




## Licenciatura em Eng.<sup>a</sup> Informática

### Projecto Final de Curso – 3º Ano – 2º Semestre

### Relatório de Projecto



Clínica  
São João de Deus  
São João de Deus | Grupo de Saúde

Cuidar Melhor

Manual

Log Out admin

**Menu**

- [Insira um Helpdesk](#)
- [Ver pedidos submetidos](#)
- [Ver todos os pedidos](#)
- [Ver contas](#)
- [Adicionar Utilizador](#)
- [Custos](#)
- [Inserir campos](#)
- [Dados da Conta](#)

Nome do Utilizador:

Email:

Localização:

Telefone/Extensão:

Departamento:

**Informação do problema**

Categoria:

Título do problema

Descrição:

Insira uma imagem:(Opcional)

**Engenharia Informática**  
**3º Ano - 2º Semestre**

Sérgio Marques Nº 20071633



## Índice

Índice .....	3
Resumo .....	4
Abstract.....	5
Agradecimentos .....	6
Índice das Figuras .....	7
Introdução .....	8
Nome da aplicação.....	8
Descrição do Projecto .....	8
Tecnologias utilizadas .....	9
Método.....	12
Abordagem Metodológica .....	12
Fluxograma.....	13
Casos de Uso .....	14
Modelo de dados.....	15
Diagrama de Sequência .....	17
Resultados.....	18
Conclusão .....	24
Bibliografia.....	25

## Resumo

O Trabalho Final de Curso tem como objectivo a implementação de uma aplicação para a resolução de Trouble Tickets na Clínica São João de Deus. O desafio consiste na utilização de uma aplicação sem custos existente no mercado. Apesar dos sistemas de trouble ticket estarem amplamente difundidos e estudados, nenhum satisfaz todos os requisitos pedidos pelo cliente. Dessa forma, construí uma aplicação de raiz, com a possibilidade de alteração da aplicação graficamente, podendo cada empresa adaptar esta aplicação às suas necessidades.

Uma plataforma de pedidos é uma ferramenta fundamental para empresas, permitindo a resolução de problemas submetidos pelos utilizadores, fomentando uma proximidade entre o utilizador e o técnico de uma forma organizada e estruturada. O facto da aplicação ser online, implica que não se instale nada nos computadores dos utilizadores, sendo apenas necessário que estes tenham um browser.

A escolha da linguagem para a aplicação em causa foi o PHP. O PHP é uma linguagem de programação de computadores interpretada, livre e muito utilizada para gerar conteúdo dinâmico na World Wide Web, e como cada vez mais se utiliza aplicações via Web, penso que faz sentido utilizar uma linguagem deste tipo.

Apenas abordámos o PHP numa cadeira ao longo do curso (Sistemas Embebidos) e de forma superficial, uma vez que o programa da cadeira estava mais focado em JAVA (Applets). Assim sendo, este Projecto Final de Curso é também um desafio pessoal, porque pretendo aprender esta linguagem que julgo ser bastante útil no que diz respeito a plataformas online. Para poder ter bases para este projecto utilizei o livro *Head First PHP & MySQL*, que me ajudou bastante quer no código PHP, quer na instalação, manutenção da base de dados Mysql como também nas queries á mesma.

## Abstract

My final report aims to implement an application for the resolution of Trouble Tickets in Clínica São João de Deus. The challenge is to use an application on the market without costs. Despite all the trouble ticket system that have been widely circulated and studied, none meets all the requirements requested by the client. So I build an application from scratch, with the possibility of changing the application graphically, meaning that each company can adapt this application for your needs.

A requests platform is an essential tool for companies, allowing problems resolution submitted by users and leveraging a better relationship between the user and the technical in an organized and structured way. Since the application is online, there is no need to install anything on users' computers; users just should have a browser.

The language that has been chosen for this application was PHP. PHP is a computers programming language, interpreted, *free* and widely used to generate dynamic content on the World Wide Web, and since we use more often web applications, I think makes sense to use this language in this application.

We only talked about PHP in one class through the all course (Embedded Systems) and superficially because the program was specially focused in JAVA (Applets). Therefore, this final project is also a personal challenge, because I want to learn this language that I think is very useful to apply in online platforms. In order to have basis for this project I used the book Head First PHP & MySQL, which helped me a lot either in the PHP code, or the installation, maintenance of the database Mysql and also the queries to the data base.

## **Agradecimentos**

Gostaria de agradecer à minha família por me apoiar ao longo de todo este trabalho, em todos os momentos que precisei, a todos os níveis.

Gostaria também de agradecer à Universidade Lusófona que me ajudou a cumprir um desejo antigo, tirar uma licenciatura em Engenharia Informática.

Por último quero agradecer aos meus colegas de curso que também me ajudaram em muitas ocasiões.

A todos, muito obrigado.

## Índice das Figuras

Figura 1 – Informação do PHP .....	9
Figura 2 – Informação Mysql .....	10
Figura 3 – Informação do Apache .....	11
Figura 4 – Modelo em Cascata .....	12
Figura 5 – Fluxograma .....	13
Figura 6 – Casos de uso .....	14
Figura 8 – Diagrama de Sequência.....	17
Figura 9 – Criar conta.....	18
Figura 10 – Fazer o login.....	18
Figura 11 – Submeter pedido .....	19
Figura 12 – Ver estado do pedido.....	20
Figura 13 – Ver resumo do Pedido.....	20
Figura 14 – Alterar password .....	20
Figura 15 – Resolver pedidos .....	21
Figura 16 – Editar pedidos .....	21
Figura 17 – Adicionar/Remover campos.....	22
Figura 18 – Ver custos dos técnicos .....	22
Figura 19 – Editar/Apagar contas dos utilizadores.....	23

## Introdução

### Nome da aplicação

O nome dado à aplicação do projecto foi Resolve. É um nome simples e elucidativo da função para a qual foi construída, resolver pedidos submetidos pelos utilizadores.

### Descrição do Projecto

Este Projecto Final de Curso consiste numa solução *open source* que permite a submissão de pedidos por parte dos utilizadores. Depois de analisar várias aplicações que existem no mercado, tais como o SugarCRM ou OTRS, verifiquei que nenhuma delas se adequava aos requisitos que foram pedidos pelo cliente. Por essa razão, construí uma aplicação que respondesse às necessidades do cliente.

Uma plataforma de pedidos consiste numa aplicação que recebe pedidos dos utilizadores, encaminha-os para os respectivos técnicos e permite que estes resolvam os mesmos.

Segundo o cliente, a aplicação tem de ter três categorias de pedidos (HelpDesk, Económico e Manutenção), que são diferenciadas quando o utilizador escolhe o tipo de pedido. A ideia é que cada pedido seja visto apenas pelos técnicos que estão atribuídos à resolução da categoria do pedido. Acho que é interessante o envio de mail quando os pedidos são criados, quer para quem o criou, quer para quem vai resolver. Na empresa onde trabalho, temos uma aplicação semelhante que não envia mail e faz-nos falta para sermos notificados sempre que um pedido é submetido. Também seria útil quando o fecho do pedido, o utilizador receber mail.

Outra situação é a possibilidade de anexar imagens aos pedidos, uma vez que muitos utilizadores onde eu trabalho, fazem pedidos sobre um “alegado” erro que lhes apareceu e quando eu vou averiguar o erro já lá não está e torna-se complicado poder perceber o problema submetido pelo utilizador.



## Tecnologias utilizadas

Para o desenvolvimento desta solução, optei pela utilização da linguagem PHP 5.3.1 (<http://www.php.net/>). O PHP é uma linguagem orientada a objectos, sendo a ideal para uma aplicação com as características pedidas, também indicada para conteúdo dinâmico na World Wide Web. Segue em baixo algumas imagens do resumo criado pela função `phpinfo()` que podemos recorrer para ver o que temos instalado no servidor que contém a aplicação:


<div> <div>PHP Version 5.3.1</div>  </div>	
System	Windows NT C1364 5.1 build 2600 (Windows XP Professional Service Pack 3) i586
Build Date	Nov 20 2009 17:20:57
Compiler	MSVC6 (Visual C++ 6.0)
Architecture	x86
Configure Command	cscrip /nologo configure.js "--enable-snapshot-build"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626,TS,VC6
PHP Extension Build	API20090626,TS,VC6
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
Registered PHP Streams	https, ftps, php, file, glob, data, http, ftp, compress.zlib, compress.bzip2, phar, zip
Registered Stream Socket Transports	tcp, udp, ssl, sslv3, sslv2, tls
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*, bzip2.*

Figura 1 – Informação do PHP

Em complementaridade usei a base de dados Mysql 5.1 (<http://www.mysql.com/>), que nos permite guardar todos os dados da aplicação. Desta forma podemos mais tarde consultar a informação dos pedidos de uma forma rápida, estruturada e segura. A base de dados Mysql é a base de dados free mais utilizada no mundo, estando mesmo ao nível de muitas bases de dados pagas tais como a Microsoft SQL Server. As principais características do Mysql são:

- 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++, C#, Visual Basic, Python, Perl, PHP, ASP e Ruby)
- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de hardware;
- Facilidade de uso;
- É um Software Livre com base na GPL;
- Replicação facilmente configurável;
- Interfaces gráficas (MySQL Toolkit) de fácil utilização cedidos pela MySQL

### mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	5.1.41

Directive	Local Value	Master Value
mysql.allow_local_infile	On	On
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	<i>no value</i>	<i>no value</i>
mysql.default_password	<i>no value</i>	<i>no value</i>
mysql.default_port	3306	3306
mysql.default_socket	MySQL	MySQL
mysql.default_user	<i>no value</i>	<i>no value</i>
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off

Figura 2 – Informação Mysql

Como servidor HTTP usei o Apache (<http://www.apache.org/>). Existem outras opções no mercado tais como o IIS, mas uma vez que este projecto consiste na aplicação de soluções *open source*, fiquei limitado na escolha. No entanto, o Apache é dos melhores servidores de HTTP (54,68% dos sites mundiais usam Apache), sendo na minha opinião bem melhor do que o IIS. O Apache permite-nos uma configuração mais avançada e corre em qualquer sistema operativo ao contrário do IIS que só corre em ambientes Windows e está mais limitado nas configurações.

### apache

Apache Version	Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l mod_autoindex_color PHP/5.3.1 mod_apreq2-20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
Apache API Version	20051115
Server Administrator	postmaster@localhost
Hostname:Port	localhost:80
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	No
Server Root	C:/xampp/apache
Loaded Modules	core mod_win32 mpm_winnt http_core mod_so mod_actions mod_alias mod_asis mod_auth_basic mod_auth_digest mod_authn_default mod_authn_file mod_authz_default mod_authz_groupfile mod_authz_host mod_authz_user mod_cgi mod_dav mod_dav_fs mod_dav_lock mod_dir mod_env mod_headers mod_include mod_info mod_isapi mod_log_config mod_mime mod_negotiation mod_rewrite mod_setenvif mod_ssl mod_status mod_autoindex_color mod_php5 mod_perl mod_apreq2

Figura 3 – Informação do Apache

De salientar ainda que todas estas aplicações foram instaladas através de um único pacote chamado XAMPP (<http://www.apachefriends.org/en/xampp.html>). Por ultimo, o trabalho foi desenvolvido com a ferramenta Netbeans 6.9.1 e no que diz respeito á linguagem HTML, usada juntamente com o PHP, usei uma aplicação online chamada de **w3schools.com** em <http://www.w3schools.com/>

## Método

### Abordagem Metodológica

Foram definidas várias etapas, para o desenvolvimento desta aplicação de modo a garantir uma construção estruturada e coerente. Por isso decidi adoptar por um modelo de desenvolvimento que me obrigasse a cumprir um conjunto de tarefas sequenciais no tempo. O modelo usado para esta aplicação foi o modelo em cascata, como podemos verificar na figura em baixo.

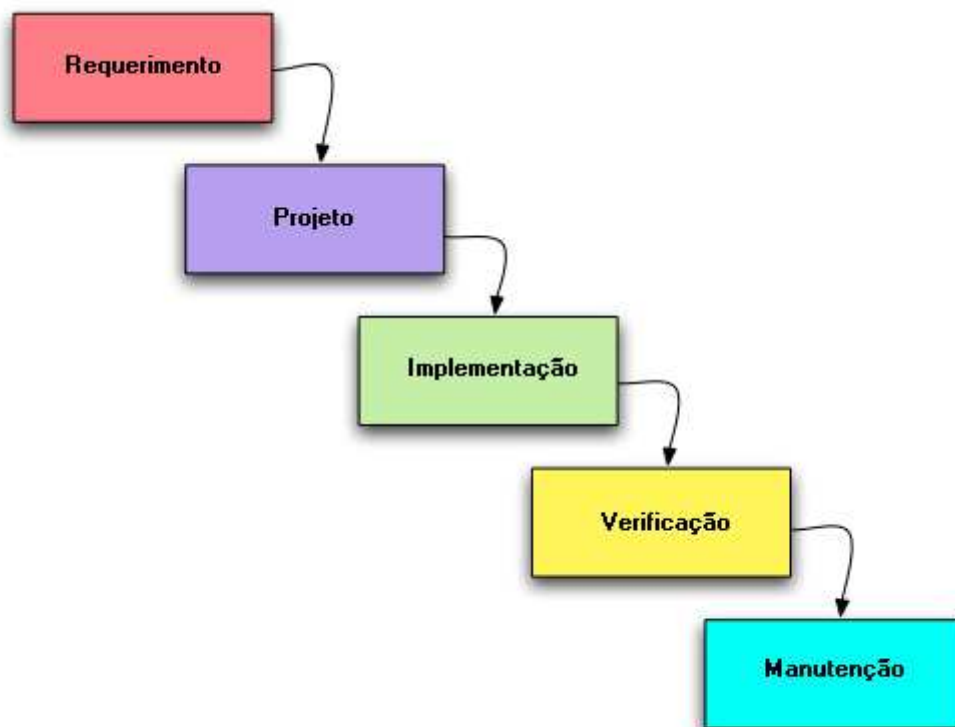


Figura 4 – Modelo em Cascata

Este modelo segue uma abordagem ‘top-down’, e tem como principal característica a sequência de actividades estabelecendo uma ordem, onde cada fase “cai” em cascata para a seguinte apenas se estiver terminada. Todas as actividades identificadas nas fases do modelo são fundamentais e estão na ordem certa

## Fluxograma

Um fluxograma é a sequência de passos necessários para que se possa atingir a automação de processos da aplicação, de acordo com um conjunto de regras definidas. Neste caso, após reunião com o Cliente, este definiu o seguinte fluxograma:

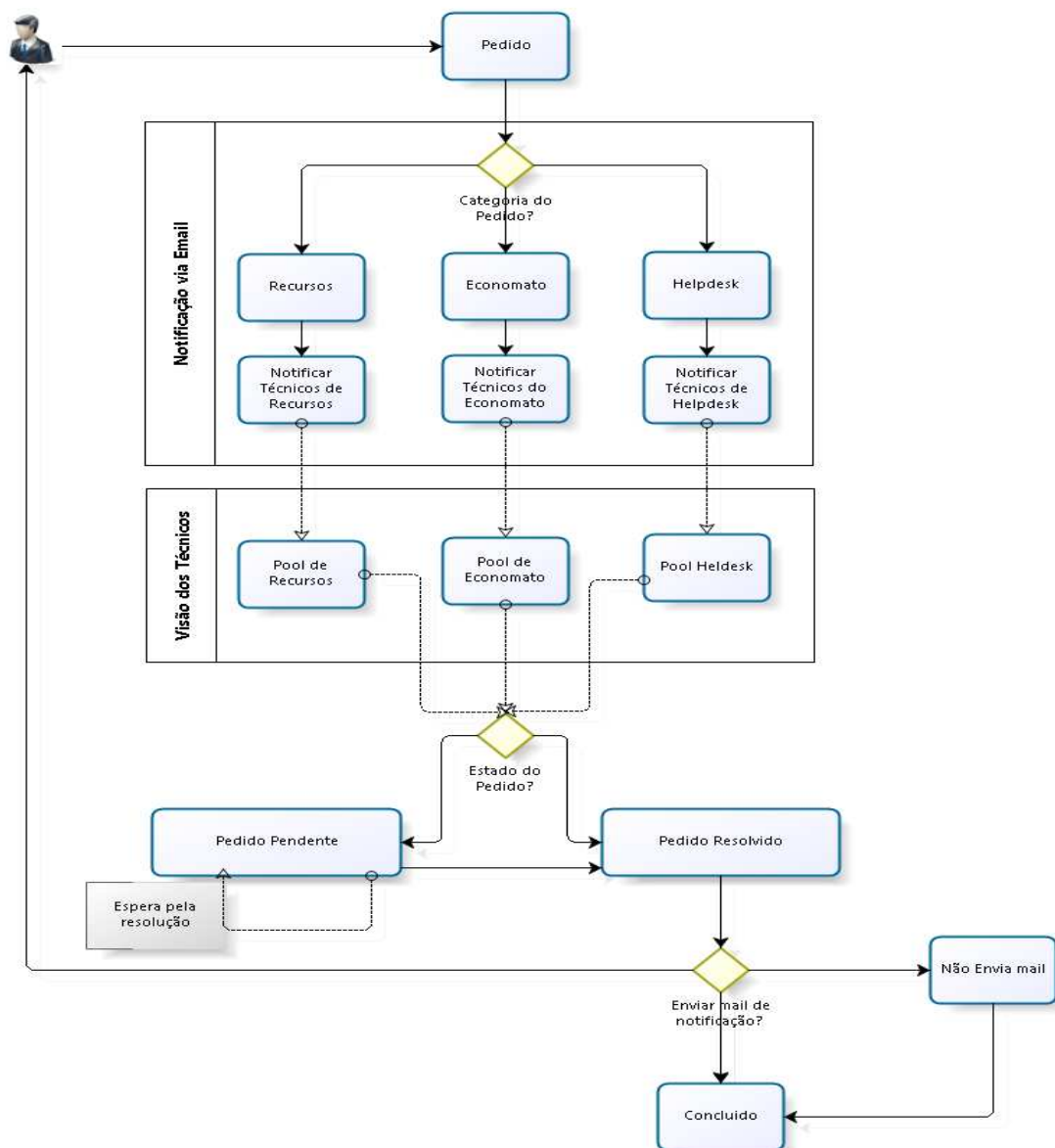


Figura 5 – Fluxograma

Como podemos observar, o utilizador submete um pedido que pode ter três categorias: HelpDesk, Economato e Manutenção. Após a submissão do pedido, todos os técnicos

que têm como função a categoria do pedido, vão ser notificados via correio electrónico. Depois cada técnico que entre na sua pool, só vai ver os pedidos da sua categoria, ou seja, um técnico de HelpDesk apenas vê os pedidos submetidos com a categoria HelpDesk. De seguida os técnicos resolvem ou não os pedidos, podendo estes ficar em estado pendente até serem resolvidos. Quando se coloca o estado “Pendente” ou “Resolvido”, os técnicos têm a opção de enviar ou não uma notificação ao utilizador.

## Casos de Uso

Um caso de uso é “uma sequência de acções que um ou mais actores realizam num sistema de modo a obterem um resultado particular”. O modelo de casos de uso permite capturar os requisitos de um sistema através do detalhe de todos os cenários que os utilizadores podem realizar. Os casos de uso, mais que iniciar a modelação de requisitos de um sistema, dirigem/conduzem todo o processo de desenvolvimento. Na figura em baixo está o modelo de casos de uso da aplicação desenvolvida.

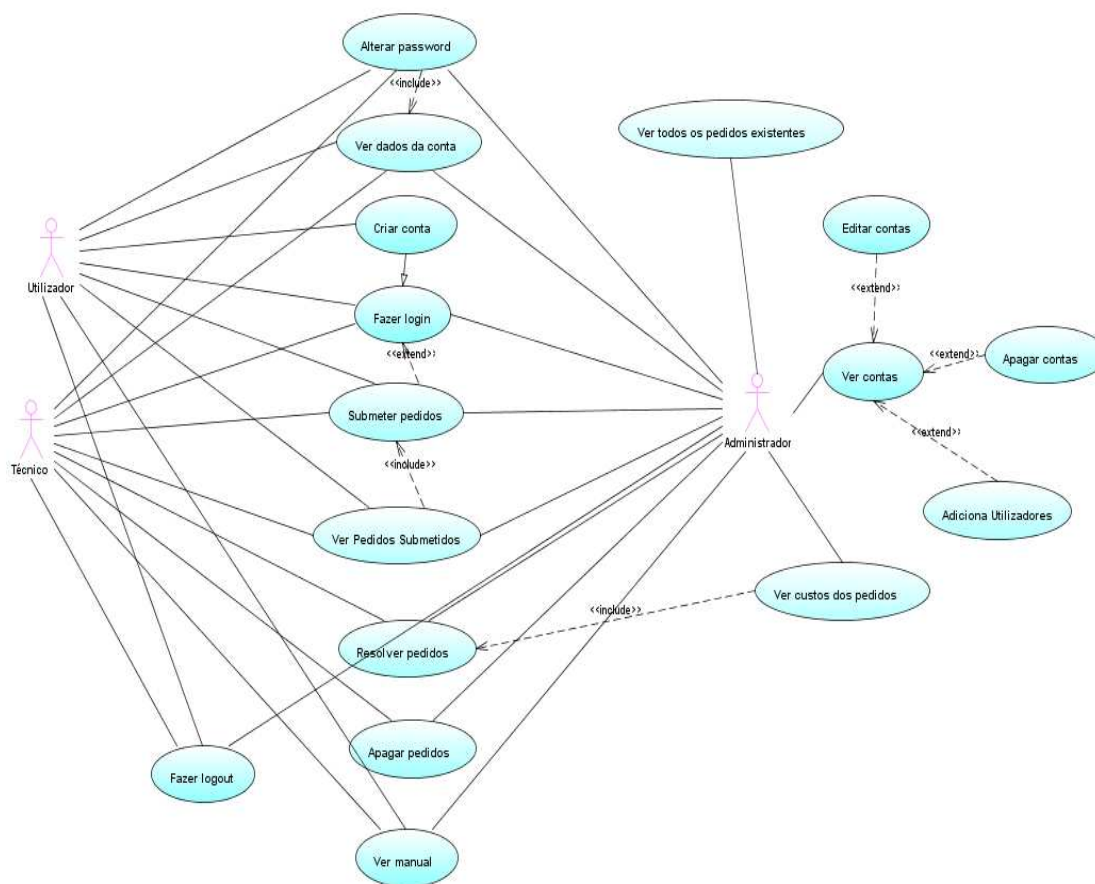


Figura 6 – Casos de uso

Como podemos verificar, os actores desta aplicação vão ter várias tarefas em comum. Os utilizadores vão ter as mesmas tarefas que os técnicos excepto na resolução e remoção dos pedidos. Outra diferença entre estes actores é que os utilizadores podem criar uma conta para utilizar a aplicação mas os técnicos não criam contas, é-lhes criada uma pelo administrador com privilégios de resolução de pedidos. Os administradores têm mais funções, permitindo-lhes fazer toda a gestão da aplicação.

## Modelo de dados

Depois do levantamento e análise de requisitos, percebeu-se quais as necessidades existentes permitindo desenvolver e estruturar a base de dados. Segue em baixo a figura com o modelo de dados da aplicação

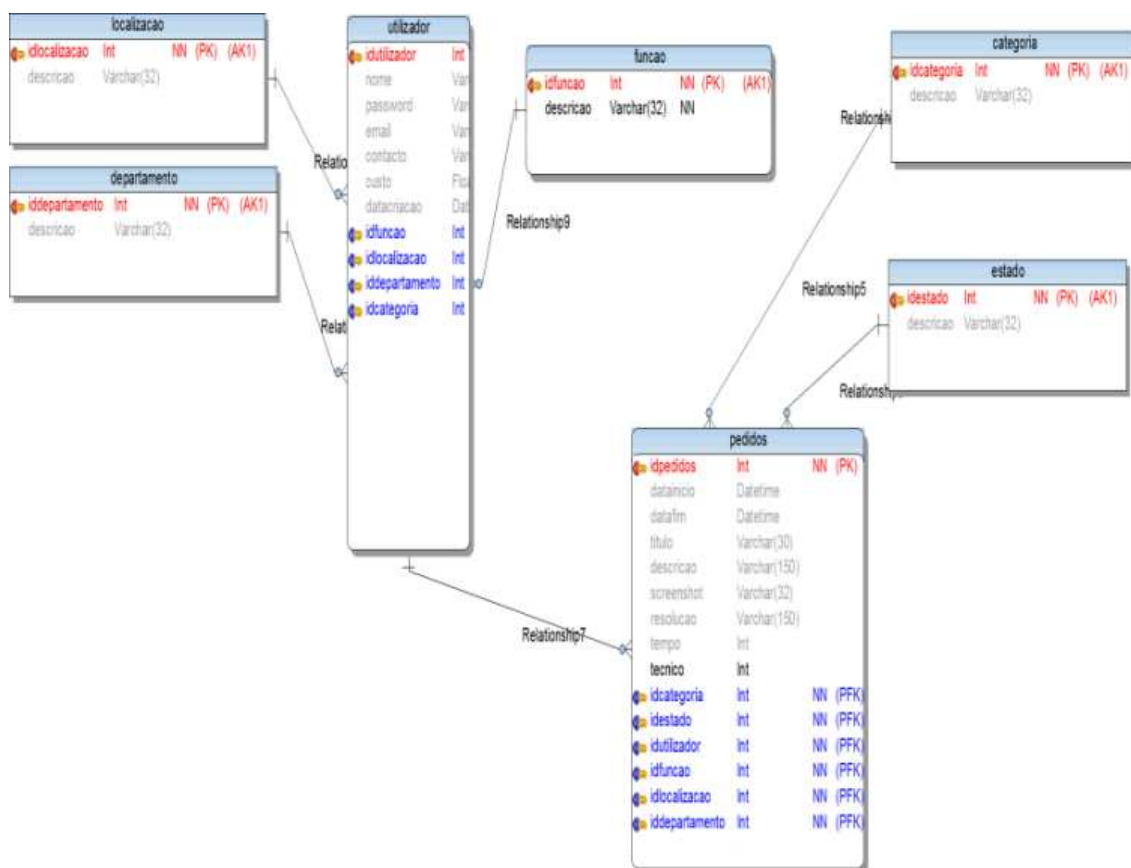


Figura 7 – Modelo de Dados

- Tabela Localização – Esta tabela contém a descrição dos locais que existem na empresa. Cada utilizador está associado apenas a uma localização e uma localização pode ter associado vários utilizadores, logo, a tabela Localização tem uma cardinalidade com a tabela de utilizadores de 1 para N (1..N).
- Tabela Departamento – Esta tabela contém a descrição dos departamentos que existem na empresa. Cada utilizador está associado apenas a um departamento e um departamento pode conter vários utilizadores. A tabela Departamento tem uma cardinalidade com a tabela de utilizadores de 1 para N (1..N).
- Tabela Função – Esta contém a descrição das funções dos funcionários na empresa. Cada funcionário pode ter apenas uma função e uma função pode ser associada a vários funcionários. Esta tabela tem uma cardinalidade com a tabela de utilizadores de 1 para N (1..N).
- Tabela categoria – Esta tabela contém a descrição da categoria dos pedidos. Cada pedido está associado a uma categoria e uma categoria pode estar associado a vários pedidos, logo esta tabela tem uma cardinalidade com a tabela pedidos de 1 para N (1..N).
- Tabela Estado – Esta tabela contém os estados dos pedidos. Cada pedido está associado um estado e um estado pode estar associado a vários pedidos, logo esta tabela tem uma cardinalidade com a tabela pedidos de 1 para N (1..N).
- Tabela Pedidos – Esta tabela contém toda a informação dos pedidos e está relacionada com a Tabela Utilizador, podendo cada pedido ser submetido apenas por um utilizador mas um utilizador pode submeter vários pedidos. A tabela Pedidos tem uma cardinalidade com a tabela utilizador de N para 1 (N..1).
- Tabela Utilizador – Esta tabela contém toda a informação dos utilizadores e está relacionado com a tabela Pedidos. Um utilizador pode submeter vários pedidos enquanto que um pedido só pode ser submetido apenas por um utilizador. A tabela Utilizador tem uma cardinalidade com a tabela Pedidos de 1 para N (1..N).



## Diagrama de Sequência

O diagrama de sequência representado na figura em baixo mostra a interacção quando um pedido é submetido por parte do utilizador.

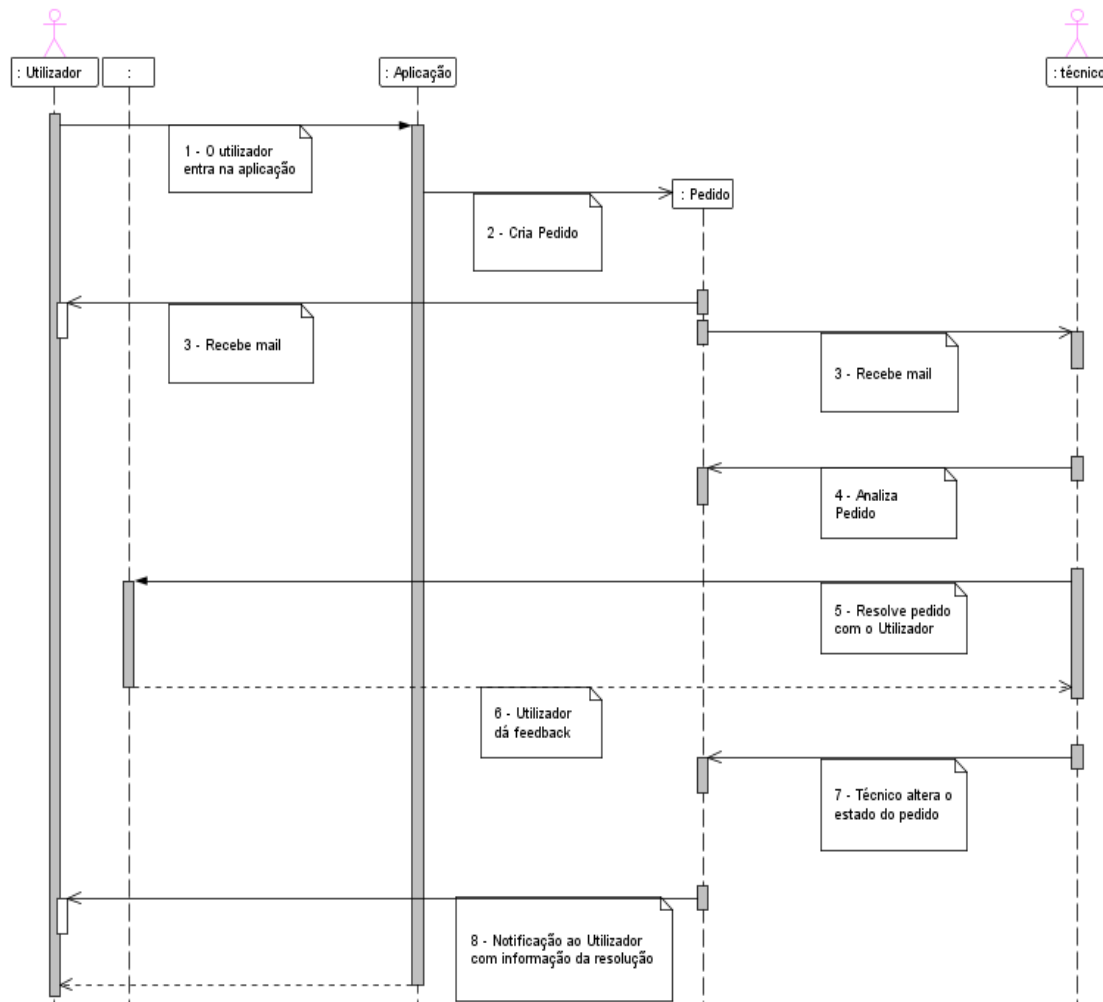


Figura 8 – Diagrama de Sequência

1. Utilizador faz login e entra na aplicação
2. Utilizador cria um pedido
3. É enviada notificação ao utilizador e técnicos que correspondam à categoria do pedido submetido
4. O técnico vai ver os detalhes do pedido para definir uma estratégia de resolução

5. O técnico vai entrar em contacto com o utilizador para poder perceber melhor o pedido submetido ou para confirmar o fecho do mesmo junto do utilizador
6. O utilizador confirma o fecho do pedido
7. O técnico altera o estado do pedido, para concluído
8. O utilizador recebe mail com a indicação de que o pedido foi fechado

## Resultados

O funcionamento desta aplicação está de acordo com o que foi requisitado pelo cliente. Toda a arquitectura da aplicação é *open source*, sem custos e bastante pratica de usar, bastando apenas colocar o nome definido pelo cliente no browser. O utilizador tem de entrar no Resolve, criar uma conta e de seguida fazer o login.

Clínica São João de Deus  
São João de Deus | Grupo de Saúde

Cuidar Melhor

[Login](#) [Criar conta](#) [Manuais](#)

**Menu**  
[Faça o login](#)

Nome: teste  
Email: teste@localhost  
Localização: Edifício 2  
Contacto: 1567  
Departamento: Executivo  
Password: \*  
Confirme a Password: \*

[Registar](#)

Para mais informações, é favor contactar o Departamento Informático através do seguinte nº 21856462 ou enviar mail para [informatica@localhost](mailto:informatica@localhost)

Figura 9 – Criar conta

Clínica São João de Deus  
São João de Deus | Grupo de Saúde

Cuidar Melhor

[Login](#) [Criar conta](#) [Manuais](#)

**Menu**  
[Faça o login](#)

Nome: teste  
Password: \*

[Log in](#)

Para mais informações, é favor contactar o Departamento Informático através do seguinte nº 21856462 ou enviar mail para [informatica@localhost](mailto:informatica@localhost)

Figura 10 – Fazer o login

Após o login, os Top Menus que tinha antes de se autenticar na aplicação, vão desaparecer e o menu que está no lado esquerdo vai ser populado com as funções consoante as contas (Administrador e Técnicos vão ter mais opções no menu)

De seguida o utilizador pode submeter um pedido ou consultar algum, se tiver já anteriormente submetidos pedidos.

The screenshot shows the 'Cuidar Melhor' web application interface. At the top left is the logo for 'Clínica São João de Deus' with the tagline 'São João de Deus | Grupo de Saúde'. At the top right is the text 'Cuidar Melhor' and a 'Log Out teste' link. Below the header is a green navigation bar with a 'Menu' icon and a 'Mamal' label. The main content area is divided into two columns. The left column contains a 'Menu' section with links: 'Insira um Helpdesk', 'Ver pedidos submetidos', and 'Dados da Conta'. The right column contains a form for submitting a request. The form fields are: 'Nome do Utilizador:' (teste), 'Email:' (teste@localhost), 'Localização:' (Edifício 2), 'Telefone/Extensão:' (1567), 'Departamento:' (Executivo), 'Categoria:' (Economato), 'Título do problema' (Papel), and 'Descrição:' (Preciso de duas resmas de papel A3). There is also an optional field for 'Insira uma imagem:' with a 'Browse...' button. At the bottom of the form is a 'Criar Helpdesk' button. A footer bar at the bottom contains the text: 'Para mais informações, é favor contactar o Departamento Informático através do seguinte nº 21856462 ou enviar mail para informatica@localhost'.

Figura 11 – Submeter pedido

Após a submissão de pedido aparece uma mensagem a dizer que foi submetido e de seguida o utilizador vai receber um mail com um resumo do pedido

Quando se submete o pedido, também temos a opção de colocar uma imagem sobre o problema que estamos a reportar. Esta situação é opcional quando se cria o pedido. O que não é opcional é o não preenchimento dos campos 'Título' e 'Descrição'. Se o utilizador não preencher os dois campos referidos, a aplicação mostra uma notificação de erro a alertar o utilizador sobre esse facto.

Após ter então enviado um pedido, o utilizador pode ver o resumo do mesmo. Ver o resumo do pedido é útil porque os técnicos podem não ter fechado o pedido (estado Pendente) e colocarem informação no mesmo.



Figura 12 – Ver estado do pedido



Figura 13 – Ver resumo do Pedido

Por ultimo, o utilizador também pode ver os dados da sua conta e alterar a password.



Figura 14 – Alterar password

Os técnicos têm todas as opções anteriores mais a gestão dos pedidos, podendo resolver e eliminá-los. Como se pode observar, após o login, o menu ficou com mais um opção, a opção ‘Ver Pedidos do Utilizadores’.



Figura 15 – Resolver pedidos



Figura 16 – Editar pedidos

O Administrador tem como opções todas as anteriormente mostradas, mais a função de criar contas, departamentos, funções, locais, categorias e estados dos pedidos e ainda ver os custos das resoluções dos pedidos (o valor custo é atribuído nas contas dos técnicos quando estes são criados pelo Administrador).

Para mais informações, é favor contactar o Departamento Informático através do seguinte nº 21856462 ou enviar mail para informatica@localhost

Figura 17 – Adicionar/Remover campos

Tecnico: admin Ver custos

Id pedido	Título	Data Inicio	Data Fim	Descrição	Resolução	Tempo(min)	Euros	Custo(hora)	Total
45	teclado	2010-10-11 22:48:42	2010-10-11 22:51:32	não funciona	Por resolver	11	5	€0,92	
44	Rato	2010-10-11 22:45:52	2010-10-11 23:21:07	Está a funcionar mal	Por resolver	120	5	€10,00	€10,92

Figura 18 – Ver custos dos técnicos


**Clínica São João de Deus**  
 São João de Deus | Grupo da Saúde

**Cuidar Melhor**  
[Manual](#) [Log Out admin](#)

**Menu**

- [Insira um Helpdesk](#)
- [Ver pedidos submetidos](#)
- [Ver todos os pedidos](#)
- [Ver contas](#)
- [Adicionar Utilizador](#)
- [Contas](#)
- [Insere campos](#)
- [Dados da Conta](#)

**Gerir contas**

Nome	Email	Departamento	Localização	Contacto	Função	Custo	Opções	
admin	admin@localhost	Administrativo	Edifício 2	1537	Administrador	5	<a href="#">Editar</a>	<a href="#">Apagar</a>
rs	rs@localhost	Informatica	Edifício 1	2358	Economato	5	<a href="#">Editar</a>	<a href="#">Apagar</a>
sfin	sfin@localhost	Financeiro	Edifício 2	9512	Helpdesk	5	<a href="#">Editar</a>	<a href="#">Apagar</a>
sonia	sonia@localhost	Informatica	Edifício 2	123	Utilizador	0	<a href="#">Editar</a>	<a href="#">Apagar</a>
teste	teste@localhost	Executivo	Edifício 2	1567	Utilizador	0	<a href="#">Editar</a>	<a href="#">Apagar</a>
vf	vf@localhost	Executivo	Edifício 2	1535	Recursos	5	<a href="#">Editar</a>	<a href="#">Apagar</a>

Figura 19 – Editar/Apagar contas dos utilizadores

## Conclusão

As novas tecnologias e o desenvolvimento de software têm cada vez mais impacto na estrutura das organizações. Numa época em que se fala tanto em crise, as aplicações *free* são as que fazem mais sentido nas empresas de um ponto de vista económico. Todas as ferramentas utilizadas neste projecto são *free*, desde o IDE utilizado para desenvolver o código (Netbeans), a base de dados (Mysql), o servidor de WEB (Apache) e a linguagem (PHP). Com este projecto aprendi como se desenvolve uma aplicação desde a fase de levantamento e análise de requisitos até à implementação. Espero no futuro implementar projectos deste género na instituição onde trabalho, Câmara Municipal da Amadora.

A arquitectura do projecto é muito simples de implementar, bastando apenas instalar o XAMPP, que ficamos logo com o PHP, Mysql e Apache. De seguida temos de configurar o acesso à base de dados e populá-la. Para isso temos uma ferramenta chamada phpMyAdmin, que permite, graficamente, gerir toda a base de dados evitando assim ter de o fazer através da linha de comandos.

Todos os requisitos foram cumpridos, ou seja, a aplicação cumpre os objectivos exigidos pelo Cliente. As funcionalidades, anteriormente descritas, estão todas implementadas, e, no que diz respeito à interface, penso que é prática e adequada às necessidades dos utilizadores.



## Bibliografia

1. Lynn Beighley, Michael Morrison, O'Reilly Media *Head First PHP & MySQL*, 2008
2. How to create Web Sites - <http://www.w3schools.com/>
3. Linguagem PHP - <http://www.php.net/>
4. Servidor Apache - <http://www.apache.org/>
5. Base de dados Mysql - <http://www.mysql.com/>
6. XAMPP - <http://www.apachefriends.org/en/xampp.html>