



Universidade do Porto
Faculdade de Engenharia
FEUP



Eduardo Alberto Almeida Espinheira Gomes

SIMEH
Sistema Integrado de Manutenção
de Edifícios de Habitação
no Instituto da Construção

IC
GOMe

LEIC
Porto, 2003

**Faculdade de Engenharia da Universidade do Porto
Licenciatura em Engenharia Informática e Computação**



**SIMEH
Sistema Integrado de Manutenção de Edifícios de
Habitação no Instituto da Construção**

Relatório do Estágio Curricular da LEIC 2002 / 03

Eduardo Alberto Almeida Espinheira Gomes

Orientador na FEUP: Eng. Ademar Aguiar
Orientador no Instituto da Construção: Prof. Rui Calejo

Novembro de 2003

UNIVERSIDADE DO PORTO
Faculdade de Engenharia
BIBLIOTECA M
N.º 78228
CDU 004(0473)
Data 10, 9, 2004

*Ao meu Tio Zé,
cuja memória é suficiente para
me incentivar a conseguir mais e melhor.*

*Aos meus pais e irmão,
que tornaram o meu caminho
mais fácil de percorrer,
permitindo que aprendesse
com todos os meus erros.*

Agradecimentos

Estas palavras pretendem agradecer a todos aqueles que contribuíram para que este estágio de final de curso reflectisse as competências adquiridas ao longo destes últimos anos e tenha permitido que este se realizasse nas melhores condições.

Agradeço ao Eng. Ademar Aguiar que mostrou, ao longo deste estágio, o mesmo apoio e disponibilidade que o caracterizou durante as várias cadeiras em que foi meu docente.

Aos meus professores: Augusto Sousa, Magalhães Cruz, Jorge Freire de Sousa e Jaime Villate que não se limitaram ao desempenho das suas funções de docência, apresentando uma acessibilidade e disponibilidade que me marcou positiva e profundamente.

Ao Prof. Rui Calejo pelo esforço dispendido na criação de condições que permitissem o melhor desenvolvimento do projecto e obtenção de um bom ambiente de trabalho.

Às Engenheiras Marisa Antunes e Alexandra Correia pelas intermináveis horas de discussão e apoio que permitiram alcançar o sistema como se encontra actualmente.

Ao Programa de Estágios no Ensino Superior PRODEP Medida 3 Acção 3.2, financiamento nº 306.012/03 – Concurso nº 1/3.2/PRODEP/2003, que financiou o meu estágio.

Ao secretariado da LEIC: Fátima, Diana, Sandra e Mónica cujo apoio e boa disposição, face a todos os problemas, permitiu tornar este meu percurso académico mais agradável e livre de obstáculos.

Aos meus vários colegas, cuja contribuição para ultrapassar esta etapa da minha vida não poderia deixar de ser mencionada.

Aos meus amigos, que não necessitam de ser nomeados, cujo apoio, amizade e compreensão ao longo destes anos nunca poderia ser completamente descrito nestas páginas.

Aos meus familiares, em especial aos meus tios e primos favoritos, cujo orgulho e apoio me permitiram acreditar que as piores situações são sempre passageiras e de que tinha capacidade para ultrapassar todas as dificuldades que se me deparavam.

A uma pessoa muito especial, cuja intervenção na minha vida me permite sentir a vida mais colorida.

Resumo

Este relatório documenta o estágio curricular realizado no Instituto da Construção no âmbito da Licenciatura em Engenharia Informática e Computação. O estágio, intitulado SIMEH – Sistema Integrado de Manutenção de Edifícios de Habitação Social, consistiu no desenvolvimento de uma ferramenta de auxílio à gestão do parque habitacional, nomeadamente manutenções e reclamações, afecto à GaiaSocial EM.

O estágio foi desenvolvido em 3 fases distintas do SIMEH: Protótipo, Aplicação e Web.

A primeira, SIMEH – PROTÓTIPO, consistiu numa fase pré-estágio cujo objectivo era desenvolver apenas um protótipo funcional para uma apreciação inicial e estabelecimento de um rumo a seguir no desenvolvimento de um sistema desta natureza. Esta fase foi rapidamente ultrapassada tendo sido atingidos os seus principais objectivos: nivelar a comunicação entre as várias partes integrantes da equipa de desenvolvimento, aprofundar o conhecimento sobre a linguagem do domínio, e definir uma metodologia de trabalho.

A segunda fase, SIMEH – APLICAÇÃO, representa o levantamento de requisitos e definição de arquitectura do sistema para a implementação da interface Win32 do sistema. O SIMEH – APLICAÇÃO consiste numa aplicação Win32, instalada nos gabinetes de apoio social e na sede da GaiaSocial, que permite o acesso aos vários módulos e serviços necessários para a gestão de empreendimentos. No final desta fase, a necessidade de uma implementação mais célere tornou imperativa a divisão da equipa de desenvolvimento pelas duas interfaces do sistema (aplicação e Web). Sendo assim, uma das equipas prosseguiu com o desenvolvimento do SIMEH – APLICAÇÃO, tendo a restante equipa iniciado o SIMEH – WEB.

A terceira fase, SIMEH – WEB, consistiu no levantamento de requisitos, definição de arquitectura e implementação de uma interface Web para o sistema. O SIMEH – WEB consiste num meio de submissão e gestão de reclamações por parte dos moradores e da própria assistente social, sendo o seu acesso permitido de qualquer local. A metodologia utilizada nesta fase permitiu atingir uma fase de maturação do produto final, com vista à introdução de critérios de usabilidade no desenho das interfaces.

De um ponto de vista global do trabalho é possível concluir que o sistema atinge os objectivos iniciais a que se propôs. São necessários, no entanto, ainda alguns ajustes recorrentes de uma utilização intensiva por parte dos quadros técnicos da GaiaSocial. O sistema permite ainda abranger mais módulos para alcançar a agregação, não só dos serviços de manutenção, mas também de toda a gestão da própria empresa municipal.

Índice de Conteúdos

Agradecimentos	3
Resumo	4
1 Introdução	8
1.1 Instituições Envolvidas	8
1.1.1 JuniFEUP	8
1.1.2 Instituto da Construção	9
1.1.3 GaiaSocial E. M.	9
1.2 SIMEH na GaiaSocial.....	10
1.3 Apresentação do SIMEH	10
1.4 Estrutura do Relatório	11
2 SIMEH	13
2.1 Descrição	13
2.2 Abordagem e Metodologia	14
3 Escolhas Tecnológicas.....	16
3.1 Servidor Web	16
3.1.1 Internet Information Services (IIS).....	16
3.1.2 Apache	17
3.1.3 Comparativo e selecção	18
3.2 Linguagem da interface Aplicaçional	19
3.2.1 Visual Basic	19
3.2.2 Delphi	20
3.2.3 Comparativo e selecção	21
3.3 Linguagem da interface web.....	22
3.3.1 mod_perl	22
3.3.2 ASP	23
3.3.3 ASP.NET	24
3.3.4 PHP	25
3.3.5 Comparativo e selecção	26
3.4 Base de Dados.....	27
3.4.1 Microsoft Access	27
3.4.2 Oracle Database	28
3.4.3 Comparativo e selecção	28

4	Arquitectura do SIMEH.....	30
4.1	Arquitectura Lógica.....	30
4.2	Arquitectura Física.....	31
5	Desenvolvimento do SIMEH — Protótipo.....	33
5.1	Introdução	33
5.2	Gerir Cadastros	35
5.2.1	Gerir Cadastro Técnico.....	36
5.2.2	Gerir Cadastro Social.....	40
5.2.3	Gerir Cadastro Económico	41
5.3	Gerir Manutenções.....	42
5.4	Gerir Intervenções	43
5.5	Gerir Reclamações.....	44
	Desenvolvimento do SIMEH — Aplicação	45
5.6	Introdução	45
5.7	Gerir Sistema	49
5.8	Gerir Recursos	51
5.9	Gerir Montagem do Empreendimento	52
5.9.1	Gerir Montagem da Arquitectura.....	52
5.9.2	Gerir Bd Tipologias	55
5.9.3	Gerir Bd Construtiva.....	58
5.9.4	Gerir Montagem Construtiva.....	59
5.10	Gerir Cadastros	62
5.10.1	Gerir Cadastro Técnico.....	63
5.10.2	Gerir Cadastro Económico	65
5.10.3	Gerir Cadastro Social.....	67
5.11	Gerir Manutenções.....	68
5.12	Gerir Intervenções	68
5.13	Gerir Reclamações.....	70
5.14	Gerir Pesquisa.....	71
5.15	Gerir Ferramentas	71
5.16	Gerir Ajuda.....	71
6	Desenvolvimento SIMEH – WEB	72
6.1	Introdução	72
6.2	Registar Novo Utilizador.....	74

6.3	Pedir Reenvio da Palavra-chave	75
6.4	Gerir Reclamações de Morador	76
6.4.1	Visualizar Reclamações	79
6.4.2	Submeter Reclamação	79
6.4.3	Consultar dados das Reclamações	81
6.5	Gerir Reclamações do Empreendimento	82
6.5.1	Submeter nova Reclamação	83
6.5.2	Consultar Reclamações	85
6.5.3	Triar Reclamações	86
7	Conclusão	88
	Referências	89
	Anexos	92
	Anexo A – Unified Modeling Language	93
	Anexo B – Critérios de Usabilidade	94

1 Introdução

O estágio abrange o período entre 3 de Março e 29 de Agosto de 2003. Consistiu no desenvolvimento de um sistema informático para apoio à gestão do parque habitacional social de Gaia. Existem 3 empresas envolvidas neste projecto: JuniFEUP, Instituto da Construção (IC) e GaiaSocial EM. Numa fase embrionária do projecto foi desenvolvido um protótipo e mais tarde nas duas fases posteriores foi especificado e implementado o produto. Esse produto denomina-se Sistema Integrado de Manutenção de Edifícios de Habitação (SIMEH).

1.1 Instituições Envolvidas

Em finais de Janeiro foi firmado um contrato entre a JuniFEUP e o IC, com o objectivo de especificar e implementar um protótipo de um sistema, cujo desenvolvimento e posterior manutenção seria mantida por uma empresa exterior. Ao longo das várias reuniões a empresa exterior decidiu abandonar o projecto, tendo sido elaborado este estágio curricular para dar continuação ao desenvolvimento do sistema.

A elaboração do estágio curricular terminou o vínculo com a JuniFEUP, tendo sido apresentado a totalidade do contexto do projecto em curso. O cliente final do IC era a GaiaSocial que necessitava de um produto para a gestão dos empreendimentos. O estágio decorreu maioritariamente nas instalações da JuniFEUP, gentilmente cedidas pela direcção da mesma.

1.1.1 JuniFEUP

A ideia de lançar uma iniciativa de índole empresarial pelos alunos da Faculdade de Engenharia da Universidade do Porto (FEUP) nasceu em 1999 na Licenciatura de Gestão e Engenharia Industrial. O seu principal objectivo foi responder a um desafio lançado pelo então Director de Licenciatura, Prof. Alcibiades Guedes, a um conjunto de alunos da referida Licenciatura. Os alunos desde logo aderiram, aos quais se juntaram, numa fase posterior, a Direcção da FEUP, a Área de Integração Profissional e alunos da Licenciatura de Engenharia Informática e Computação, para além de um conjunto de individualidades da FEUP que constituíram o grupo de Tutores. Hoje este grupo de pessoas constitui a JuniFEUP [1] que é um projecto de toda a comunidade da FEUP.

O tempo decorrido até ao início das actividades serviu sobretudo para estudar a melhor forma de implementar o projecto na FEUP. Foram analisados projectos semelhantes de outras Universidades nacionais e estrangeiras, para melhor se poderem equacionar as vantagens e desvantagens do desenvolvimento de um projecto desta natureza na FEUP. As conclusões deste estudo foram as melhores, pois um projecto desta natureza funciona como um pólo dinamizador onde os alunos empreendedores terão condições de desenvolver os seus projectos, e onde existirão oportunidades de interagir com a comunidade empresarial.

A JuniFEUP pretende ser um parceiro estratégico cujo objectivo é reduzir a distância entre o meio empresarial e o meio estudantil, promovendo actividades em conjunto com empresas, disponibilizando serviços e promovendo o aparecimento de ideias inovadoras. A JuniFEUP disponibiliza todos os meios que tem à sua disposição para criar as condições necessárias ao sucesso das iniciativas dos seus membros.

A JuniFEUP tem como missão permitir a todos os seus membros aplicar os conhecimentos adquiridos na FEUP em projectos, favorecendo a aproximação do meio universitário ao meio empresarial.

1.1.2 Instituto da Construção

Foi em 1989 que surgiu a ideia de criar um elo de ligação mais estreito entre a indústria da construção e a faculdade. O objectivo era contribuir directamente para uma melhoria significativa do sector da construção. O departamento de Engenharia Civil da FEUP, que dispunha desde logo de um corpo docente altamente qualificado, destacando-se um elevado número de doutorados, podia fornecer o *know-how* necessário.

Esse elo chama-se Instituto da Construção (IC) [2], que se apresenta como uma associação científica e técnica sem fins lucrativos, destinada a estabelecer laços directos entre o meio académico e o meio empresarial que serve.

Tem como associados a Universidade do Porto (UP), a Comissão de Coordenação da Região Norte (CCRN), a Associação dos Industriais da Construção Civil e Obras Públicas do Norte (AIOCCOPN), a Associação Portuguesa de Comerciantes de Materiais de Construção (APCMC), a Electricidade de Portugal (EDP), a Caixa Geral de Depósitos (CGD), o Instituto Nacional de Habitação (INH), a Câmara Municipal do Porto (CMP) e a Associação dos Portos do Douro e Leixões (APDL).

O IC presta serviços através da investigação aplicada e desenvolvimento experimental, da certificação de qualidade de produtos de construção, do comportamento físico e mecânico de produtos de construção, do desenvolvimento de novos processos/métodos ou materiais de construção, da divulgação de conhecimentos no âmbito de construção, da promoção de cursos, conferências, encontros ou congressos e da publicação ou divulgação de bibliografias especializadas.

O IC oferece oportunidade de resolução de problemas de elevado grau de dificuldade técnica, desenvolvimento e avaliação científica de novos processos, métodos e materiais, ajuda na melhoria da qualidade e da produtividade, projectos específicos de investigação aplicada e actividades correntes de certificação e de auditoria técnica.

1.1.3 GaiaSocial E. M.

A GaiaSocial [3] é uma empresa municipal que iniciou a sua actividade em 2000 e que surgiu da necessidade de gerir a Habitação Social do concelho de Vila Nova de Gaia.

A GaiaSocial adoptando, desde o primeiro momento, uma relação de grande proximidade com os municípios, tem como principais objectivos promover uma gestão integrada e participada nos empreendimentos, nomeadamente através de administrações de condomínio, promover a melhoria das condições de vida, promover uma adequada administração patrimonial e social, designadamente organizando e mantendo actualizado o cadastro dos bens imóveis e uma base de dados relativa aos residentes, assegurar a manutenção dos edifícios e respectivos espaços exteriores, promover as acções de cobrança de rendas dos fogos municipais, promover acções de formação e informação junto das populações dos empreendimentos, e promover a execução de obras de reparação nas habitações e edifícios necessárias para garantir um bom estado de conservação.

Para cumprir os vários objectivos a que se propunha, concilia esforços e reúne os vários intervenientes do processo, desenvolvendo condições especiais para o seu atendimento e acompanhamento. Dispõe para isso de uma equipa altamente especializada cujo principal objectivo é servir com eficiência, facilidade e rapidez as pessoas que se dirigem à GaiaSocial. O relacionamento entre entidades bancárias, empreiteiros e promotores é gerido de modo a que a relação com as famílias se desenrole em perfeito entendimento.

1.2 SIMEH na GaiaSocial

O Instituto da Construção (IC) comprometeu-se a desenvolver um produto para a GaiaSocial que permitisse auxiliar na gestão do parque habitacional social de Gaia.

O problema da GaiaSocial prende-se com a grande quantidade de habitações, cerca de 3000, e com a ineficiência na resolução dos muitos problemas decorrentes da manutenção e reabilitação dos seus vários empreendimentos. Existem outros problemas como a gestão dos habitantes e indivíduos, perfazendo um total de cerca de 27000 indivíduos, que procuram a atribuição de novas habitações.

A inexistência de recursos humanos técnicos necessários para desempenhar as várias tarefas de manutenção e intervenção obriga a GaiaSocial a contratar empresas exteriores. Esta solução origina uma dificuldade acrescida na gestão das muitas reclamações diariamente recebidas na GaiaSocial.

O produto a ser desenvolvido pelo IC pretende organizar as várias tarefas desempenhadas actualmente e facilitar a gestão financeira através da utilização de várias metodologias de previsão na manutenção de edifícios. Para alcançar essas metas o produto a ser desenvolvido necessitava de cadastrar todos os dados sociais e económicos, assim como gerir a documentação de cada empreendimento. Ao mesmo tempo permitia organizar as várias tarefas relacionadas com as reclamações, manutenções e intervenções decorrentes das várias patologias encontradas. A agregação de todos estes serviços permitiu a definição do nome da ferramenta a desenvolver: Sistema Integrado de Manutenção de Edifícios de Habitação (SIMEH).

1.3 Apresentação do SIMEH

O projecto SIMEH surge no Instituto da Construção após um conjunto de acontecimentos.

No estabelecimento de um protocolo entre o Instituto da Construção (IC) e a GaiaSocial EM (GS) fica designada a prestação de serviços por parte do IC consistindo na criação de um sistema de suporte à gestão dos empreendimentos de habitação social da GS.

É iniciado um trabalho de investigação, pelo grupo de trabalho chefiado pelo Prof. Rui Calejo, com o objectivo de definir e estruturar um sistema que se adequasse ao pedido da GaiaSocial. Após um vasto estudo prévio o Instituto da Construção inicia então a implementação de um pequeno protótipo, denominado EdiGest – Gestor de Edifícios, através do Eng. Pedro Ramos enquanto que também era iniciado o trabalho de normalização de conceitos e desenvolvimento de relatórios obtidos das visitas periódicas aos empreendimentos e à própria GaiaSocial pelas Engenheiras Marisa Antunes e Alexandra Correia. A ausência forçada do programador, devido a outros

compromissos, provocou a necessidade de encontrar outras soluções para a análise e implementação do sistema.

Através de um contacto inicial, estabelecido com a JuniFEUP, ficou estabelecido um contrato para o desenvolvimento de um novo protótipo EdiGest (futuramente referido como SIMEH – PROTÓTIPO), em Janeiro de 2003. A dimensão do projecto, o aumento constante de novos requisitos e o interesse demonstrado pela equipa da JuniFEUP dá origem à criação de um estágio curricular.

O estágio consistia num aproveitamento do trabalho já realizado para a criação do SIMEH – Sistema Integrado de Manutenção de Edifícios de Habitação.

O sistema apresenta duas interfaces do SIMEH: Aplicação e Web.

A interface Aplicação consiste numa aplicação Win32 que permite a 3 tipos de utilizadores (administrador, técnico e assistente social) percorrer os vários módulos do sistema para permitir a introdução ou tratamento dos vários dados dos empreendimentos. O sistema deve ser passível de evolução através da introdução de novos módulos para acrescer novas funcionalidades e serviços.

A interface Web, acessível aos moradores e assistentes sociais, consiste num meio de acesso através da Internet, que visa permitir a submissão de reclamações por parte dos seus utilizadores, assim como o tratamento posterior por parte das assistentes sociais. Esta interface deve apresentar uma boa escalabilidade para permitir, no futuro, integrar mais módulos.

1.4 Estrutura do Relatório

Este relatório está dividido em 8 capítulos, bibliografia e 2 anexos.

Neste capítulo 1 é feita uma breve introdução às várias instituições envolvidas no projecto, assim como aos objectivos do estágio. O objectivo deste capítulo é a obtenção de uma melhor compreensão do contexto deste estágio no seio da empresa.

No capítulo 2 é apresentado o SIMEH e seus objectivos. São abordados quais os problemas concretos que motivaram o desenvolvimento deste sistema, no enquadramento da GaiaSocial e do Instituto da Construção, e qual a solução pensada para o resolver.

No capítulo 3 é justificada a utilização da plataforma de desenvolvimento escolhida. São abordadas as várias escolhas tecnológicas através de uma análise comparativa entre os vários produtos presentes no mercado para cada um dos problemas que surgiram.

No capítulo 4 é descrita a arquitectura final do sistema, abordando as várias arquitecturas intermédias das fases: Protótipo, Aplicação e Web.

No capítulo 5 é referida a primeira fase de estágio: SIMEH – PROTÓTIPO. Esta fase de desenvolvimento é abordada através da especificação do protótipo desenvolvido do ponto de vista dos requisitos e base de dados.

No capítulo 6 é feita a apresentação da segunda fase de estágio: SIMEH – APLICAÇÃO. Assim como aconteceu no capítulo anterior, o desenvolvimento é apresentado através da especificação total do sistema.

No capítulo 7 é descrita a terceira e última fase de estágio: SIMEH – WEB. Seguindo a estrutura dos capítulos anteriores o desenvolvimento é apresentado através dos vários pontos de vista e diagramas de modelação de projectos de software.

No capítulo 8 são apresentadas as conclusões retiradas no decorrer do projecto, bem como as perspectivas de desenvolvimentos futuros no sistema desenvolvido durante o estágio.

No Anexo A é apresentada a norma Unified Modeling Language (UML) utilizada na descrição de cada uma das fases de estágio.

No Anexo B são apresentados os critérios de usabilidade utilizados no SIMEH – WEB.

2 SIMEH

Este capítulo pretende dar ao leitor uma visão geral dos problemas actuais com que se depara a GaiaSocial, nomeadamente com manutenções, intervenções e reclamações. Abordam-se ainda as tarefas que se realizaram nas fases do estágio assim como a metodologia utilizada.

2.1 Descrição

Os objectivos que regem o desenvolvimento do SIMEH foram ambiciosos e o poder e flexibilidade já atingidos pelo sistema são factores que lhe permitem solucionar grande parte dos problemas com que a GaiaSocial se depara. Para isso, contribuíram os diversos módulos que actualmente compõem a solução e que abordam a base de muitos dos problemas.

O problema principal decorrente da gestão de empreendimentos prende-se com as constantes manutenções e intervenções. Torna-se simples traçar um plano de manutenção adequado a um dado material. No entanto, num parque habitacional de cerca de 3000 habitações, cada uma com centenas de materiais a exigir planos de natureza diferente, torna-se bastante mais difícil. Mas o principal problema é representado pelas intervenções que consistem em manutenções forçadas, que não são previsíveis, e sobre as quais se sobrepõe o mesmo problema da dimensão do parque habitacional. A acrescer aos problemas já existentes, a GaiaSocial utiliza uma política de manutenção que a confronta com a inexistência de recursos humanos técnicos necessários, disponíveis nos seus quadros, para desempenhar essas mesmas tarefas. Este facto traduz-se num aumento significativo do custo, na ausência de uma gestão cuidada.

O problema das manutenções torna necessário o conhecimento e definição de vários conjuntos de dados: arquitectura do empreendimento, montagem construtiva e planos de manutenção.

2.1.1 Arquitectura do Empreendimento

A definição da arquitectura do empreendimento cria duas necessidades: a montagem da arquitectura e a definição de uma base de dados de tipologias. A montagem da arquitectura é obtida através da definição dos vários dados associados a cada empreendimento e conseqüente contagem dos blocos, entradas, pisos e fracções que o compõe. No entanto, é necessário associar a cada tipo de fracção uma tipologia previamente criada. Uma tipologia consiste na indicação da estrutura de uma fracção pelos vários compartimentos nos quais se descrevem dados importantes como área e perímetro, entre outros, sem esquecer os componentes (portas, envidraçados e janelas) que são partilhados entre as várias divisões. A criação de uma base de dados de tipologias facilita a posterior associação entre as tipologias e fracções que partilham uma mesma estrutura.

2.1.2 Montagem Construtiva

A montagem construtiva consiste na associação entre a arquitectura e estrutura de cada fracção com um conjunto de materiais e elementos, ou seja, declara a existência não apenas de uma porta num dado compartimento, mas indica, exactamente, qual a porta

(fornecedor, custo, tipo, dimensões, ...) que lhe está associada. Esta etapa torna necessária a existência de uma base de dados construtiva com todas as soluções e elementos construtivos para realizar essa associação.

2.1.3 Planos de Manutenção

Os planos de manutenção consistem na planificação de um conjunto de manutenções necessários para o bom funcionamento, ou prolongamento do tempo de vida de um dado material ou elemento. O conhecimento dos vários planos permite prever os problemas que podem surgir e agregar manutenções para poupar no custo do tempo de vida do empreendimento.

As intervenções surgem no seguimento da submissão de uma reclamação. Uma reclamação pode ser definida como a constatação de uma patologia (problema) num dado local, material e/ou elemento. Isto significa que é necessário um módulo de reclamações para servir de base ao módulo de intervenções. Este módulo assenta em conjuntos de dados anteriormente referidos como o conhecimento da arquitectura e montagem de um empreendimento para permitir localizar as patologias e associá-las a um dado elemento e local.

As reclamações implicam um conhecimento prévio dos habitantes das moradias geridas pela GaiaSocial definindo assim um cadastro social. No entanto, o procedimento de uma reclamação é suportado por dois tipos de colaborador da empresa municipal: assistente social e técnico. A assistente social vai confirmar e tratar as submissões das reclamações. No entanto, cabe ao técnico o processamento das restantes fases do procedimento: análise e accionamento dos mecanismos de resolução. De outro modo, as reclamações assentam sobre um conhecimento prévio de alguns dos recursos humanos da GaiaSocial.

Esta descrição do problema demonstra que a resolução assenta sobre um conjunto de dados basilares para a boa utilização do cruzamento dos dados dos vários conjuntos. Isto, por sua vez, prova a complexidade da própria definição do problema.

2.2 Abordagem e Metodologia

O projecto dividiu-se em 3 fases distintas do SIMEH: Protótipo, Aplicação e Web.

Na primeira, SIMEH – PROTÓTIPO, foram estabelecidos alguns requisitos superficiais nas diversas reuniões e disponibilizado o protótipo do EdiGest inicialmente implementado. A abordagem escolhida consistiu na implementação de um produto, semelhante ao EdiGest, ao qual se acresceram mais funcionalidades que semanalmente iam sendo aprofundadas. A metodologia de desenvolvimento utilizada consistia em pequenos ciclos de especificação, validação e implementação. Nesta fase a função de analista e co-arquitecto era desempenhada por Eduardo Espinheira, enquanto que o Nuno Serrano desempenhava as funções de co-arquitecto e implementador. Isto permitiu uma melhor compreensão da dimensão do problema, da linguagem de domínio e uma mais fácil apreensão de conceitos. Esta fase decorreu entre Janeiro e Fevereiro de 2003 não fazendo por isso parte do estágio.

Na segunda fase, SIMEH – APLICAÇÃO, iniciada em Março, foram alteradas as metas e estabelecidos novos objectivos e prioridades. Isto representou uma mudança da filosofia de implementação e uma redefinição da abordagem. O trabalho foi recomeçado

aproveitando ao máximo os conhecimentos adquiridos com o protótipo anterior. No entanto, a alteração dos objectivos deu origem a uma nova definição de arquitectura e especificação. A metodologia manteve pequenos ciclos de especificação com um aumento substancial dos ciclos de validação e implementação. As tarefas mantiveram a mesma atribuição. O estado da especificação da interface Aplicação do SIMEH e a necessidade de um desenvolvimento mais célere permitiu uma separação da equipa de estágio a meio do mês de Maio. Logo depois começou o desenvolvimento do SIMEH – WEB.

A terceira fase representa o desenvolvimento da uma interface Web do SIMEH. Esta fase consistiu numa especificação completa do SIMEH – WEB e na definição da arquitectura assente na mesma estrutura basilar da interface SIMEH – APLICAÇÃO. A metodologia utilizada permitiu longos ciclos de especificação, implementação e validação. Todas as tarefas relacionadas com esta fase foram efectuadas por Eduardo Espinheira, enquanto que a restante equipa de desenvolvimento procedia com outras funções na interface aplicação. Isto permitiu o refinamento e maturação do produto final. Uma reestruturação do produto permitiu também um aumento substancial da escalabilidade para agregar no futuro novos módulos e funcionalidades.

Fases	Início	Fim	Jan 2003		Feb 2003		Mar 2003		Apr 2003		May 2003		Jun 2003		Jul 2003		Aug 2003			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Simeh - Protótipo	13-01-2003	28-02-2003	[Barra de progresso]																	
Simeh - Aplicação	03-03-2003	16-05-2003	[Barra de progresso]																	
Simeh - Web	19-05-2003	29-08-2003	[Barra de progresso]																	

3 Escolhas Tecnológicas

Para o desenvolvimento do SIMEH foi necessário proceder à escolha de um conjunto de 4 tecnologias, nomeadamente: servidor web, linguagem da interface aplicacional, linguagem da interface web e base de dados. Para cada uma das escolhas são apresentadas as alternativas viáveis e uma secção final de comparação e escolha.

Na fase SIMEH – PROTÓTIPO foi dada a liberdade de escolha da linguagem de programação, sendo no entanto sugerido o Microsoft Visual Basic e o Borland Delphi. Esta fase ainda não apresentava interacção com a base de dados tendo sido a escolha relegada para fases posteriores.

Na fase SIMEH – APLICAÇÃO foi necessário escolher a base de dados, sendo utilizado o Microsoft Access numa fase inicial e sendo o SGBD do Oracle imposto como requisito final.

Na fase SIMEH – WEB foi necessário proceder à escolha de duas tecnologias: linguagem da interface e servidor web. A existência nos servidores da GaiaSocial do IIS e o Apache, permitiram limitar o leque de possibilidades existentes. A escolha da linguagem não apresentou qualquer tipo de imposição sendo dada ampla liberdade à equipa de desenvolvimento.

3.1 Servidor Web

A criação de uma interface web do SIMEH completamente autónomo permitiu que a escolha do servidor web fosse efectuada entre as duas possibilidades existentes nos servidores da GaiaSocial: Internet Information Services (IIS) e Apache. O IIS é o servidor web utilizado pela GaiaSocial para a página institucional que apresenta, no entanto a instalação do SGBD da Oracle vem acompanhada do Apache.

3.1.1 Internet Information Services (IIS)

Internet Information Services (IIS) [6] é um servidor web da Microsoft, com uma parcela pequena, mas significativa, de mercado da ordem dos 24,44% dos servidores activos [81]. É entregue gratuitamente com todas as versões do Microsoft Windows Server 2003 [7] e disponibiliza uma infra-estrutura de serviços web fiável, maneável e escalável. O IIS 6.0 permite às organizações, de qualquer dimensão, disponibilizar de uma forma rápida e simples sítios web e aplicações [8]. O IIS 6.0 permite gerir facilmente o servidor, aumentar a segurança dos sistemas de informação e providenciar uma plataforma de produtividade no desenvolvimento.

Para aumentar a disponibilidade dos servidores, o IIS tem como base uma arquitectura com tolerância a falhas e monitorização de processos o que aumenta de uma forma significativa a confiabilidade do servidor web. Além do mais, cada problema existente numa aplicação não causa problemas noutra aplicação, ou no próprio servidor.

Para reduzir o tempo gasto em administração dos sítios web e aplicações, o IIS disponibiliza um conjunto de ferramentas assim como um ficheiro XML de configuração, que pode ser modificado sem ter de reiniciar o servidor. Estas vantagens permitem a um mesmo administrador gerir um número maior de servidores sem um acréscimo significativo de complexidade [9].

Com o objectivo de aumentar a segurança, o servidor da Microsoft diminui as zonas de exposição propensas a ataque através de opções de segurança agressivas. Além disso, os módulos de autenticação e autorização foram melhorados.

O IIS alterou a arquitectura que tinha anteriormente com vista a isolar aplicações e sítios web em pequenas unidades chamadas *application pools*. Este isolamento permite conter a propagação de erros e facilita a resolução dos mesmos [10].

Uma das vantagens mais referidas desde a última actualização tem sido o aumento de escalabilidade através da possibilidade de cada servidor gerir e manter muitos mais sítios web e aplicações. Para esse efeito o IIS não pré-aloca os recursos internos no arranque do servidor. Isto permite que um maior conjunto de processos seja executado concorrentemente. Esta opção de alocação de recursos também melhora, de uma forma drástica, o reiniciar e encerrar dos próprios servidores e processos.

Outra das vantagens é o casamento das tecnologias de servidores web e do desenvolvimento de aplicações. A Microsoft, naturalmente, agregou as potencialidades da recente tecnologia .NET com o IIS salientando que a utilização do conjunto rentabiliza de uma forma transparente as respectivas utilizações [11].

3.1.2 Apache

O servidor HTTP¹ Apache [12] é um projecto cujo objectivo é desenvolver e manter um servidor HTTP *open-source*² para os sistemas operativos modernos incluindo o UNIX (e clones) e o Windows NT. O projecto visa disponibilizar um servidor eficiente, seguro e escalável que forneça os serviços necessários tendo em conta os standards HTTP.

O Apache tem sido o servidor web mais utilizado desde Abril de 1996. Num estudo efectuado pela Netcraft [13] observou-se que o Apache era o servidor escolhido por cerca de 67,28% dos sites disponíveis na Internet em Agosto de 2003, tornando-o o servidor web dominante no mercado mundial.

O servidor HTTP Apache é um projecto da Apache Software Foundation [14]. Está por isso afecto à licença de utilização da Apache [15].

A arquitectura modular do Apache permite adicionar ou remover funcionalidades, de acordo com o que é requerido. Como qualquer servidor web, suporta CGIs e FastCGI para estabelecer uma interface com programas externos e linguagens de *scripting*.

Cada sítio web é considerado pelo Apache como um *virtual host*, o que permite que uma só máquina sirva pedidos para muitos domínios distintos, usando um conjunto de directivas de configuração completamente diferentes para cada um.

A modularidade (API) do Apache, assim como todos os factores de segurança, performance e robustez, são constantemente reafirmados, pela fatia de mercado que representa.

A distribuição e facilidade de acesso ao código fonte permite a resoluções de problemas de forma imediatas, através da vasta comunidade de utilizadores. Além de permitir de uma forma segura e ponderada o crescimento constante de um produto de software estável e seguro [16].

¹ HyperText Transfer Protocol

² Software de código aberto

A quantidade de entidades de renome mundial que utilizam o Apache é bastante vasta.. Como exemplo, podem ser referidas as seguintes entidades: Amazon (<http://www.amazon.com>), Cable & Wireless (<http://cw.net>), Verio (<http://www.verio.net>), About.com (<http://www.about.com>), Valueweb (<http://www.valueweb.net>) e TierraNet (<http://www.tierranet.com>). Os sítios web referidos foram seleccionados a partir de uma lista de servidores com maior fiabilidade [17].

3.1.3 Comparativo e selecção

Para estabelecer uma comparação entre os dois servidores web torna-se importante salientar alguns pormenores técnicos que estabelecem fiabilidade e qualidade de serviço prestado.

O Apache usa um modelo de processos pai-filho. Neste, o pai não serve os pedidos, mas garante que existe pelo menos um processo filho a atender ao pedido, ou seja, se alguma coisa acontecer ao processo filho que está a servir o pedido o processo pai, de uma forma rápida, soluciona o problema alocando outro processo filho para essa tarefa. No caso do IIS, caso o processo que estava a servir o pedido vá abaixo a solução não é imediatamente alcançada. No entanto, a actual independência entre processos evita um problema antigo que consistia na necessidade de reiniciar o servidor de cada vez que um determinado número de processos falhava [18].

A nível de suporte torna-se difícil o IIS competir com o Apache devido, essencialmente, ao próprio desenvolvimento de cada um dos servidores. O Apache, ao ser o servidor web mais utilizado e fomentar que cada utilizador o configure e solucione os problemas que vá encontrando, leva a que os utilizadores adquiram um grau de conhecimento superior. Essa filosofia, quando multiplicada pelos milhões de utilizadores existentes, permite criar uma comunidade de suporte muito pró-activa na resolução de problemas. Além disso, o IIS já se tornou famoso por todos os erros que têm sido tornados públicos regularmente. Nesse aspecto, o Apache domina através da confiança: são muito poucos os erros encontrados, além de serem imediatamente absorvidos e solucionados por uma comunidade de milhões de programadores e analistas.

No entanto, a maior parte dos administradores de servidores Windows agradecem que o processo de configuração do servidor web tenha uma interface gráfica o que torna o sistema mais amigável. O Apache é configurado através de um ficheiro de texto implicando que a sua manutenção tenha que ser feita por alguém mais especializado, o que pode não ser verdade no caso do IIS.

A nível de custos torna-se difícil de comparar. O IIS é disponibilizado com as versões avançadas de Windows. Isto significa que o custo do IIS é praticamente nulo tendo em conta que se encontra pronto a usar. Por outro lado, o Apache para além de se mostrar como mais estável, mais rápido e fiável é ainda gratuito. Foi escolhido pela Oracle para ser instalado com todos os seus produtos a partir de uma determinada gama.

Na GaiaSocial, foi permitida uma certa flexibilidade de escolhas tendo em conta que tinham um sistema operativo Windows (IIS) e uma base de dados Oracle (Apache). Tendo em conta as características mencionadas a escolha recaiu facilmente no Apache.

3.2 Linguagem da interface Aplicacional

A escolha da linguagem da interface aplicacional foi limitada de uma forma ligeira, através de sugestão por parte da instituição de estágio de duas possibilidades: Visual Basic e Borland Delphi. Ao longo de reuniões consecutivas foi debatido o problema até se encontrar uma solução.

A principal dificuldade consistia na ausência de competências por parte de uma das empresas intervenientes para continuar o desenvolvimento e fornecer a assistência técnica necessária ao produto. No entanto foi alcançado um acordo de forma a escolher a linguagem que melhor se adequava à complexidade particular do sistema a ser desenvolvido.

3.2.1 Visual Basic

O Visual Basic [19] é uma linguagem de programação com uma longa lista de fiéis seguidores. Apesar do seu elevado custo disponibiliza, de uma forma surpreendente, um ambiente para criação de programas adequado para os programadores amadores e profissionais [20].

O Visual Basic faz parte de um conjunto de ferramentas da Microsoft (Visual Studio [21]) que permite a utilização de um ambiente gráfico de desenvolvimento que é muito intuitivo. Tem sido a ferramenta de eleição para todos os indivíduos que necessitam de desenvolver uma interface gráfica Windows de uma forma rápida e com o mínimo de conhecimentos de programação.

O Visual Basic faz parte de uma família de aplicações RAD³. Estas aplicações surgem da necessidade de criar aplicações cada vez mais complexas, em menos tempo. O Visual Basic surgiu como uma ferramenta pioneira neste campo, no sistema operativo Windows, pela facilidade de utilização [22].

Ao contrário de outras ferramentas (como o Visual C++) [23] é permitido ao utilizador clicar e arrastar pequenos componentes de interface que geram automaticamente código pelo programador. A configuração de pequenos componentes é o que permite a libertação de todo o potencial do Visual Basic. As interfaces são montadas como um puzzle, permitindo obter interfaces agradáveis e apelativas ao utilizador. Este tipo de filosofia na montagem de interfaces foi tão bem aceite pelos programadores que rapidamente contagiou a forma de desenvolver aplicações com outras ferramentas.

Para facilitar o desenvolvimento do código necessário o Visual Basic é dotado de uma quantidade quase infindável de pequenas ferramentas e utilitários das quais se pode destacar, por exemplo, o *Intellisense Function* [24] que completa automaticamente e em tempo real, à medida que se escreve o código, muitas das propriedades dos componentes.

São muitas as características a salientar na ferramenta em questão: desde a facilidade na geração de código SQL complexo no acesso a base de dados ou o acesso aos resultados de uma forma imediata. No entanto, uma das características mais salientadas consiste na incorporação dos *drivers*⁴ nativos OLE DB⁵, que permitem o acesso rápido ao SQL

³ Rapid Application Development

⁴ Controladores

⁵ Object Linking and Embedding Databases

Server e Oracle. Porém, as capacidades de acesso a bases de dados não se limitam a estas duas anteriormente referidas, porque o Visual Basic é compatível com os standards ODBC.

Outra característica importante prende-se com a interacção natural e esperada com outras ferramentas da Microsoft, tais como o IIS, através de linguagens como o ASP a referir posteriormente.

No entanto, existem alguns problemas que apenas podem ser resolvidos através de componentes externos. Apesar desses mesmos componentes obrigarem a um custo adicional, facilitam de muitas e variadas formas o desenvolvimento correcto e optimizado de aplicações [25].

3.2.2 Delphi

O Delphi [26] é um dos produtos mais conceituados da Borland [27]. Consiste num IDE⁶, ou seja, um ambiente integrado de desenvolvimento, composto por compilador, editor de texto, ferramenta de depuração de erros e algumas outras ferramentas de apoio.

Assim como o Visual Basic, o Delphi é uma ferramenta RAD. Isto significa, como já foi referido anteriormente, que, à medida que os componentes vão sendo seleccionados, o Delphi, neste caso, escreve o código fonte automaticamente. Os componentes, em geral, incluem classes de propriedades muito utilizadas e que se relacionam com outros objectos [28].

Apesar do que possa parecer à priori o Delphi não é, por si só, uma linguagem, mas sim um IDE para a linguagem Object Pascal [29]. O Object Pascal é a linguagem Pascal com uma metodologia de programação orientada por objectos associada. Claro está que este tipo de metodologia traz muitas vantagens.

Para manter compatibilidade com a recentemente lançada plataforma .NET, o Delphi teve o seu compilador alterado de forma a permitir a migração fácil das aplicações Win32 anteriores para a plataforma actual. É possível utilizar objectos COM⁷ proprietários do Delphi em aplicações .NET.

Uma das mais recentes inovações da Borland foi o desenvolvimento do Kylix. Este consiste num IDE em tudo semelhante ao Delphi; no entanto, foi criado para desenvolver aplicações em Linux. A interacção entre os dois foi pensada de forma a permitir a reutilização total do código desenvolvido em qualquer um dos IDEs. Para tornar possível essa reutilização são necessários alguns cuidados óbvios, como, por exemplo, garantir que os componentes usados num deles estão disponíveis da mesma forma no outro. Este novo IDE vem gratuitamente com praticamente todas as distribuições do Delphi.

Outra das novidades do Delphi consiste na alteração da arquitectura de desenvolvimento (MDA⁸) cujo objectivo é tornar mais rápida a passagem da fase de concepção para a fase de produção.

⁶ Integrated Development Environment

⁷ Component Object Model

⁸ Model Driven Architecture

Seguindo as novas filosofias de desenvolvimento de engenharia de software, o Delphi permite uma integração transparente com ferramentas da Rational Rose [30][31], mais concretamente o ModelMaker [32], utilizadas para modelar aplicações através de UML.

Outra das funcionalidades prende-se com a integração constante de componentes comerciais com a distribuição do Delphi. Isto vem diminuir o custo habitual de compra de componentes proprietários. No entanto, a facilidade de desenvolver componentes próprios continua a ser uma das características fundamentais desta ferramenta.

Para facilitar o desenvolvimento de código fonte por parte dos programadores, o Delphi é dotado de dezenas de ferramentas tais como: auto-complete, modelos de código, dicas e sugestões, preenchimento automático de parâmetros, visualização da estrutura dos componentes e objectos entre muitas outras.

Uma das ferramentas agregadas ao Delphi mais premiadas e conceituadas é o Interbase [33]. Esta ferramenta permite uma interacção com qualquer tipo de base de dados. A optimização na ligação e funcionalidades acrescidas são as características que tornaram o Interbase como uma referência incontestada.

Existem muitas formas de atestar as vantagens e qualidades do Delphi. Uma dessas formas é a quantidade de prémios recebidos desde o seu aparecimento.

3.2.3 Comparativo e selecção

A comparação entre o Visual Basic e o Delphi pode ser feita de várias formas analisando algumas das suas características. Alguns dos critérios apresentados não foram referidos anteriormente devido à sua natureza mais técnica; no entanto, por serem importantes num comparativo desta índole não se podem deixar à margem [22].

Um critério comum numa análise desta natureza é a performance. Para a comparar nas duas ferramentas deve ser analisada a linguagem de base de cada uma delas. O Delphi usa o Object Pascal enquanto que o Visual Basic utiliza o Microsoft Basic. A performance do Delphi é notória simplesmente porque gera executáveis compilados, enquanto que o Visual Basic produz código semi-interpretado, isto é, o Delphi é constituído por um compilador de código nativo optimizado em vez de um interpretador de pseudo-código, obrigatoriamente mais lento, usado por produtos como o Visual Basic. Isto resulta numa execução dez ou vinte vezes mais rápida dos executáveis produzidos pelo Delphi. O *linker*⁹ usado pelo compilador permite a optimização de segmentos de código, reduzindo em cerca de 30% o tamanho do executável. Isto permite, mais uma vez, um ganho de velocidade considerável.

Outra área em que o Delphi apresenta uma velocidade substancialmente maior é no acesso a bases de dados. A camada de base de dados do Visual Basic é implementada via ODBC em oposição ao motor de base de dados da Borland usado no Delphi. As opções de conectividade disponíveis numa e noutra ferramenta apresentam mais uma vez uma vantagem clara na utilização do Delphi.

O outro lado de uma análise de performance prende-se com a velocidade de desenvolvimento de aplicações. Como já foi referido anteriormente, ambas as ferramentas são RAD o que implica facilidade na transição do *design* inicial e

⁹ ferramenta de ligação de módulos numa fase de tradução de código fonte

prototipagem para a implementação final e instalação. Uma ferramenta RAD disponibiliza vários elementos para aumentar a velocidade de desenvolvimento requerida tais como: um ambiente gráfico de desenvolvimento, blocos ou componentes de alto nível e acesso directo, ou via objectos, a segmentos de código contextualizados. Em Windows, a estrutura de uma aplicação é frequentemente construída à volta de um GUI com comportamentos da aplicação a serem desencadeados por mensagens ou eventos. No entanto, a panóplia de soluções apresentadas pelo Visual Basic fica aquém daquela apresentada pelo Delphi em pormenores importantes tais como: disponibilidade de muitos mais componentes e controlos, melhoria nas capacidades de localização dos objectos, melhoria na modificação das listas de propriedades, funções de eventos partilhadas, galeria extensa de modelos de projectos e código e uma interface integrada na fase de *debugging*¹⁰ e inspecção de objectos.

Em conclusão, numa análise de ferramentas RAD disponíveis no mercado as duas escolhidas nesta comparação são claramente as melhores. No entanto, numa análise mais cuidada, o Delphi emerge como uma referência entre os dois.

3.3 Linguagem da interface web

A ampla liberdade oferecida à equipa de estágio permitiu observar quais as possibilidades existentes no mercado, após a escolha do servidor web Apache. As possibilidades encontradas foram: mod_perl, JSP, ASP, ASP.Net e o PHP.

O mod_perl é um módulo do apache baseado na linguagem perl, que apresenta um potencial de desenvolvimento quase ilimitado, o facto ter sido desenvolvido para o Apache torna-o uma das escolhas mais fortes.

O JSP (JavaServer Pages) [77] é uma tecnologia disponibilizada pela Sun, amplamente utilizada no desenvolvimento web. No entanto, esta linguagem é retirada do leque de escolhas, devido a experiências anteriores com a mesma. Esta decisão baseia-se apenas na escolha pessoal do futuro programador.

O ASP e o ASP.Net são duas tecnologias da Microsoft baseadas em filosofias diferentes, mas que têm sido largamente utilizadas na implementação de sítios web com complexidades consideráveis.

O PHP é uma linguagem *open-source* que tem vindo a conquistar adeptos e acrescentando funcionalidades para competir no mercado competitivo de desenvolvimento web.

A análise das tecnologias é efectuada através de uma pequena apresentação dos aspectos técnicos considerados relevantes e consequente análise comparativa no contexto do projecto de estágio.

3.3.1 mod_perl

O mod_perl [36] é um módulo do Apache. Permite o casamento de duas tecnologias de maturidade e qualidade comprovadas: Apache e Perl.

Ao contrário de outras tecnologias de implementação de interfaces Web, o mod_perl não se limita a esse objectivo. Esta linguagem de programação permite, de uma forma

¹⁰ fase de depuração de erros

simples e directa, interagir com a totalidade da API¹¹ do Apache. Desta forma, podem ser criados e configurados scripts CGI¹² que são executados a uma velocidade 100 vezes superior à velocidade de execução normal. De igual modo, a interacção entre o servidor e a linguagem oferece a facilidade de criação de módulos personalizados para estender e melhorar o próprio servidor web.

O potencial de utilização do mod_perl é praticamente ilimitado devido à possibilidade de acesso a todas as fases do ciclo de execução do script. Existem muitos pacotes construídos a partir da infra-estrutura do mod_perl que auxiliam ao desenvolvimento de sítios web escaláveis e de fácil manutenção. Para tornar ainda mais acessível a implementação de aplicações ou pacotes baseados no mod_perl, existe um número considerável de modelos e application frameworks disponíveis no CPAN¹³ [37]. Além das vantagens referidas, o mod_perl permite ligações persistentes à base de dados e a capacidade de armazenar dados de sessão durante o processo de execução [38]. No entanto, existem algumas contrapartidas a referir das quais se destacam: o elevado uso de memória por parte do mod_perl e a curva de aprendizagem considerada, normalmente, como acentuada e longa [38].

Tendo em conta a natureza do desenvolvimento da própria linguagem, o suporte é realizado através de listas de correio electrónico, livros e sítios web. Estes canais constituem a interface entre a comunidade de programadores e utilizadores do mod_perl.

Assim como para todos os projectos da Apache Software Foundation [14] a licença de utilização do mod_perl está definida na *Apache License*.

Para comprovar as qualidades inerentes à escolha do mod_perl como linguagem de desenvolvimento de uma interface web podem ser referidas algumas entidades utilizadoras, como por exemplo: PAUSE¹⁴ (<http://pause.perl.org>), IMDB¹⁵ (<http://www.imdb.com>), WIRED Digital (<http://home.wired.com>), Webmonkey (<http://www.webmonkey.com>), Slashdot (<http://slashdot.org>), Macromedia (<http://www.macromedia.com>), ArtToday (<http://www.arttoday.com>) e Adobe (<http://www.adobe.com>).

3.3.2 ASP

Active Server Pages, ou ASP [40][41] como é mais conhecido, é uma tecnologia que permite aos programadores desenvolverem páginas web dinâmicas e interactivas.

A linguagem vulgarmente utilizada no desenvolvimento do ASP é o VBScript¹⁶ [42] embora também possa ser utilizado o Jscript¹⁷ [43].

Como o ASP foi inicialmente introduzido pela Microsoft no seu próprio servidor web (IIS) [44], que corre em Windows 2000 / XP Pro / NT4, é o servidor que o melhor rentabiliza. No entanto, para colmatar esta limitação, a Sun [45] desenvolveu um produto chamado Chil!Soft [46] que permite que os ASP possam ser executados noutros

¹¹ Application Programming Interface

¹² Common Gateway Interface

¹³ Comprehensive Perl Archive Network

¹⁴ Perl Authors Upload Server

¹⁵ Internet Movie DataBase

¹⁶ Visual Basic Script

¹⁷ Versão de JavaScript da Microsoft

servidores web tais como Apache, I-Planet [47], Zeus [48], Red Hat Server [49], etc. usando vários sistemas operativos incluindo Linux [50], Solaris [51], Hp-UX [52], AIX [53], etc.

Por vezes, torna-se importante separar algumas funcionalidades da totalidade do sítio web. Para solucionar esse género de necessidades o ASP permite compilar componentes autónomos. Os componentes ASP são blocos de código que podem ser requisitados posteriormente separando assim uma ou mais funcionalidades do todo. Esta organização no desenvolvimento do próprio sítio web permite, não só uma reutilização do código, como também uma robustez e escalabilidade muito maiores [54].

O ASP é um componente de desenvolvimento do IIS e é, por isso, livre de ser usado com o IIS.

Para atestar a qualidade da linguagem em análise basta referir que o sítio mais visitado nos últimos anos é desenvolvido e gerido com ASP: o sítio é o da Microsoft (<http://www.microsoft.com>).

No entanto, de forma a salientar a facilidade na integração do Apache com o ASP, podem ser referidas também: FreeLotto (<http://www.freelotto.com>), WebTime (<http://webtime-project.net>), Apache Hello World Benchmarks (<http://chamas.com/bench/>), American Gamers (<http://www.americangamers.com>), AlterCom Advanced Web Hosting (<http://altercom.com>) e o Red Hat Support (<http://www.redhat.com/apps/support/>) [55].

3.3.3 ASP.NET

O ASP.NET [56] é uma tecnologia inserida na plataforma .NET. A tecnologia ASP.NET apresenta várias características atraentes, tanto para o programador que a usa no desenvolvimento, como para o utilizador final. Apesar de manter o mesmo objectivo do ASP apresenta algumas diferenças fundamentais, tais como o desenvolvimento através de um ambiente orientado a objectos, disponibilizando uma melhoria significativa de velocidade e escalabilidade para sítios web de média e grande dimensão.

Existem várias características e funcionalidades importantes de referir numa análise cuidada desta tecnologia; uma dessas características é a performance. Para alcançar a performance actual, o ASP.NET usa técnicas de *caching* e compila o código a executar [57].

Antigamente, o código era interpretado cada vez que o sítio era visitado por um utilizador, mas agora a página a ser visualizada, já tem o código compilado. O *caching* consiste em armazenar informação numa determinada zona de memória que vai ser posteriormente utilizada. O ASP.NET permite aos programadores definir áreas do sítio web a utilizar essa técnica de *caching*. Além dessas áreas, também podem ser definidos dados de uma base de dados para utilizar a mesma técnica. Esta técnica permite rentabilizar o espaço de memória utilizado, assim como aumentar a velocidade de acesso a uma determinada informação total ou parcialmente estática.

Para permitir uma maior disponibilidade do sítio web o ASP.NET foi dotado de ferramentas de gestão automática de memória e processos que permitem a recuperação de erros.

A principal vantagem desta tecnologia, e de toda a *framework* .NET, é a liberdade de escolha da linguagem a utilizar. Cada programador pode desenvolver a aplicação ou interface web em uma ou mais linguagens disponibilizadas por esta plataforma. As linguagens existentes são: Visual Basic, C#, JScript, C++, J# , APL, Cobol, Component Pascal, Eiffel, Fortran, Haskell, Mercury, Oberon, Perl, Python, Scheme, SmallScript, TMT Pascal, etc. Esta característica permite reduzir ou anular a curva de aprendizagem necessária, na utilização da tecnologia .NET [58].

A utilização do ASP.NET é completamente gratuita. O ambiente de desenvolvimento a utilizar pode ir do simples Notepad [59] ou Editplus [60], de preço nulo ou reduzido, até editores premiados e de preço considerável como o Visual Studio .NET [61].

Apesar de tudo, esta apresenta uma limitação considerável em relação a servidores de alojamento. Apenas aceita servidores com sistemas operativos e serviço de alojamento web da Microsoft.

Para confirmar a qualidade do desenvolvimento de sistemas através da plataforma .NET, mais especificamente o ASP.NET, podem ser citados as seguintes organizações: Bank of Nova Scotia (<http://www.scotiabank.com/>), Pacific Life (<http://www.pacificlife.com/>) Corillian (<http://www.corillian.com/>), MSN (<http://www.msn.com/>), Compaq (<http://www.compaq.com/>), Siemens (<http://www.sea.siemens.com/>), Dell, Accenture (<http://www.accenture.com/>), Arthur Andersen (<http://www.andersen.com/>) National Cancer Institute (<http://www.cancer.gov/>) e Ford [56].

3.3.4 PHP

O PHP [62] é uma linguagem de *scripting* open-source de uso geral, executada do lado do servidor e especialmente concebida para o desenvolvimento de aplicações Web.

Actualmente, apesar dos objectivos claros do desenvolvimento do PHP, a linguagem sofreu um alargamento nos campos de acção; apresenta as seguintes áreas: *scripting* do lado do servidor, *scripting* de linha de comandos e aplicações GUI¹⁸ do lado do cliente.

As vantagens inerentes a um desenvolvimento com esta linguagem prendem-se essencialmente com a facilidade de programação, apoiada em centenas de referências bibliográficas e comunidades na Internet, mas especialmente com a flexibilidade oferecida pelo próprio PHP. O PHP pode ser utilizado em praticamente todas as plataformas e é amplamente suportado pela maioria dos servidores web actuais. A natureza da sua própria implementação permite-lhe ser configurado como módulo para a maioria dos servidores e como CGI comum para outros.

Além da liberdade de escolha do ambiente de execução, o PHP permite a possibilidade de utilizar uma programação estrutural [63] ou orientada a objectos [64]. Na actual versão, muitas bibliotecas de código e grandes aplicações foram escritas somente usando OOP¹⁹.

Outra característica a realçar é a possibilidade de gerar imagens, ficheiros PDF²⁰ e animações Flash²¹ de uma forma dinâmica.

¹⁸ Graphical User Interface

¹⁹ Object Oriented Programming

²⁰ Portable Document Format

²¹ Formato de animações vectoriais da Macromedia

A mais forte característica do PHP é o suporte a uma ampla variedade de bases de dados tais como dBase [65], Interbase [66], PostgreSQL [67], Sybase [68], ODBC²² [69], Informix [70], Oracle entre outros. Adicionalmente, o PHP suporta o ODBC permitindo aceder a qualquer base de dados que obedeça a esta norma.

O PHP permite, através da utilização de extensões *open-source*, aumentar o potencial de utilização para fins mais específicos tais como sistemas de pagamento *online*, outras funcionalidades de e-commerce, instanciação de objectos em JAVA [71], protocolos de comunicação, processamento de texto, interpretador de documentos XML, funções de pesquisa, compressão, calendário, tradução, etc [72][73].

Existem alguns projectos relacionados com o PHP que se tornam importantes de referir: o PEAR e a Iniciativa da Garantia de Qualidade do PHP.

O PEAR²³ [74] é o repositório de extensões PHP e de Aplicação: consiste numa versão das classes base do PHP que são desenvolvidas para futuramente distribuir em extensões C e PHP. O projecto nasceu em 2000, com um largo número de programadores, com o objectivo de desenvolver os alicerces para permitir ao PHP, rentabilizar o código existente e acrescentando, cada vez mais, novas funcionalidades. Hoje em dia, o PEAR inclui acesso a bases de dados, controlo de *caches*, cálculos matemáticos avançados e e-commerce²⁴.

A Iniciativa da Garantia de Qualidade do PHP [75] foi criada, também em 2000, em resposta às críticas sobre as versões do PHP não terem sido testadas o suficiente para ambientes de produção profissionais. Os programadores inseridos nesta iniciativa melhoram o código base do próprio PHP tendo em vista não apenas resolver erros, mas também aumentar significativamente a performance da própria linguagem.

Para assegurar a facilidade de utilização do PHP para projectos de dimensão variável podem ser referidos alguns sítios web tais como: easyDNS (<http://www.easydns.com/>), StoneKeep Consulting (<http://www.keystone.com/>), FreshMeat (<http://www.freshmeat.net/>), RegSoft (<http://www.regsoft.com/>), VB-World (<http://www.vb-world.net/>), Canadian Police Insignia Collectors (<http://www.cpica.on.ca/>), Nerede (<http://www.nerede.com/>), TrialSport (<http://www.trialsport.de/>), dealNews (<http://dealnews.com/>), Family Health Network (<http://www.familyhealth.com.au/>), Indianapolis (<http://www.indy500.com/>), ZooMedia (<http://www.zoomedia.com/>), InfoTrain (<http://www.infotrain.com/>) e WNDU (<http://www.wndu.com/>).

3.3.5 Comparativo e selecção

Após uma breve explicação de cada uma das tecnologias torna-se simples fazer uma análise comparativa.

Em relação à integração com o servidor web escolhido as escolhas diminuem abruptamente. O ASP e o ASP.NET apresentam-se como tecnologias que ou não permitem nenhuma interacção ou então necessitam de software proprietário para permitir uma ligação transparente ao próprio servidor. Neste ponto, o mod_perl destaca-se de uma forma muito positiva devido à própria origem do mesmo. No entanto, a integração entre o PHP e o Apache não apresenta qualquer tipo de obstáculos.

²² Open Database Connection

²³ PHP Extension Add-on Repository

²⁴ Electronic Commerce

Em relação a integração com o sistema de gestão de base de dados escolhido em cada uma das fases nenhuma das linguagens apresenta qualquer tipo de dificuldade.

Analisando a complexidade de cada linguagem, obtêm-se um equilíbrio relativamente frágil. O ASP.NET destaca-se de longe do ASP e apresenta as mesmas vantagens do PHP. No entanto, o mod_perl apresenta características que o permite classificar como demasiado complexo para o problema em questão.

A nível de facilidade de utilização e curva de aprendizagem o PHP, o ASP e o ASP.NET destacam-se pela positiva. Isto indica que o mod_perl, sendo escolhido, acarretaria um tempo de aprendizagem da linguagem. Com as condicionantes de tempo de um projecto deste tipo e dimensão, o mod_perl ganha aqui um ponto negativo que se torna difícil de colmatar.

Numa análise mais próxima entre ASP, ASP.NET e PHP torna-se necessário referir algumas características importantes. O ASP e o ASP.NET suportam múltiplas linguagens de programação. Este tipo de soluções torna a arquitectura inerentemente mais lenta e mais consumidora de memória. O código gerado por cada uma das linguagens apresenta tamanhos diferentes. O PHP destaca-se como o mais sucinto, seguido do ASP.NET e só depois o ASP [76].

Tendo em conta as necessidades actuais e potenciais do sistema a ser desenvolvido, o ASP e ASP.NET apresenta mais problemas devido à necessidade de utilização de software proprietário para implementar algumas das funcionalidades. Estas funcionalidades encontram-se disponíveis nas restantes linguagens (mod_perl e PHP).

A escolha lógica acaba por ser o PHP, especialmente devido às condicionantes de servidor que eliminam as opções do ASP e ASP.NET.

3.4 Base de Dados

A escolha de base de dados apresentava um requisito adicional: a obrigatoriedade do Oracle. Esta condicionante devia-se à existência prévia de infra-estruturas no ambiente de instalação. Apesar de tudo não foi disponibilizada em tempo útil a informação necessária acerca da versão e configuração do Oracle. Esta situação reforçou a necessidade de dividir o desenvolvimento em duas fases. Numa primeira fase seria utilizada uma base de dados que apresentasse os requisitos necessários. Numa segunda e última fase seria feita a migração da estrutura e dados para a base de dados Oracle existente na GaiaSocial.

A base de dados escolhida para a primeira fase foi o Microsoft Access. De seguida como ditava a imposição foi utilizado o sistema de gestão de base de dados da Oracle.

3.4.1 Microsoft Access

O Microsoft Access [4] é a base de dados disponibilizada pela Microsoft em vários pacotes de software dos quais se destaca o mais comumente utilizado o Microsoft Office. A acessibilidade e facilidade de utilização desta ferramenta tem vindo a conquistar adeptos no desenvolvimento de produtos de pequena e média dimensão.

O Access tem vindo a apresentar um aumento de funcionalidades para acompanhar o surgir de novas tecnologias, como por exemplo: o suporte de XML. O público-alvo, menos técnico, a que se destina não tem descurado um conjunto de funcionalidades

necessárias para utilizadores mais avançados, tais como: ligação à várias fontes de dados desde bases de dados em Access ou diferentes até folhas de Excel.

A interoperabilidade entre as várias ferramentas do pacote a que normalmente pertence, tem acrescentado ainda mais funcionalidades, tais como: facilidade de publicação e importação e exportação de dados em formatos normalizados.

Um vasto conjunto de características tem mantido o Microsoft Access como uma ferramenta de trabalho amplamente utilizada e com uma apresentação e facilidade de utilização incomparável.

3.4.2 Oracle Database

A Oracle Database é um dos produtos da Oracle mais conceituados no mercado dos sistemas de gestão de bases de dados a nível mundial.

A capacidade da base de dados em lidar com um grande número de dados de uma forma de tal modo eficiente que o desempenho obtido, só pode ser comparado às várias ferramentas que orbitam em torno do produto.

Uma das características por vezes negativa que pode ser apontada à Oracle prende-se com a falta de adequação dos produtos desenvolvidos para uma grande parte dos mercados. Isto porque a complexidade apresentada pela própria base de dados supera, normalmente em larga escala a complexidade do problema.

A variedade de soluções e todo um conjunto de funcionalidades que permitem a resolução de qualquer tipo de problema tem permitido à Oracle manter o lugar de prestígio alcançado até agora.

3.4.3 Comparativo e selecção

A análise das soluções apresentadas conduziu uma análise das competências técnicas apresentadas no contexto do desenvolvimento de um produto da natureza do SIMEH.

Um dos critérios em que se baseou a escolha foi a facilidade de criação e implementação de uma base de dados em Microsoft Access. A curva de aprendizagem é quase nula e as funcionalidades requeridas no processo de implementação são mínimas. No caso da criação de uma base de dados Oracle a totalidade de opções e passos necessários é muito superior. Isto implicaria uma longa curva de aprendizagem numa fase em que se tornava necessário apresentar resultados de uma forma expedita.

Outro critério importante consiste na disponibilidade do Microsoft Access em todos os computadores existentes no ambiente de desenvolvimento permitindo, dessa forma, uma interacção mais directa com os dados. Por exemplo, se fosse detectada alguma inconsistência a facilidade de utilização e a interface amigável do Access permitiriam a correcção e visualização dos possíveis erros. No caso do Oracle [5] seria necessário instalar a base de dados no computador destino sendo necessária a instalação de um cliente Oracle ou a instalação completa do pacote. Para além disto a visualização dos dados carece de uma interface amigável e intuitiva.

O espaço ocupado por uma base de dados Access é praticamente nulo quando comparado com o de uma base de dados Oracle. Isto facilita o transporte e controlo de versões da própria base de dados.

Uma base de dados em Access é totalmente concentrada num só ficheiro o que, mais uma vez, permite uma portabilidade superior. O Oracle necessita de um conjunto de ficheiros e de alguns critérios na passagem de uma base de dados de um computador para outro.

A execução do Microsoft Access, para manipular os dados, é de uma simplicidade limitativa. No entanto, essa limitação também apresenta vantagens. O Access tem requisitos de memória e de processador muito baixos, comparativamente ao que é, actualmente, disponibilizado pelos computadores. Pelo contrário, o Oracle tem interfaces muito pesadas e com uma complexidade muito superior à requerida para uma base de dados da natureza da do SIMEH.

Porém, existem produtos que apresentam uma maior interoperabilidade com o Oracle por utilizarem SQL como linguagem nativa de manipulação da base de dados. Este facto facilitaria a migração de dados. Neste caso, essa transição foi rapidamente resolvida através da criação de uma ferramenta, pela equipa de estágio, que permitia a leitura dos dados da base de dados em Access e consequente criação de scripts, que permitiam a introdução dos dados na base de dados Oracle.

Em suma, o Microsoft Access apresentou-se como um meio para atingir um fim, obedecendo a todos os critérios que eram exigidos numa fase inicial de idealização e prototipagem da aplicação final.

Como foi referido anteriormente, a base de dados final foi o Oracle. O Oracle é um dos sistemas de gestão de bases de dados comercial mais utilizados e conceituados. No caso da base de dados do SIMEH, o Oracle apresenta muito mais funcionalidades do que aquelas que alguma vez serão exigidas. No entanto, a quantidade de dados a serem manipulados apresentam uma ordem de grandeza que justifica a utilização desta ferramenta.

4 Arquitectura do SIMEH

Os vários requisitos da arquitectura do SIMEH foram sendo apresentados ao longo de várias reuniões. Foram estabelecidos 3 cenários de utilização: utilizador na GaiaSocial, utilizador no gabinete de apoio social e morador. Existem duas interfaces: aplicação e web. A interface aplicação corre em máquinas com o sistema operativo Windows 2000 e utilizam o browser Internet Explorer para a interface web. A base de dados encontra-se na GaiaSocial e é partilhada pelas duas interfaces.

A definição da arquitectura de um sistema recorre a vários tipos de representações (Anexo A).

Numa primeira abordagem, é analisada a arquitectura lógica do sistema. O diagrama de pacotes permite visualizar a organização em módulos lógicos, especificação de interfaces e dependências entre eles sendo, por isso, o escolhido para a modelação em questão.

De seguida, numa segunda abordagem, é analisada a arquitectura física do sistema. O diagrama de distribuição permite capturar a topologia de *hardware* do sistema sobre a qual são executados os componentes de *software* sendo, por isso, escolhido para a modelação em questão.

4.1 Arquitectura Lógica

A arquitectura de um sistema que integra duas interfaces apresenta algumas particularidades nem sempre vulgares nos sistemas desta natureza. Existem 4 camadas e praticamente todas dependem exclusivamente da camada imediatamente inferior.

A primeira camada é a camada de Apresentação. Esta camada encontra-se nas duas interfaces da aplicação. No SIMEH – WEB esta camada é apresentada através de um browser web que executa código XHTML e CSS. No SIMEH – APLICAÇÃO esta camada é representada através de uma aplicação que corre numa plataforma Win32 implementada com o Borland Delphi e componentes visuais.

A segunda é a de Lógica de Negócio, esta camada agrega todas as funcionalidades representativas do sistema. Existe uma interdependência entre as camadas das duas interfaces, devido à utilização de algumas funcionalidades presentes no SIMEH – APLICAÇÃO por parte do SIMEH – WEB. No SIMEH – WEB esta camada foi implementada através de XHTML e PHP. No SIMEH – APLICAÇÃO é desenvolvida através do Borland Delphi.

A terceira é a de Ligação à Base de Dados, esta camada permite estabelecer a ligação com a estrutura de dados subjacente ao sistema. No SIMEH – WEB traduziu-se na utilização do módulo OCI, distribuído com o PHP. No SIMEH – APLICAÇÃO foi utilizado um componente de ligação ao Oracle.

A quarta e última é a de Base de Dados. Esta camada baseia-se no SGBD do Oracle, funcionando como um repositório de toda a informação do SIMEH.

Torna-se importante referir que na fase SIMEH – PROTÓTIPO, a arquitectura utilizada era semelhante à parte que se refere ao SIMEH – APLICAÇÃO. Não havendo no entanto ainda uma interacção com a base de dados.

Outro dado importante prende-se com o facto de que durante o desenvolvimento, a estrutura de base de dados basilar do sistema foi o Microsoft Access, que de igual modo era partilhado pelas duas interfaces. No entanto esta arquitectura lógica representa a arquitectura final ou pretendida do sistema.

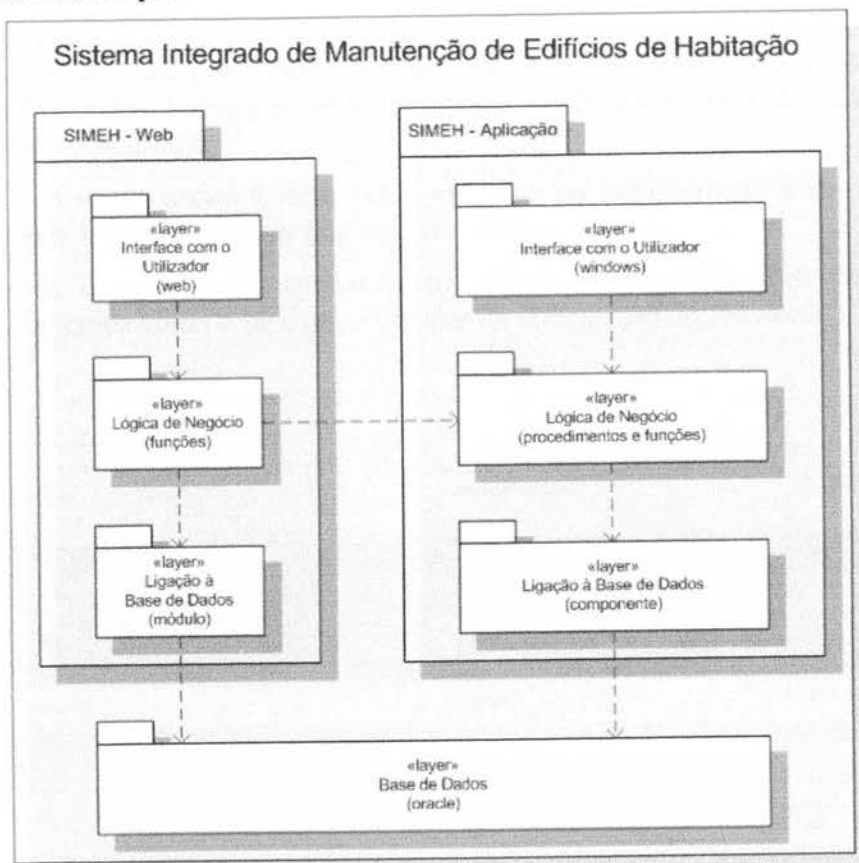


Figura 1: Diagrama de pacotes da arquitectura lógica

4.2 Arquitectura Física

A arquitectura física apresenta um diagrama com uma complexidade adicional, devido à agregação dos componentes com objectivos semelhantes. As várias camadas encontram neste diagrama a sua disposição.

Analisando do ponto de vista dos utilizadores, existem 4 perfis de utilização: Administrador, Técnico, Assistente Social e Morador. Apenas o morador é que se encontra limitado nos acessos ao sistema, podendo apenas interagir com o SIMEH – WEB através de um browser. Os outros utilizadores podem aceder às duas interfaces: Aplicação e Web.

Do ponto de vista físico, também se torna importante referir as máquinas que cada utilizador usa para aceder ao SIMEH. O Administrador e o Técnico acedem de máquinas da própria rede interna da GaiaSocial. Enquanto que o Assistente Social acede de um dos vários gabinetes de apoio social afectos a cada empreendimento. O morador, por sua vez, acede de qualquer máquina com ligação à Internet.

A máquina servidora apresenta as restantes camadas descritas no diagrama da arquitectura lógica com apenas um pacote, Páginas Web Dinâmicas, que pertence não

só à camada de apresentação como à de lógica de negócio. Este pacote representa o servidor Web (Apache) que permite gerar as páginas que se apresentam nos browsers das várias máquinas.

Para facilitar a modelação os restantes pacotes agregam os vários componentes das restantes camadas, excepto o do repositório de informação. O pacote de componentes de lógica de negócio é composto pelos módulos das duas interfaces, no SIMEH – WEB foi utilizado o PHP como linguagem de programação das funções sendo utilizado o Delphi para o SIMEH – APLICAÇÃO.

A ligação à base de dados é feita pelos módulos ou componentes disponíveis para estabelecer esta ligação, nas duas linguagens utilizadas.

Na camada de Base de dados apenas se encontram o servidor de Base de Dados da Oracle que funciona como o repositório basilar de informação do sistema.

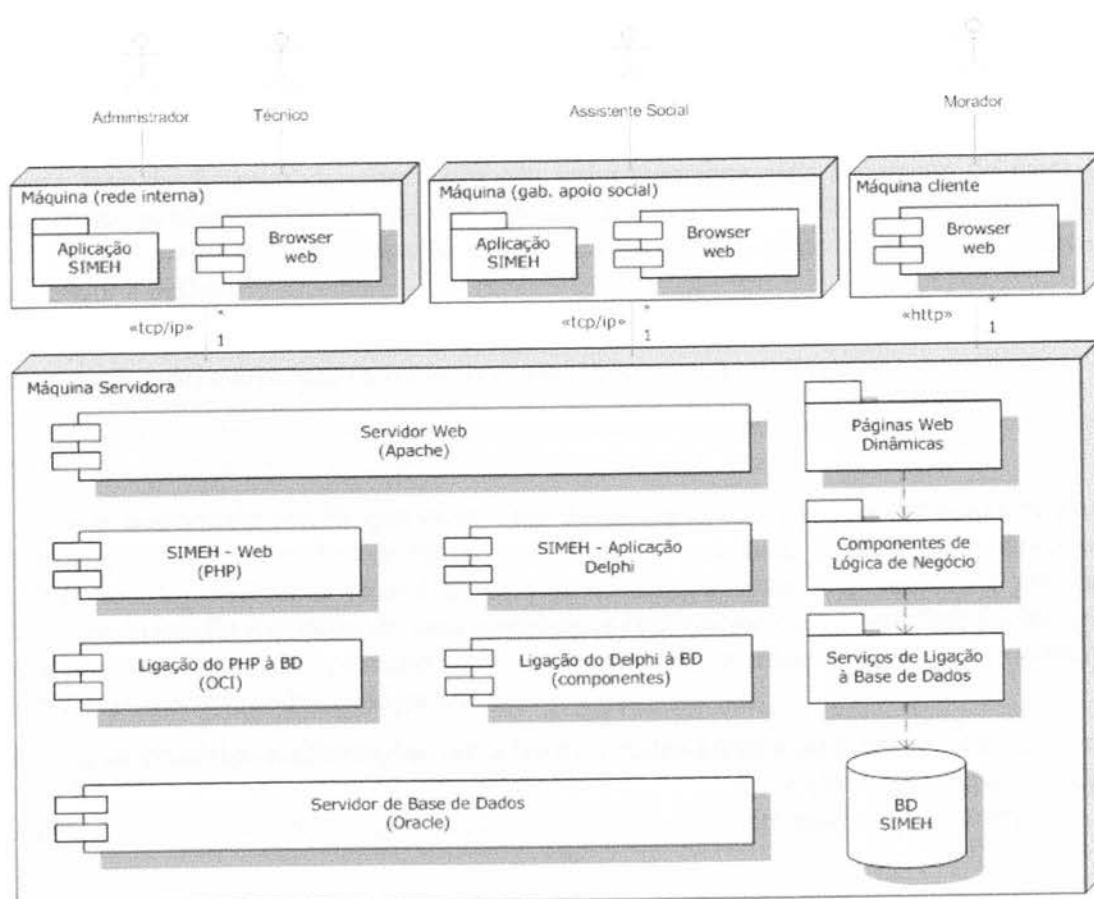


Figura 2: Diagrama de distribuição da arquitectura física

5 Desenvolvimento do SIMEH — Protótipo

A fase de protótipo, apesar de poder ser considerada uma fase pré-estágio, permitiu contextualizar o projecto SIMEH. Nesta fase, atingiram-se dois objectivos: aprendeu-se uma parte da linguagem do domínio e definiu-se um ponto de partida para o SIMEH.

Ao longo de várias reuniões foram discutidas estratégias e houve uma tentativa de continuar o trabalho existente. Após uma análise cuidada do trabalho desenvolvido foi necessário definir um novo ponto de partida e redefinir objectivos.

O protótipo desenvolvido pretendia simplesmente tornar visível algumas das escolhas e conclusões definidas no decurso das várias reuniões de especificação de requisitos.

A análise de requisitos seguinte não demonstra a evolução dos requisitos apresentados, mas sim uma base para o projecto que viria a ser continuado posteriormente.

Para especificar os requisitos do protótipo foram usados diagramas de casos de uso (Anexo A). Estes diagramas permitem mostrar as funcionalidades do sistema ignorando a sua organização interna. De igual forma, ajudam a especificar o contexto do sistema e capturar os requisitos funcionais do mesmo.

Nesta fase do desenvolvimento ainda não tinha sido feita uma separação de áreas de utilização, pelo que existe apenas um tipo de utilizador (actor). No entanto já tinha sido referida a necessidade de autenticação no sistema, assim como os possíveis tipos de utilizador a definir.

O objectivo final deste sistema é integrar uma quantidade de serviços e fontes de informação necessárias para a correcta gestão de um empreendimento.

O protótipo a desenvolver deve ter uma apresentação do tipo *Single Document Interface* (SDI) para facilitar a leitura da informação a consultar.

Torna-se importante referir que nesta fase do projecto não foi tido em conta nenhum critério de usabilidade (Anexo B) na construção da interface. Esta situação deve-se a dois factos: foi dado mais valor à escolha da informação a ser disponibilizada em cada menu em oposição à criação de uma interface usável e acessível; a interface foi baseada num protótipo feito anteriormente tendo como exemplo as imagens dos botões do painel de ferramentas disponibilizado na altura.

Na fase de protótipo a interacção com a base de dados ainda é inexistente. Isto deve-se à metodologia de desenvolvimento aplicada (pequenos ciclos de especificação, implementação e validação), que manteve a base de dados em mutação constante.

5.1 Introdução

Os 3 aspectos mais importantes para uma gestão correcta são: cadastro técnico, cadastro social e cadastro económico. Cada um destes permite visualizar a totalidade das informações obtidas a partir dos outros módulos. Alguns módulos, a implementar posteriormente, vão fornecer a base para os cálculos e visualizações referidas.

Este protótipo é constituído por dois painéis de acesso às ferramentas e opções que o compõem.

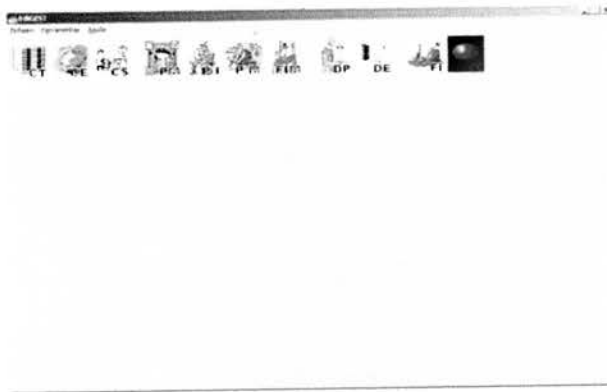


Figura 3: Interface principal do Protótipo

Um dos acessos é constituído pelo painel superior, que é habitual em qualquer aplicação Win32. Este painel é constituído pelas seguintes opções: Ficheiro, Ferramentas e Ajuda. Cada uma destas opções consiste num agregado de opções.

A opção de menu *Ficheiro* apresenta como opções: *Imprimir* e *Sair*. Como o próprio nome indica a opção *Imprimir* permite transpôr para papel o que está a ser visualizado no ecrã. No entanto, nesta fase de desenvolvimento, esta opção não chegou a ser implementada. A opção *Sair* permite fechar a janela do protótipo. Esta opção foi obviamente implementada, visto não necessitar de nenhum critério especial ou aprofundamento de requisitos.

A opção de menu *Ferramenta* apresenta como escolha única *Opções*. Este menu e a opção que o compõe são habituais em aplicações. Neste caso, permite configurar opções de ligação, utilização ou opções intrínsecas à aplicação. No entanto, nesta fase de desenvolvimento, não chegou a ser especificada e implementada, tendo sido apenas sugerida pela equipa de análise e implementação.

A opção de menu *Ajuda* é composta pelos seus dois habituais componentes: *o acerca de* e *o ajuda do projecto*. *O acerca de* permite informar o utilizador das informações de desenvolvimento e produção do programa. A opção de ajuda é uma das mais importantes de qualquer aplicação, visto facilitar a utilização e reduzir a curva de aprendizagem necessária. No entanto, assim como outras opções, foi relegada para segundo plano tendo em conta a fase de desenvolvimento em que se inseria.

O outro painel principal, que pode ser apelidado de barra de ferramentas, é constituído pelos seguintes botões: *Cadastro Técnico*, *Cadastro Económico*, *Cadastro Social*, *Procedimentos de Manutenção*, *Plano de Intervenções*, *Plano de Manutenção*, *Fichas de Inspeção e Manutenção*, *Diagnóstico Preliminar*, *Diagnóstico Específico*, *Fichas de Intervenção e Reclamações*. Os botões anteriormente referidos podem, e devem, ser agregados em 4 grupos fundamentais: *Gerir Cadastros*, *Gerir Manutenções*, *Gerir Intervenções* e *Gerir Reclamações*.

As funcionalidades referidas encontram-se representadas no diagrama de casos de uso que se segue.

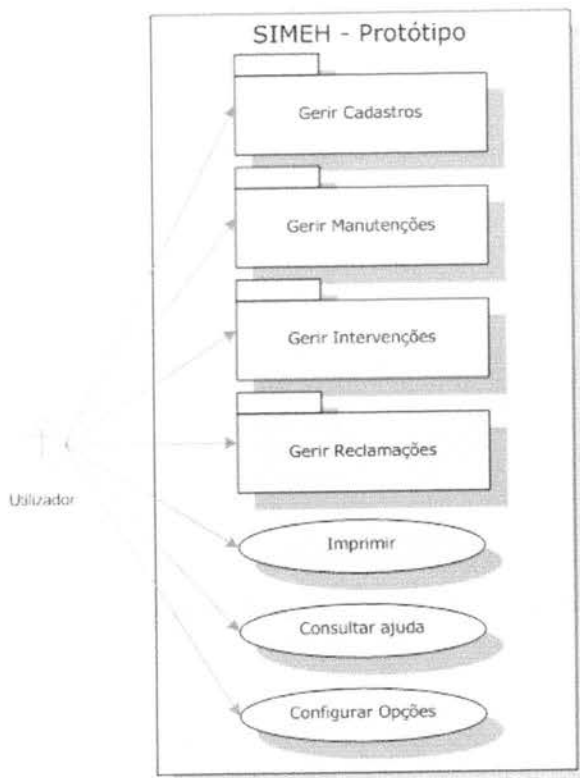


Figura 4: Diagrama de casos de uso de alto nível do protótipo

5.2 Gerir Cadastros

Este item é, como foi referido anteriormente, uma agregação de várias funcionalidades. A necessidade desta agregação deve-se à complexidade potencial do programa a desenvolver.

No protótipo desenvolvido os cadastros representam a informação, geralmente tabular, recolhida dos dados anteriormente inseridos. Estas visualizações permitem obter uma visão facilitada de cada dado individual sem prescindir da visão da totalidade dos dados.

O pacote de casos de utilização Gerir Cadastros é constituído por: Cadastro Técnico, Cadastro Social e Cadastro Económico.

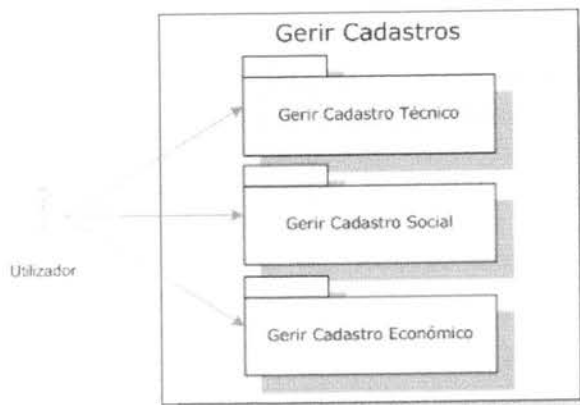


Figura 5: Diagrama de casos de uso do pacote Gerir Cadastros

5.2.1 Gerir Cadastro Técnico

O pacote de casos de utilização Cadastro Técnico agrega outros dois pacotes: Gerir Arquivo de Projecto e Gerir Dados Cadastrais.

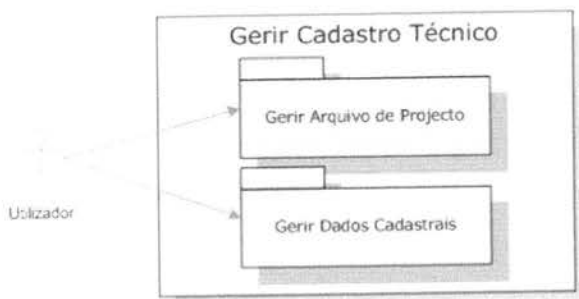


Figura 6: Diagrama de casos de uso do pacote Gerir Cadastro Técnico

Gerir Arquivo de Projecto

O pacote de casos de utilização Arquivo de Projecto agrega outros dois pacotes: Gerir Projecto e Gerir Documentos. Este pacote serve como repositório de toda a documentação associada a um dado empreendimento. A transição entre as duas opções é feita pela pressão do botão respectivo que se encontra no topo esquerdo do menu.



Figura 7: Diagrama de casos de uso do pacote Gerir Arquivo de Projecto

O pacote de casos de utilização Gerir Projecto permite inserir, visualizar, editar e eliminar documentos. As funcionalidades referidas anteriormente não apresentam ainda um procedimento ou forma definidas tendo sido apenas registados os requisitos em si. A árvore que se encontra no lado esquerdo representa a normalização usada na organização dos documentos. Os documentos armazenados neste menu são de carácter técnico, como por exemplo: o desenho da peça de uma das tubagens usadas na drenagem de águas e esgotos.



Figura 8: Diagrama de casos de uso do pacote Gerir Projecto

Foram referidos alguns pormenores acerca de cada ficheiro. Devem ter associados os seguintes pormenores: um ícone, um nome e uma descrição. O ícone deve representar o tipo de ficheiro. A descrição, definida no momento da inserção, será disponibilizada na opção acerca. O nome deverá ser normalizado. O ficheiro pode ser aberto para consulta não tendo sido definida a necessidade, ou não, da consulta ser dentro do programa. Os ficheiros podem ser adicionados a qualquer nível da árvore para permitir uma maior flexibilidade na adaptação à normalização imposta.

A implementação alcançada neste menu apenas serviu para facilitar a visualização do produto final e testar a conjugação dos elementos em si.



Figura 9: Interface do Arquivo de Projecto do módulo Projecto

O pacote de casos de utilização Gerir Documentos permite inserir, visualizar, editar e eliminar documentos. As funcionalidades referidas anteriormente não apresentam ainda um procedimento ou forma definidas tendo sido apenas registados os requisitos em si. A árvore que se encontra no lado esquerdo representa a normalização usada na organização dos documentos.

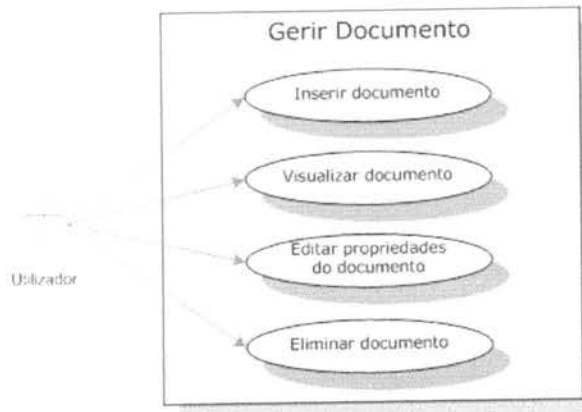


Figura 10: Diagrama de casos de uso do pacote Gerir Documento

Os pormenores definidos para o menu anterior, Gerir Projecto, são também aplicáveis neste menu.

A implementação alcançada neste menu apenas serviu para facilitar a visualização do produto final e testar a conjugação dos elementos entre si.

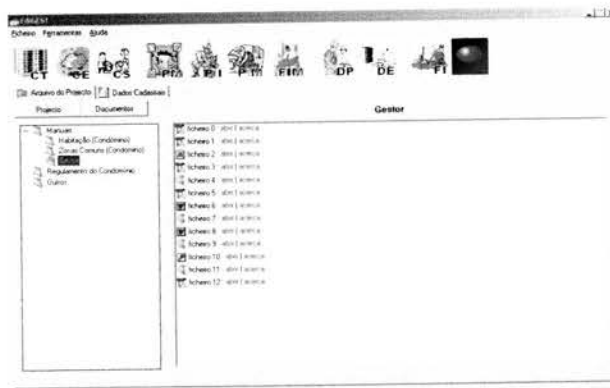


Figura 11: Interface do Arquivo de Projecto do módulo Documentos

Gerir Dados Cadastrais

O pacote de casos de utilização Dados Cadastrais agrega outros pacotes: Gerir Caracterização dos Custos, Gerir Caracterização Construtiva e Gerir Caracterização das Intervenções. A transição entre estas opções é feita através da *combo-box* situada em cima do lado direito.

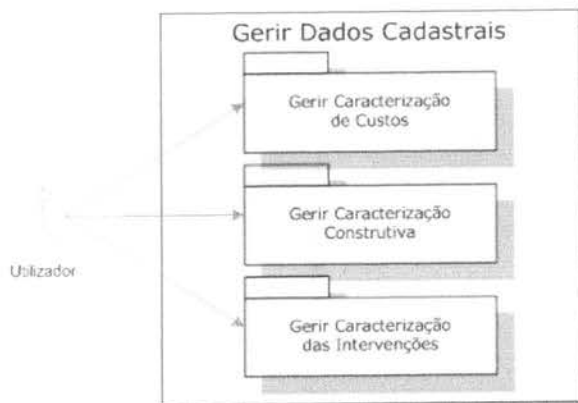


Figura 12: Diagrama de casos de uso do pacote Gerir Dados Cadastrais

Dos pacotes acima mencionados apenas a caracterização dos custos foi especificada. Os outros partilhavam a mesma interface com alguns pormenores diferentes, que acabaram por ser relegados para segundo plano nesta fase de desenvolvimento.

O pacote de casos de utilização Gerir Caracterização dos Custos permite introduzir a localização da fracção a analisar, inserir, visualizar, editar e eliminar os campos da tabela visualizada. A escolha da fracção é feita através das *combo-boxes* que se encontram acima da tabela.

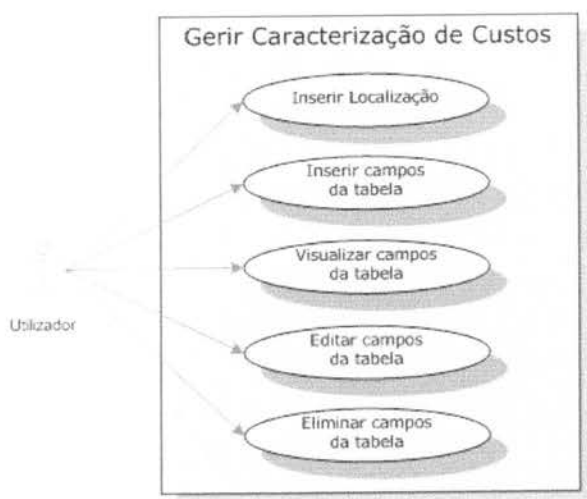


Figura 13: Diagrama de casos de uso do pacote Gerir Caracterização de Custos

As funcionalidades de inserção, edição e eliminação, referidas anteriormente, não apresentam ainda um procedimento ou forma definidas tendo sido apenas registados os requisitos em si. A tabela e respectivos componentes tiveram a interface baseada nas tabelas normalizadas definidas pelos elementos da equipa associada ao projecto SIMEH. Cada registo é constituído por dois elementos: um deles era texto desnormalizado e outro era um conjunto de caracteres cuja legenda seria fornecida posteriormente. O segundo elemento, por ser normalizado, devia ser inserido através de uma *combo-box*.

A implementação alcançada neste menu apenas serviu para facilitar a visualização do produto final e testar a conjugação dos elementos entre si.

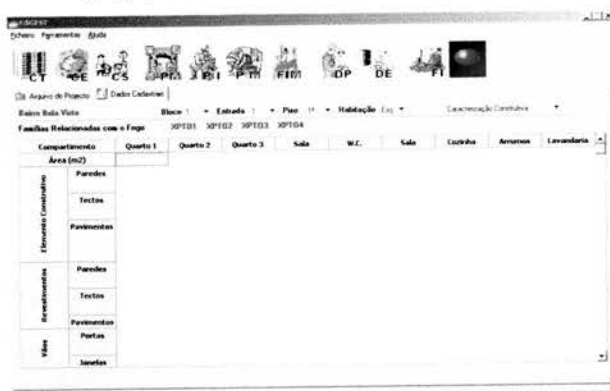


Figura 14: Interface do Cadastro Técnico do módulo Dados Cadastrais

5.2.2 Gerir Cadastro Social

O pacote de casos de utilização Cadastro Social permite inserir, visualizar, editar e eliminar dados do indivíduo; permite também visualizar e editar dados do agregado familiar.

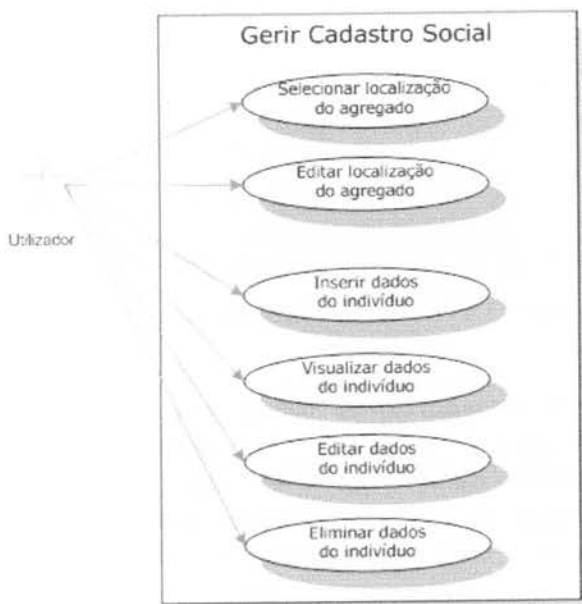


Figura 15: Diagrama de casos de uso do pacote Gerir Cadastro Social

Nesta fase de desenvolvimento, a interface deste menu foi completamente baseada nas tabelas obtidas da normalização efectuada pela equipa do SIMEH. Logo, existem vários procedimentos e formas que não foram devidamente especificados tendo sido apenas registados os respectivos requisitos.

A interface sugerida apresenta vários campos com *check-boxes* e permite visualizar os dados do indivíduo e do agregado a que pertence.

The screenshot shows a software window titled 'Cadastro Social'. It contains several input fields: 'Nome', 'Idade', 'Sexo', 'Ligação ao Arrendatário', 'A/NA', 'Observações', and 'Ref.'. Below these are two tables for educational qualifications. The first table, 'Habilitações Literárias', lists levels from 1 to 8. The second table, 'Categoria Profissional', lists levels from 1 to 3. Both tables have columns numbered 1 to 10, likely representing different social indicators or reference points.

Figura 16: Interface do Cadastro Social

5.2.3 Gerir Cadastro Económico

O pacote de casos de utilização Cadastro Económico agrega outros dois pacotes: Gerir Custo Inicial e Gerir Custo de Ciclo de Vida.

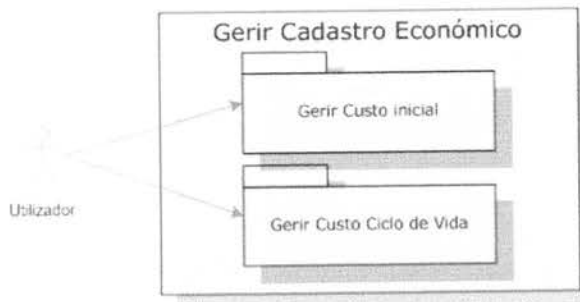


Figura 17: Diagrama de casos de uso do pacote Gerir Cadastro Económico

Estes dois pacotes de casos de uso, apesar de partilharem algumas semelhanças na interface, são especificados em fases diferentes, porque o custo inicial de um projecto pode ser definido independentemente, sendo um módulo autónomo. Pelo contrário, o custo de ciclo de vida é um custo baseado nas obras de intervenção, manutenção e outros custos genéricos agregados, o que obriga a uma consulta de informação de módulos ainda não especificados.

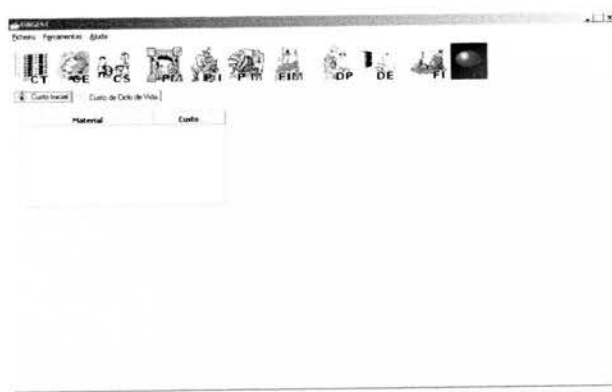


Figura 18: Interface do Cadastro Económico

Gerir Custo Inicial

O pacote de casos de utilização Gerir Custo Inicial permite inserir, visualizar, editar e eliminar custos. As funcionalidades referidas anteriormente não apresentam ainda um procedimento ou forma definidas tendo sido apenas registados os requisitos em si. A cada custo são indicadas duas características: a designação do material e o custo associado. Não foi indicada nenhuma forma de organização dos custos apresentados.

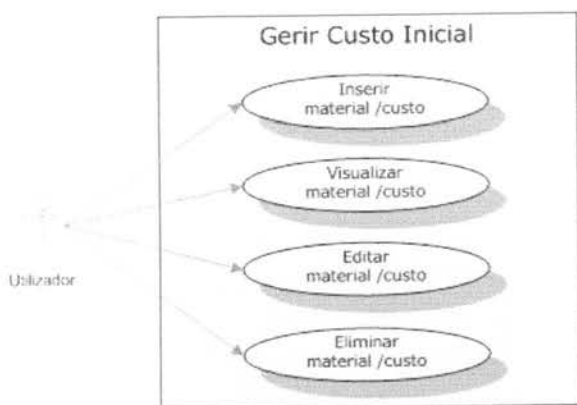


Figura 19: Diagrama de casos de uso do pacote Gerir Custo Inicial

A implementação alcançada neste menu apenas serviu para facilitar a visualização do produto final e testar a conjugação dos elementos entre si. Foi apenas implementado, neste protótipo, o Custo Inicial relegando o Custo Ciclo de Vida para uma fase posterior.

5.3 Gerir Manutenções

O pacote de casos de utilização Gerir Manutenções agrega 3 outros pacotes: Gerir Procedimentos de Manutenção, Gerir Plano de Manutenção e Gerir Fichas de Inspeção e Manutenção.

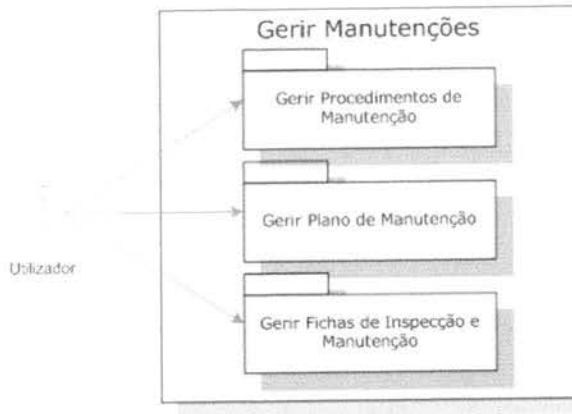


Figura 20: Diagrama de casos de uso do pacote Gerir Manutenções

Nenhum dos pacotes de casos de utilização acima mencionados foi especificado com mais profundidade. Isto não permite tecer comentários ou informações acerca de qualquer um deles.

5.4 Gerir Intervenções

O pacote de casos de utilização Gerir Intervenções agrega 4 outros pacotes: Gerir Plano de Intervenção, Gerir Diagnóstico Preliminar, Gerir Diagnóstico Específico e Gerir Fichas de Intervenção.

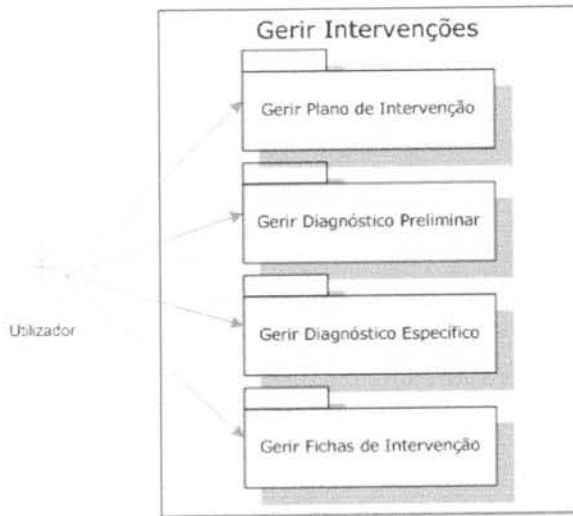


Figura 21: Diagrama de casos de uso do pacote Gerir Intervenções

Nenhum dos pacotes de casos de utilização acima mencionados foi especificado com mais profundidade. Isto não permite tecer comentários ou informações acerca de qualquer um deles.

5.5 Gerir Reclamações

O pacote de casos de utilização Gerir Reclamações nunca chegou a ser especificado neste protótipo. Foi, no entanto, referido que grande parte do módulo de reclamações iria ser disponibilizado na Internet.

Desenvolvimento do SIMEH — Aplicação

A fase aplicação foi sem dúvida uma das mais importantes do estágio, tendo as tarefas necessárias sido separadas pelos vários membros da equipa de desenvolvimento.

A fase correspondente ao SIMEH – APLICAÇÃO, consistiu numa especificação de requisitos e definição de uma arquitectura e estrutura basilar para facilitar o trabalho a ser executado pelo resto da equipa de desenvolvimento. De salientar o trabalho que foi desenvolvido tendo em conta a complexidade e dimensão do problema.

Assim como foi feito para a fase de protótipo, para especificar os requisitos foram usados diagramas de casos de uso. Possibilitando desta forma compreender o sistema no que diz respeito a funcionalidades e contexto.

Nesta fase de desenvolvimento já tinham sido definidos os tipos de utilizador do SIMEH — Aplicação. A definição desses perfis de utilização serviu também para definir as áreas acessíveis por cada um dos utilizadores. No entanto, a autenticação ainda não limitava os acessos a cada um dos menus, tendo ficado esse objectivo para a parte final do desenvolvimento do projecto.

Existem 3 tipos de utilizadores: administrador, técnico e assistente social. O administrador é o utilizador que gere a totalidade do sistema mantendo o seu bom funcionamento e tendo acesso a todos os menus. O técnico é o utilizador cuja principal função é preencher alguns dos principais repositórios de informação do sistema tendo acesso a todos os menus, excepto os que se referem ao cadastro social por uma questão de confidencialidade de dados sensíveis. O assistente social é um dos utilizadores para os quais se destina o sistema tendo acesso a todos os menus necessários à sua função dentro da GaiaSocial.

O sistema a desenvolver deve ter uma apresentação do tipo *Multiple Document Interface* (MDI) para permitir a comparação ou cruzamento de informação de documentos diferentes.

Assim como na fase anterior, o objectivo primordial era definir a informação a apresentar em cada um dos menus e utilizar um esboço de estrutura visual. Logo, não foi tido em conta nenhum critério de usabilidade.

5.6 Introdução

O sistema a ser desenvolvido baseia-se num conceito simples de introdução de dados em repositórios de informação para posterior visualização, utilização e manipulação pelos utilizadores finais.

O sistema pode ser dividido em 10 módulos fundamentais: Gerir Sistema, Gerir Recursos, Gerir Montagem do Empreendimento, Gerir Cadastros, Gerir Manutenções, Gerir Intervenções, Gerir Reclamações, Gerir Pesquisa, Gerir Ferramentas e Gerir Ajuda.

O diagrama permite visualizar as áreas do sistema a que cada utilizador tem direito de aceder.

Os tipos de utilizadores definidos surgiram em conformidade com a estrutura organizacional da GaiaSocial. O administrador tem direito de aceder à totalidade dos módulos, visto ser este o actor que instala e gere o sistema. Este tipo de utilizador é normalmente definido para separar as funções de utilizador das de gestor de sistema. É normalmente este o utilizador que supervisiona a totalidade de funcionalidades e utilização do sistema. O técnico, da mesma forma que o administrador, tem direito de aceder à totalidade dos módulos. No entanto, dentro de alguns desses módulos não pode aceder a todas opções disponíveis. O técnico surge no conjunto de actores como uma figura que introduz grande parte da informação técnica do sistema. Apesar de, na realidade, as competências deste actor serem separadas devido à diversidade de funções que desempenha nesta representação esta função encontra-se agregada apenas para facilitar a modelação do sistema. O assistente social tem direito de aceder apenas a alguns módulos do SIMEH — Aplicação. Os módulos aos quais tem acesso são: Gerir Sistema, Gerir Cadastros e Gerir Reclamações. Este actor é um dos utilizadores principais deste sistema e representa os funcionários que trabalham no apoio aos empreendimentos da GaiaSocial. Esse apoio é dado não só a nível técnico (por exemplo através do encaminhamento de reclamações) como a nível social. Devido a esse facto, o SIMEH — Aplicação apresenta uma forte componente de cadastro social, que vai ser explicado com maior pormenor em secções posteriores.

De notar que uma parte da definição das áreas de utilização se deve ao facto da base de dados do SIMEH armazenar dados sociais sensíveis, sendo por isso o acesso restringido a pessoas qualificadas ou que necessitem da informação recolhida para desempenhar a sua função.

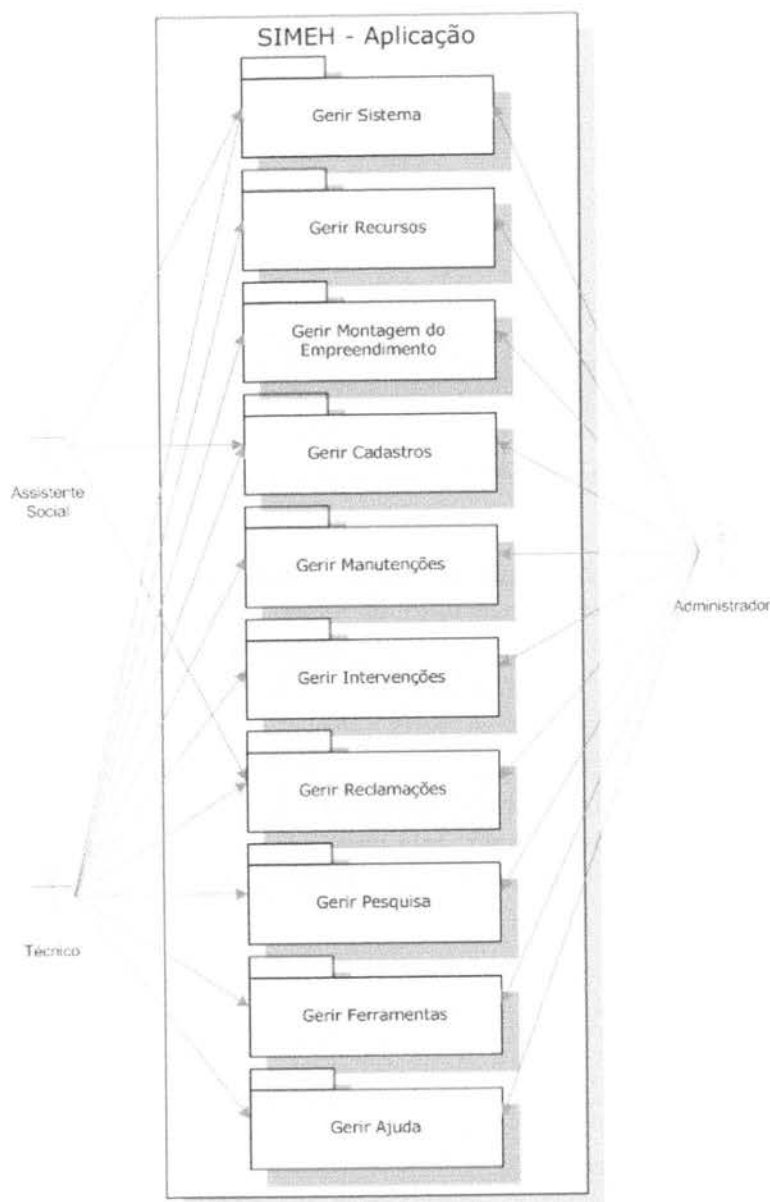


Figura 22: Diagrama de casos de uso de alto nível da aplicação

Como já foi referido, a fase aplicação seguiu-se à fase de protótipo, não aproveitando muito mais do que alguns conceitos da fase anterior. A barra de ferramentas foi retirada devido ao grande acréscimo de menus e funcionalidades. A estrutura dos menus ficou completamente diferente. O sistema ficou com uma interface mais indicada para tirar partido da arquitectura MDI, dispondo de maior espaço de visualização.

No entanto, foram referidos alguns pormenores de usabilidade que deverão ser implementados mais tarde como por exemplo: teclas de atalho ou até mesmo recuperar a barra de ferramentas, personalizando as funcionalidades a que tem acesso.

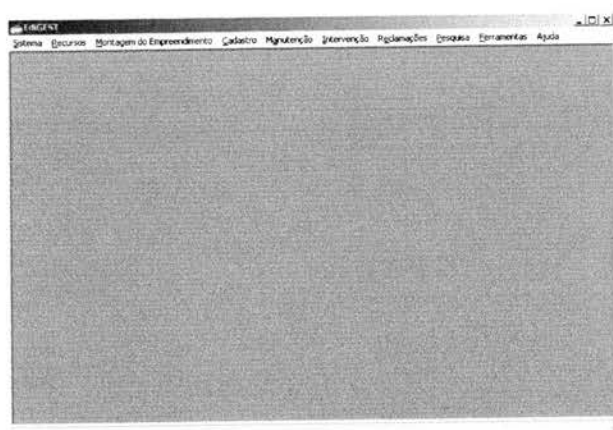


Figura 23: Interface principal do SIMEH — Aplicação

O diagrama que se segue representa o modelo de classes de alto nível da totalidade do sistema. O sistema apresenta 13 áreas de informação consistindo esta separação na melhor forma de compreender a complexidade do SIMEH. As áreas são: permissões da aplicação, cadastro económico, recursos humanos, empreendimento, arquivo de projecto, permissões da web, reclamações, cadastro social, tipologias, montagem construtiva, base de dados construtiva, patologias e intervenções. Em cada uma das explicações dos módulos seguintes uma ou mais destas áreas vão ser abordadas, sendo explicado o conceito de cada uma delas.

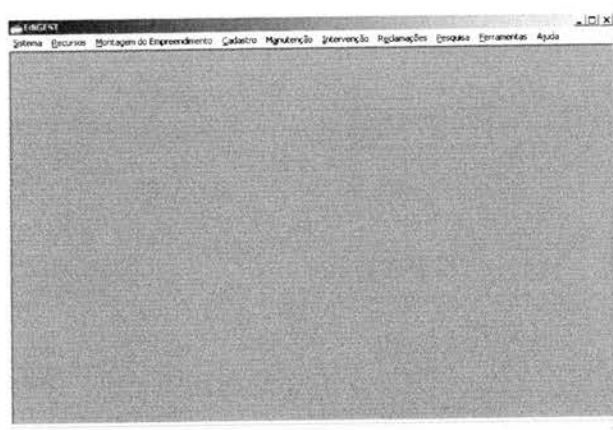


Figura 23: Interface principal do SIMEH — Aplicação

O diagrama que se segue representa o modelo de classes de alto nível da totalidade do sistema. O sistema apresenta 13 áreas de informação consistindo esta separação na melhor forma de compreender a complexidade do SIMEH. As áreas são: permissões da aplicação, cadastro económico, recursos humanos, empreendimento, arquivo de projecto, permissões da web, reclamações, cadastro social, tipologias, montagem construtiva, base de dados construtiva, patologias e intervenções. Em cada uma das explicações dos módulos seguintes uma ou mais destas áreas vão ser abordadas, sendo explicado o conceito de cada uma delas.

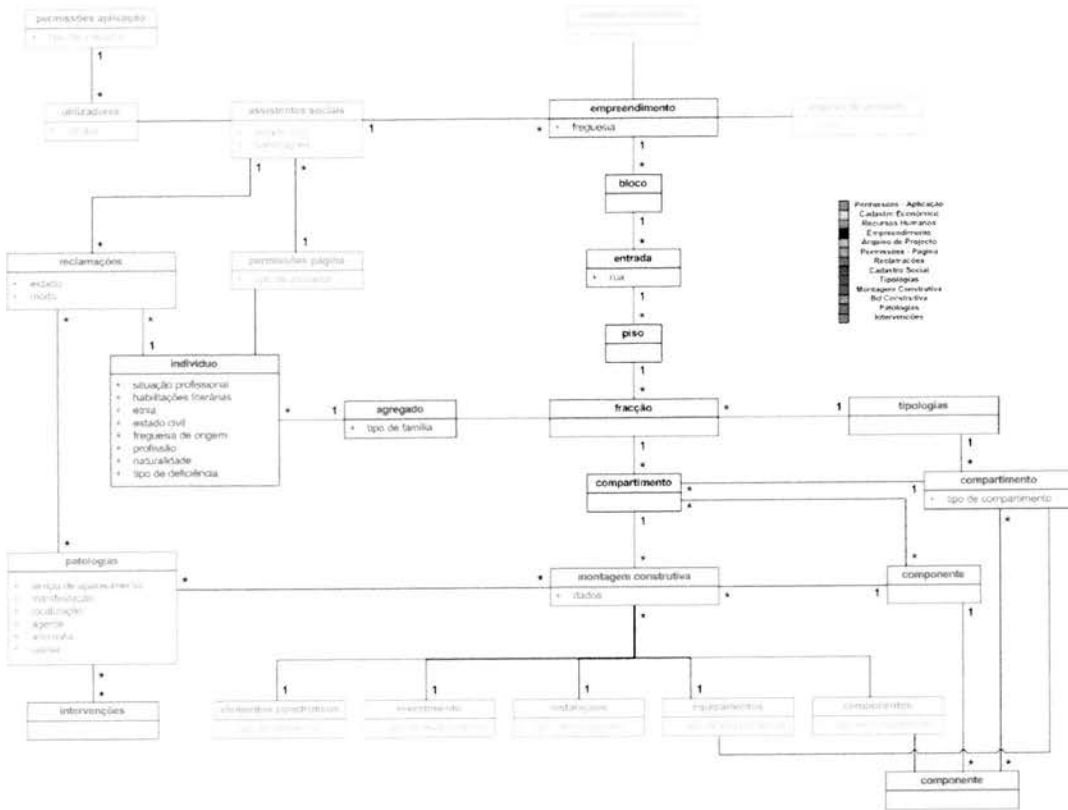


Figura 24: Modelo de Classes de alto nível da aplicação

5.7 Gerir Sistema

O módulo Gerir Sistema permite executar as funcionalidades vulgares numa aplicação Win32. Este módulo é composto por dois pacotes de casos de uso e um caso de uso. Os pacotes de casos de uso são: Gerir Permissões e Gerir Configuração. O caso de uso é o Sair.

O pacote Gerir Permissões permite definir os vários perfis de utilização e introduzir os utilizadores da GaiaSocial. Este módulo pode ser acrescido de mais funcionalidades. No entanto, para isso necessitamos do *feedback* de uma utilização mais intensiva do sistema. Esta opção só pode ser utilizada pelo administrador de sistema.

O pacote Gerir Configuração ainda não foi definido na sua totalidade, tendo a sua existência sido já definida. Neste módulo vão ser agregadas todas as funcionalidades necessárias à instalação do sistema, assim como as que permitam personalizar o sistema. Essas funcionalidades podem ser desde a definição da rede em que se encontra ou pormenores de ligação até pormenores de teclas de atalho e barras de ferramentas. Esta opção, neste momento, só pode ser utilizada pelo administrador de sistema. No entanto, se a configuração apresentar as opções de personalização referidas anteriormente vai ser necessário separar os dois módulos e tornar o módulo de personalização acessível a todos os perfis de utilização.

A opção Sair permite encerrar a aplicação. Esta opção está acessível, como seria de esperar, a todos os perfis de utilização.

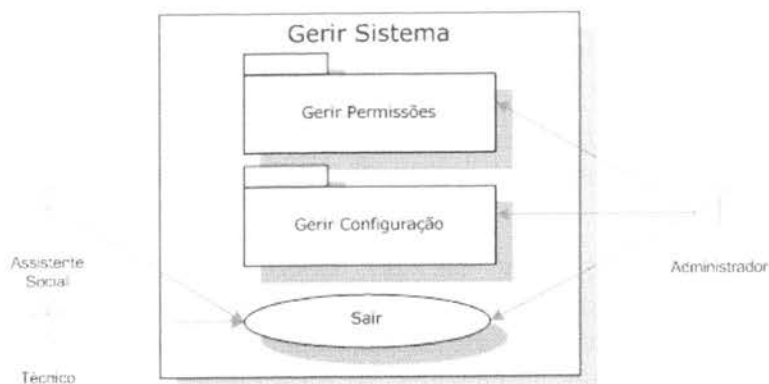


Figura 25: Diagrama de casos de uso do pacote Gerir Sistema

A interface definida até agora permite num passo simples: introduzir, editar ou remover utilizadores.

A introdução de utilizadores é feita numa janela à parte que permite introduzir: *login*, palavra-chave, tipo de utilizador e o cargo que desempenha. É importante salientar que os dois níveis de definição de utilizadores (cargo e tipo de utilizador) foram separados propositadamente para permitir uma maior flexibilidade e adequação à realidade existente no ambiente da GaiaSocial.

A edição e remoção são feitas através da mesma janela da introdução, sendo o caso da remoção sujeito a uma confirmação subsequente.

Este menu não pretende substituir a parte de recursos humanos de uma forma definitiva, mas sim definir utilizadores e perfis de utilização. Deve ser sujeito a alterações aquando de uma definição mais concreta dos recursos humanos afectos ao SIMEH.

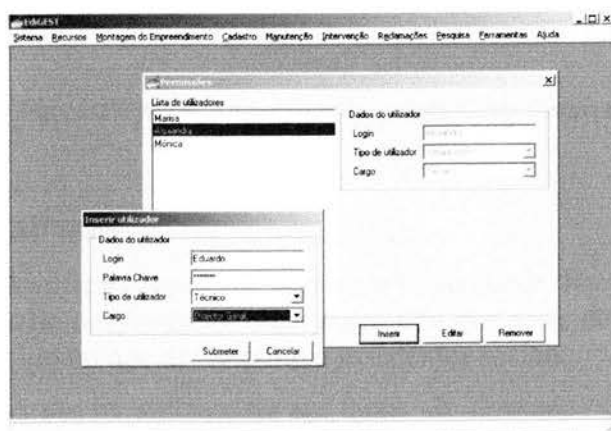


Figura 26: Interface do Gerir Permissões do pacote Gerir Sistema

O modelo de classes representado na figura seguinte permite representar os conceitos relacionados com este módulo e permitem obter uma visão da base de dados do ponto de vista do pacote de casos de uso Gerir Sistema.

Como foi referida na representação das funcionalidades presentes neste pacote os conceitos a representar estão relacionados com permissões, recursos humanos e elementos do cadastro social.

Existem dois tipos de permissões: aplicação e web. Estes dois tipos identificam as duas interfaces do sistema desenvolvido. Cada uma destas permissões apresenta um atributo típico: tipo de utilizador. Este atributo pretende especificar o perfil de utilização dos utilizadores registados.

Dos recursos humanos, como vai ser visto posteriormente, apenas as assistentes sociais é que já se encontram completamente especificadas e implementadas. O conceito de utilizador nos recursos humanos permitiu introduzir dois níveis de definição. Um primeiro nível de definição de utilizadores representa o perfil de utilização e o segundo representa o cargo representado pelo utilizador. Isto permite, como foi referido anteriormente, uma flexibilidade na atribuição de cargos e competências suficientemente independente para se adequar à realidade da GaiaSocial.

A ligação do cadastro social à permissão da interface web vai ser mais desenvolvida posteriormente na secção referente ao desenvolvimento do SIMEH — Web. No entanto, esta ligação permite a introdução de moradores cadastrados.

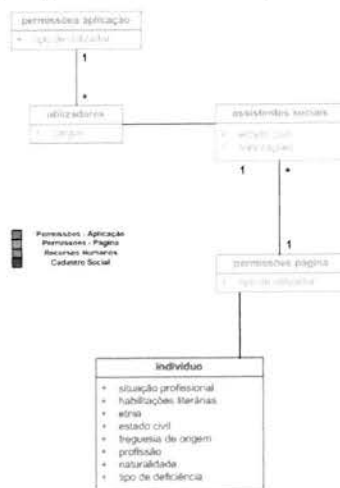


Figura 27: Modelo de classes do pacote Gerir Sistema

5.8 Gerir Recursos

O módulo *Gerir Recursos* permite organizar a totalidade dos recursos humanos da GaiaSocial. Por falta de um *feedback* mais específico por parte do cliente este módulo foi apenas desenvolvido no que diz respeito a assistentes sociais.

As assistentes sociais são os utilizadores aos quais são anexados dados de várias áreas da base de dados, daí a necessidade de uma implementação mais célere.

Devido ao estado ainda pouco desenvolvido deste módulo ainda não existia uma interface disponível.

O modelo de classes representado na figura seguinte permite representar a ligação entre duas classes. As classes são: *assistentes sociais* e *utilizadores*. Esta

ligação que já foi brevemente referida anteriormente permite definir um modelo a seguir pelo resto das classes a colocar neste módulo. Aquando da introdução de mais classes, tais como administradores ou técnicos, a ligação à classe utilizadores deve ser estabelecida da mesma forma como as assistentes sociais. Esta ligação define dois níveis de definição de utilizadores que permite não só uma independência de cargo e tipo de utilizador, como foi referido anteriormente, mas também uma facilidade na agregação de novas classes representativas de funções desempenhadas na GaiaSocial.



Figura 28: Modelo de classes do pacote Gerir Recursos

5.9 Gerir Montagem do Empreendimento

O módulo Gerir Montagem do Empreendimento é um dos mais importantes do SIMEH — Aplicação. Este módulo agrega os repositórios de informação centrais do sistema. É composto por 4 pacotes de casos de uso.

Todas as opções deste módulo podem ser acedidas tanto pelo técnico como pelo administrador.

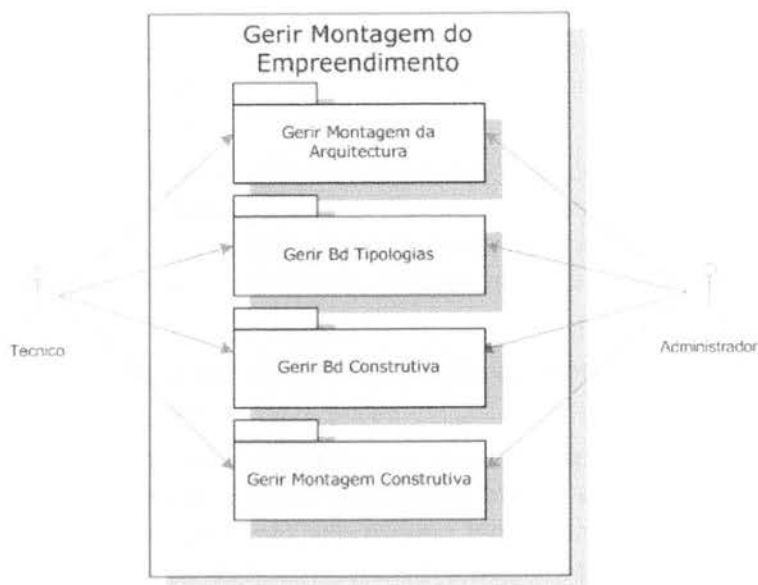


Figura 29: Diagrama de casos de uso do pacote Gerir Montagem do Empreendimento

5.9.1 Gerir Montagem da Arquitectura

O pacote Gerir Montagem da Arquitectura torna possível ao técnico introduzir todos os dados referentes à estrutura habitacional. Este pacote é composto por dois pacotes: Inserir Nova Montagem e Alterar Montagem.

O pacote *Inserir Nova Montagem* consiste na introdução da estrutura do empreendimento assim como os dados que lhe estão associados. Os dados introduzidos são, por exemplo, o nome do empreendimento e os dados do gabinete social que lhe está afecto. A estrutura do empreendimento tem por base o número de blocos, o número de entradas, o número de pisos, o número de fracções e outros dados acessórios. Esta opção pode ser acedida pelo técnico e pelo administrador.

O pacote *Alterar Montagem* consiste na alteração da estrutura previamente introduzida no pacote anterior. A alteração de uma estrutura interfere com muitas tabelas pelo que qualquer alteração está sujeita a uma confirmação. Esta opção pode ser utilizada tanto pelo técnico como pelo administrador.

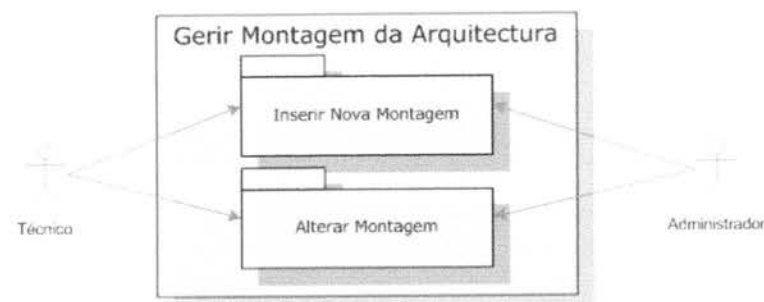


Figura 30: Diagrama de casos de uso do pacote Gerir Montagem da Arquitectura

As interfaces que se seguem mostram os 5 passos necessários para a introdução da estrutura do empreendimento. Basicamente estes passos formam um *wizard* de criação, vulgar nas aplicações Win32.

O primeiro passo consiste na introdução dos dados considerados pertinentes para uma correcta identificação do empreendimento. De notar que em cima existe uma barra que assinala o estado em que se encontra a introdução da nova montagem. O espaço do lado esquerdo foi reservado para a introdução de uma imagem exemplificativa da acção a decorrer.

O segundo passo permite a introdução do número de blocos e na definição do nome pelo qual vão ser identificados.

O terceiro passo consiste na introdução de divisões existentes em cada bloco. O conceito de divisão permite definir, para cada bloco, quantas divisões estruturais independentes existem, por exemplo, um mesmo bloco habitacional pode corresponder a 3 estruturas completamente diferentes que estão anexadas significando isto que temos 1 bloco e 3 divisões. Depois de definir as divisões define-se o número de entradas estabelecendo qual é a principal e definindo o número associado a cada uma.

O quarto passo serve para introduzir o número de pisos elevados, térreos e enterrados associados a cada entrada principal, porque a identificação de uma estrutura habitacional inserida num bloco é sempre identificada pela entrada principal.

O quinto passo permite introduzir o número de fracções existentes em cada um dos pisos previamente definidos. Este passo só acaba quando for efectuado para todos os pisos de todas as entradas de todos os blocos. Cada fracção vai ter uma identificação.

No entanto, no caso de ter apenas 2 divisões é inserida automaticamente a designação de “direito” e “esquerdo”, visto ser esta a designação mais comum utilizada nestas situações.

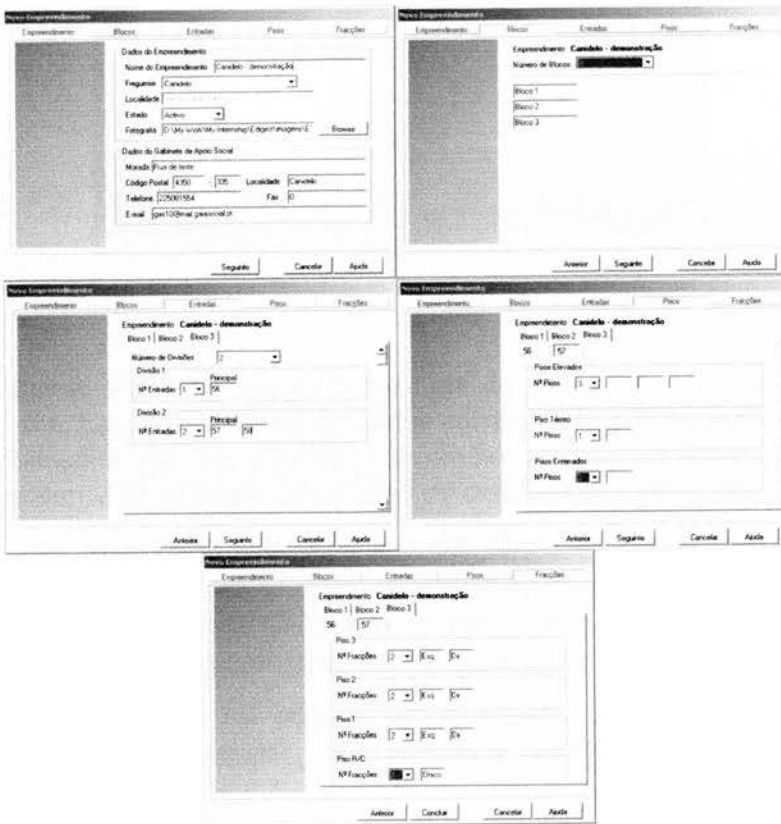


Figura 31: Interfaces do Inserir Nova Montagem do pacote Gerir Montagem da Arquitectura

A interface do Alterar Montagem é diferente do Inserir Nova Montagem por motivos óbvios. Neste caso a interface disponibiliza toda a informação previamente inserida. Para permitir uma visão estruturada, recorreu-se à árvore que aparece do lado esquerdo, sendo o lado direito reservado para os dados do empreendimento ou dados provenientes da selecção de algum item do lado esquerdo.

Esta interface permite inserir, editar ou remover qualquer parte ou até mesmo a totalidade da estrutura inserida.

Tendo em conta a importância dos passos que podem ser efectuados neste menu, todas as acções passam por uma confirmação do utilizador.

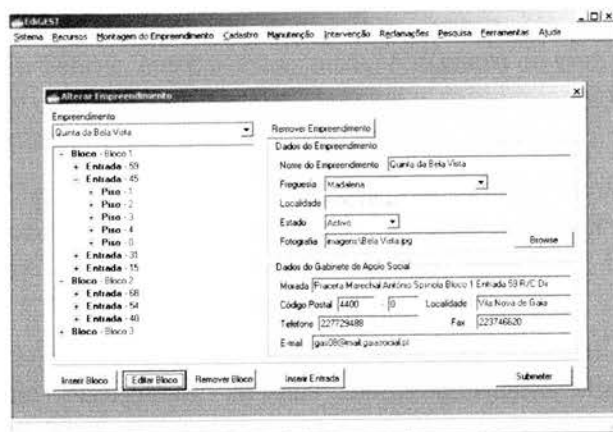


Figura 32: Interface do Alterar Montagem do pacote Gerir Montagem da Arquitectura

O diagrama que se segue representa a estrutura de dados que serve de base para os menus anteriores. Esta área da base de dados corresponde à parte mais importante do SIMEH. Os dados referentes ao empreendimento nos seus vários níveis são usados para ligar a todas as outras áreas consoante o nível de detalhe que é desejado.

Os conceitos que constituem este diagrama são fáceis de prever. Existe uma classe `Empreendimento` que agrega todas as informações pertinentes de identificação deste. Existe uma classe agregada que se designa por `bloco`, com os dados que o definem. De seguida continua o nível de detalhe para `entrada`, `piso` e `fracção`.

Esta estrutura indica o que foi referido anteriormente. Um empreendimento agrega blocos, um bloco agrega entradas, uma entrada é composta por pisos, cada piso é constituído por fracções.

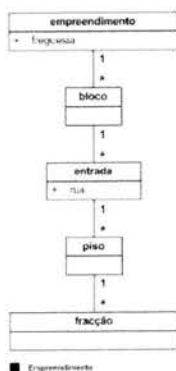


Figura 33: Modelo de classes do pacote Gerir Montagem da Arquitectura

5.9.2 Gerir Bd Tipologias

O pacote `Gerir Bd Tipologias` constitui o repositório de dados de tipologias. Este pacote é constituído por 3 pacotes de casos de uso: `Gerir Tipologias de Habitação`, `Gerir Tipologias de Zonas Comuns` e `Gerir Tipologias de Zonas Exteriores`.

Uma tipologia é a definição estrutural de um dado espaço. No caso de uma habitação, uma tipologia permite definir, entre outras coisas, o número de compartimentos, os tipos a que pertencem e os componentes (portas, janelas e envidraçados) que são partilhados por um ou mais compartimentos.

O pacote *Gerir Tipologias de Habitação* permite construir a totalidade de tipologias que definem uma habitação. Essa definição vai adicionar mais dados à caracterização por compartimentos e medidas de uma habitação. Este pacote pode ser acedido pelo técnico e pelo administrador.

O pacote *Gerir Tipologias de Zonas Comuns* constitui a base de dados das tipologias das zonas partilhadas de um empreendimento. Essas zonas comuns correspondem a áreas tais como escadas entre pisos, *hall* de entrada de um bloco, etc. Este pacote pode ser acedido pelo técnico e pelo administrador. Este módulo ainda não foi implementado tendo sido apenas referido na estrutura dos menus.

O pacote *Gerir Tipologias de Zonas Exteriores* permite definir e editar tipologias referentes às áreas exteriores do empreendimento tais como entradas e jardins. Este pacote pode ser acedido pelo técnico e pelo administrador. Este módulo ainda não foi implementado tendo sido apenas referido na estrutura dos menus.

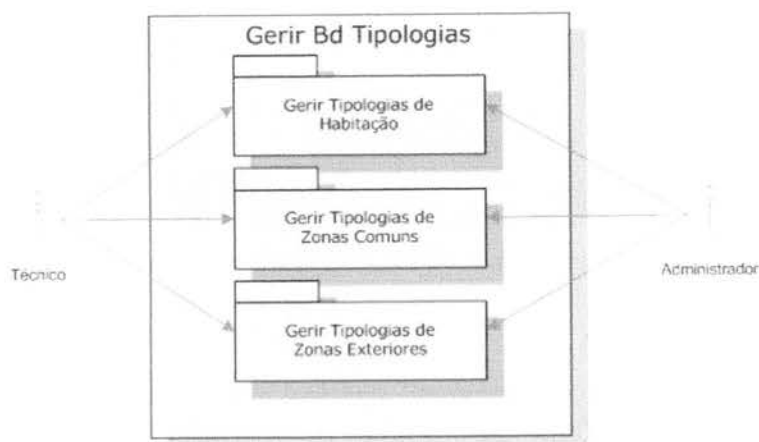


Figura 34: Diagrama de casos de uso do pacote Gerir Bd Tipologias

A interface do *Gerir Tipologias de Habitação* tem como objectivos caracterizar uma tipologia de habitação através de dados que o definam.

Ao inserir uma tipologia é necessário inserir, inicialmente, o número de quartos da tipologia, porque os nomes das tipologias estão normalizados (T+número de quartos variante + número de tipologias existentes +1). De seguida é definida a quantidade de compartimentos existentes para cada tipo e os bairros onde vai ser aplicada esta tipologia.

Para completar a inserção de uma tipologia é necessário inserir, para cada um dos compartimentos definidos, o pé-direito, a área útil, o perímetro, o número de janelas, o número de portas e de envidraçados. Para cada um dos componentes (portas, janelas e envidraçados) é necessário indicar se são partilhados entre vários compartimentos ou

não. Este passo permite uniformizar a designação de cada um dos componentes evitando designações diferentes para um mesmo componente.

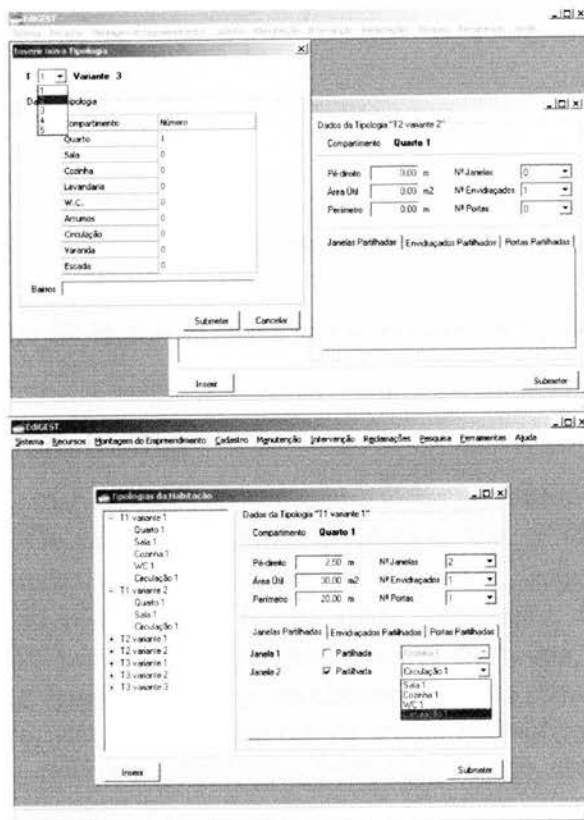


Figura 35: Interfaces do Gerir Tipologias de Habitação do pacote Gerir Montagem do Empreendimento

As classes pertencentes a este módulo caracterizam tudo o que foi referido anteriormente acerca deste módulo. A definição de uma tipologia é constituída essencialmente por dados da tipologia e um conjunto de compartimentos. Cada compartimento apresenta dados próprios assim como uma classificação tipificada (quarto, circulação, wc, sala, etc.). A caracterização de cada compartimento é acompanhada pela definição dos componentes que a compõem. Cada um destes componentes pode pertencer apenas a um ou mais compartimentos, daí a relação entre as classes ser de “muitos para muitos”.



Figura 36: Modelo de classes do pacote Gerir Bd Tipologias

5.9.3 Gerir Bd Construtiva

O pacote Gerir Bd Construtiva armazena todos os dados de materiais utilizados no empreendimento. Os materiais usados dividem-se em 5 famílias: elementos construtivos (paredes, lajes, coberturas, etc.), revestimentos (tintas, azulejos, cortiças, etc.), instalações (gás, electricidade, ventilação, águas residuais, etc.), equipamentos (lava-loiça, esquentadores, tanques de roupa, armários embutidos, etc.) e componentes (portas, janelas e envidraçados). Para cada material, consoante a família a que pertence, são introduzidos outros dados acessórios como, por exemplo, fotografia, designação, referência, descrição, fornecedor, custo, etc.

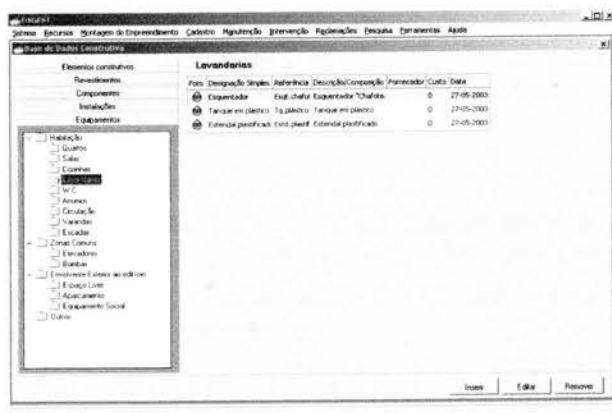


Figura 37: Interface da Base de Dados Construtiva do pacote Montagem do Empreendimento

Através da divisão entre os vários materiais torna-se fácil vislumbrar a estrutura de classes que deverá compor este módulo. Como se verifica no diagrama seguinte cada família de materiais vai constituir uma e uma só classe sem nenhuma ligação entre elas.

É importante salientar que devido à definição de tipo de componente fazer parte da base de dados construtiva existe uma relação obrigatória entre a classe (família) componentes e o conceito de componente da tipologia.



Figura 38: Modelo de classes do pacote Gerir Bd Construtiva

5.9.4 Gerir Montagem Construtiva

O pacote Gerir Montagem Construtiva permite fazer duas das mais importantes associações do sistema. A primeira associa a montagem da arquitectura à base de dados de tipologias. A segunda associa a montagem da arquitectura à base de dados construtiva. Este módulo é constituído por 4 pacotes de casos de uso: Gerir Associação de Tipologias, Gerir Montagem da Habitação, Gerir Montagem de Zonas Comuns e Gerir Montagem de Zonas Exteriores. Todos estes pacotes podem ser acedidos pelo técnico e pelo administrador de sistema.

O pacote Gerir Associação de Tipologias permite fazer uma das duas associações referidas no início desta secção: a ligação entre a montagem da arquitectura e a base de dados de tipologias. Esta associação é feita ao nível da fracção definindo para cada uma a tipologia que se lhe adequa. Isto permite adicionar ao empreendimento todos os dados que faltavam para o conhecimento de toda a estrutura base, ou seja, foi construído um empreendimento com blocos, entradas, pisos, fracções e agora foi introduzido o nível de detalhe seguinte: compartimentos e dados que o definem como o pé-direito ou área útil.

Neste pacote ainda pode ser feita outra associação. No entanto, na base de dados de tipologias ainda não foram especificadas e implementadas as partes de zonas comuns e exteriores. As associações que faltam são: a associação entre o empreendimento e as tipologias das zonas exteriores e comuns.

O pacote de casos de uso Montagem da Habitação permite fazer a outra associação referida anteriormente, ou seja, a associação entre o empreendimento e a base de dados construtiva. Esta associação permite obter mais um nível de detalhe na construção do empreendimento, porque além de ter sido obtida toda a informação estrutural obtém-se a informação construtiva.

O pacote de casos de uso Gerir Montagem de Zonas Comuns permite caracterizar as zonas comuns existentes através de tipologias assim como foi feito para a habitação. No entanto, como este pacote depende directamente de dois pacotes que não estão ainda implementados este também não o está.

O pacote de casos de uso Gerir Montagem de Zonas Exteriores permite associar a cada zona exterior, previamente inserida, uma tipologia que a caracterize. No entanto, assim como aconteceu com as zonas comuns, os módulos nos quais este assenta ainda não estão implementados, o que não permite implementar este também.

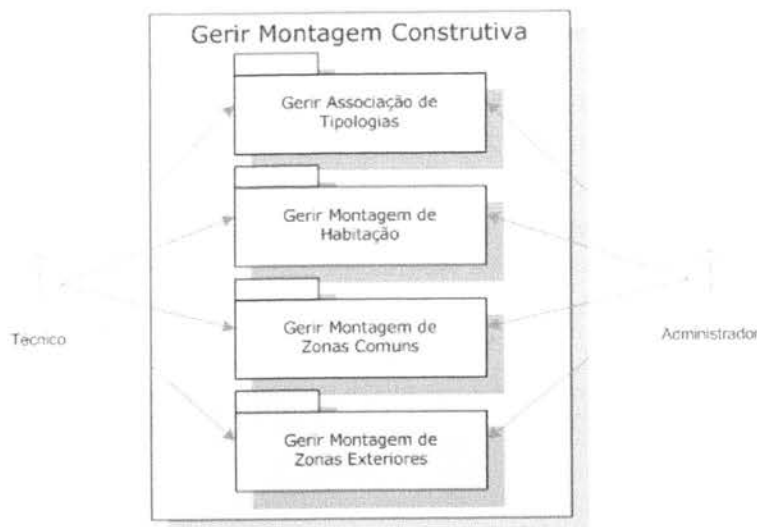


Figura 39: Diagrama de casos de uso do pacote Gerir Montagem Construtiva

A interface do módulo Gerir Associação de Tipologias é composto por um menu superior que se vai tornar comum a muitos módulos para escolher o empreendimento, o bloco, a entrada, o piso e, se necessário, a fracção para depois lhe associar uma tipologia.

O procedimento consiste em escolher o empreendimento com o nível de detalhe desejado (empreendimento, bloco, entrada, piso ou fracção) para listar as opções de associação correspondentes. De seguida, a lista pode ser ordenada por qualquer um dos critérios listados. O penúltimo passo consiste em escolher quais as fracções que se deseja associar. Só então se poderá escolher a tipologia que o identifica.

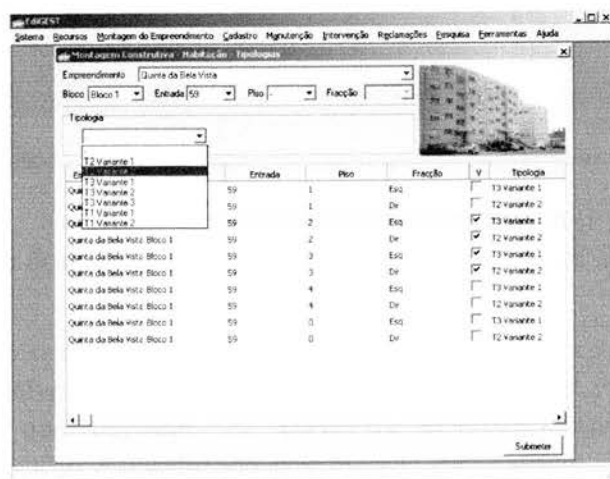


Figura 40: Interface do Gerir Associação de Tipologias do pacote Montagem Construtiva

O módulo Gerir Montagem de Habitação necessita de dois passos. O primeiro passo apresenta um menu muito similar ao da Associação de Tipologias. No entanto, foi acrescentado o campo Compartimento. Ao seleccionar este campo, que

se encontra ao lado da Tipologia, passa-se para o passo seguinte. É importante salientar que estão a ser seleccionadas várias fracções para se fazer a montagem construtiva e que, neste passo, estamos a escolher não um compartimento de uma fracção, mas sim um compartimento que é comum a todas as fracções seleccionadas na listagem que se encontra na parte inferior do ecrã.

O passo seguinte consiste em escolher das várias famílias de materiais os elementos que se encontram no compartimento escolhido. Desta forma, facilita-se a montagem de um empreendimento de grandes dimensões.

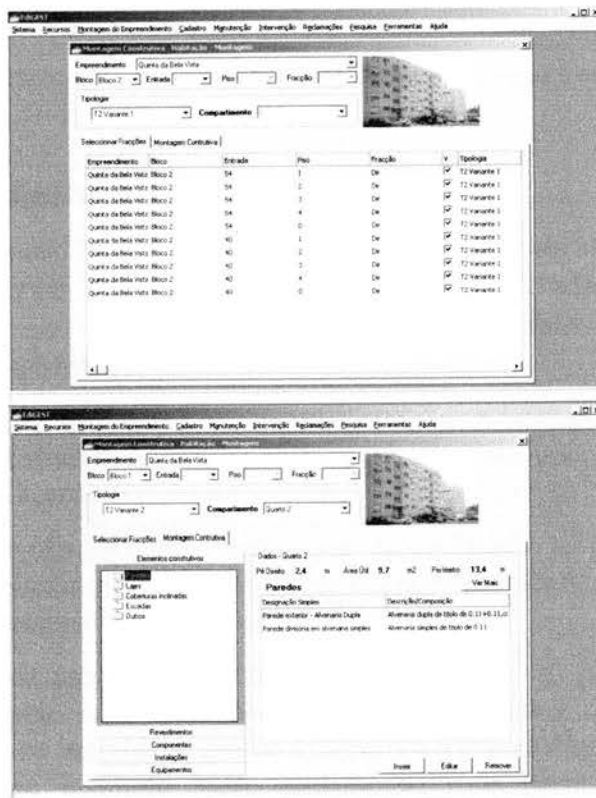


Figura 41: Interface do Gerir Montagem de Habitação do pacote Montagem Construtiva

O modelo de classes deste módulo tão importante agrega classes de 4 áreas da base de dados.

A área que pode ser considerada como base de tudo é a do empreendimento que introduz o primeiro nível de detalhe. O segundo nível de detalhe foi introduzido com o compartimento que também faz parte da área do empreendimento. O repositório de informação é constituído pela Base de Dados Construtiva que agrega a área Tipologias como já foi referido. A ligação entre o empreendimento e a base de dados construtiva é feita através da classe montagem construtiva. A área montagem construtiva é também constituída por uma classe componente que permite definir a relação que existe entre os componentes partilhados, o compartimento e a montagem construtiva.

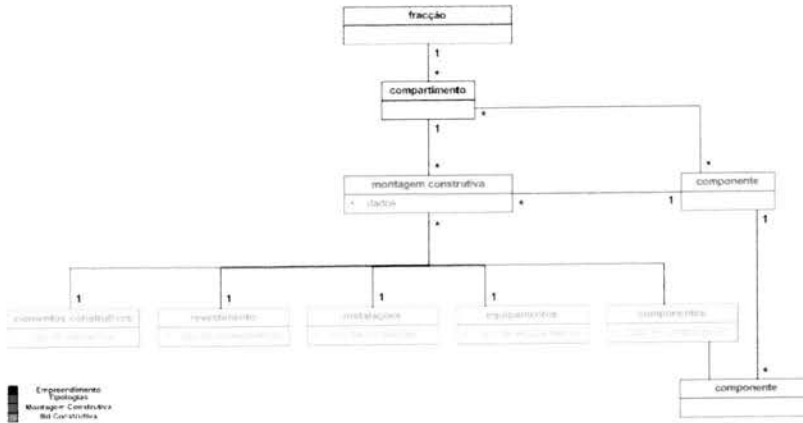


Figura 42: Modelo de classes do pacote Gerir Montagem Construtiva

5.10 Gerir Cadastros

O pacote de casos de uso Gerir Cadastros é um pacote de alto nível que agrega 3 pacotes distintos: Gerir Cadastro Técnico, Gerir Cadastro Económico e Gerir Cadastro Social. Estes 3 pacotes referem-se às áreas técnica, económica e social do SIMEH.

O pacote de casos de uso Gerir Cadastro Técnico pode ser acedido pelo técnico, pelo administrador e pela assistente social. Este pacote vai ser detalhado posteriormente nesta secção.

O pacote de casos de uso Gerir Cadastro Económico só pode ser acedido pelo técnico e pelo administrador devido aos dados económicos confidenciais deste módulo. Este pacote vai ser detalhado posteriormente nesta secção.

O pacote de casos de uso Gerir Cadastro Social só pode ser acedido pela assistente social e pelo administrador de sistema. Os dados contidos neste pacote detalham demasiado os indivíduos que se encontram registados na GaiaSocial, por isso é que o seu acesso é muito mais restringido. Este pacote vai ser detalhado posteriormente nesta secção.

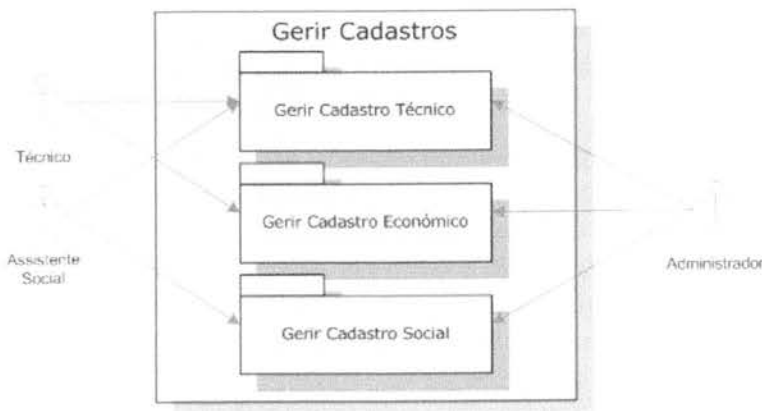


Figura 43: Diagrama de casos de uso do pacote Gerir Cadastros

5.10.1 Gerir Cadastro Técnico

O pacote de casos de uso Gerir Cadastro Técnico permite manipular dois repositórios de dados relacionados com o empreendimento. Este pacote é composto por dois outros: Gerir Dados Cadastrais e Gerir Arquivo de Projecto. Os dois pacotes podem ser acedidos por qualquer um dos utilizadores do sistema.

O pacote de casos de uso Gerir Dados Cadastrais representa uma agregação de pontos de vista sobre o empreendimento. Este módulo é o que apresenta maior expansibilidade apesar de apresentar apenas 3 opções na fase de desenvolvimento em que se encontra: caracterização construtiva, caracterização patológica e caracterização das intervenções. Nenhuma destas opções chegou a ser completamente implementada tendo sido apenas iniciado um esboço da interface. Um dos objectivos deste sistema é acumular, neste menu, a totalidade de visões possíveis a fazer sobre um empreendimento.

O pacote de casos de uso Gerir Arquivo de Projecto representa um repositório de documentação relativa a um dado empreendimento. Este módulo divide-se em dois: projecto e documentos. O módulo de projecto permite uma consulta sobre todos os documentos técnicos referentes ao projecto como, por exemplo, peças desenhadas, plantas, etc. O módulo de documentos agrega todos os documentos genéricos relacionados com um empreendimento tais como: manual do condomínio, manual da habitação, etc.

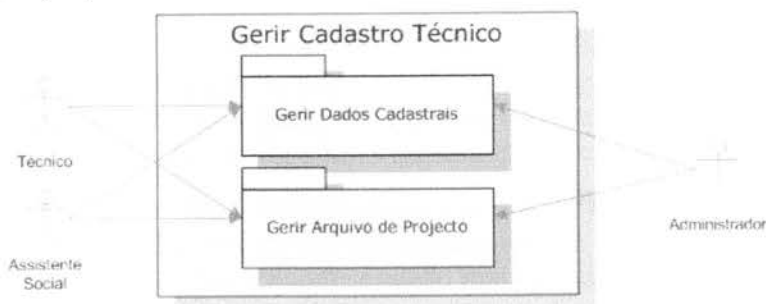


Figura 44: Diagrama de casos de uso do pacote Gerir Cadastro Técnico

Como foi referido anteriormente, a interface do Gerir Dados Cadastrais não se encontra numa fase adiantada de desenvolvimento. Logo, foi apenas implementado um esboço com poucas funcionalidades de módulo.

Esta interface apresenta um menu de selecção comum a várias interfaces do SIMEH no qual se permite uma escolha do empreendimento, bloco, entrada, piso e fracção. O nível de detalhe não se limita à fracção sendo necessário definir o compartimento sobre o qual se debruça uma das 3 visões à escolha. As opções devem permitir uma navegabilidade simples e deve ser deixada em aberto a possibilidade de acrescentar novas opções (visões). As alterações efectuadas vão ser etiquetadas com uma data e um motivo.

Cada uma destas visões vai apresentar um conjunto de dados que corresponde ao cruzamento de dados de várias áreas diferentes da base de dados. Pelo facto de não ter

sendo completamente especificado nenhum dos 3 módulos não se torna possível indicar quais os dados que devem ser visualizados em cada um dos casos.



Figura 45: Interface do Gerir Dados Cadastrais do pacote Gerir Cadastro Técnico

A interface do Gerir Arquivo de Projecto necessita de quatro passos para escolher o documento a ser visualizado. O primeiro passo consiste na escolha entre: projecto ou documentos, visto representaram duas áreas documentais diferentes. O segundo passo incide na escolha do grupo de documentos que se pretende consultar que se encontra estruturado em árvore para permitir uma facilidade de organização e visualização do conjunto. O terceiro passo já é efectuado no menu do lado direito onde se selecciona, na lista apresentada, qual é o documento que se pretende consultar. O quarto passo é apenas o *click* na opção abrir que se encontra à frente da designação do documento. Este *click* executa o programa adequado à abertura do ficheiro escolhido.

Neste menu existem outras opções importantes a referir. Uma destas opções consiste na visualização de mais dados acerca do documento que se encontra seleccionado tais como data, autor, nome, etc. No entanto, a opção mais importante deste menu é a opção de pré-visualização do documento. Como o próprio nome indica, esta opção permite observar, numa janela do lado direito do menu, a capa do documento a ser aberto para acrescentar uma confirmação do documento que foi escolhido para ser aberto.

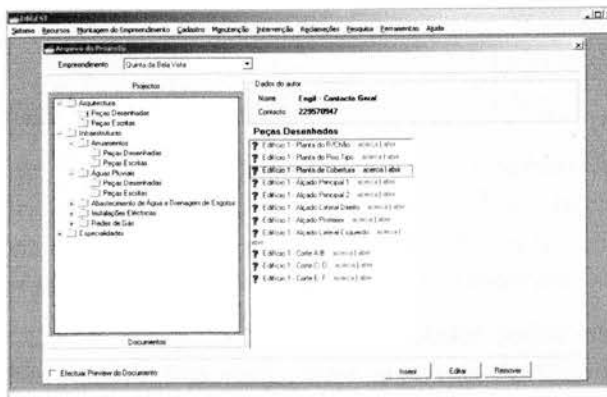


Figura 46: Interface do Gerir Arquivo de Projecto do pacote Gerir Cadastro Técnico

O modelo de classes representativo deste módulo apresenta 7 áreas distintas da base de dados.

A área central deste módulo é a que se refere ao empreendimento, porque, como foi referido anteriormente, todas as visões se debruçam sobre um ou vários pontos do empreendimento. Existe uma associação entre a classe de dados de empreendimento e a parte de arquivo de projecto do qual se destaca o atributo autor.

Devido ao nível de detalhe necessário para cada uma das opções torna-se necessário aceder à montagem construtiva e consequente base de dados construtiva.

A área de patologias é associada ao material escolhido na montagem construtiva. Cada patologia apresenta uma ou mais fichas de intervenção que se lhe associam.

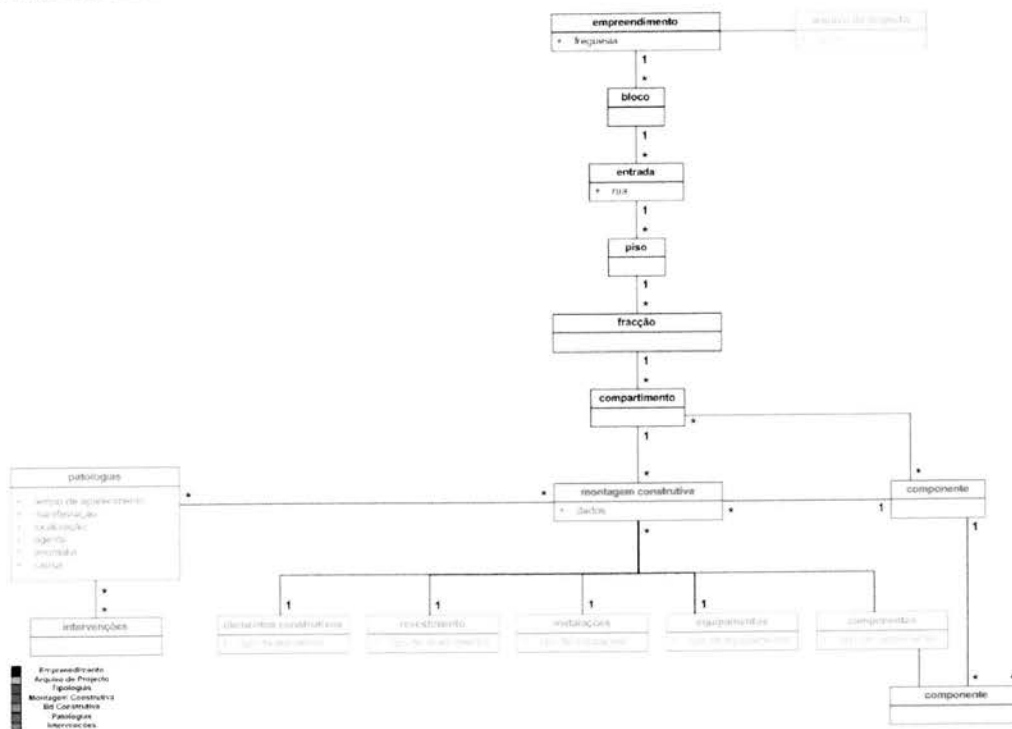


Figura 47: Modelo de classes do pacote Gerir Cadastro Técnico

5.10.2 Gerir Cadastro Económico

O pacote de casos de uso Gerir Cadastro Económico apresenta dois outros: Gerir Custo Inicial e Gerir Custo Ciclo de Vida. Este módulo agrega todas as visões contabilísticas sobre o empreendimento: criação e ciclo de vida. Este módulo pode ser utilizado pelo técnico e pelo administrador.

O pacote Gerir Custo Inicial permite guardar todos os dados associados à criação de um empreendimento nos seus vários níveis de detalhe. Podem ser guardados dados desde custo de materiais até mão-de-obra.

O pacote Gerir Custo Ciclo de Vida permite guardar todos os dados de custos que se seguem à criação do empreendimento, ou seja, manutenções e intervenções cujos conceitos vão ser explicados posteriormente.

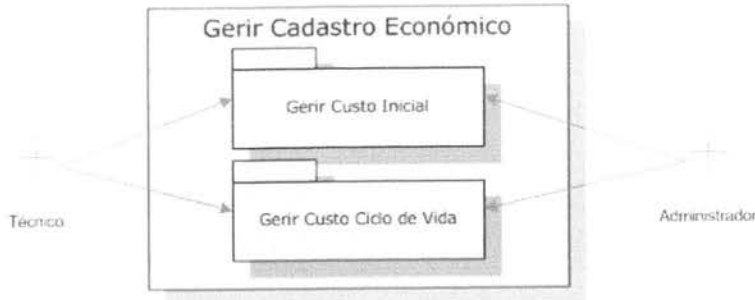


Figura 48: Diagrama de casos de uso do pacote Gerir Cadastro Económico

A interface de cada um destes módulos é semelhante não se encontrando ainda com todas as funcionalidades implementadas neste fase de desenvolvimento.

Depois de escolher o empreendimento em questão temos acesso a uma árvore localizada do lado esquerdo com uma pequena catalogação dos custos que, após seleccionar o item que se deseja, apresenta do lado direito uma tabela com os totais parciais e finais respectivos.

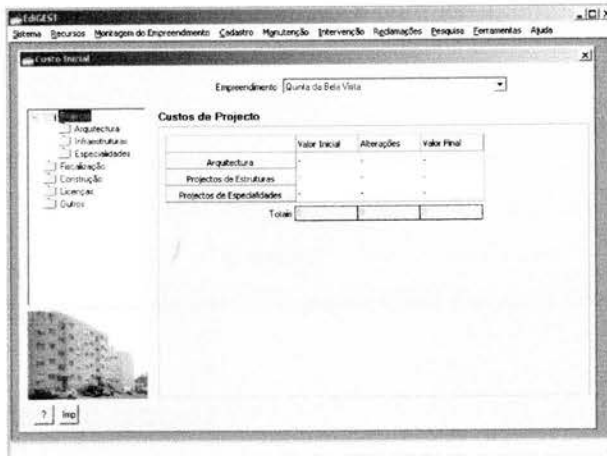


Figura 49: Interface do Gerir Custo Inicial do pacote Gerir Cadastro Económico

A fase de desenvolvimento em que se encontra este menu ainda não permitiu implementar a introdução dos custos tendo sido apenas desenhada a interface, a árvore relativa a cada um dos grupos e a ligação à base de dados para a escolha do empreendimento ou nível de detalhe (bloco, entrada, piso e fracção).

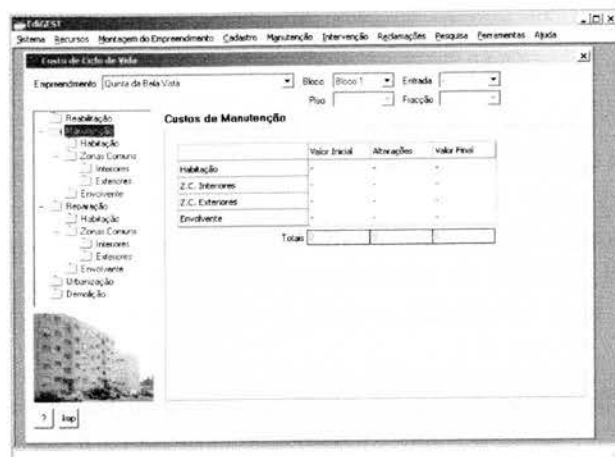


Figura 50: Interface do Gerir Custo de Ciclo de Vida do pacote Gerir Cadastro Económico

O modelo de classes deste pacote encontra-se em desenvolvimento. A classe empreendimento fica agregada à classe do cadastro económico. No entanto, não permite detalhar os custos introduzidos. Esta ligação apenas permite introduzir os custos do módulo Gerir Cadastro Inicial.

Para introduzir as ligações necessárias ao módulo Gerir Custo de Ciclo de Vida é necessário explorar conceitos ainda não completamente especificados como, por exemplo, manutenções, patologias e intervenções. Além disso, como foi referido anteriormente, permite uma anexação dos custos a vários graus de detalhe (empreendimento, bloco, entrada, piso, fracção e compartimento).



Figura 51: Modelo de classes do pacote Gerir Cadastro Económico

5.10.3 Gerir Cadastro Social

O módulo relativo ao Cadastro Social tem como objectivo gerir todos indivíduos que habitam ou que, simplesmente, se encontram registados na GaiaSocial a aguardar habitação.

Para cumprir esse objectivo todos os dados considerados significativos para observar a evolução dos vários agregados na GaiaSocial são armazenados numa das áreas da base de dados. Isto para que a GaiaSocial possa conseguir gerir não só o parque habitacional social de Vila Nova de Gaia, mas conhecer mais dos habitantes que pretende servir.

Devido à confidencialidade dos dados armazenados é obrigatória a restrição de acessos a este módulo.

A interface deste módulo ainda não se encontra implementada.

O diagrama de classes deste módulo encontra-se em desenvolvimento sendo composto actualmente apenas por: uma classe *indivíduo*, uma classe *agregado* e uma classe *fracção*.

A ligação entre *agregado* e *indivíduo* permite definir a família a que cada habitante pertence. A ligação do *agregado* familiar a uma *fracção* tem como objectivo definir a localização do *agregado* em questão. É, no entanto de salientar, que o conceito de família não foi completamente explorado no SIMEH, isto é, um *agregado* é um conjunto de indivíduos que habitam a mesma *fracção* no qual se registam os graus de parentesco.

A classe *indivíduo* guarda uma infinidade de dados acerca do indivíduo desde as habilitações literárias até à situação económica.

A classe *agregado* reúne os dados sócio-económicos dos vários indivíduos permitindo uma visão facilitada da realidade com que cada família convive.



Figura 52: Modelo de classes do pacote Gerir Cadastro Social

5.11 Gerir Manutenções

O pacote de casos de uso *Gerir Manutenções* permite planear e agregar os planos de manutenção existentes em cada empreendimento. Isto permite rentabilizar as equipas de manutenção e reduzir os custos.

Uma manutenção consiste na intervenção periódica sobre um dado material ou zona do empreendimento efectuada por equipa normalmente subcontratada. A vantagem de um bom plano de manutenção consiste no bom funcionamento e consequente durabilidade de cada material.

Torna-se importante referir que uma correcta gestão de manutenções significa o abatimento mais significativo do custo de ciclo de vida de um empreendimento.

Este módulo ainda não se encontra numa fase de desenvolvimento que permita o desenho das bases de dados que servem de repositório da informação necessária. Igualmente não foram obtidos dados suficientes para a correcta especificação de uma interface gráfica.

5.12 Gerir Intervenções

O pacote de casos de uso *Gerir Intervenções* permite guardar as manutenções efectuadas a um dado material que se encontram fora do tempo normal, ou seja, as que

consistem na resolução de problemas que surgiram fora do plano de manutenção habitual do material em questão.

Este pacote é constituído por vários pacotes de casos de uso dos quais apenas o Gerir Bd Patologias é que foi requerido neste fase de desenvolvimento.

O pacote Gerir Bd Patologias consiste, assim como outros pacotes com o nome semelhante, num repositório de dados. Os dados armazenados neste módulo consistem nas fichas de patologias passíveis de serem encontradas nos materiais utilizados num empreendimento. Este módulo pode ser acedido pelo técnico e pelo administrador.

Estas fichas permitem a identificação correcta do problema existente fornecendo dados importantes acerca dos agentes que o produziram, assim como permitem descobrir a intervenção mais adequada a cada situação.

A forma de detecção de patologias pode ocorrer de várias formas das quais se destacam as reclamações. Este meio de reclamação será posteriormente abordado.

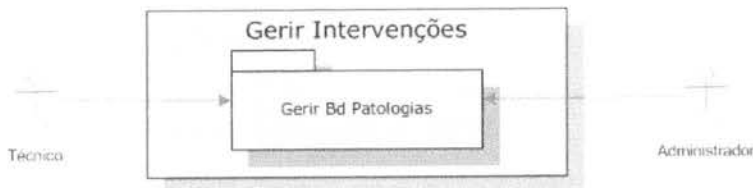


Figura 53: Diagrama de casos de uso do pacote Gerir Intervenções

O modelo de classes deste módulo agrega cinco áreas da base de dados. A área das patologias faz, obrigatoriamente, parte deste modelo no qual se indicam alguns dos atributos mais relevantes desta classe. A cada patologia estão associadas uma ou mais fichas de intervenção. Por isso torna-se imprescindível a ligação entre as duas classes. Cada patologia está associada a um elemento presente na montagem construtiva de um empreendimento, o que justifica a presença das classes respectivas. O meio por excelência de detecção de problemas (patologias) é efectuado através de reclamações que apresentam um estado definido e que podem ser submetidos por vários meios.



Figura 54: Modelo de classes do pacote Gerir Intervencões

5.13 Gerir Reclamações

O pacote de casos de uso Gerir Reclamações tem como objectivo fazer a ponte entre a interface Web e a interface aplicação do SIMEH.

As reclamações são submetidas através da interface Web. Neste módulo alguns dos dados recolhidos vão ser sujeitos a tratamento. A reclamação não identifica, na totalidade, o material ou zona a que se refere. Essa associação é efectuada neste módulo.

O processo de uma reclamação passa por vários estados ao longo da sua resolução. No entanto, na fase de desenvolvimento em que este módulo se encontra, apenas algumas foram mencionadas. O primeiro estado é o de Submetida, por se seguir à submissão por parte do morador ou da assistente social. A assistente social vai deferir ou indeferir a reclamação efectuando desta forma a passagem ao estado Deferido ou Indeferido, respectivamente. A fase que este módulo efectua é a fase Em Análise. As seguintes, como foi referido anteriormente, ainda não foram especificadas.

Não foi ainda elaborado nenhuma interface para este módulo.

O diagrama de dados deste pacote apresenta a relação que existe entre 3 classes de 3 áreas distintas. A primeira área é a das reclamações onde se salienta os atributos que foram referidos anteriormente: a cada reclamação está associado um indivíduo (não obrigatoriamente) e uma assistente social que se vai ocupar dessa reclamação e algumas das fases que se seguem à submissão.

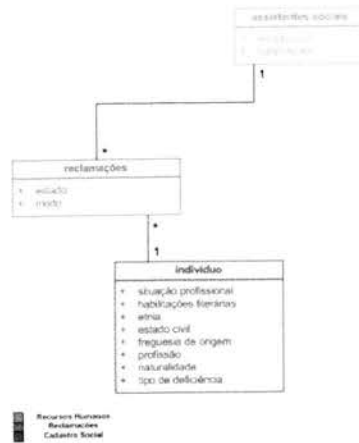


Figura 55: Modelo de classes do pacote Gerir Reclamações

5.14 Gerir Pesquisa

O pacote de casos de uso Gerir Pesquisa pretende agregar pesquisas personalizáveis. Estas têm em vista a obtenção de dados específicos considerados pertinentes para um estudo e acompanhamento de cada um dos empreendimentos registados na base de dados do SIMEH.

Este módulo não foi especificado daí não se encontrar ainda implementado, nem apresentar nenhum modelo de dados de suporte.

5.15 Gerir Ferramentas

O pacote de casos de uso Gerir Ferramentas tem como objectivo permitir o acesso a ferramentas que podem ser consideradas importantes para o sistema.

Este género de menus torna-se vulgar neste tipo de soluções informáticas. No entanto, não foi discutido o conteúdo deste pacote tendo sido apenas mencionada a sua existência.

5.16 Gerir Ajuda

O pacote de casos de uso Gerir Ajuda pretende, no futuro, agregar um conjunto de manuais que permitam a utilização correcta e intensiva deste sistema.

Devido à intensidade com que o sistema foi especificado e implementado a geração de material de apoio ao software desenvolvido foi relegada para segundo plano.

6 Desenvolvimento SIMEH – WEB

A fase Web foi a última fase do estágio e coincidiu com a separação da equipa de estágio no desenvolvimento dos vários módulos. Tendo sido praticamente terminada a especificação do SIMEH – APLICAÇÃO, uma das equipas permaneceu na implementação e acréscimo de requisitos enquanto que a outra equipa se encarregava do SIMEH – WEB.

A fase SIMEH – WEB partiu de uma especificação genérica relativamente curta retirada de várias reuniões, tendo a fase de implementação sido completamente terminada em tempo útil. A fase seguinte consistiu na validação e refinamento do trabalho desenvolvido, sendo esperado ainda o *feedback* de utilizadores reais.

Nesta fase de estágio, além das reuniões com a equipa de desenvolvimento, foram efectuadas algumas apresentações para apresentar o trabalho e receber sugestões da instituição cliente.

Em oposição ao que aconteceu com as outras fases, a dimensão dos requisitos a implementar permitiu a definição correcta dos perfis de utilização do SIMEH – WEB. Estes perfis de utilização vão definir os utilizadores que vão ter acesso a cada uma das áreas.

Existem apenas 2 tipos de utilizadores: assistente social e morador. A assistente social é a utilizadora que recorre a esta interface para a auxiliar nas fases iniciais de uma reclamação. O morador tem nesta interface a oportunidade de preencher reclamações e consultar o estado em que as mesmas se encontram.

O objectivo de cada um destes utilizadores é comum: auxiliar a preencher a base de dados com os dados necessários a uma boa gestão, sendo de salientar que nesta interface Web apenas se aplica ao módulo de Reclamações.

Dada a correcta definição de requisitos foram tidos em conta alguns critérios de usabilidade (Anexo B) para auxiliar os utilizadores, em especial o morador, a conseguir tirar partido deste módulo.

6.1 Introdução

A interface a ser desenvolvida permite utilizar alguns dos dados introduzidos no SIMEH – APLICAÇÃO e gerir as fases iniciais do módulo Reclamações.

O sistema pode ser dividido de várias formas, quer pelos seus dois utilizadores ou pelos módulos que o compõem.

O sistema é composto por 2 áreas fundamentais: morador e assistente social. Como o próprio nome indica cada uma destas é relativa aos dois tipos de utilizadores da interface.

A área da assistente social é apenas composta pelo pacote de casos de uso Gerir Reclamações do Empreendimento. O acesso a este pacote necessita de autenticação para ser utilizado.

A área do morador é composta por 3 pacotes de casos de uso: Gerir Reclamações do Morador, Registrar Novo Utilizador e Pedir

Reenvio de Palavra-chave. Destes pacotes apenas o Gerir Reclamações do Morador necessita de autenticação para a sua utilização.

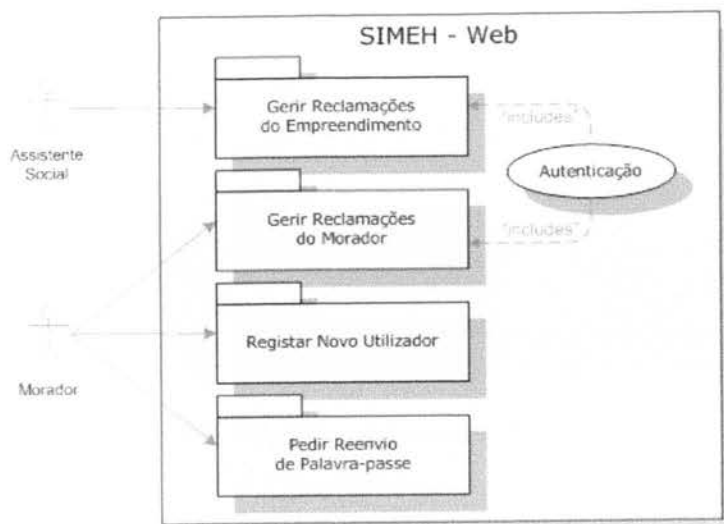


Figura 56: Diagrama de casos de uso de alto nível do SIMEH – WEB

Uma das principais características desta fase, que permite uma demarcação das anteriores, é o facto do seu desenvolvimento ter permitido um refinamento do trabalho desenvolvido. Esse refinamento torna-se imperativo para melhorar a interface de utilização do morador, porque este tipo de utilizador representa um conjunto de pessoas exteriores aos quadros da GaiaSocial ao qual não se torna possível dar formação para uma utilização intensiva. Daí a utilização de critérios de usabilidade para oferecer uma interface apelativa, escalável, simples e intuitiva.

A interface principal do SIMEH apresenta as funcionalidades que seriam de esperar a partir dos requisitos fornecidos: uma autenticação para o acesso às áreas com informações sensíveis e uma opção de registo e outra de pedido de reenvio de palavra-chave.

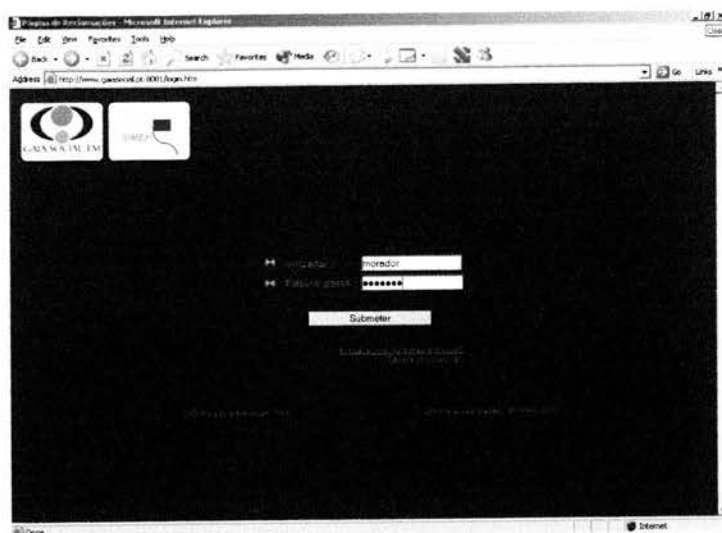


Figura 57: Interface principal do SIMEH – WEB

O modelo de classes do SIMEH – WEB é composto apenas por uma pequena parte da base de dados do sistema. É composto por 5 áreas: cadastro social, recursos humanos, reclamações, permissões página e empreendimento. Todas as ligações entre estas classes foram explicadas anteriormente, sendo novamente desenvolvidas na explicação dos módulos que se seguem.

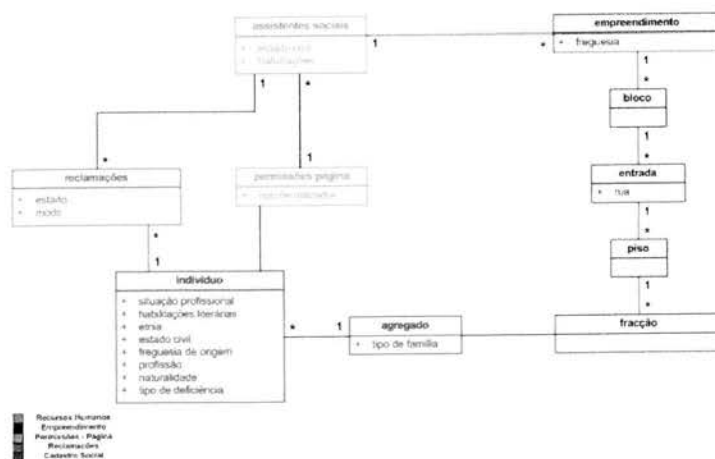


Figura 58: Modelo de classes de alto nível do SIMEH – WEB

6.2 Registrar Novo Utilizador

O módulo Registrar Novo Utilizador permite aos moradores procederem ao registo para a utilização da interface Web do SIMEH. Como seria de esperar, o registo efectuado através deste módulo ainda passa pela assistente social para ser autorizado. O módulo correspondente para a aceitação do pedido referido é feito na aplicação.

A interface deste pacote é constituída por campos de dados que permitem confirmar a identificação do utilizador que se pretende registar.

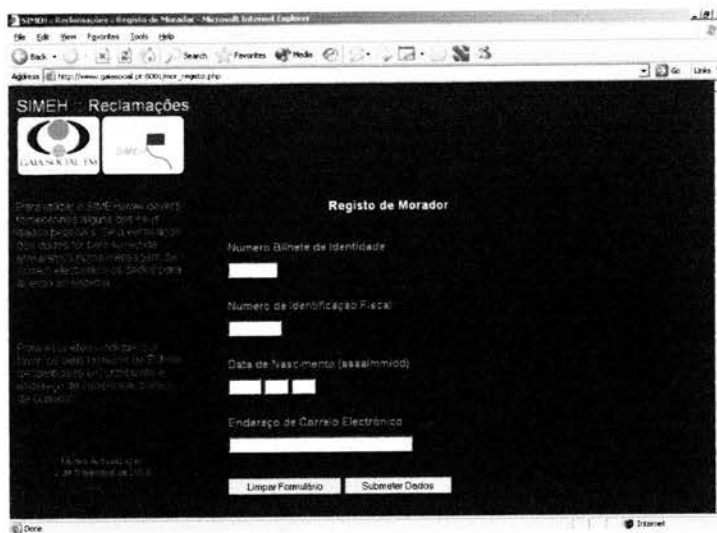


Figura 59: Interface do pacote Registrar Novo Utilizador

O modelo de classes deste módulo apenas acede às áreas: cadastro social e pagina permissões. A relação entre estas classes é óbvia: após a correcta identificação do indivíduo a classe página permissões permite uma nova entrada, num estado ainda não autorizado.

Posteriormente o estado do registo é alterado para permitir o acesso ao SIMEH – WEB.



Figura 60: Modelo de classes do módulo Registrar Novo Utilizador

6.3 Pedir Reenvio da Palavra-chave

O pacote Pedir Reenvio da Palavra-chave consiste numa funcionalidade que permite retirar trabalho dos gestores da interface Web do SIMEH. A situação de perda ou esquecimento da palavra-chave previamente atribuída pode ser resolvida mediante o pedido de reenvio da mesma para o indivíduo que se registou. Este módulo permite automatizar essa tarefa.

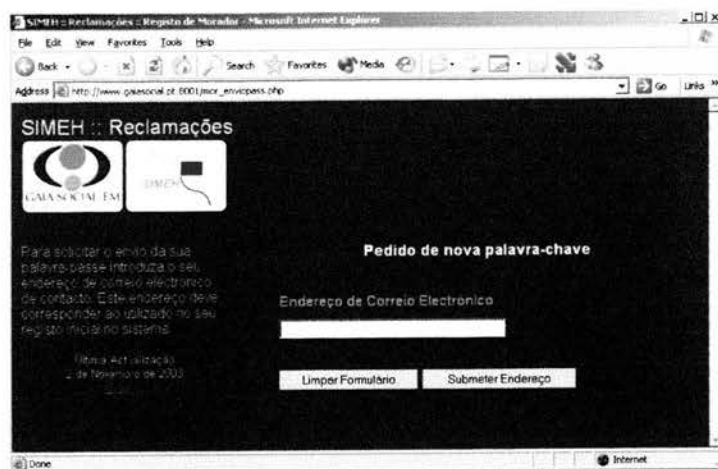


Figura 61: Interface do pacote Pedir Reenvio da Palavra-chave

O modelo de classes deste módulo acede apenas a uma área constituída por apenas uma classe: `permissões página`, porque os dados de contacto necessários para esta funcionalidade encontram-se todos nesta classe.



Figura 62: Modelo de classes do módulo Pedir Reenvio da Palavra-chave

6.4 Gerir Reclamações de Morador

O pacote `Gerir Reclamações de Morador` é o mais importante do SIMEH – WEB. Este pacote interage com os indivíduos registados na GaiaSocial, permitindo-lhes registar as reclamações e consultar o estado em que as mesmas se encontram.

Este pacote é composto por 2 pacotes e 3 casos de uso interdependentes. Os 2 pacotes são: `Submeter Reclamação` e `Consultar Dados de Reclamações`. Os casos de uso são: `Visualizar Reclamações`, que permite, `Editar Reclamação` e `Eliminar Reclamação`.

As opções deste módulo só podem ser acedidas pelo morador.

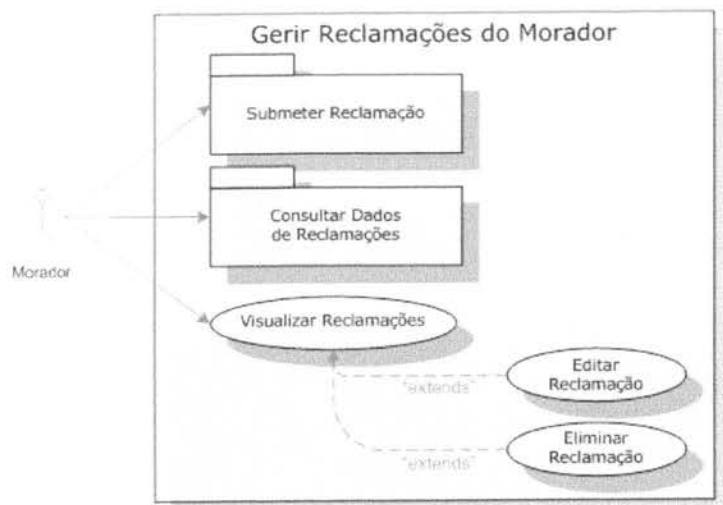


Figura 63: Diagrama de casos de uso do pacote Gerir Reclamações de Morador

A interface deste pacote pode ser dividida em várias áreas.

A área que se destaca das demais é composta pela lista de reclamações que se encontra no centro da interface. A lista apenas exibe um determinado número de reclamações de cada vez, sendo a navegação efectuada na área de menu que se encontra do lado esquerdo. Para cada reclamação é apresentado o código atribuído, a data de submissão e o estado em que se encontra. Esta área refere-se ao caso de uso *Visualizar Reclamações*, que permite também que caso a reclamação se encontre no estado “Submetida” seja dada a possibilidade ao morador de *Editar Reclamação* e *Eliminar Reclamação*.

A área do lado esquerdo pode ser definida como a área de menu, visto ser aqui que se disponibilizam algumas das opções desta interface. Após ser feita a autenticação a opção que se encontra escolhida como padrão é a consulta (*Visualizar Reclamações*). No entanto, é também neste menu que se encontra a possibilidade de *Submeter Reclamações*. A opção seguinte permite navegar nas reclamações apresentadas na lista central. Também faz parte desta área um pequeno gráfico que permite obter uma visão geral dos estados das várias reclamações submetidas pelo morador. Destacada das demais opções, também neste menu, encontra-se a opção de *Sair*.

A área superior desta interface apresenta, não só os logótipos do sistema e da entidade, como apresenta, também, dados acerca do morador tais como: nome, código interno, empreendimento a que pertence e a fotografia.

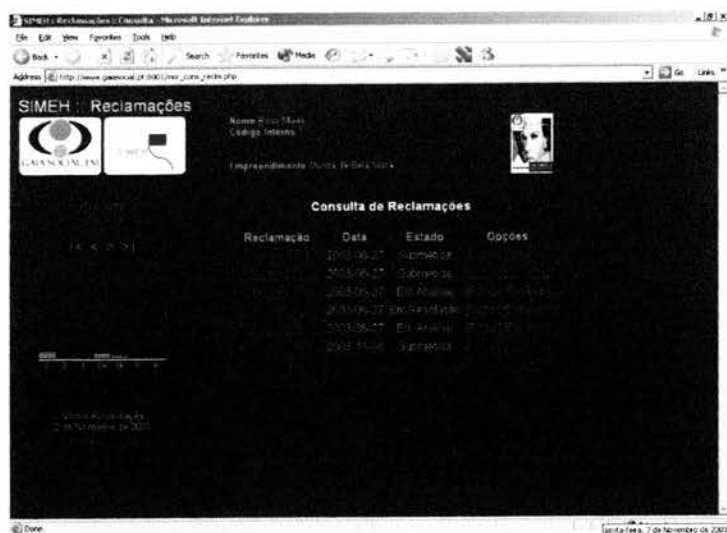


Figura 64: Interface do pacote Gerir Reclamações de Morador

O modelo de classes deste módulo apresenta praticamente todas as áreas do SIMEH – WEB: reclamações, permissões página, cadastro social e empreendimento.

A área permissões página é utilizada para efectuar a correcta autenticação, identificando o indivíduo registado no cadastro social. Para descobrir os dados de localização é necessário utilizar a ligação entre o agregado familiar e a fracção. Através do resto das classes do empreendimento descobrir a localização completa do indivíduo. Esta informação torna-se imperativa para efectuar grande parte da submissão de reclamações.

A utilização da área reclamações permite a consulta, submissão e consequente gestão das reclamações do morador.

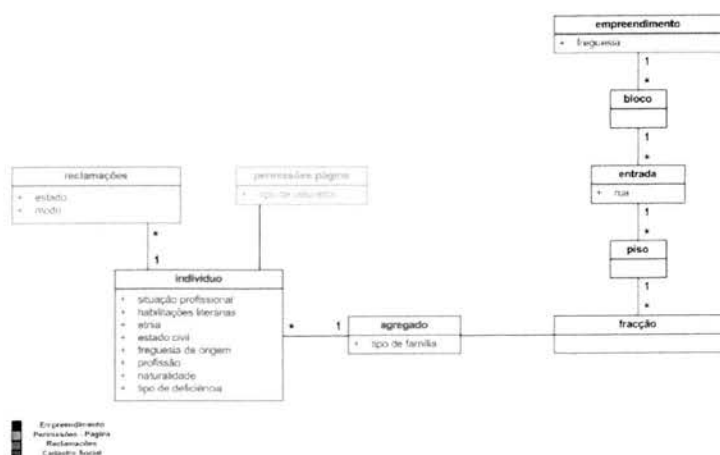


Figura 65: Modelo de classes do módulo Gerir Reclamações de Morador

6.4.1 Visualizar Reclamações

O pacote *Visualizar Reclamações* encontra-se seleccionado quando o morador tem acesso a esta interface. Como foi referido no item anterior é composto por 3 áreas: superior, menu e listagem. Os constituintes das áreas já foram apresentados anteriormente, porque, como foi referido, é este o pacote de casos de uso que se encontra “aberto” aquando de uma autenticação válida no sistema.

6.4.2 Submeter Reclamação

O pacote *Submeter Reclamação* permite ao morador introduzir os dados necessários à geração de uma ficha de reclamação.

Os dados necessários dividem-se em 3 categorias: descrição, localização e observações. Existem mais dados que se tornam necessários mas estes já se encontram determinados aquando da primeira autenticação. Esses dados são: data, identificação do morador, identificação do agregado e identificação da fracção, piso, entrada e bloco a que pertence. A apresentação das categorias dos dados é referida na explicação da interface.

A interface deste pacote é composta pelas mesmas áreas que a anterior para manter o máximo de informação necessária e constante ao utilizador do sistema. Isto significa que todos os constituintes da área superior e do menu se encontram da mesma forma. Na área, anteriormente, de listagem encontram-se agora um novo módulo de inserção de dados.

O módulo de inserção de dados, como já foi referido, é constituído por: descrição, localização e observações.

A categoria descrição representa apenas a apresentação informal de descrever a reclamação, usando para isso duas áreas: texto e fotografia. A área de texto permite introduzir um texto até 500 caracteres para auxiliar os serviços da GaiaSocial a compreender a dimensão do problema. A área da fotografia permite aos serviços técnicos da GaiaSocial evitar uma viagem de um técnico especializado ao local para obter a informação da patologia presente na localização abaixo descrita.

A categoria localização permite localizar de uma forma normalizada o local de manifestação do problema. Para isso são apresentadas 4 opções: apartamento, zonas comuns internas, zonas comuns externas e outras.

A categoria observações permite a introdução de dados, numa área de texto, que possam não ter sido apresentados em categorias anteriores como, por exemplo: a localização do problema ou até mesmo a periodicidade com que o problema surge.

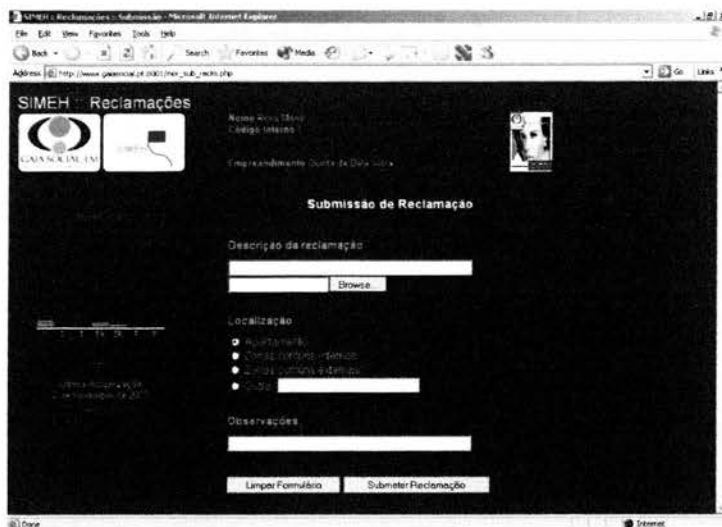


Figura 66: Interface do pacote Submeter Reclamação

O modelo de classes deste módulo compreende todas as classes de introdução de reclamações: todas as que permitem identificar convenientemente o morador, assim como a assistente social que terá que receber o pedido efectuado.

Para efectuar a submissão de reclamações são necessárias as áreas: recursos humanos, empreendimento, reclamações e cadastro social. A área de reclamações é a que vai agregar a identificação dos intervenientes na reclamação. Para isso é necessária a classe indivíduo do cadastro social e a classe assistentes sociais dos recursos humanos. A partir da classe agregado e classes do empreendimento é possível identificar, na totalidade, a localização do problema, sendo de seguida a classe empreendimento da área empreendimento que permite saber qual a assistente social que vai receber o pedido de reclamação.

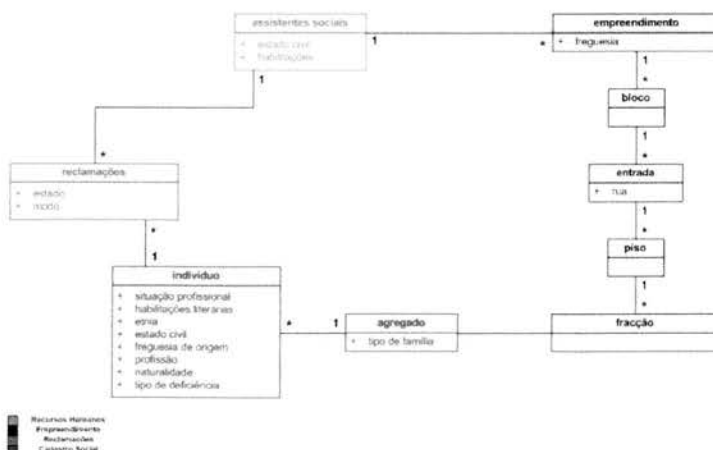


Figura 67: Modelo de classes do módulo Submeter Reclamação

6.4.3 Consultar dados das Reclamações

O pacote de casos de uso Consultar dados das Reclamações é aquele permite conhecer todos os dados acerca de uma reclamação que não se encontram expressos na listagem do pacote Visualizar Reclamações.

Os dados apresentados encontram-se divididos nas mesmas categorias que se encontram na submissão: descrição, localização e observações. Foi adicionado o campo reclamação.

A área reclamação apresenta 4 dados: código interno da reclamação, estado em que se encontra, data e o modo como foi submetida.

A área descrição apresenta o texto introduzido anteriormente na submissão e a fotografia do problema.

A área localização indica os dados normalizados introduzidos na submissão.

A área observações apresenta o texto opcional introduzido anteriormente.

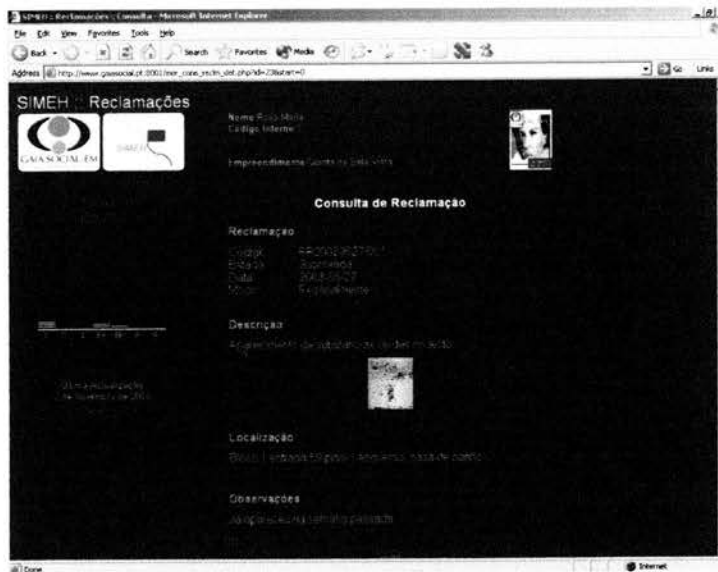


Figura 68: Interface do pacote Consultar dados das Reclamações

O modelo de classes deste módulo corresponde apenas a 2 áreas: cadastro social e reclamações. São apenas estas áreas por dois motivos. Primeiro, porque o resto das áreas, apesar de utilizadas, servem apenas para as partes da interface que se mantêm constantes não sendo necessário referi-las neste diagrama. Segundo, porque na classe reclamações foram introduzidos todos os dados necessários a esta listagem, inclusive a localização.

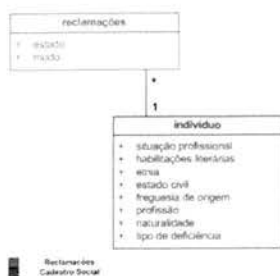


Figura 69: Modelo de classes do módulo Consultar dados das Reclamações

6.5 Gerir Reclamações do Empreendimento

O pacote Gerir Reclamações do Empreendimento permite à assistente social participar no processo de reclamações do sistema global SIMEH.

Este pacote é composto por 3 pacotes e 1 caso de uso. Os 3 pacotes são: Submeter Reclamação, Triar Reclamações e Consultar Reclamações. O caso de uso é Consultar dados de Reclamação que funciona como uma extensão do pacote Consultar Reclamações.

As opções deste módulo só podem ser acedidas pela assistente social.

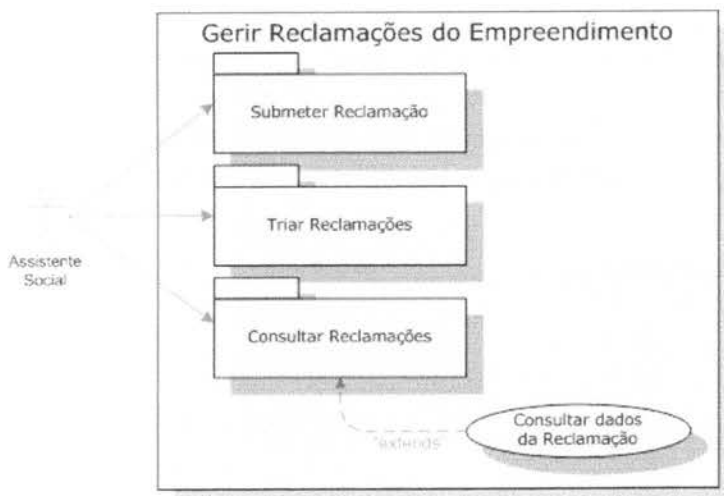


Figura 70: Diagrama de casos de uso do pacote Gerir Reclamações do Empreendimento

A interface deste pacote é simples para permitir a agregação futura de opções.

Esta interface é constituída apenas por um menu que apresenta as opções. As opções são: Submissão de nova Reclamação, Consulta de Reclamações, Triagem de Reclamações e Sair.

A estrutura da interface não é constante: o menu que se encontra no centro é posteriormente desviado para o canto superior direito para permitir que a área central apresente o item escolhido.

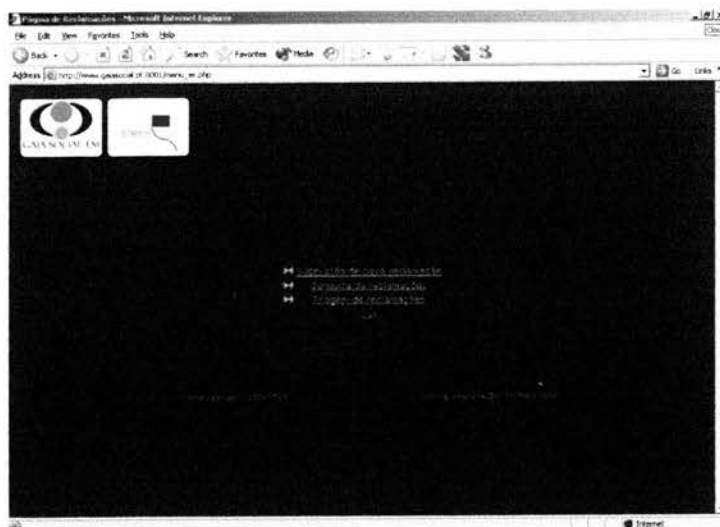


Figura 71: Interface do pacote Gerir Reclamações do Empreendimento

O modelo de classes deste módulo apresenta todas as áreas da base de dados do SIMEH – WEB: cadastro social, recursos humanos, reclamações, empreendimento e permissões página.

Assim como aconteceu no pacote Gerir Reclamações do Morador a área permissões página permite a correcta autenticação e validação dos utilizadores desta interface. A diferença reside apenas no facto de que a ligação mais importante é estabelecida entre a assistente social e o/os empreendimento(s) que lhe estão afectos. Cada reclamação submetida por um morador (cadastro social) é recebida pela assistente social do empreendimento a que pertence.

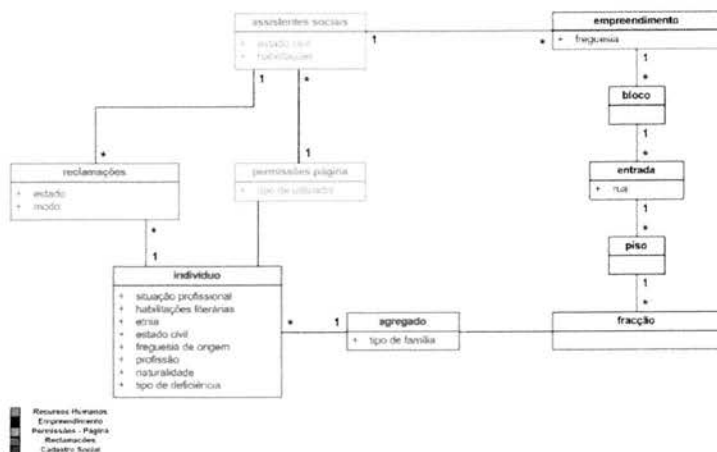


Figura 72: Modelo de classes do módulo Gerir Reclamações do Empreendimento

6.5.1 Submeter nova Reclamação

O pacote Submeter nova Reclamação permite introduzir os dados de uma reclamação. O objectivo deste pacote é submeter as reclamações que são recebidas em

cada um dos gabinetes de apoio social pelos seguintes modos: pessoalmente, carta, Internet e telefone.

Ao contrário do que acontece na área do morador o reclamante não está identificado. Assim, o SIMEH – APLICAÇÃO, que já se encontra dotado dos mecanismos de pesquisa de indivíduos, fornece o código interno do morador em questão.

A interface deste pacote apresenta 2 áreas: menu e exibição. Na área de menu, situada na parte superior direita, encontram-se as várias opções que foram apresentadas ao entrar no pacote Gerir Reclamações de Empreendimento. Na área de exibição é apresentada um formulário de introdução de dados, visto ter sido este o item escolhido.

A introdução de dados divide-se em 5 categorias: reclamante, modo, descrição, localização e observações.

A categoria reclamante permite identificar univocamente o morador que submeteu uma dada reclamação. Para este efeito é introduzido o código interno do indivíduo.

A categoria modo tem como objectivo indicar qual o modo de submissão que foi escolhido pelo morador para entrar em contacto com o gabinete de apoio social do empreendimento.

A categoria descrição representa, assim como na submissão do morador, uma apresentação informal da reclamação. É constituída por uma área de texto e uma área de submissão de um ficheiro de imagem.

A categoria localização permite fornecer os dados necessários na indicação do local do problema que é relatado.

A categoria observações fornece uma área de texto para indicar possíveis dados que possam ser considerados pertinentes na submissão da reclamação.

The screenshot displays a web browser window with the address <http://www.gabinete.de.riiit.technico.pt/riiit.php>. The page features a dark background with white text and form elements. At the top left, there are logos for 'LUMANKINE, I.M.' and 'SIMEH'. The main content area is titled 'Informação Geral' and includes a 'Número' field with a text input and a 'Designação do Reclamante' field. Below this, there is a 'Modo' section with radio buttons for 'pessoalmente', 'carta', 'Internet', and 'telefone'. The 'Descrição' section contains a large text area and a 'Browse' button for image upload. The 'Localização' section has radio buttons for 'apartamento', 'Trabalho (comércio/indústria)', 'Zona de habitação informal', and 'outro', followed by a text input field. The 'Observações' section has a large text area. At the bottom, there are two buttons: 'Limpar Formulário' and 'Submeter Reclamação'.

Figura 73: Interface do pacote Submeter nova Reclamação

O modelo de classes deste módulo é em tudo semelhante ao da submissão de reclamações da área do morador apesar da utilização das classes ser radicalmente diferente.

As áreas que compõem este módulo são 4: recursos humanos, reclamações, cadastro social e empreendimento.

A área de recursos humanos faz parte deste módulo porque a submissão é efectuada pela própria assistente social que, por sua vez, também se encarrega do passo que se segue.

A área de reclamações era obrigatória neste módulo, já que é na classe que a compõe que se armazenam a maioria dos dados inseridos nos formulários.

A área de cadastro social e empreendimento, apenas para confirmação dos dados inseridos.

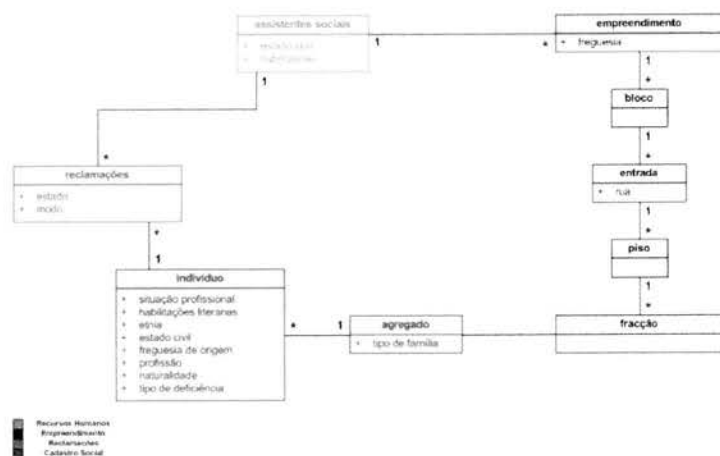


Figura 74: Modelo de classes do módulo Submeter nova Reclamação

6.5.2 Consultar Reclamações

O pacote de casos de uso Consultar Reclamações permite à assistente social visualizar a totalidade das reclamações dos empreendimentos que gere.

Os dados são apresentados numa tabela simples com 3 campos: código da reclamação, data e estado. O código da reclamação é uma ligação para uma página gerada dinamicamente que mostra, com mais pormenor, todos os dados da reclamação. Esta interface é semelhante à que é apresentada na área do morador.

A interface deste módulo apresenta as mesmas áreas que foram apresentadas para o módulo anterior, sendo a área de exibição substituída pela tabela anteriormente referida. Foi tido em conta que existem várias funcionalidades que vão ser implementadas futuramente daí a simplicidade na apresentação e interacção.

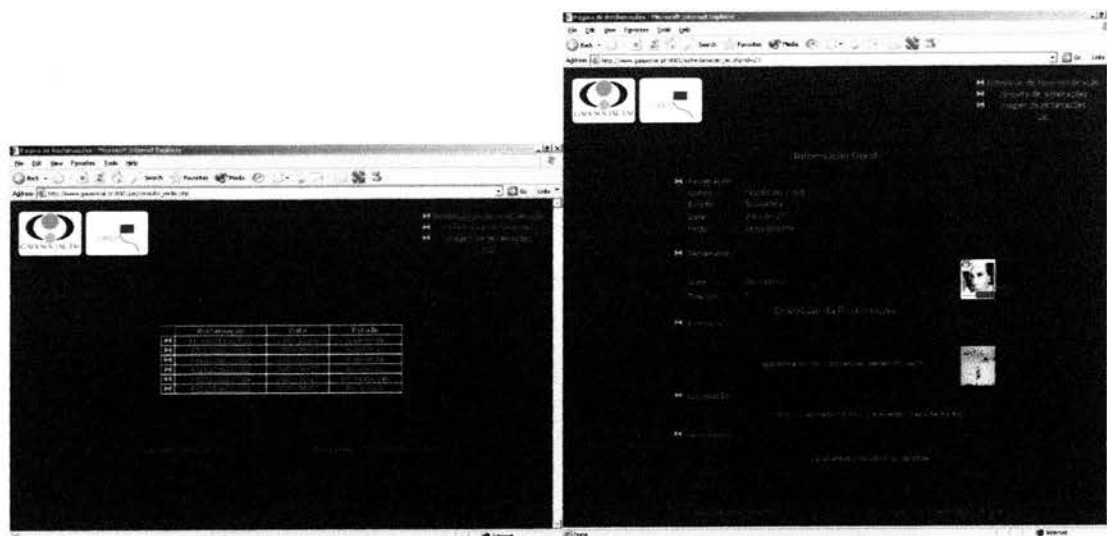


Figura 75: Interfaces do pacote Consultar Reclamações e do Consultar dados das Reclamações

O modelo de dados referente ao módulo Consultar Reclamações é semelhante ao da área do morador. Apresenta apenas 3 áreas: recursos humanos, reclamações e cadastro social.

A área de recursos humanos, representada pela classe assistente social, permite a identificação do utilizador e conseqüente filtragem do total de dados que se encontram na classe reclamações.

A área de cadastro social apenas é utilizada no módulo Consultar dados das Reclamações para indicar os dados do reclamante.

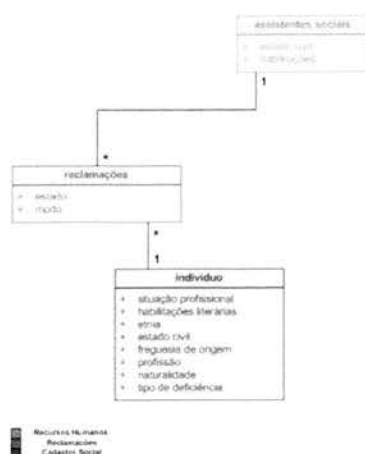


Figura 76: Modelo de classes do módulo Consultar Reclamações

6.5.3 Triar Reclamações

O pacote de casos de uso Triar Reclamações permite filtrar as reclamações que se encontram submetidas. Esta triagem, denominada 1ª análise, é efectuada pela assistente

social, para poupar trabalho ao técnico que vai analisar as reclamações, através da escolha das reclamações que são, de facto, válidas.

A interface é bastante simples e intuitiva exibindo uma lista de reclamações que se encontram no estado “submetida”. Apresenta, para cada uma delas, a opção de passagem ou não à fase seguinte.

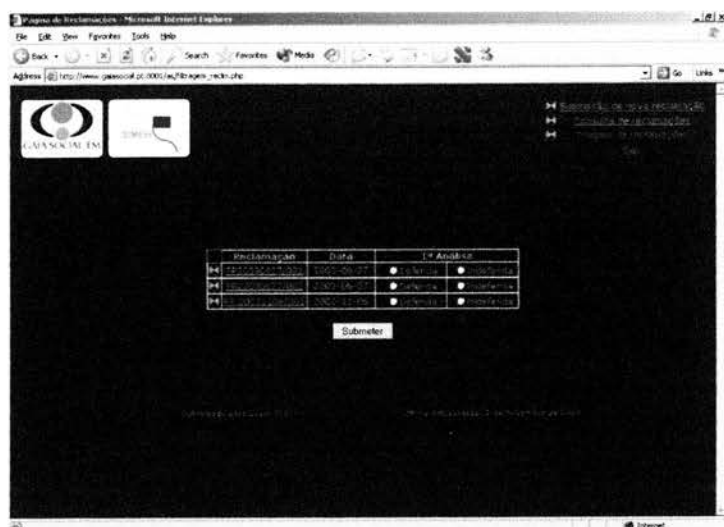


Figura 77: Interface do pacote Triar Reclamações

O modelo de classes deste módulo é apenas composto por uma área: *reclamações*, porque a fase de triagem corresponde apenas à alteração de um dos atributos desta classe.



Figura 78: Modelo de classes do módulo Triar Reclamações

7 Conclusão

No estágio referido foram estabelecidas, por várias vezes, metas e objectivos a atingir nas várias etapas e fases. Esses objectivos foram largamente atingidos, tendo o trabalho desenvolvido sido amplamente reconhecido pela instituição de estágio (IC) e pela instituição cliente (GaiaSocial).

O objectivo principal do estágio era desenvolver uma ferramenta de gestão para auxiliar à gestão de empreendimentos e, em especial, à sua manutenção. Uma ferramenta deste tipo necessita de um número muito grande de estruturas de dados de várias áreas para conseguir definir uma base de trabalho para desempenhar essa tarefa.

A necessidade da definição de uma grande quantidade de módulos para o desenvolvimento desta ferramenta obrigou a equipa de desenvolvimento a estabelecer prioridades e metas de menor prazo, para o correcto crescimento de uma estrutura de dados de base.

A principal característica do SIMEH é a integração dos vários módulos e o cruzamento dos dados pertencentes às várias bases de dados, permitindo assim usufruir das vantagens de obter uma visão mais globalizante do problema.

Do ponto de vista académico este estágio permitiu aplicar e aprofundar conhecimentos num vasto número de áreas da informática permitindo constatar, como seria de esperar, que as várias áreas de conhecimento interagem entre si no decorrer de um desenvolvimento correcto na área de engenharia.

Um dos maiores obstáculos consistiu na adequação do sistema a um cliente final com objectivos contraditórios ou por vezes, inexistentes, que obrigou a alterações ou abandono de algum trabalho previamente desenvolvido em mais do que uma fase de desenvolvimento.

O sistema desenvolvido encontra-se actualmente a ser amplamente testado na GaiaSocial, sendo aguardados os resultados da sua utilização intensiva para continuar a trabalhar na fase de maturação do produto. Devido ao potencial do projecto desenvolvido haverá sempre mais requisitos a explorar. Logo, a equipa de estágio vai manter o desenvolvimento da ferramenta, assim como das mais diversas aplicações paralelas.

O estágio pautou-se por condições excepcionais: na liberdade do local de trabalho, flexibilidade de horário, sigilo no decurso do desenvolvimento e acesso a dados sensíveis e confidenciais.

Entre as várias melhorias que podem ser referidas: a alteração dos vários módulos para permitir a manutenção de qualquer tipo de edifícios ou bloco estrutural é, sem dúvida, a principal. Esta melhoria permite a utilização de um sistema deste tipo na própria Faculdade de Engenharia da Universidade do Porto onde os benefícios se tornariam rapidamente evidentes.

Referências

- [1] JuniFEUP – Empresa Júnior da FEUP, Página Oficial - <http://www.junifeup.pt>
- [2] Instituto da Construção, Página Oficial- <http://www.fe.up.pt/~ic>
- [3] GaiaSocial E.M., Página Oficial - <http://www.gaiasocial.pt>
- [4] Microsoft Corporation, Página Oficial do Microsoft Access - <http://www.microsoft.com/office/access/default.asp>
- [5] Oracle Corporation, Página Oficial do SGBD Oracle 9i - <http://www.oracle.com/ip/dep/otn/database/oracle9i/>
- [6] Microsoft Corporation, Página Oficial do IIS - <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/iis/default.asp>
- [7] Microsoft Corporation, Página Oficial do Microsoft Windows Server 2003 - <http://www.microsoft.com/windowsserver2003/default.mspx>
- [8] Microsoft Corporation, Artigo da Microsoft acerca do IIS do win2003 server - <http://www.microsoft.com/windowsserver2003/evaluation/overview/technologies/iis.mspx>
- [9] Microsoft Corporation, Artigo técnico da Microsoft acerca do IIS - <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowsserver2003/proddocs/server/iiswelcome.asp>
- [10] Microsoft Corporation, Página Oficial do Microsoft Windows 2000 - <http://www.microsoft.com/windows2000/technologies/web/default.asp>
- [11] Microsoft Corporation, Artigo técnico acerca do .NET no Windows 2003 - <http://www.microsoft.com/windowsserver2003/evaluation/overview/dotnet/dotnet.mspx>
- [12] Apache Software Foundation, Página Oficial do servidor web da Apache - <http://httpd.apache.org/>
- [13] NetCraft Ltd, Página Oficial do Netcraft - <http://news.netcraft.com/>
- [14] Apache Software Foundation, Página Oficial - <http://www.apache.org/>
- [15] Apache Software Foundation, Licença de Utilização do software desenvolvido no Apache - <http://www.apache.org/LICENSE.txt>
- [16] ObjectMania! Consulting inc, Artigo da ObjectMania acerca de software open-source - <http://www.objectmania.com/expertise.jsp>
- [17] Netcraft Ltd, Lista de servidores com maior fiabilidades em Junho de 2003 - http://uptime.netcraft.com/perf/reports/performance/Hosters?tn=june_2003&reverse=0
- [18] Techtarget, Artigo na SearchWin2000 acerca do IIS e Apache - http://searchwin2000.techtarget.com/featuredTopic/0,290042,sid1_qci833246,00.html
- [19] Microsoft Corporation, Página Oficial do Visual Basic - <http://msdn.microsoft.com/vbasic/>
- [20] F.O.S. inc, Faq do iniciante do Visual Basic - <http://www.thebestweb.com/VBFAQS/index.asp>, 1997
- [21] Microsoft Corporation, Página Oficial do Visual Studio - <http://msdn.microsoft.com/library/en-us/vsintro/html/vsstartpage.asp>
- [22] Borland Corporation, Artigo comparativo do delphi vs vb, da Borland Community - <http://community.borland.com/article/0,1410,15689,00.html>
- [23] Microsoft Corporation, Página Oficial do Visual C++ - <http://msdn.microsoft.com/library/en-us/vccedit98/HTML/vcstartpage.asp>
- [24] M.Peresada, Página não-oficial de componentes de reports para o Delphi - http://www.torry.net/tools_reports.htm

- [25] JúpiterMedia Corporation, Artigo técnico acerca do Visual Basic 6 no WinPlanet - <http://www.winplanet.com/winplanet/reviews/661/1/>
- [26] Borland Corporation, Página Oficial do Borland Delphi - <http://www.borland.com/delphi/>
- [27] Borland Corporation, Página Oficial da Borland - <http://www.borland.com/>
- [28] Gladiston Santana, Artigo não-oficial de apresentação do Delphi - http://www.gladisto.hpg.ig.com.br/artigos/oque_e_delphi/oquee_delphi.htm, 2001
- [29] About inc, Coleção de artigos acerca do ObjectPascal Delphi.About.com - <http://delphi.about.com/cs/objectpascalide/>
- [30] IBM Corporation, Página dos produtos da Rational Rose - <http://www.rational.com/products/rose/>
- [31] Michael Moors, Página não-Oficial da Rational Rose - <http://www.rationalrose.com/>
- [32] ModelMaker Tools BV, Página Oficial do componente ModelMaker para Delphi - <http://www.modelmakertools.com/>
- [33] Borland Corporation, Página Oficial do Borland Interbase - <http://www.borland.com/interbase/>
- [34] Artigo a premiar o Delphi como "Produto do Ano" da revista checa de software SwNet - <http://swnet.cz/index.php?ID=18803>
- [35] Borland Corporation, Lista de prémios atribuídos ao Borland Delphi - <http://www.borland.com/delphi/awards/index.html>
- [36] Apache Software Foundation, Página Oficial do mod_perl - <http://perl.apache.org>
- [37] Sun Microsystems inc, Página Oficial do Comprehensive Perl Archive Network - <http://www.cpan.org/>
- [38] CooperMcGregor inc, Slide 80 da apresentação do Apache no LinuxWorld - <http://www.coopermcgregor.com/insight/conferences/linuxworld/slide80.html>
- [39] CooperMcGregor inc, Slide 79 da apresentação do Apache no LinuxWorld - <http://www.coopermcgregor.com/insight/conferences/linuxworld/slide79.html>
- [40] Microsoft Corporation, Página Oficial do ASP - <http://msdn.microsoft.com/library/default.asp?url=/nhp/default.asp?contentid=28000522>
- [41] Bruce Corkhill, Coleção de ligações acerca dos ASP no Webwizguide - <http://www.webwizguide.com/asp/default.asp>
- [42] Microsoft Corporation, Página Oficial do VbScript - <http://msdn.microsoft.com/library/en-us/script56/html/vtoriVBScript.asp>
- [43] Microsoft Corporation, Página Oficial do JScript - <http://msdn.microsoft.com/scripting/jscript/>
- [44] Bruce Corkhill, Tutorial acerca do ASP na Webwizguide - http://www.webwizguide.com/asp/tutorials/what_is.asp.asp
- [45] Sun Corporation, Página Oficial da Sun - <http://www.sun.com/>
- [46] Sun Corporation, Página Oficial da ChiliSoft - <http://www.chilisoft.com>
- [47] Sun Corporation, Página Oficial do I-Planet - <http://iplanet.com>
- [48] Zeus Technology, Página Oficial do Zeus - <http://www.zeus.com>
- [49] Red Hat inc, Página Oficial do Red Hat - <http://www.redhat.com>
- [50] Linux Online inc, Página Oficial do Linux - <http://www.linux.org>
- [51] Sun Corporation, Página Oficial do Solaris - <http://www.sun.com/solaris/>
- [52] HP Corporation, Página Oficial do Hp-UX - <http://www.hp.com/products1/unix/operating/>
- [53] IBM Corporation, Página Oficial do AIX - <http://www.ibm.com/servers/aix/>
- [54] Jupitermedia Corporation, Página Oficial da 4GuysFromRolla - <http://www.4guysfromrolla.com>
- [55] Chamas Enterprises inc, Página Oficial de integração do ASP no Apache - <http://www.apache-asp.org/>

- [56] Microsoft Corporation, Página Oficial do ASP.NET - <http://asp.net/>
- [57] BrillianceWeb inc, Artigo da BrillianceWeb acerca das vantagens do ASP.NET - http://www.brillianceweb.com/betterwebdesign/tips_52.aspx
- [58] Microsoft Corporation, Página Oficial do .NET - <http://www.microsoft.com/net/>
- [59] Microsoft Corporation, Página Oficial do Notepad - <http://www.notepad.org/>
- [60] ES-Computing, Página Oficial do Editplus - <http://www.editplus.com/>
- [61] Microsoft Corporation, Página Oficial do Visual Studio .NET - <http://msdn.microsoft.com/vstudio/default.asp>
- [62] PHP Group, Página Oficial do PHP - <http://www.php.net>
- [63] Programação estruturada - <http://www.nota6.hpg.ig.com.br/estruturada.doc>
- [64] Luiz Paulo Maia, Edurado Morelli, Artigo acerca da Programação Orientada a Objectos - http://www.training.com.br/pub_prog_oo.htm
- [65] dBase inc, Página Oficial do dBase - <http://www.dbase.com/>
- [66] Borland Corporation, Pagina Oficial do Interbase - <http://www.interbase.com/>
- [67] PostgreSQL Global Development Group, Página Oficial do postgresql - <http://www.postgresql.org/>
- [68] Sybase inc, Página Oficial do SyBase - <http://www.sybase.com/>
- [69] Microsoft Corporation, Página acerca do ODBC - http://msdn.microsoft.com/library/devprods/vs6/visualc/vccore/core_database_topics_28.odbc.29.htm
- [70] IBM Corporation, Página Oficial d do Informix - <http://www.informix.com/>
- [71] Sun Microsystems Inc, Página Oficial do Java - <http://java.sun.com/>
- [72] PHP Documentation Group, Apresentação das características no manual Online do Php - http://pt2.php.net/manual/pt_BR/printwn/features.php
- [73] PHP Documentation Group, Apresentação das funcionalidades no manual Online do Php - http://pt2.php.net/manual/pt_BR/printwn/intro-whatcando.php, 2003
- [74] PHP Group, Página Oficial do PEAR - <http://pear.php.net/>
- [75] PHP developer's network , Página Oficial - <http://www.devnetwork.net/>
- [76] John Lim, 7 Reasons why PHP is better than ASP - http://php.weblogs.com/php_asp_7_reasons , 2000
- [77] Sun Microsystems Inc, Página Oficial do JSP - <http://java.sun.com/products/jsp/>
- [78] Brett McLaughlin, JSP technology – friend or foe? - <http://www-106.ibm.com/developerworks/library/w-friend.html?dwzone=web> , 2000
- [79] John Lim , What is Wrong with Java Server Pages - <http://php.weblogs.com/jsp> , 2001
- [80] Jason Hunter , The Problems with JSP - <http://www.servlets.com/soapbox/problems-jsp.html> , 2000
- [81] NetCraft Ltd, Estudo de Utilização de Servidores web Activos - http://news.netcraft.com/archives/2003/08/01/august_2003_web_server_survey.html , 2003

Anexos

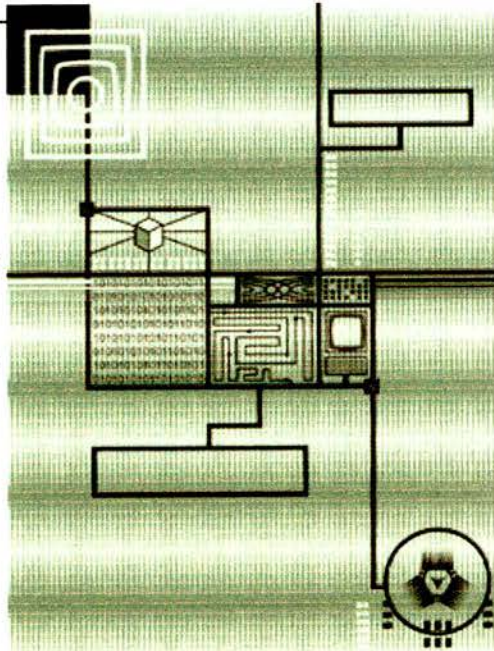
Anexo A – Unified Modeling Language

Nas páginas seguintes apresenta-se o documento intitulado “UML basics: An introduction to the Unified Modeling Language” cujo autor é Donald Bell da IBM Global Services. O documento original pode ser encontrado em http://www.therationaledge.com/content/jun_03/f_umlintro_db.jsp.

UML basics: An introduction to the Unified Modeling Language

by [Donald Bell](#)
 IBM Global Services

Way back in the late twentieth century - 1997 to be exact -- the Object Management Group (OMG) released the Unified Modeling Language (UML). One of the purposes of UML was to provide the development community with a stable and common design language that could be used to develop and build computer applications. UML brought forth a unified standard modeling notation that IT professionals had been wanting for years. Using UML, IT professionals could now read and disseminate system structure and design plans -- just as construction workers have been doing for years with blueprints of buildings.



It is now the twenty-first century -- 2003 to be precise -- and UML has gained traction in our profession. On 75 percent of the resumes I see, there is a bullet point claiming knowledge of UML. However, after speaking with a majority of these job candidates, it becomes clear that they do not truly know UML. Typically, they are either using it as a buzz word, or they have had a sliver of exposure to UML. This lack of understanding inspired me to write this quick introduction to UML 1.4. When you are finished reading you will not have enough knowledge to put UML on your resume, but you will have a starting point for digging more deeply into the language.

A little background

As I mentioned, UML was meant to be a unifying language enabling IT professionals to model computer applications. The primary authors were

- ▶ [subscribe](#)
- ▶ [contact us](#)
- ▶ [submit an article](#)
- ▶ [rational.com](#)
- ▶ [issue contents](#)
- ▶ [archives](#)
- ▶ [mission statement](#)
- ▶ [editorial staff](#)

Jim Rumbaugh, Ivar Jacobson, and Grady Booch, who originally had their own competing methods (OMT, OOSE, and Booch). Eventually, they joined forces and brought about an open standard. (Sound familiar? A similar phenomenon spawned J2EE, SOAP, and Linux.) One reason UML has become a *standard* modeling language is that it is programming-language independent. (UML modeling tools from IBM Rational are used extensively in J2EE shops as well in .Net shops.) Also, the UML notation set is a language and not a methodology. This is important, because a language, as opposed to a methodology, can easily fit into any company's way of conducting business without requiring change.

Since UML is not a methodology, it does not require any formal work products (i.e., "artifacts" in IBM Rational Unified Process® lingo). Yet it does provide several types of diagrams that, when used within a given methodology, increase the ease of understanding an application under development. There is more to UML than these diagrams, but for my purposes here, the diagrams offer a good introduction to the language and the principles behind its use. By placing standard UML diagrams in your methodology's work products, you make it easier for UML-proficient people to join your project and quickly become productive. The most useful, standard UML diagrams are: use case diagram, class diagram, sequence diagram, statechart diagram, activity diagram, component diagram, and deployment diagram.

It is beyond the scope of this introductory article to go into great detail about each type of diagram. Instead, I will provide you with enough information for a general understanding of each one and then supply more details in later articles.

Use-case diagram

A use case illustrates a unit of functionality provided by the system. The main purpose of the use-case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases. Use-case diagrams generally show groups of use cases -- either all use cases for the complete system, or a breakout of a particular group of use cases with related functionality (e.g., all security administration-related use cases). To show a use case on a use-case diagram, you draw an oval in the middle of the diagram and put the name of the use case in the center of, or below, the oval. To draw an actor (indicating a system user) on a use-case diagram, you draw a stick person to the left or right of your diagram (and just in case you're wondering, some people draw prettier stick people than others). Use simple lines to depict relationships between actors and use cases, as shown in Figure 1.

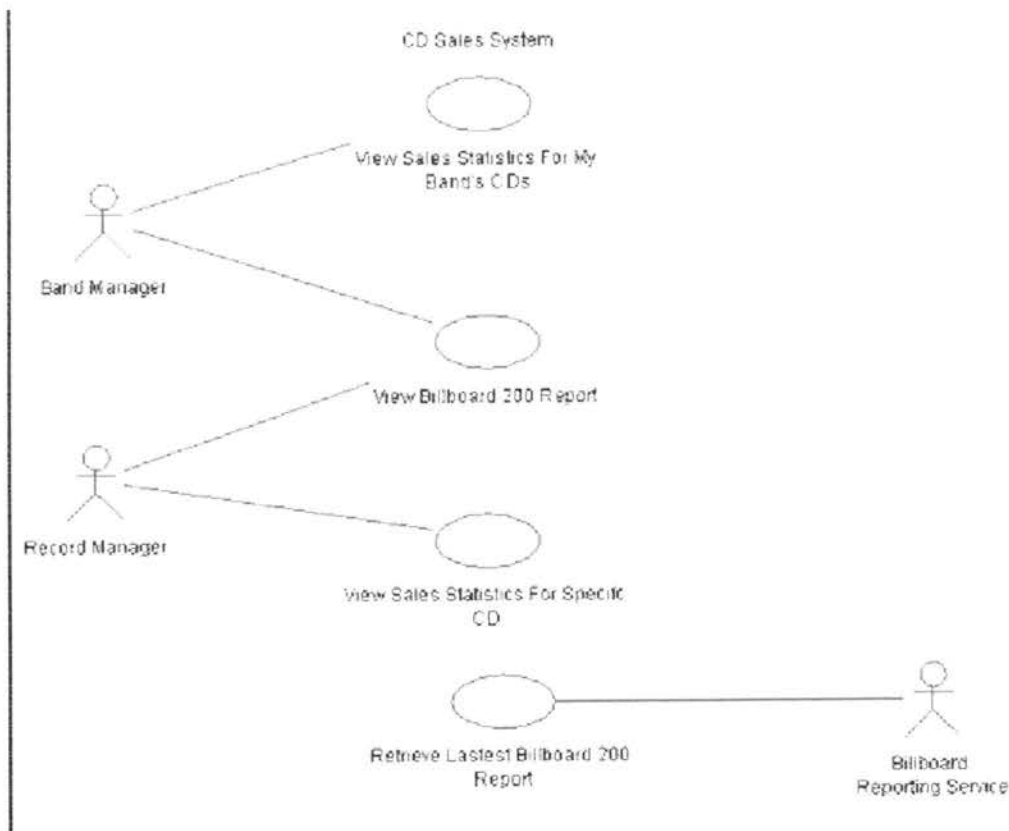


Figure 1: Sample use-case diagram
[Click to enlarge](#)

A use-case diagram is typically used to communicate the high-level functions of the system and the system's scope. By looking at our use-case diagram in Figure 1, you can easily tell the functions that our example system provides. This system lets the band manager view a sales statistics report and the Billboard 200 report for the band's CDs. It also lets the record manager view a sales statistics report and the Billboard 200 report for a particular CD. The diagram also tells us that our system delivers Billboard reports from an external system called Billboard Reporting Service.

In addition, the absence of use cases in this diagram shows what the system *doesn't* do. For example, it does not provide a way for a band manager to listen to songs from the different albums on the Billboard 200 - - i.e., we see no reference to a use case called Listen to Songs from Billboard 200. This absence is not a trivial matter. With clear and simple use-case descriptions provided on such a diagram, a project sponsor can easily see if needed functionality is present or not present in the system.

Class diagram

The class diagram shows how the different entities (people, things, and data) relate to each other; in other words, it shows the static structures of the system. A class diagram can be used to display logical classes, which are typically the kinds of things the business people in an organization talk

about -- rock bands, CDs, radio play; or loans, home mortgages, car loans, and interest rates. Class diagrams can also be used to show implementation classes, which are the things that programmers typically deal with. An implementation class diagram will probably show some of the same classes as the logical classes diagram. The implementation class diagram won't be drawn with the same attributes, however, because it will most likely have references to things like Vectors and HashMaps.

A class is depicted on the class diagram as a rectangle with three horizontal sections, as shown in Figure 2. The upper section shows the class's name; the middle section contains the class's attributes; and the lower section contains the class's operations (or "methods").

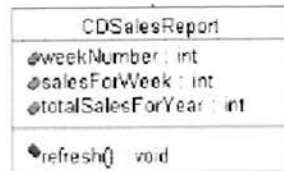


Figure 2: Sample class object in a class diagram

In my experience, almost every developer knows what this diagram is, yet I find that most programmers draw the relationship lines incorrectly. For a class diagram like the one in Figure 3, you should draw the inheritance relationship¹ using a line with an arrowhead at the top pointing to the super class, and the arrowhead should be a *completed triangle*. An association relationship should be a solid line if both classes are aware of each other and a line with an *open arrowhead* if the association is known by only one of the classes.

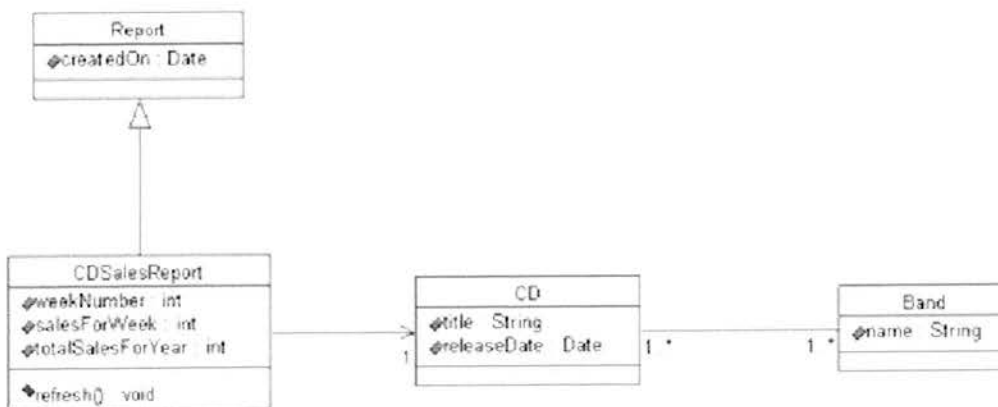


Figure 3: A complete class diagram, including the class object shown in Figure 2

[Click to enlarge](#)

In Figure 3, we see both the inheritance relationship and two association relationships. The CDSalesReport class inherits from the Report class. A CDSalesReport is associated with one CD, but the CD class doesn't know anything about the CDSalesReport class. The CD and the Band classes both know about each other, and both classes can be associated to one or more of each other.

A class diagram can incorporate many more concepts, which we will cover later in this article series.

Sequence diagram

Sequence diagrams show a detailed flow for a specific use case or even just part of a specific use case. They are almost self explanatory; they show the calls between the different objects in their sequence and can show, at a detailed level, different calls to different objects.

A sequence diagram has two dimensions: The vertical dimension shows the sequence of messages/calls in the time order that they occur; the horizontal dimension shows the object instances to which the messages are sent.

A sequence diagram is very simple to draw. Across the top of your diagram, identify the class instances (objects) by putting each class instance inside a box (see Figure 4). In the box, put the class instance name and class name separated by a space/colon/space " : " (e.g., myReportGenerator : ReportGenerator). If a class instance sends a message to another class instance, draw a line with an open arrowhead pointing to the receiving class instance; place the name of the message/method above the line. Optionally, for important messages, you can draw a dotted line with an arrowhead pointing back to the originating class instance; label the return value above the dotted line. Personally, I always like to include the return value lines because I find the extra details make it easier to read.

Reading a sequence diagram is very simple. Start at the top left corner with the "driver" class instance that starts the sequence. Then follow each message down the diagram. Remember: Even though the example sequence diagram in Figure 4 shows a return message for each sent message, this is optional.

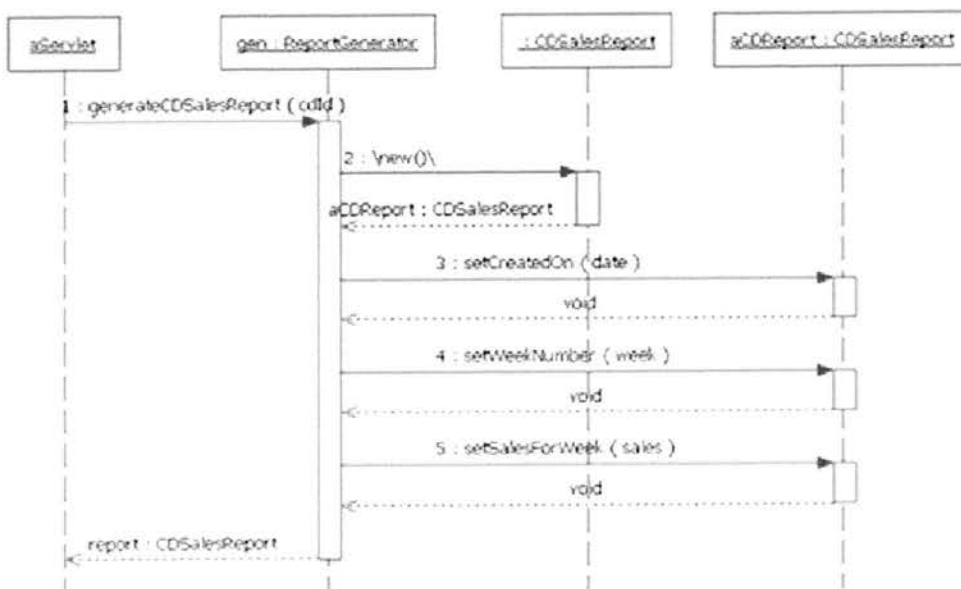


Figure 4: A sample sequence diagram

[Click to enlarge](#)

By reading our sample sequence diagram in Figure 4, you can see how to create a CD Sales Report. The aServlet object is our example driver. aServlet sends a message to the ReportGenerator class instance named gen. The message is labeled generateCDSalesReport, which means that the ReportGenerator object implements this message handler. On closer inspection, the generateCDSalesReport message label has cdId in parentheses, which means that aServlet is passing a variable named cdId with the message. When gen instance receives a generateCDSalesReport message, it then makes subsequent calls to the CDSalesReport class, and an actual instance of a CDSalesReport called aCDReport gets returned. The gen instance then makes calls to the returned aCDReport instance, passing it parameters on each message call. At the end of the sequence, the gen instance returns aCDReport to its caller aServlet.

Please note: The sequence diagram in Figure 4 is arguably too detailed for a typical sequence diagram. However, I believe it is simple enough to understand, and it shows how nested calls are drawn. Also, with junior developers, sometimes it is necessary to break down sequences to this explicit level to help them understand what they are supposed to do.

Statechart diagram

The statechart diagram models the different states that a class can be in and how that class transitions from state to state. It can be argued that every class has a state, but that every class shouldn't have a statechart diagram. Only classes with "interesting" states -- that is, classes with three or more potential states during system activity -- should be modeled.

As shown in Figure 5, the notation set of the statechart diagram has five basic elements: the initial starting point, which is drawn using a solid circle; a transition between states, which is drawn using a line with an open arrowhead; a state, which is drawn using a rectangle with rounded corners; a decision point, which is drawn as an open circle; and one or more termination points, which are drawn using a circle with a solid circle inside it. To draw a statechart diagram, begin with a starting point and a transition line pointing to the initial state of the class. Draw the states themselves anywhere on the diagram, and then simply connect them using the state transition lines.

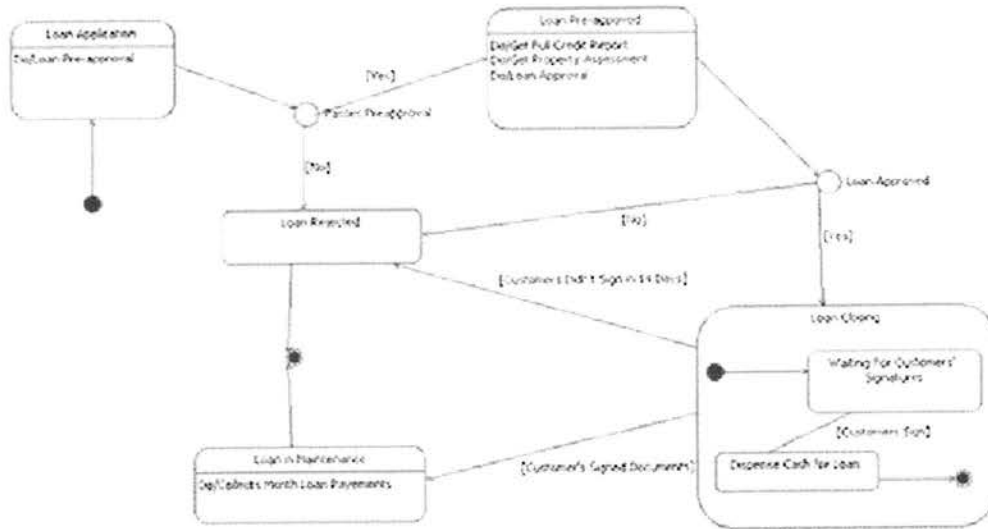


Figure 5: Statechart diagram showing the various states that classes pass through in a functioning system
[Click to enlarge](#)

The example statechart diagram in Figure 5 shows some of the potential information they can communicate. For instance, you can tell that loan processing begins in the Loan Application state. When the pre-approval process is done, depending on the outcome, you move to either the Loan Pre-approved state or the Loan Rejected state. This decision, which is made during the transition process, is shown with a decision point -- the empty circle in the transition line. By looking at the example, a person can tell that a loan cannot go from the Loan Pre-Approved state to the Loan in Maintenance state without going through the Loan Closing state. Also, by looking at our example diagram, a person can tell that all loans will end in either the Loan Rejected state or the Loan in Maintenance state.

Activity diagram

Activity diagrams show the procedural flow of control between two or more class objects while processing an activity. Activity diagrams can be used to model higher-level business process at the business unit level, or to model low-level internal class actions. In my experience, activity diagrams are best used to model higher-level processes, such as how the company is currently doing business, or how it would like to do business. This is because activity diagrams are "less technical" in appearance, compared to sequence diagrams, and business-minded people tend to understand them more quickly.

An activity diagram's notation set is similar to that used in a statechart diagram. Like a statechart diagram, the activity diagram starts with a solid circle connected to the initial activity. The activity is modeled by drawing a rectangle with rounded edges, enclosing the activity's name. Activities can be connected to other activities through transition lines, or to decision points that connect to different activities guarded by conditions of the decision point. Activities that terminate the modeled process are connected to a termination point (just as in a statechart diagram).

Optionally, the activities can be grouped into swimlanes, which are used to indicate the object that actually performs the activity, as shown in Figure 6.

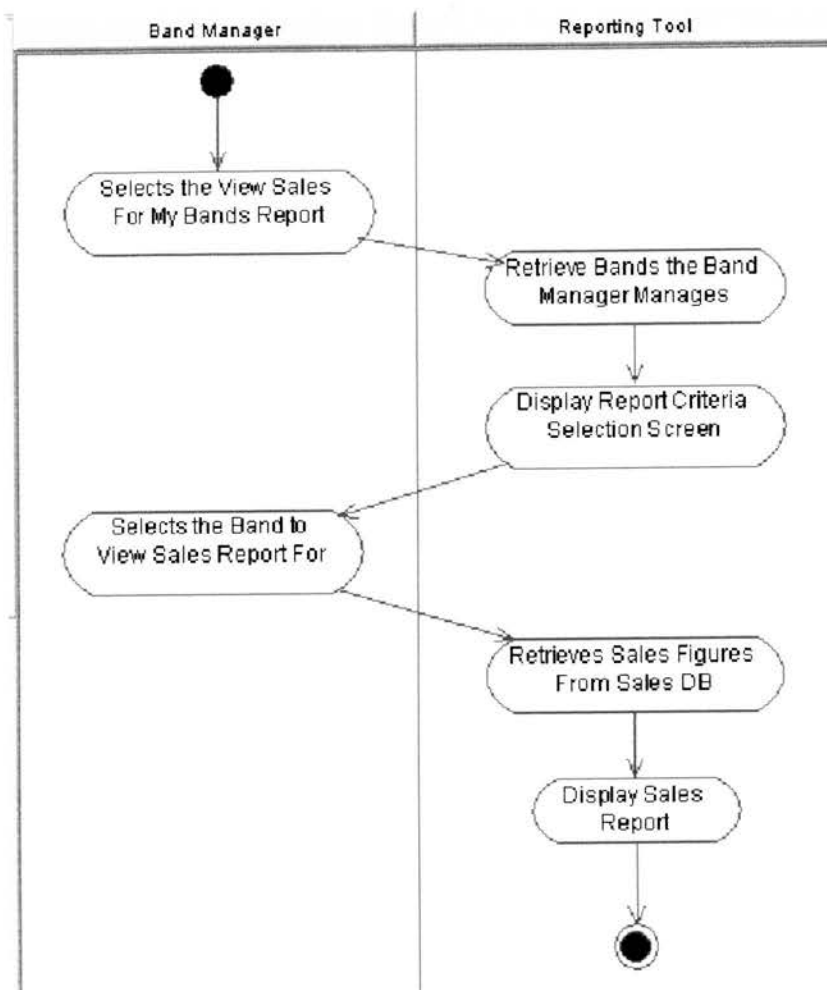


Figure 6: Activity diagram, with two swimlanes to indicate control of activity by two objects: the band manager, and the reporting tool

In our example activity diagram, we have two swimlanes because we have two objects that control separate activities: a band manager and a reporting tool. The process starts with the band manager electing to view the sales report for one of his bands. The reporting tool then retrieves and displays all the bands that person manages and asks him to choose one. After the band manager selects a band, the reporting tool retrieves the sales information and displays the sales report. The activity diagram shows that displaying the report is the last step in the process.

Component diagram

A component diagram provides a physical view of the system. Its purpose is to show the dependencies that the software has on the other software components (e.g., software libraries) in the system. The diagram can be

shown at a very high level, with just the large-grain components, or it can be shown at the component package level.²

Modeling a component diagram is best described through an example. Figure 7 shows four components: Reporting Tool, Billboard Service, Servlet 2.2 API, and JDBC API. The arrowed lines from the Reporting Tool component to the Billboard Service, Servlet 2.2 API, and JDBC API components mean that the Reporting Tool is dependent on those three components.

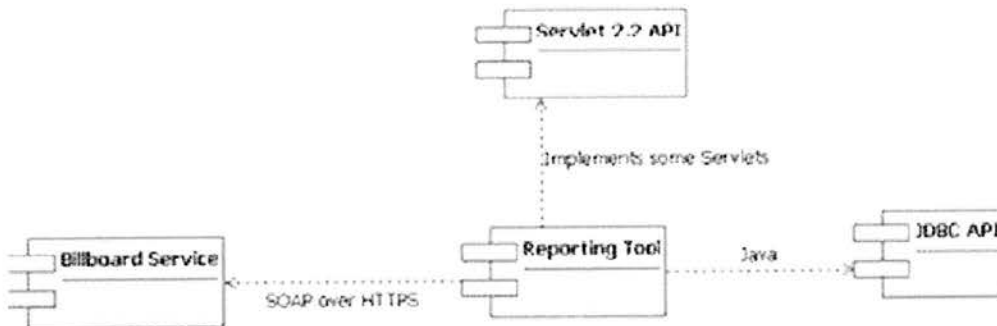


Figure 7: A component diagram shows interdependencies of various software components the system comprises
[Click to enlarge](#)

Deployment diagram

The deployment diagram shows how a system will be physically deployed in the hardware environment. Its purpose is to show where the different components of the system will physically run and how they will communicate with each other. Since the diagram models the physical runtime, a system's production staff will make considerable use of this diagram.

The notation in a deployment diagram includes the notation elements used in a component diagram, with a couple of additions, including the concept of a node. A node represents either a physical machine or a virtual machine node (e.g., a mainframe node). To model a node, simply draw a three-dimensional cube with the name of the node at the top of the cube. Use the naming convention used in sequence diagrams: [instance name] : [instance type] (e.g., "w3reporting.myco.com : Application Server").

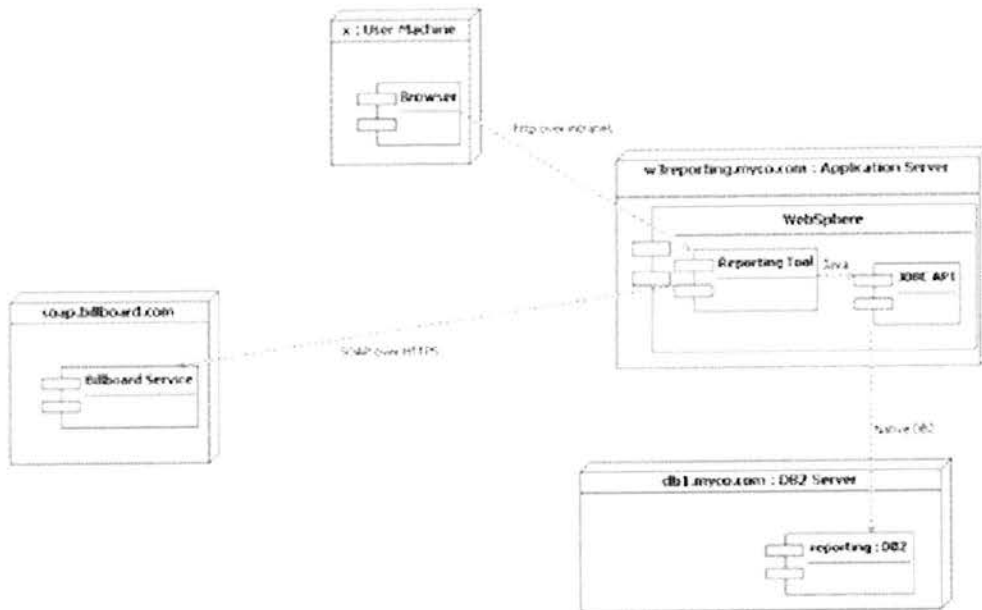


Figure 8: Deployment diagram. Because the Reporting Tool component is drawn inside of IBM WebSphere, which in turn is drawn inside of the node w3.reporting.myco.com, we know that users will access the Reporting Tool via a browser running on their local machine, and connecting via HTTP over their company's intranet.

[Click to enlarge](#)

The deployment diagram in Figure 8 shows that the users access the Reporting Tool by using a browser running on their local machine and connecting via HTTP over their company's intranet to the Reporting Tool. This tool physically runs on the Application Server named w3reporting.myco.com. The diagram shows the Reporting Tool component drawn inside of IBM WebSphere, which in turn is drawn inside of the node w3.reporting.myco.com. The Reporting Tool connects to its reporting database using the Java language to IBM DB2's JDBC interface, which then communicates to the actual DB2 database running on the server named db1.myco.com using native DB2 communication. In addition to talking to the reporting database, the Report Tool component communicates via SOAP over HTTPS to the Billboard Service.

Conclusion

Although this article provides only a brief introduction to Unified Modeling Language, I encourage you to start applying the information you have learned here to your own projects and to dig more deeply into UML. There are several software tools that help you to integrate UML diagrams into your software development process, but even without automated tools, you can use markers on a whiteboard or paper and pencils to draw your UML diagrams and still achieve benefits.

Notes

¹ For more information on inheritance and other object-oriented principles, see <http://java.sun.com/docs/books/tutorial/java/concepts/inheritance.html>

² The phrase component package level is a programming language-neutral way of referring to class container levels such as .Net's namespaces (e.g., System.Web.UI) or Java's packages (e.g., java.util).

Resources

<http://www.uml.org> -- The official UML Web site.

<http://www.rational.com/uml/resources/documentation/index.jsp> --Offers several different versions of the actual UML specification.

<http://www.rational.com/rose> --Information on IBM Rational Rose,[®] A commercial UML modeling tool.

<http://www.rational.com/xde> --Information on IBM Rational XDE,[®] a commercial UML modeling tool that is integrated with IBM's Eclipse development platform.

<http://argouml.tigris.org> --Information on Argo UML, an .open source UML modeling tool built in Java.

<http://uml.sourceforge.net/index.php> -- Information on Umbrello UML Modeller, an open source UML modeling tool for KDE.



For more information on the products or services discussed in this article, please click [here](#) and follow the instructions provided. Thank you!

Anexo B – Critérios de Usabilidade

Há várias formas de avaliar a usabilidade de interfaces. Uma delas é a verificação das heurísticas de Nielsen [http://www.useit.com/papers/heuristic/heuristic_list.html]:

Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

Recognition rather than recall

Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.





FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000078228

