

Universidade Lusófona de Humanidades e Tecnologias
Licenciatura em Engenharia Informática
Departamento de Comunicação, Artes e Tecnologias da Informação

Relatório Final

Software de Gestão de Clínicas

Discente:	Hugo Castro, N.º 20064150
Orientador:	Prof. António Frazão

Lisboa, Dezembro de 2007

Índice	Pág.
0. Abstract	2
1. Introdução	3
2. Objectivo	4
3. Descrição do Projecto.	4
4. Arquitectura.	8
5. Modelo de Objectos.	8
5.1. Diagrama de Classes	9
5.1.2. Descrição das Classes	10
5.1.2.1. Utilizadores e Permissões	10
5.1.2.2. Médicos, Especialidades, Entidades e Agenda	10
5.1.2.3. Consultas, Agenda, Pacientes e Entidades	10
5.2. Modelo de Implementação	11
6. Modelo Funcional	12
7. Conclusão	16
8. Bibliografia	18
9. Anexos	19

0. Abstract

Reaching the final phase of the degree in Computer Engineering, came the idea of developing a web application, to be held in the class of Project. This idea came from the fact that there is a niche market to explore with respect to web applications for clinical management. The proposal is based on an on-line application of Clinical Management. It is, with no doubt, a very innovative idea, and can be a very useful and convenient tool. This software aims to make all the operational management of the clinic, the patients, the doctors and their agenda, office appointments, among others.

The application is a vital tool in the daily management of a company. All its functioning mode is geared to improve the performance of enterprises, increasing the productivity of collaborators, minimizing the time for work and thus facilitating the routine tasks.

The application uses new technologies such as the case of Servlets and use of the database MySQL, as are technologies that stand out in the world of work.

The technologies used are quite appropriate, since the Java together with the database MySQL give enough strength, flexibility and security to the application. Another important aspect is that these technologies are "free".

In this document are described the functions of the Software.

1. Introdução

Atingindo a fase final da Licenciatura em Engenharia Informática, surgiu uma ideia de elaboração de uma aplicação web, a ser realizada na cadeira de Projecto. Esta ideia veio do facto de haver um nicho de mercado por explorar no que diz respeito a aplicações web para gestão de clínicas. A proposta baseia-se numa aplicação on-line de Gestão de Clínicas. É sem dúvida uma ideia bastante inovadora, podendo ser uma ferramenta muito útil e cómoda, usando as novas tecnologias para simplificar o dia a dia de uma clínica.

Depois de alguma investigação numa clínica, tentando perceber as suas necessidades, e falando com as pessoas que trabalham diariamente com uma aplicação semelhante, fiquei com uma ideia mais clara do que é esperado de uma aplicação para a gestão de clínicas.

Mas esta aplicação não é só uma boa ferramenta. Sob o meu ponto de vista, foi também uma boa oportunidade para aplicar várias matérias leccionadas ao longo dos 3 anos de licenciatura, bem como, aplicar boas práticas de Engenharia de Programação, contribuindo para o sucesso de uma aplicação bem estruturada.

2. Objectivo

Com este software pretende-se efectuar toda a gestão operacional da clínica, dos pacientes, dos médicos e da sua agenda, das marcações de consultas, entre outros.

O programa usa as novas tecnologias e tem como principal objectivo simplificar o dia a dia de uma clínica, “user friendly”

A aplicação é uma ferramenta imprescindível na gestão diária de uma empresa. Todo o seu modo de funcionamento está orientado para melhorar a actuação das empresas, aumentando a produtividade dos colaboradores, minimizando os tempos de trabalho e facilitando assim as tarefas rotineiras.

Neste documento estão descritas as funcionalidades do Software.

O objectivo deste documento é de descrever o trabalho realizado pelo aluno Hugo Castro no desenvolvimento deste projecto nomeadamente a análise funcional, desenho, desenvolvimento do código e documentação.

3. Descrição do Projecto

O presente projecto passou por várias fases que em conjunto permitiram chegar ao produto final que é a aplicação em si. Essas fases passaram:

- pelo estudo das tecnologias envolvidas no desenvolvimento da aplicação;
- pela análise das funcionalidades que a aplicação terá de ter;
- pelo desenho da aplicação de modo a satisfazer as funcionalidades requeridas na análise funcional;
- pelo desenvolvimento da aplicação utilizando as tecnologias estudadas na implementação do desenho da mesma;
- pelo documento do projecto.

Para o desenvolvimento da aplicação foi escolhida a tecnologia Java. Esta linguagem é caracterizada pela sua portabilidade, pois os programas Java correm em todas as plataformas desde que haja uma Java Virtual Machine.

A linguagem Java foi projectada tendo em vista os seguintes objectivos:

- Orientação a objectos - Baseado no modelo de Smalltalk e Simula67;
- Portabilidade - Independência de plataforma - "write once run anywhere";
- Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP ;
- Segurança - Pode executar programas via rede com restrições de execução;

Além disso, podem-se destacar outras vantagens apresentadas pela linguagem:

- Sintaxe similar a Linguagem C/C++.
- Facilidades de Internacionalização – Suporta nativamente caracteres Unicode;
- Simplicidade na especificação, tanto da linguagem como do "ambiente" de execução (JVM);
- É distribuída com um vasto conjunto de bibliotecas (ou APIs);
- Possui facilidades para criação de programas distribuídos e multitarefa (múltiplas linhas de execução num mesmo programa);
- Desalocação de memória automática por processo de colector de lixo (garbage collector);
- Carga Dinâmica de Código - Programas em Java são formados por uma colecção de classes armazenadas independentemente e que podem ser carregadas no momento de utilização.

Também foi usado O Hibernate para gerar os DAO.

O Hibernate é um framework para o mapeamento objecto-relacional escrito na linguagem Java, mas também é disponível em .Net como o nome Nhibernate. Este programa facilita o mapeamento dos atributos entre uma base tradicional de dados

relacionais e o modelo objecto de uma aplicação, mediante o uso de arquivos (XML) para estabelecer esta relação.

Hibernate é um software livre de código aberto distribuído com a licença LGPL. O objectivo do Hibernate é diminuir a complexidade entre os programas Java, baseado no modelo orientado a objectos, que precisam trabalhar com um banco de dados do modelo relacional (presente na maioria dos SGDBs). Em especial, no desenvolvimento de consultas e actualizações dos dados.

A sua principal característica é a transformação das classes em Java para tabelas de dados (e dos tipos de dados Java para os SQL). O Hibernate gera as chamadas SQL e retira ao programador o trabalho manual da conversão dos dados resultante, mantendo o programa portátil para quaisquer bancos de dados SQL, porém causando um pequeno aumento no tempo de execução.

Nas questões relacionadas para o gerenciamento de transacções e na tecnologia de acesso à base de dados são de responsabilidade de outros elementos na infra-estrutura do programa. Apesar de existirem API no Hibernate para possuir operações de controle transaccional, ele simplesmente delegará estas funções para a infra-estrutura na qual foi instalada.

No caso de aplicações construídas para serem executadas em servidores de aplicação, o gerenciamento das transacções é realizado segundo o padrão JTA. Já nas aplicações standalone, o programa delega o tratamento transaccional ao driver JDBC, que é o caso desta aplicação.

Hibernate pode ser utilizado em aplicações Java standalone ou em aplicações Java EE, utilizando servlets ou sessões EJB beans.

Na aplicação são geradas páginas HTML dinâmicas com a ajuda de Servlets programadas em Java. Uma Servlet é uma classe Java que é carregada dinamicamente e executada por um servidor WEB especial que será um servidor de servlets.

Para o aspecto gráfico foram usadas duas ferramentas:

A User Interface (YUI) Library que é um conjunto de utilitários e controlos, escrito em JavaScript, para construir interactivamente aplicações web usando técnicas como scripts DOM, DHTML e AJAX. A YUI Library também inclui vários recursos a CSS. Todos os componentes da YUI Library foram liberadas como código aberto sob uma licença BSD e são livres para todos os usos.

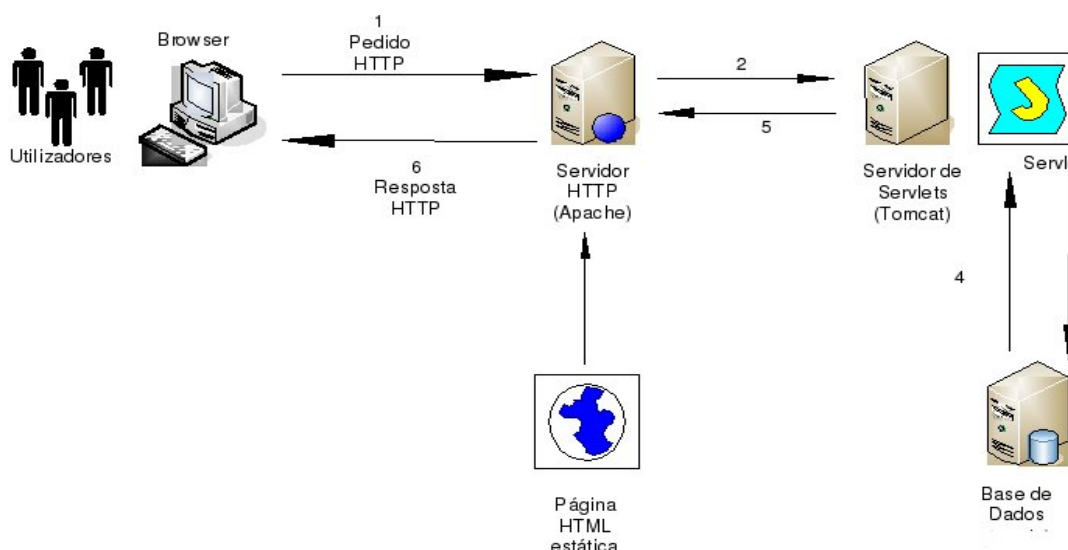
E o Mootools que é uma framework de JavaScript compacta, modular e Orientada a Objectos, desenhado para programadores de JavaScript. Permite escrever um código poderoso, flexível, and cross-browser com elegância, bem documentado e coerente API. Mootools é um código que respeita normas rigorosas e é livre.

No que toca à base de dados, foi usado o MySql 5.0 que tem como principais características:

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

4. Arquitectura

Acesso ao software de Gestão pela Internet, através do protocolo http. O software e dados ficam centralizados em servidores dedicados, permitindo fácil manutenção e acesso distribuído nas filiais, fornecedores, clientes e eventualmente em terminais móveis. Todo o acesso pelos vários utilizadores será verificado em termos de permissões e perfis de acesso.

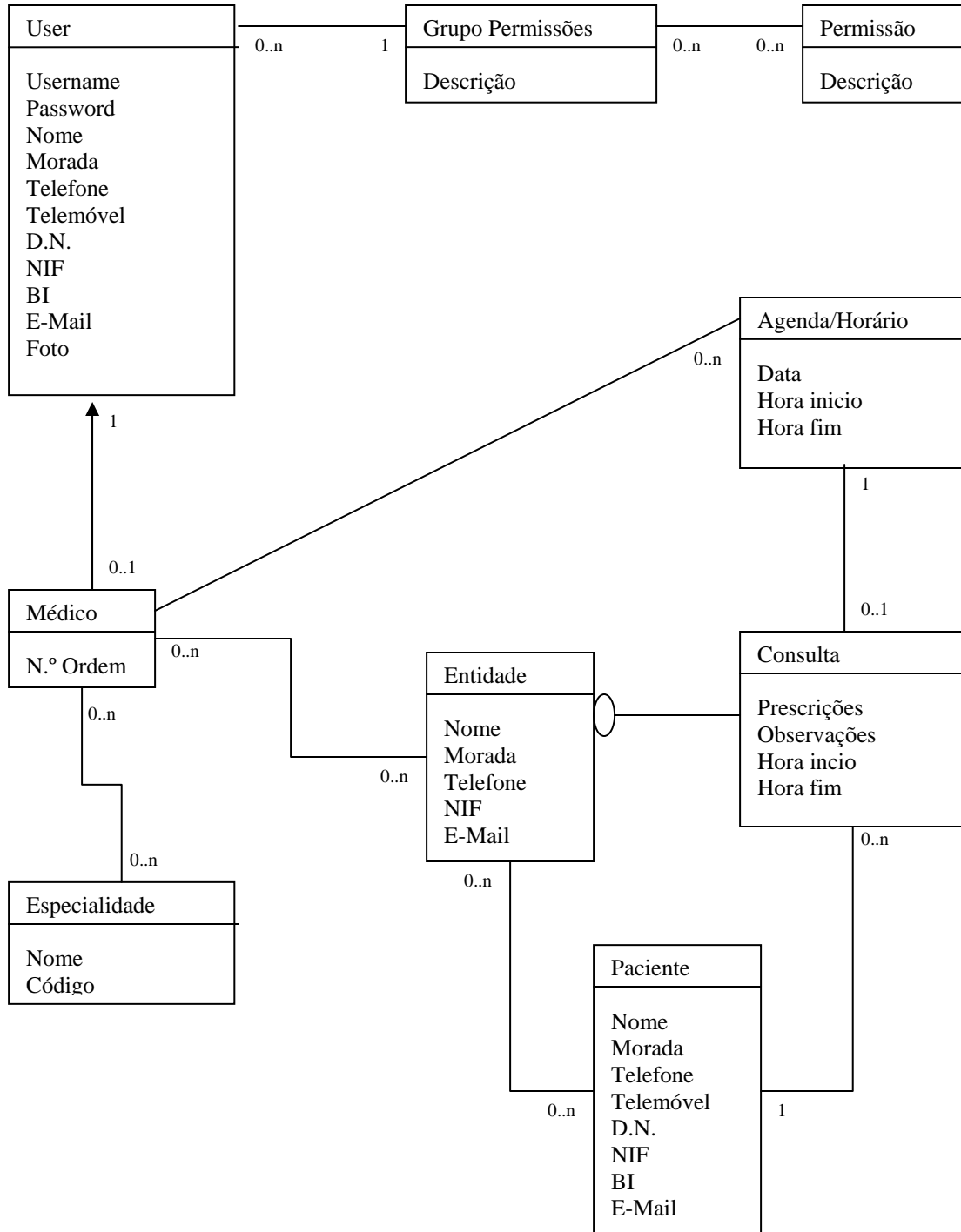


A servlet é carregada para o servidor na primeira vez que é solicitada a sua execução. Da sua execução resulta um “ficheiro” HTML que é devolvido ao servidor e deste é enviado para o cliente. Depois de executada, a servlet permanece em memória, esperando por novo pedido. Só será retirada da memória caso o servidor fique sem memória disponível. Cada vez que é solicitada a execução de uma servlet o servidor verifica se a versão que está em memória é a mais recente (comparação por data/hora). Se não for, é carregada a partir do disco a nova versão. Desta forma evita-se a paragem/arranque do servidor para substituir versões das classes Java.

5. Modelo de objectos

Irá ser mostrado neste ponto, os modelos extraídos depois de efectuada a pesquisa junto do cliente, bem como uma descrição dos mesmos.

5.1. Diagrama de Classes:



5.1.2. Descrição das Classes:

5.1.2.1. - Utilizadores e Permissões:

Um user é associado a um grupo de permissões podendo conter 0 ou mais permissões. Um user poderá ser ou não um médico.

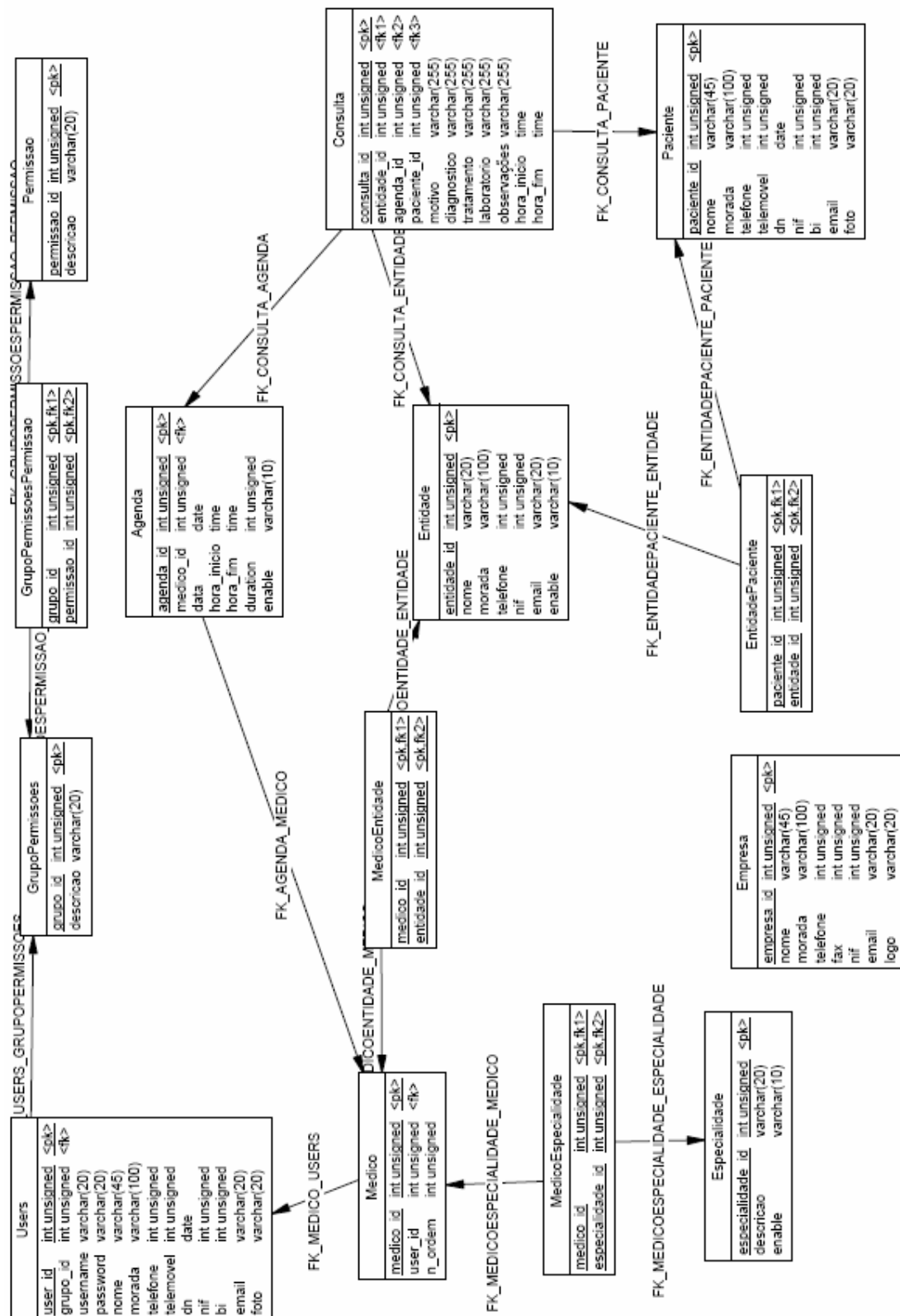
5.1.2.2. - Médicos, Especialidades, Entidades e Agenda:

O Médico poderá ser associado a 0 ou mais Especialidades e Entidades, poderá estar associado a 0 ou mais Agendas. As agendas contêm as disponibilidades inseridas pelos médicos.

5.1.2.3. - Consultas, Agenda, Pacientes e Entidades:

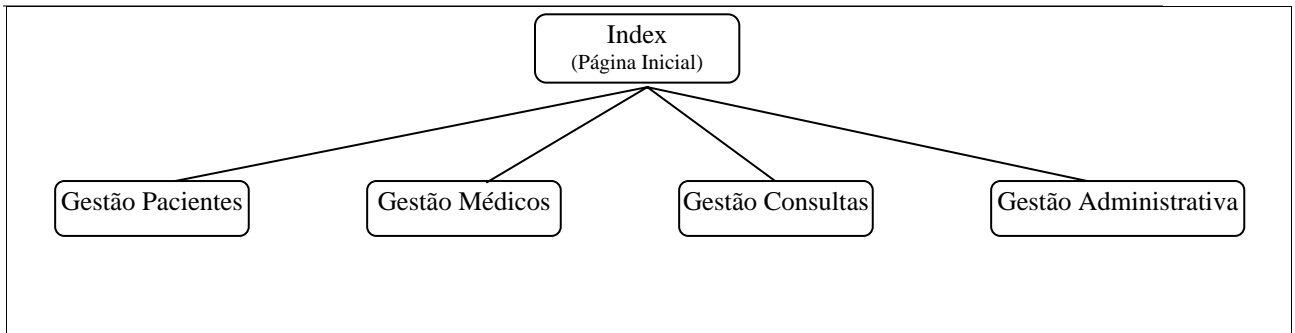
As Consultas estão associadas a uma única Agenda e a um único Paciente. Estas são basicamente o seu historial. Nelas contém todos os dados referentes a cada consulta realizada. Poderá também estar associada uma entidade à consulta, que será uma das entidades do Paciente e acordadas pelo médico. O Paciente poderá estar associado a 0 ou mais entidades e a 0 ou mais Consultas.

5.2. Modelo de implementação:

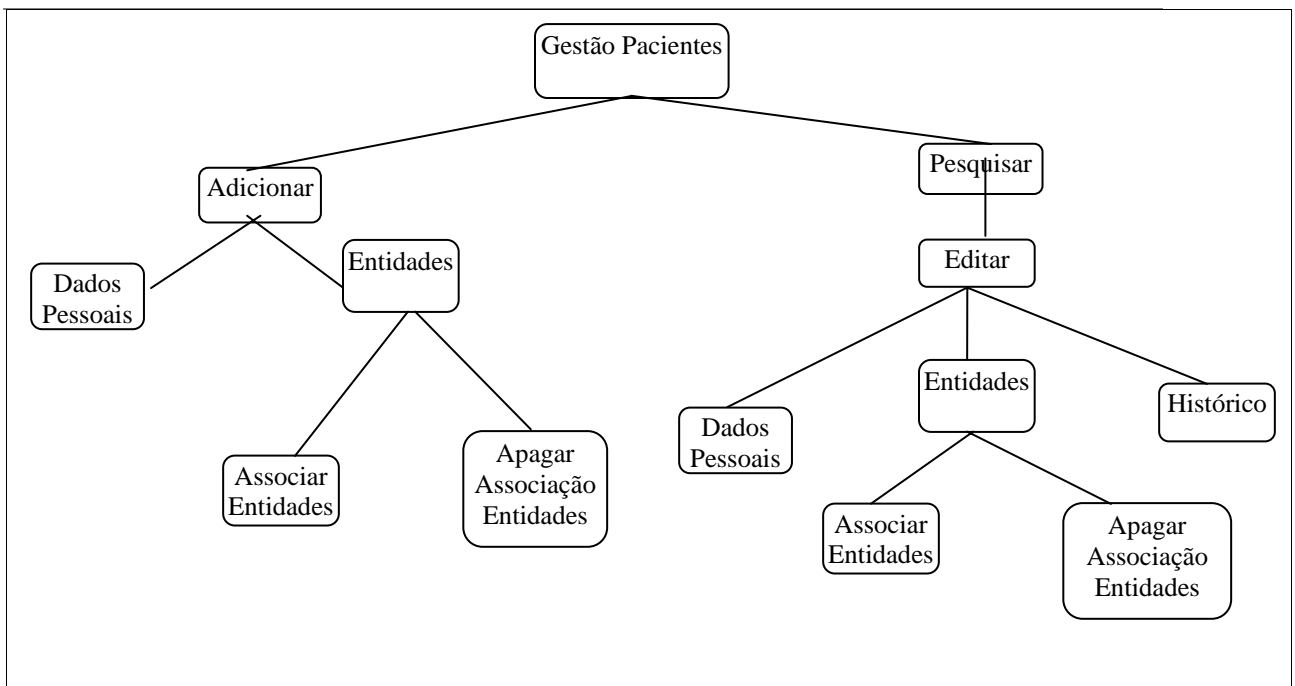


6. Modelo funcional

O modelo funcional irá mostrar-nos a navegabilidade da aplicação.



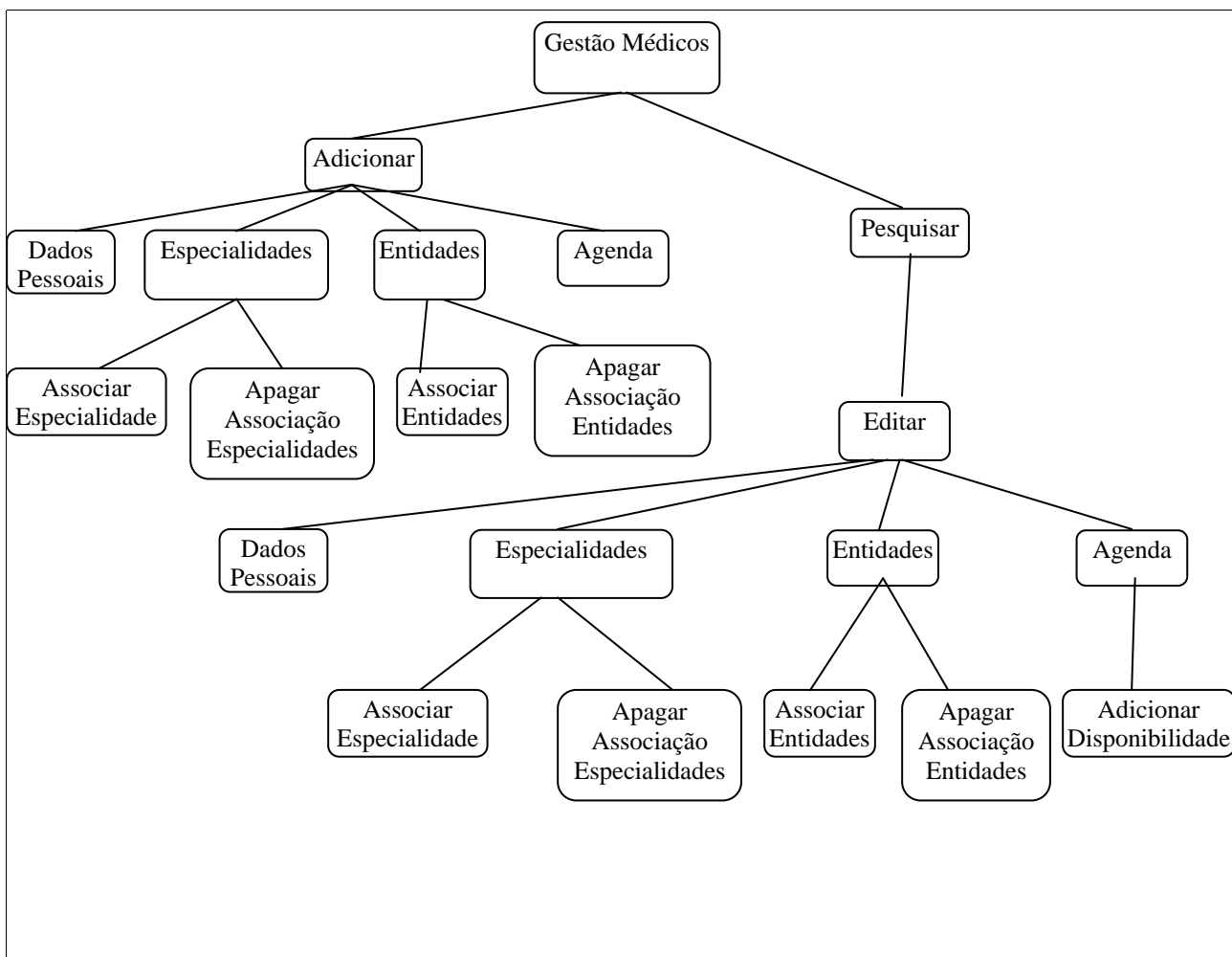
Ao aceder à aplicação, o utilizador irá ter uma página inicial onde se encontram quatro tabs que se referem aos quatro grandes grupos de gestão presente na aplicação. A tab Gestão Pacientes está seleccionada por defeito.



Se o utilizador escolher a tab Gestão de Pacientes, encontra uma pequena caixa onde poderá fazer a pesquisa do paciente ou adicionar um novo paciente.

Se pretender adicionar um paciente, após gravar os seus dados pessoais, o utilizador, poderá associar logo as entidades.

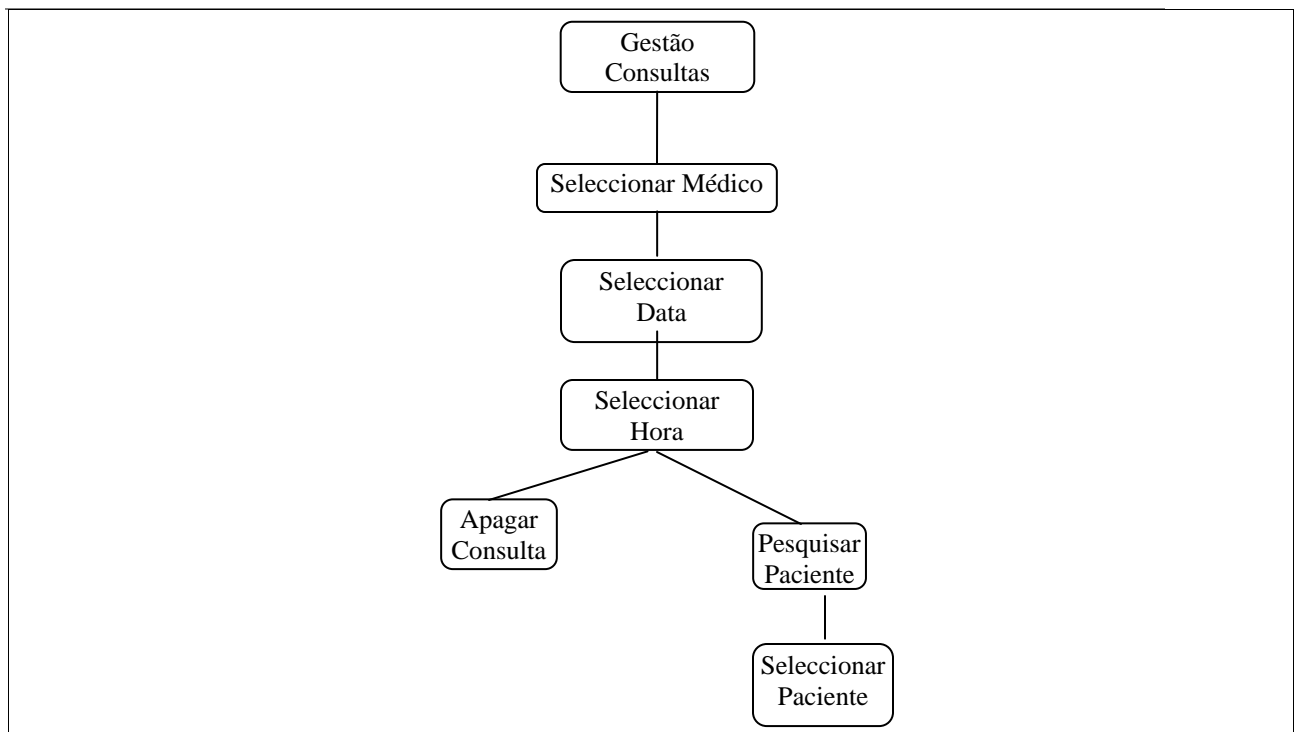
Se pretender pesquisar só terá de preencher um dos campos de pesquisa. Se não preencher nenhum campo, será mostrada a listagem de todos os pacientes. Depois de pesquisar e seleccionar um paciente o utilizador terá a possibilidade de editar os dados pessoais, associar ou desassociar uma entidade ou ter acesso a todo o histórico.



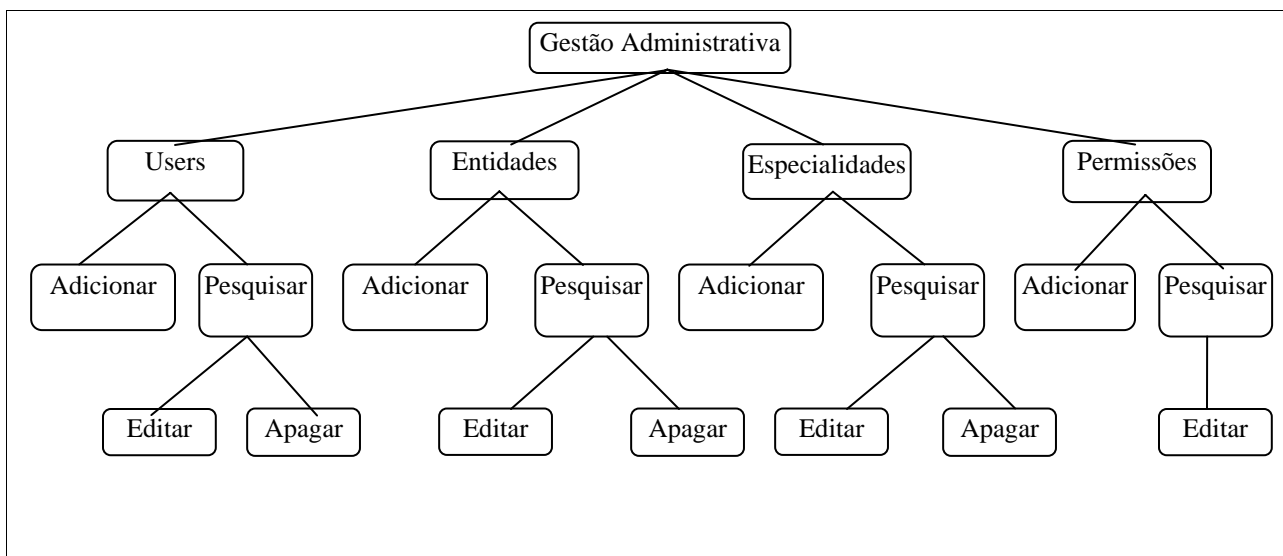
Se o utilizador escolher a tab Gestão de Médicos encontra uma pequena caixa onde poderá fazer a pesquisa do médico ou adicionar um novo.

Se pretender adicionar um médico, após gravar os seus dados pessoais, o utilizador, poderá associar logo as entidades ou especialidades e adicionar a disponibilidade na agenda.

Se pretender pesquisar só terá de preencher um dos campos de pesquisa. Se não preencher nenhum campo, será mostrada a listagem de todos médicos. Depois de pesquisar e seleccionar um médico o utilizador terá a possibilidade de editar os dados pessoais, associar ou desassociar uma entidade ou especialidade e ver ou adicionar a disponibilidade na agenda.



Se o utilizador escolher a tab Gestão de Consultas, terá de seleccionar o médico pretendido e depois irá ter uma listagem dos dias de disponibilidade do mesmo. Escolhe o dia, depois escolhe uma hora vaga e pesquisa o paciente para adicionar à consulta.



Na tab Gestão Administrativa, o utilizador terá a opção de adicionar, pesquisar, editar ou apagar Users, Entidades, Especialidades e Grupos de Permissões.

7. Conclusão

Este produto pretende ser uma ferramenta útil para qualquer tipo de Clínica, facilitando bastante o trabalho dos profissionais de saúde. A qualquer momento e em qualquer sítio, desde que haja uma ligação à Internet, o médico tem acesso a todo um historial de um paciente, bem como a sua agenda. Futuramente, esta aplicação irá ter um módulo de Gestão Financeira, ficando assim completa.

Existem ainda bastantes melhoramentos a nível estético que poderão ser feitos à aplicação, no entanto para os objectivos da cadeira de Trabalho Final de Curso e da própria aplicação não tem grande relevância.

O único ponto que estava projectado para esta aplicação e que não foi concluído diz respeito às permissões. Obviamente que esta aplicação não poderá entrar em produção sem as permissões implementadas.

O importante é o trabalho desenvolvido antes e durante o desenvolvimento da aplicação e esse trabalho preencheu claramente os objectivos pretendidos para o funcionamento da mesma. Desde a análise funcional descrita em diagramas que se baseou nas funcionalidades pretendidas pelo cliente, passando pelo planeamento do desenho até à implementação da aplicação, vai um caminho que tem de ser percorrido para se atingir um produto final robusto, que satisfaz todos os requisitos do cliente.

O papel de cliente foi desempenhado pelos colaboradores de uma clínica médica que indicou em várias reuniões todas as funcionalidades que a aplicação deveria atingir.

Todo este processo de engenharia de software foi bastante importante na compreensão de uma possível situação do mundo do trabalho. Outro aspecto muito positivo a nível pessoal foi o da implementação de muitas das matérias leccionadas ao longo da licenciatura, entre muitas outras: bases de dados, programação em objectos (em Java), etc. Outro aspecto foi o de poder aprender a utilizar novas tecnologias como o caso das Servlets e o uso de base de dados MySQL, visto que são tecnologias que vingam no mundo do trabalho.

As tecnologias usadas são bastante adequadas, visto que o Java em conjunto com a base de dados MySQL, dotam a aplicação de bastante robustez, flexibilidade e segurança. Outro aspecto importante é o de serem tecnologias “livres”.

7. Bibliografia

SUN “*The Java Tutorials*”, acedido em: <http://java.sun.com/docs/books/tutorial/>

Mysql, acedido em: <http://www.mysql.com/>

Mootools, “*the compact javascript framework*”, acedido em: <http://mootools.net/>

The Yahoo! “*User Interface Library (YUI)*”, Acedido em:
<http://developer.yahoo.com/yui/>

HIBERNATE “*Relational Persistence for Java and .NET*”, acedido em:
<http://www.hibernate.org/>

W3 Schools, acedido em: <http://www.w3schools.com/>

Anexos