDetail()
-  BindDataMaster()
-  btnAdicionar_Click()
-  btnGravar_Click()
-  Calendar1_SelectionChanged()
-  GridView1_RowCommand()
-  Page_Load()
-  SaveDetail()

UI - Visitantes

Reservas
 Class
 ↳ Page

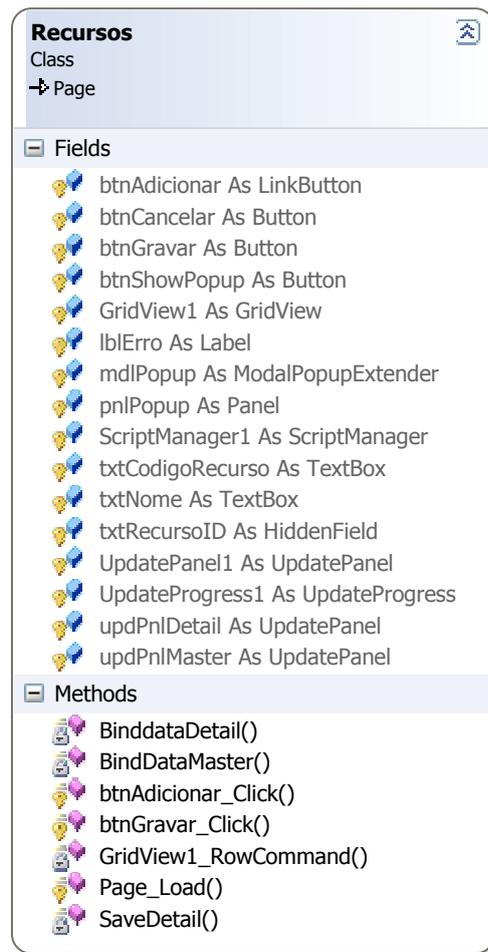
Fields

- btnAdicionar As LinkButton
- btnCancelar As Button
- btnGravar As Button
- btnShowPopup As Button
- Calendar1 As Calendar
- cmbRecurso As DropDownList
- GridView1 As GridView
- lblErro As Label
- mdlPopup As ModalPopupExtender
- pnlPopup As Panel
- ScriptManager1 As ScriptManager
- txtDataFim As TextBox
- txtDataInicio As TextBox
- txtReservaID As HiddenField
- UpdatePanel1 As UpdatePanel
- UpdateProgress1 As UpdateProgress
- updPnlDetail As UpdatePanel
- updPnlMaster As UpdatePanel

Methods

- BinddataDetail()
- BindDataMaster()
- btnAdicionar_Click()
- btnGravar_Click()
- Calendar1_SelectionChanged()
- GridView1_RowCommand()
- Page_Load()
- SaveDetail()

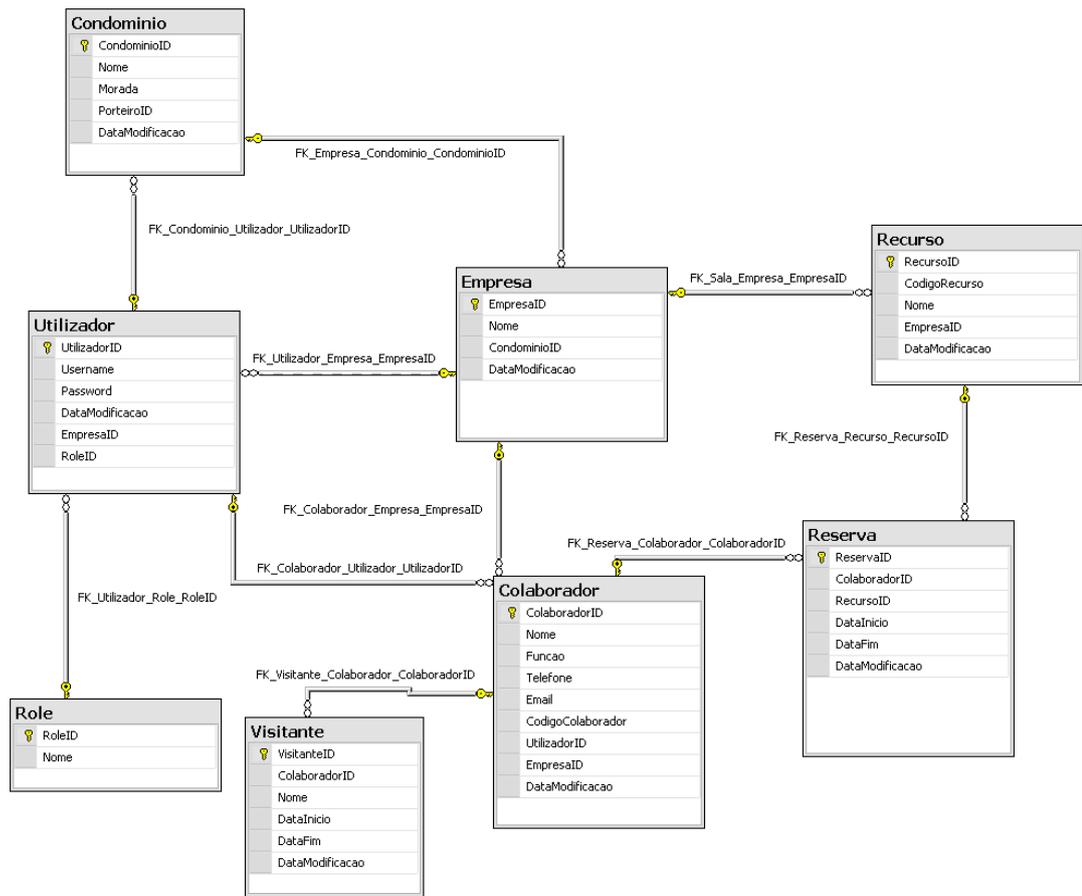
UI - Reservas



UI - Recursos

B.3 Diagramas de Base e Dados

De seguida apresentamos o diagrama de Base de Dados.



Modelo de Dados

Anexo C - Testes

Durante o processo de desenvolvimento e após conclusão do mesmo foram executados os seguintes testes à aplicação:

C.1 Formulário de Login

Confirma-se que é necessário introduzir os campos obrigatórios de “Email” e “Password”. Estes dados são gravados caso a opção “Guardar login” estiver activa. O browser também pode gravar estes campos, pois é sensível a dados de validação.

C.2 Funções de Super Administrador

Foram validadas as opções de “Editar”, “Apagar” e “Adicionar (...)”, nos seguintes itens da aplicação:

- Condomínios
- Empresas
- Utilizadores

C.3 Funções de Administrador

Foram validadas as opções de “Editar”, “Apagar” e “Adicionar (...)”, nos seguintes itens da aplicação:

- Utilizadores
- Colaboradores
- Recursos.

C.4 Funções de Utilizador

As funcionalidades de “Adicionar Reservas” e “Adicionar Visitantes” foram validadas com sucesso, através da escolha do dia no calendário auxiliar e através dos preenchimentos dos campos do formulário.

C.5 Funções de Porteiro

As funcionalidades de “Adicionar Visitantes” foram validadas com sucesso, através da escolha do dia no calendário auxiliar e através dos preenchimentos dos campos do formulário.

Anexo D - Plano de instalação

D.1 Conteúdo

Este anexo é destinado ao administrador do sistema. A instalação da aplicação é feita em duas fases:

- A instalação da base de dados;
- A instalação da aplicação.

D.2 Instalação da base de dados

- Instalar o Microsoft SQL Server 2008.
- Chamar o SQL Server Management Studio.
- Conectar ao servidor de base de dados.
- Fazer “attach...” do ficheiro MDF (e conseqüentemente LDF).

D.3 Instalação da aplicação

A instalação da aplicação deverá ser feita no servidor *web*. Este computador deve ter configurado o Internet Information Services [IIS] com o .NET Framework 3.5.

Depois de copiados os ficheiros para uma pasta deste servidor, criar uma Virtual Directory no IIS, e associar à pasta onde residem os ficheiros.

Para garantir a estabilidade criar no IIS uma Application Pool e associar a Virtual Directory à Application Pool. Em seguida colocar em modo “isolated”.

Por fim, para garantir o acesso à base de dados, verificar se a “connection string” está com as credenciais correctas para acesso à base de dados no ambiente de produção.

Anexo E - Manual de utilizador

E.1 Acesso à aplicação

Para ter acesso à aplicação, deverá abrir um *browser* como o Internet Explorer, e aceder ao site escrevendo na barra de endereço a localização, a qual deverá ser dada pelo administrador do sistema.

Neste exemplo, usa-se o computador local, na raiz do servidor Web, ficando o endereço da seguinte forma: <http://localhost:1657/default.aspx>.

E.2 Área de acesso

Ao aceder à página principal da aplicação, é solicitado a introdução da identificação e de uma senha, com a opção de gravar as informações de acesso no computador:

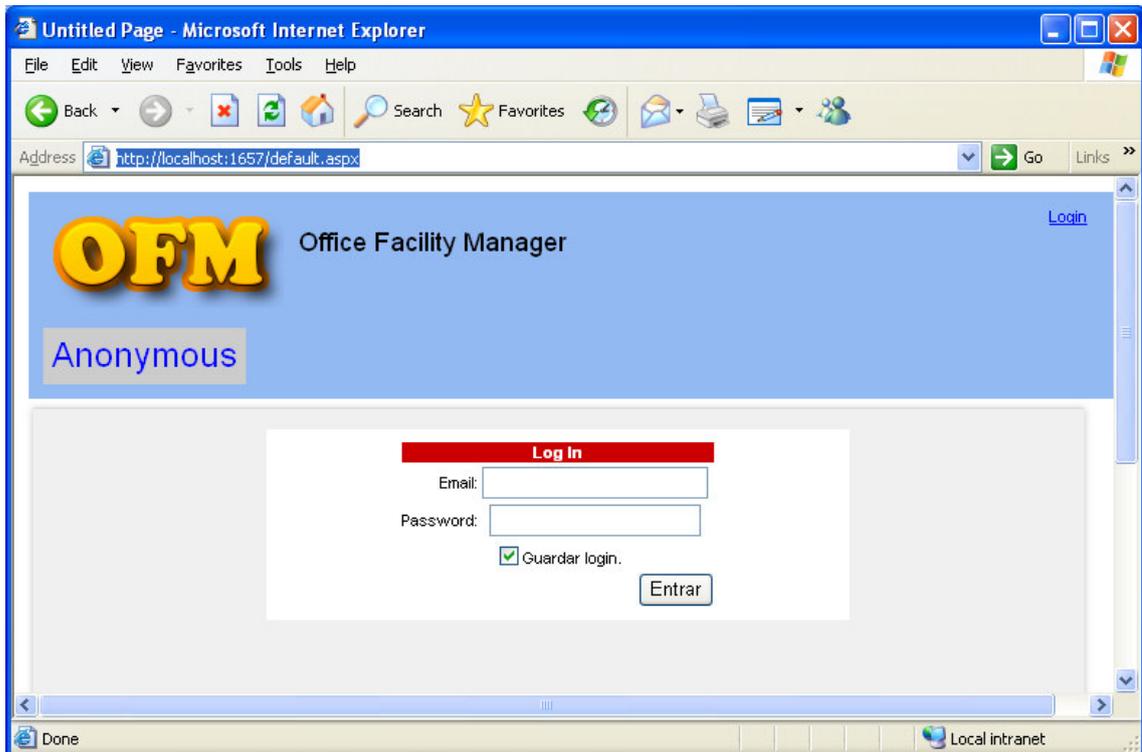


Figura 5: OFM - Área de acesso.

No caso de a senha estar incorrecta, é pedido que tente novamente:

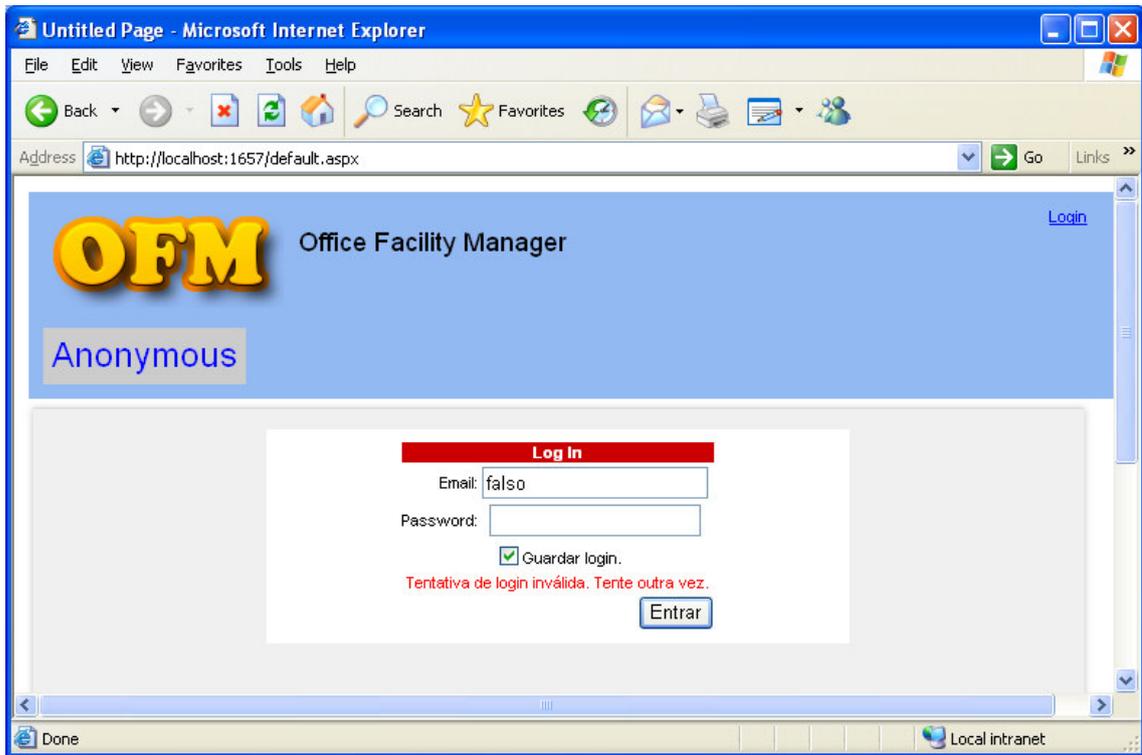


Figura 6: OFM - Senha inválida.

E.3 Área reservada

Em função do tipo de credenciais, a aplicação apresenta funcionalidades distintas, sendo indicado no topo o menu de funções principais e na lateral direita as funcionalidades associadas às funções principais. Na zona central encontra a listagem de registos em uso.

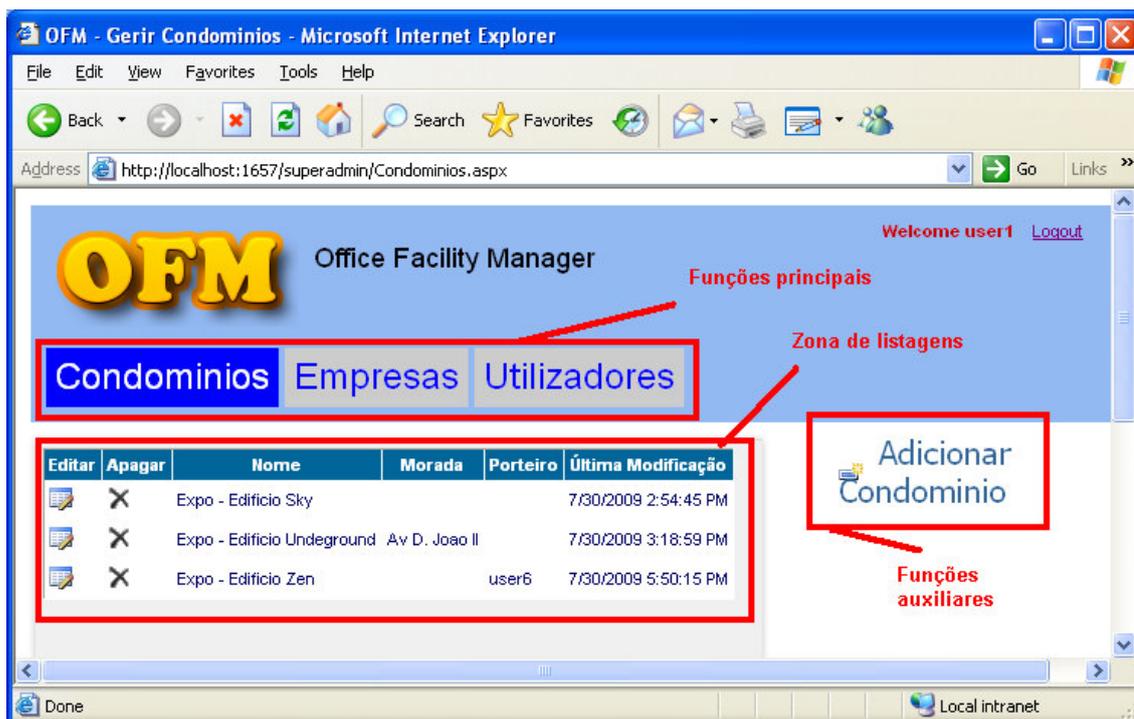


Figura 7: OFM – Funções por áreas reservadas.

Anexo F - Manual técnico

O manual técnico da aplicação da aplicação é produzido pela aplicação *VBdocman .NET 2.3* e é composto em duas partes principais:

- DAL: *Data Access Layer*. Onde estão as classes que suportam o acesso à base de dados;
- UI: *User Interface*. Onde estão os componentes que suportam a interface com o utilizador.

Esta documentação contém a descrição das classes e métodos e também inclui o código fonte (vide CD anexo).

Lisboa, 31 de Julho de 2009

Pedro Machado e Cunha Sousa e Costa & Pedro Jorge de Almeida Crespo