



Licenciatura em Engenharia Informática e de Computadores

Projeto e Seminário

Sistema de partilha entre médicos de dermatologia da ficha de utente,
incluindo fotos de sinais dermatológicos

Medical Skin Care

Luís Moreira

Hélio Cavadissa

Orientadores:

Jorge Martins

Vítor Almeida

Julho, 2018

Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores

Relatório final realizado no âmbito de Projeto e Seminário.

Semestre de Verão 2017/2018

Medical Skin Care

Hélio José Cavadissa

Luís Stephanio da Conceição Moreira

Orientadores: Jorge Martins

Vítor Almeida

Resumo

Medical Skin Care é uma aplicação Web dirigida à especialidade de dermatologia que surge com objetivo de permitir a partilha de ficha médica entre médicos. Permite que as unidades hospitalares registadas na plataforma efetuem o registo de médicos, enfermeiros e utentes. Todos os utilizadores registados na plataforma têm um perfil que pode ser acedido através de um e-mail e *password*. No caso de um médico, utente ou enfermeiro, essa *password* é gerada no momento do registo e é-lhe enviada no e-mail, sendo possível alterar quando o mesmo aceder à aplicação.

As fichas médicas dos utentes são constituídas por informações detalhadas sobre a pele do utente, incluindo as avaliações efetuadas à distância de até 20 cm. As informações são obtidas através de questionários e formulários robustos sobre cada utente, as informações incluem histórico de saúde do utente e familiar. As informações recolhidas podem ser partilhadas com outros profissionais da saúde. A plataforma respeita as normas do novo Regulamento Geral de Proteção de Dados, sendo que os dados dos utentes são apenas acedidos pelos profissionais da saúde a quem os mesmos concedem permissão. O utente tem sempre acesso à informação sobre quem tem acesso aos seus dados médicos.

Palavras Chave

Medical Skin Care, dermatologia, ficha médica, dermatoscópio, saúde, pele.

Conteúdo

Glossário.....	2
1.Introdução.....	5
1.1 Enquadramento.....	6
1.2 Projetos semelhantes.....	6
1.2.1. Dermatron	6
1.2.2. OpenMRS	6
1.3 Medical Skin Care	7
1.3.1 Objetivo	8
1.4 Estrutura do relatório.....	8
2. Requisitos Funcionais	9
2.1. Utilizadores	9
2.1.1 Administrativo	9
2.1.2 Médico.....	10
2.1.3 Enfermeiro.....	12
2.1.4 Utente	12
3. Tecnologias utilizadas no projeto.....	14
3.1 Servidor	14
3.1.1 <i>Frameworks</i> para construção de API.....	14
3.2. Base de Dados	16
3.2.1 MongoDB.....	16
3.2.2. Cifra dos dados.....	16
3.3 Microsoft Azure.....	16
3.3.1 Key Vault	16
3.3.2 Storage Service	17
3.4 <i>Client-Side</i>	18
3.4.1 Captura de Foto Através dum Dermatoscópio.	18
4. Arquitetura	20
4.1 Camada de apresentação.....	21
4.2. Camada de negócio	21
4.2.1 Cifra dos Dados.....	21
4.2.2 Armazenamento das imagens.....	21
4.2.3 Registo e Autenticação.....	21
4.3 Camada de Acesso Aos Dados.....	22
4.3.1. Modelo de Dados	22

5. Notas finais.....	24
Referências.....	25
Bibliografia	26

Glossário

No decorrer do presente documento encontram-se algumas expressões utilizadas no âmbito deste projeto. Nesta seção introduzimos essas expressões e as respectivas definições de modo a facilitar a leitura do relatório.

Ficha médica.

Quando referimos à ficha médica, estamos a falar de ficha médica específica para pacientes de dermatologias, a mesma inclui, tipo de pele, pigmentação da pele, textura, temperatura, detalhes que não aparecem numa ficha médica geral.

Grupo de Fichas médica.

Um grupo de fichas, é um agrupamento de fichas médicas com base na patologia, todas as fichas médicas de um utente com a mesma patologia fazem parte do mesmo grupo de ficha.

Perfis de utilizadores.

Existem 4 tipos de perfis de utilizadores, o administrativo que representa a entidade hospitalar, o médico, o enfermeiro e o utente (que também é referido como paciente) que são todos criados por uma unidade hospitalar (administrativo). Em algumas partes deste documento o administrativo será referido como unidade hospitalar e vice e versa.

Foto de Sinais.

São fotos adquiridas através de equipamentos especiais (dermatoscópio) que suportam uma resolução específica para sinais em diferentes tons de pele.

Unidades hospitalares.

A unidade hospitalar aqui entende-se como hospitais, clínicas ou até mesmo pequenos consultórios médicos.

Registar utilizadores a uma unidade hospitalar.

Quando nos referimos em registar um utilizador, é no sentido de criar uma conta, a unidade hospitalar (administrativo), regista os médicos, enfermeiros e utentes.

Tipos de utilizadores

- Administrativo
- Médico
- Enfermeiro
- Utente

Adicionar utilizadores a uma unidade hospitalar

Quando nos referimos a adicionar, é no sentido de criar uma relação entre o utilizador e a respetiva unidade hospitalar. A unidade hospitalar (administrativo), adiciona os médicos, enfermeiros e utentes através do email que os utilizadores usaram para o registo na aplicação.

Equipa médica.

Uma equipa médico é constituída por médicos e enfermeiros duma instituição

Permissão de Acesso.

É a permissão que um médico ou enfermeiro precisa para aceder ao grupo de fichas de um utente, são enviadas pela unidade hospitalar para utente.

Role (função)

É um identificador usado para distinguir o papel de cada utilizador, na aplicação temos 4 *roles* relevantes: *administrative, doctor, nurse, patient*. Esses *roles* correspondem aos administrativos, doutores, enfermeiros e utentes respetivamente

Questionários

De modos a ter toda informação sobre o utente, fazemos recolha de dados para efeitos estatísticos, por este motivo temos um conjunto de informações que serão adicionadas às fichas do utente, nomeadamente:

- Histórico Familiar Médico
- Histórico Social
- Histórico Sexual
- Histórico Médico

1.Introdução

Este projeto tem como objetivo a implementação de um registo de saúde eletrónico (*Electronic Health Records* (EHRs)) que inclua a ficha de utente vocacionada para apoio aos médicos de dermatologia e que permita a troca de informação entre profissionais de saúde. Face às regras de segurança cada vez mais apertadas, inclusive devido ao recentemente entrado em vigor Regulamento Geral de Proteção de Dados (RGPD)^[1] e não só, a privacidade dos dados é uma área que não pode ser descurada numa aplicação onde se lida com dados pessoais dos utentes.

Na fase preparatória do projeto, de modo a se obter mais informação sobre dermatologia houve o apoio do prof. João Maia Silva, especialista na área de dermatologia. Na reunião tida com este especialista o mesmo apontou a falta de informação sobre o histórico dos utentes como um dos maiores problemas que os profissionais da saúde enfrentam. Em Portugal os médicos do Serviço Nacional de Saúde (SNS) podem aceder ao historial clínico dos utentes através do Portal^[2] do Profissional, onde é possível ter acesso às informações sobre a medicação prescrita ao doente na sua última consulta, exames, alergias, etc. Há, no entanto, uma grande necessidade de informações mais detalhadas de acordo com o ramo da saúde. Há igualmente uma grande de necessidade de acesso ao historial médico de pacientes provenientes de outras partes do mundo e que, em muitos desses países, não há protocolos como portal do SNS.

1.1 Enquadramento

A realização do projeto resulta da necessidade de desenvolvimento de Plataformas de Partilha de Dados de Saúde e o Acesso dos Cidadãos aos seus dados de saúde, assim como a consagração do princípio da portabilidade de informação de saúde no âmbito da utilização de aplicações informáticas e de outras formas de interligação de sistemas de informação com impacto na saúde, respeitando as normas^[3] exigidas pelo RGPD.

A primeira tarefa executada no âmbito deste projeto foi o levantamento do “estado da arte” em termos de aplicações do tipo EHS/EMR gerais e específicas da dermatologia.

Existem projetos *open source* que podem ser usados para criar plataformas que respondem as necessidades dos profissionais da saúde e utentes, no entanto devido ao cuidado que se deve ter na construção de plataforma da saúde é necessário fazer uma pesquisa minuciosa sobre este tipo de aplicações. Fizemos uma pesquisa sobre as aplicações existentes nesta área tendo sobressaído duas. Apresentaremos no capítulo 1.2 os dois projetos *open source* que podem ser usados para criar plataformas com características já referidas anteriormente.

1.2 Projetos semelhantes

É importante referir que existe uma panóplia de projetos que podem ser usados como base na criação de uma plataforma para partilha de informação no entanto, tratando-se duma plataforma para uso médico onde deve se ter especial cuidado com o tratamento da informação da ficha médica do utente, surge a necessidade se filtrar as várias opções existentes. Das várias soluções existentes debruçamo-nos apenas sobre as duas aplicações que se revelaram mais interessantes e que podem ser usadas para criar uma plataforma de partilha de informação no âmbito da saúde: Dermatron e OpenMRS.

1.2.1. Dermatron [4]

É um projeto que se encontra alojado no GitHub^[5], com licença MIT, o projeto consiste numa aplicação que já incorpora pequenos módulos que permitem gerir consultas e registo de utentes, inserir e pesquisar medicamentos, agendar consultas, permite também inserir um conjunto de informação sobre um utente como lesão, sintomas, imagem capturada via dermatoscópio USB.

É possível criar módulos que permitem a partilha de dados de utentes, porém deve-se ter um especial cuidado na restrição aos acessos aos dados, pois a aplicação não fornece mecanismos de privacidade quanto aos dados, todos os dados podem ser vistos por todos o que contraria quer o RGPD quer o HIPAA (*Health Insurance Portability and Accountability Act*) americano.

1.2.2. OpenMRS ^[6]

Oficialmente designado por Open Medical Record System, é um projeto *open source* para desenvolvimento de *software* baseado em sistema de registos clínicos eletrónicos, tendo sido desenvolvido essencialmente para apoiar a prestação de cuidados de saúde nos países em desenvolvimento. O *software* cresceu a partir da crítica necessidade de ampliar os cuidados/tratamentos do vírus HIV no continente africano, mas desde o início que foi concebido como um sistema de registos clínicos eletrónicos de uso geral, podendo suportar toda a gama de tratamentos médicos.

O OpenMrs tem uma estrutura modular e robusta, tornado possível a implementação de módulos de Sistemas de Informação em Saúde que permitam partilhar informação coerente e aumentar a interoperabilidade.

É importante apontar alguns dos aspetos que tornam o OPENMRS robusto tais como a autenticação dos utilizadores, acesso baseado em privilégio, criação e manutenção de dados de utentes, incluindo dados demográficos, observações clínicas. O OPENMRS permite criar registos médicos sendo também possível exportá-los de modo a serem usados em outras ferramentas (Excel, Access). Os registos também podem ser imagens, ficheiros de som.

Devido à vasta comunidade envolvida no desenvolvimento existem muitos módulos distintos, desenvolvidos em tecnologias diferentes, dos quais destacamos o modulo desenvolvido especialmente para dermatologia SkinHelpDesk^[7].

O modulo SkinHelpDesk dá a possibilidade aos médicos dermatologistas de apresentarem o seu parecer sobre o perfil de um paciente, este perfil é denominado por *skin profile*. O *skin profile* é um conjunto de informações sobre uma patologia de um utilizador tais como problemas de pele, detalhes sobre o problema, como quando e onde começou, para que um paciente possa ver analisado o seu perfil, basta fazer um registo na aplicação e submeter o seu *skin profile*.

1.3 Medical Skin Care

Na seção 1.2 apresentamos alguns projetos que podem ser usados para criar plataformas que permitam a partilha de informação médica de utentes. São aplicações amplamente conhecidas, utilizadas em muitos países, especialmente o OpenMRS, e que apresentam uma base sólida, no caso deste último com a vantagem de poder integrar módulos construídos com tecnologias diferentes, o que nos levou a considerar a hipótese de desenvolver um módulo específico de dermatologia sobre OpenMRS. Contudo é necessário ter em conta a compatibilidade na integração dos módulos, o que implicaria tempo elevado de pesquisa e aprendizagem da nossa parte. Dado o curto prazo para execução do projeto e a longa lista de pesquisas realizadas, desde a consulta com os profissionais da saúde às pesquisas de normas sobre proteção dos dados, decidimos no nosso projeto implementar uma aplicação de raiz que permite a partilha de ficha de utentes entre médicos dermatologistas e doravante vamo-nos referir à nossa aplicação como Medical Skin Care.

O projeto consiste numa aplicação Web de apoio ao diagnóstico em dermatologia, incluindo dermatoscopia diferencial, com partilha entre médicos da ficha médica do utente e com atenção especial à privacidade dos dados do utente. Como tal o tratamento dos dados constantes na ficha médica do utente tem em consideração requisitos fundamentais, tais como: confidencialidade, integridade, autenticação e conformidade legal, nomeadamente tendo em especial consideração a privacidade dos dados. No entanto é necessário ter em consideração que o projeto não permite a integração com os sistemas de outros hospitais, como tal, é uma plataforma independente e que as unidades hospitalares deverão criar uma conta que lhes permitirá, entre outras operações, registar os seus médicos, enfermeiros e utentes.

Com a partilha da ficha médica de utentes pretendemos alargar a quantidade de informação sobre o utente a que um dermatologista terá acesso, informações como fotos de sinais, histórico familiar, histórico médico, informações essas que ajudam o dermatologista a ter uma melhor perceção sobre a patologia.

1.3.1 Objetivo

O principal objetivo é permitir a partilha da ficha médica de utentes com vista a alargar a quantidade de informação sobre o utente a que um dermatologista terá acesso, informações como fotos de sinais, histórico familiar, histórico médico, informações essas que ajudam o dermatologista a ter uma melhor perceção sobre a patologia.

1.4 Estrutura do relatório

Nos capítulos que se seguem apresentaremos uma descrição mais detalhada sobre os aspetos da implementação da nossa solução, começaremos por apresentar os requisitos funcionais da aplicação, logo a seguir apresentaremos as tecnologias utilizadas na aplicação. Depois de introduzirmos as tecnologias utilizadas, no capítulo a seguir faremos a descrição das camadas da arquitetura da aplicação. Uma vez concluída a descrição da aplicação, apresentaremos no capítulo notas finais as nossas conclusões e sugestões para continuação do trabalho.

2. Requisitos Funcionais

2.1. Utilizadores

A plataforma tem 4 tipos de utilizadores:

- Administrativo que representa a unidade hospitalar
- Médico
- Enfermeiro
- Utente.

A seguir faremos uma descrição do papel desempenhado por cada um dos utilizadores.

2.1.1 Administrativo

Um administrativo representa uma unidade hospitalar, o administrativo pode registar médicos e enfermeiros na sua unidade hospitalar, também pode adicioná-los à sua unidade hospitalar caso esses já estejam registados em outras unidades hospitalares. Todos os utilizadores precisam estar autenticados de modo a usar a aplicação, no caso dos médicos, enfermeiros e utentes quando do seu registo é gerada uma *password* (enviada para o *email* pessoal) que lhe permitirá aceder à aplicação, o utilizador poderá posteriormente alterar a *password* para uma que lhe seja mais conveniente. No caso do utente, quando do seu registo, é-lhe atribuída uma equipa médica. O acesso aos dados médicos do utente é apenas concedido à equipa. O administrativo tem apenas acesso aos dados pessoais do utente.

Relativo a médicos e enfermeiros:

- Registrar (criar uma conta)
- Integra-los numa equipa
- Remove-los duma equipa
- Adicionar um médico como líder duma equipa.

Um administrativo pode registar utentes ou adicioná-los caso já estejam registados em outra(s) unidade hospitalar. O administrativo envia um pedido de permissão para acesso aos dados do utente, neste pedido é associada uma equipa, caso o utente aceite, o acesso as fichas do utente serão limitadas aos membros da equipa associada ao pedido.

Ao adicionar um utilizador é necessário que o mesmo indique o *email* de registo na aplicação. No caso do utente, quando é adicionado, o mesmo recebe um *email* a indicar que tem uma notificação na aplicação, quando o utente aceder à aplicação poderá encontrar o pedido de acesso às suas fichas.

Relativo a utentes:

- Registrar (criar uma conta)
- Apagar (dentro da sua unidade)
- Adicionar (associar um utente à sua instituição)
- Consultar os dados pessoais do utente.

Um administrativo pode criar equipas, uma equipa é constituída por médicos e enfermeiros. A equipa tem um team Leader que é o médico principal, os restantes integrantes da equipa são

profissionais da saúde que o team leader pode consultar sem que seja necessária a permissão extra para os mesmos terem acesso aos dados do utente, uma vez que já pertencem a equipa.

Relativo à equipa médica:

- Criar uma equipa
- Eliminar
- Adicionar integrantes
- Remover integrantes.

As funções desempenhadas pelo administrativo podem ser resumidas na figura 1, o administrativo depois de ser registado na aplicação poderá adicionar ou registar médicos, utentes e enfermeiros, criar equipas, remover e adicionar profissionais à equipa.

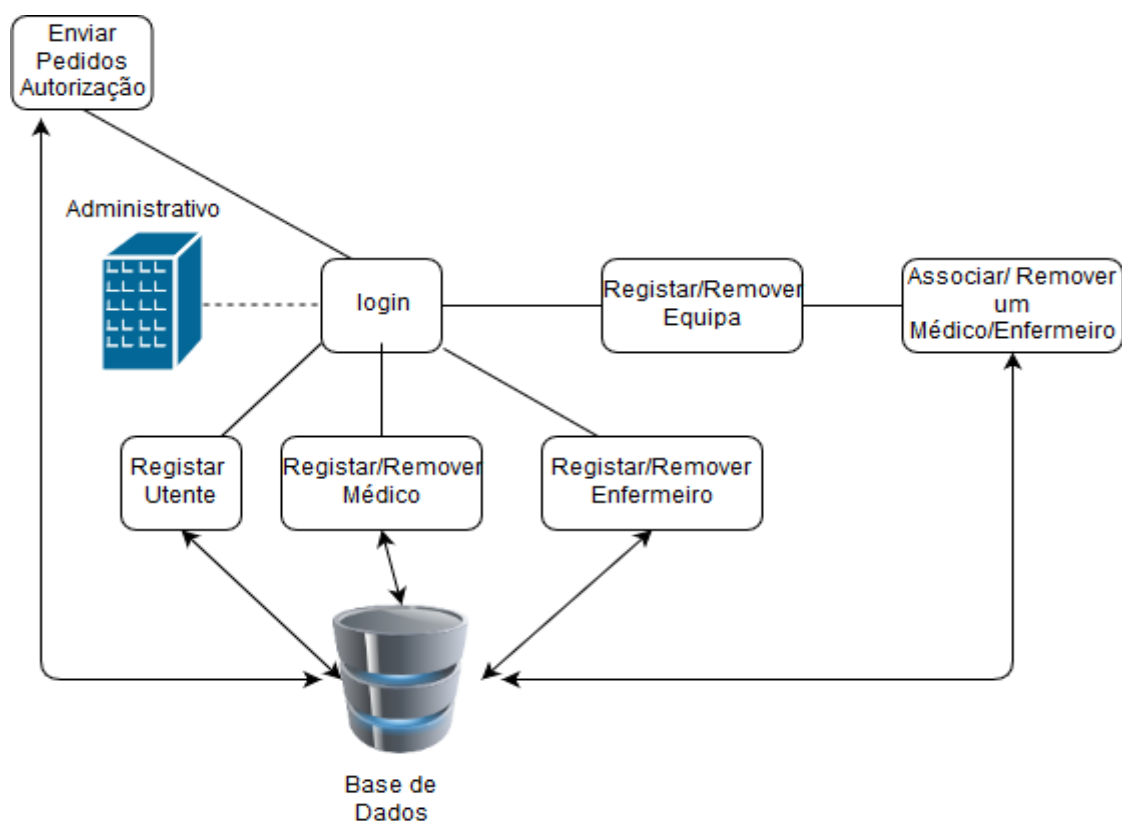


Figura 1- Representação gráfica das funções que um administrativo pode desempenhar.

2.1.2 Médico

Um médico pode estar associado a várias unidades hospitalares, dentro duma unidade hospitalar o médico pode estar inserido em várias equipas, cada uma dessas equipas tem utentes associados. O médico pode apenas criar ou consultar as fichas médicas de utentes das equipas em que ele está associado.

Resumo das funções do médico:

- Criar um grupo de fichas
- Criar uma ficha
- Pode adicionar uma ficha ao grupo de ficha de um utente
- Pode consultar as fichas dum utente

- Pode adicionar notas sobre a ficha dum utente
- Consultar informações sobre médicos e enfermeiros da mesma equipe
- Preencher questionários sobre histórico do utente.

A ficha médica do utente é constituída por informações específica de dermatologia como, por exemplo:

Avaliação à distância de 1 ou 2 metros: Contem informações relativas ao tipo de pele, fototipo, palidez, humidade, espessura e temperatura.

Avaliação à distância de 20 a 30 cm (detalhada): Contem informações sobre lesões cutâneas como, diagnóstico provisório, localização e imagens.

Para cada utente são criados questionários recolhidas apenas uma vez (quando do registo). Esses questionários são recolhidos com objetivos de serem usados em pesquisas estatísticas, são questionários que contêm informações sobre histórico familiar, histórico social, histórico sexual e histórico médico.

As funções desempenhadas pelo médico podem ser resumidas na figura 2, quando um médico é associado à um utente, o médico tem acesso aos dados do utente, o médico pode consultar os detalhes das fichas, de igual modo pode criar fichas, adicionar notas médicas como mostra a figura 2.

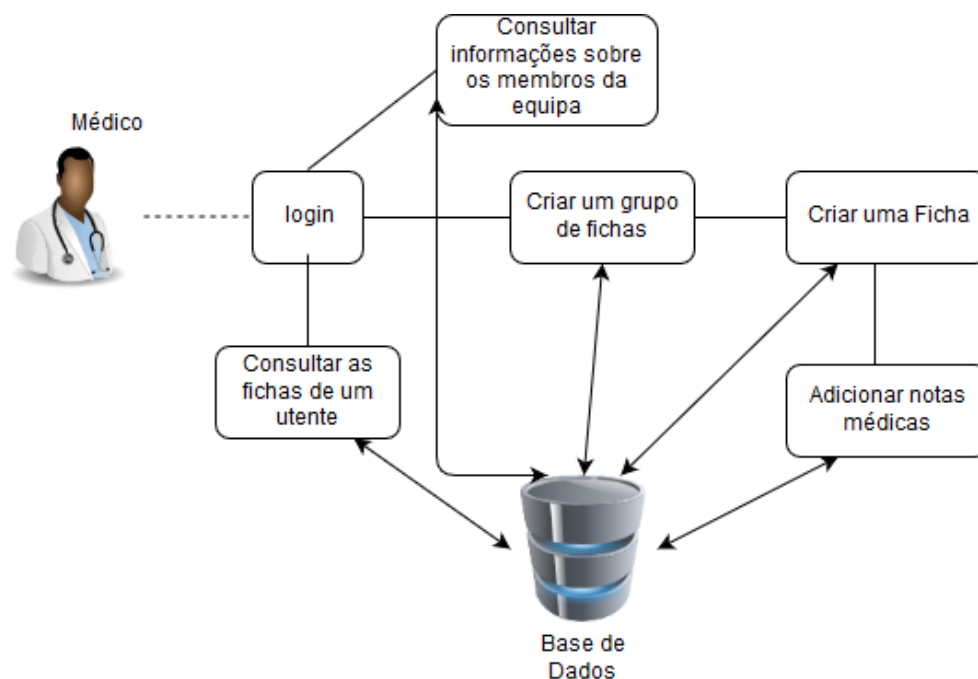


Figura 2 - Representação gráfica das funções que um médico pode desempenhar.

2.1.3 Enfermeiro

A descrição sobre os médicos pode ser aplicada de igual modo aos enfermeiros, com exceção na criação da ficha de utente, um enfermeiro apenas pode consultá-las, os enfermeiros não podem criar fichas médicas de utentes.

Resumo das funções do enfermeiro:

- Pode consultar as fichas dum utente
- Pode adicionar notas sobre a ficha dum utente
- Consultar informações sobre médicos e enfermeiros da mesma equipe
- Preencher questionários sobre histórico do utente.

As funções desempenhadas pelo enfermeiro podem ser resumidas na figura 3, quando um enfermeiro é associado à um utente, o enfermeiro tem acesso aos dados do utente, pode consultar os detalhes das fichas e adicionar notas de enfermagem figura 3.

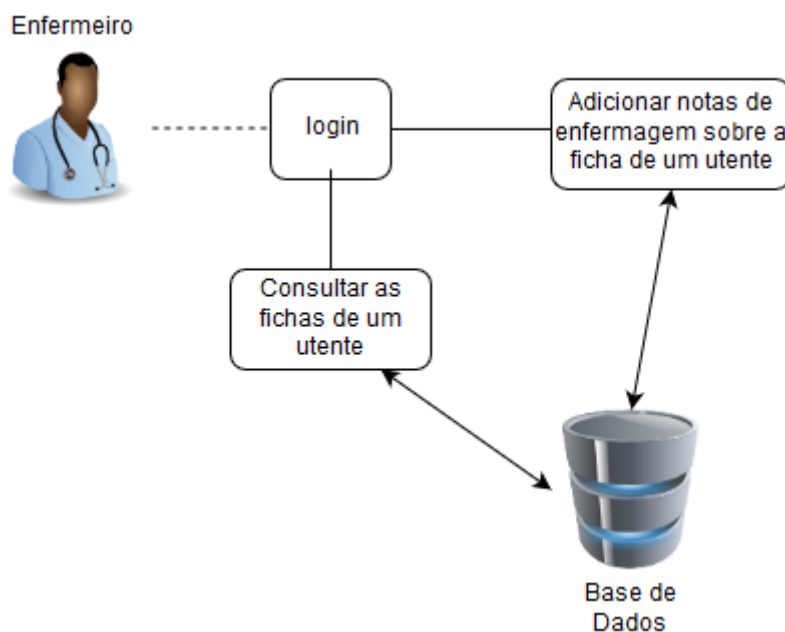


Figura 3 - Representação gráfica das funções que um enfermeiro pode desempenhar.

2.1.4 Utente

Um utente pode ser associado a várias unidades hospitalares e dentro das unidades hospitalares pode estar dentro de várias equipas, um utente tem sempre uma equipa associada, somente os integrantes desta equipa têm acesso às fichas do mesmo. Uma equipa pode criar ou consultar os grupos de fichas do utente sendo o grupo de fichas constituído por todas as fichas médicas com a mesma patologia. Quando o utente concede permissão de acesso aos seus dados, apesar do pedido ser enviado por uma unidade hospitalar, os seus dados clínicos serão apenas acedidos pela equipa associada ao pedido, a unidade hospitalar terá apenas acesso aos dados pessoais. O utente pode saber quais são as equipas que têm acesso aos seus dados.

O acesso à ficha do utente é limitado aos médicos e enfermeiros que pertençam à equipa a quem o utente concedeu a permissão de acesso. Quando um utente recebe um pedido

de permissão de acesso às suas fichas, associado ao pedido tem sempre uma equipa médica, caso o utente aceite o pedido todos os integrantes da equipa médica terão acesso as fichas do utente.

Resumo das funções do utente:

- Pode consultar as suas fichas
- Conceder permissão de acessos aos seus grupos de ficha
- Consultar equipas há quem deu permissão de acesso.

As funções desempenhadas por utentes podem ser vistas na figura 4.

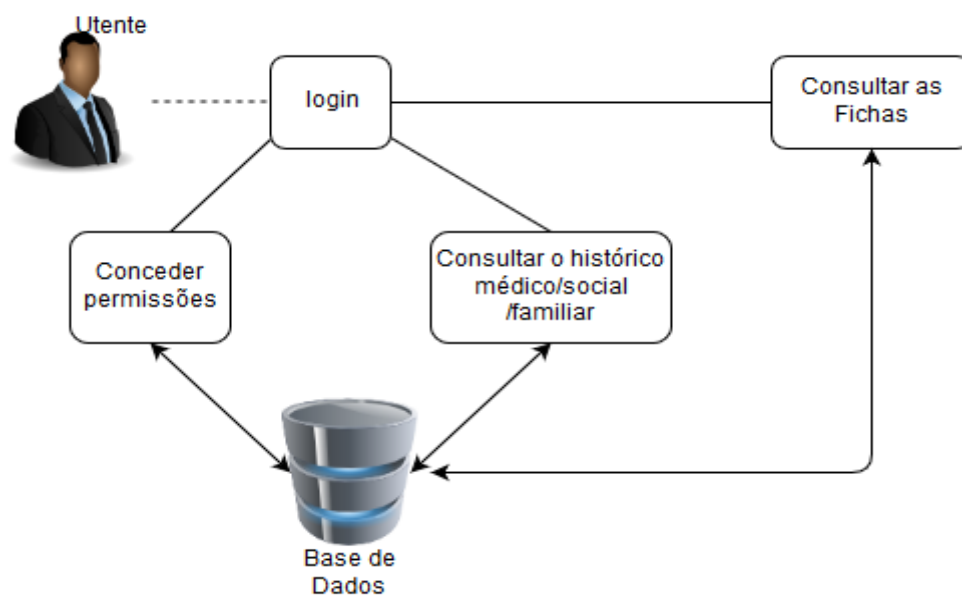


Figura 4 - Representação gráfica das funções que um utente pode desempenhar.

3. Tecnologias utilizadas no projeto

3.1 Servidor

Existe atualmente no mercado um leque de escolha vasto no que se refere às plataformas para desenvolvimento de aplicações servidoras. Tivemos em consideração três plataformas ,nomeadamente ASP.NET^[8], Java^[9] e Node.js^[10], por serem plataformas com linguagens de programação por nós dominadas. Poderíamos optar por qualquer uma delas, no entanto escolhemos a plataforma Node.js. O Node.js é de entre as três a plataforma mais rápida e simples, porém não foi este o motivo principal da escolha da plataforma, mas sim o facto de que a linguagem de programação utilizada em Node.js é o JavaScript^[11] que também é a linguagem de programação por que optámos para o desenvolvimento da aplicação cliente. É importante realçar que não é necessário que a componente servidora tenha a mesma linguagem de programação que a componente cliente, foi apenas uma opção tomada por nós de forma a evitar os constantes erros de sintaxe quando se trabalha com várias linguagens de programação distintas em simultâneo.

Na construção da aplicação servidora, o nosso objetivo é ter uma API^[12] com *endpoints* que poderão ser consumidos pela nossa aplicação cliente ou por outras aplicações.

Uma das vantagens de se utilizar a plataforma Node.js, é o elevado número de *frameworks* para o desenvolvimento de aplicações Web, devido à vasta comunidade de programadores que o JavaScript possui. A seguir mostraremos algumas dessas *frameworks* que podem ser utilizadas para a construção da API mencionada anteriormente.

3.1.1 *Frameworks* para construção de API

Como já foi referido anteriormente, devido à comunidade Node.js bastante ativa, existem muitas *frameworks* para construções de API, muitas delas com diferentes estratégias de implementação. Algumas *frameworks* seguem um conjunto de regras que permitem a implementação de uma API RESTfull^[13], de entre essas escolhemos a *framework* Loopback^[14], embora a mais popular seja o Express^[15]. Apesar do Express ser considerada a *framework* padrão de Node.js e dos recursos robustos que a *framework* oferece, o Express não oferece um padrão no que toca a organização e desenvolvimento do código, os *endpoints* para API devem ser escritos manualmente, por essas razões escolhemos o Loopback. Apresentaremos a seguir a *framework* Loopback com mais detalhe.

3.1.1.1 LoopBack

O LoopBack é uma *framework* construída sobre o Express, tem incorporado um conjunto de módulos do Node.js que podem ser usados para criar APIs REST^[16]. O LoopBack tem um conjunto de modelos predefinidos, os modelos são objetos JavaScript que representam recursos sobre os quais é possível fazer operações de CRUD. O LoopBack já tem alguns modelos embutidos como *User*, *Role* e *Application*, tal como em programação orientada a objetos, onde todos os objetos derivam da class Object, no Loopback o modelo base é o PersistedModel, ao criar um modelo que deriva de PersistedModel, passa-se a ter acesso as operações de CRUD sobre o mesmo modelo através duma API que o Loopback fornece para interagir com o modelo criado.

No exemplo 1 mostramos a criação do modelo doctor, com as propriedades Name e LastName, e a criação da API para interagir com o modelo doctor, a API está disponível no porto 8080 através do url: “/api”.

```

var loopback = require('loopback');
var app = module.exports = loopback();
var Item = loopback.createModel('Item', {
  mdescription: 'string', completed: 'boolean'
});
app.model(Item);
app.use('/api', loopback.rest());
app.listen(8080);

```

Exemplo 1 – Código para criação do modelo e da API para interagir com o mesmo.

O LoopBack oferece uma interface gráfica onde é possível ver os *endpoints* da API gerada no exemplo 1, o exemplo 2, mostra o código necessário para aceder a interface gráfica e a figura 5 mostra a interface gráfica.

```

app.start = function() {
  // start the web server
  return app.listen(function() {
    app.emit('started');
    var baseUrl = app.get('url').replace(/\/$/, "");
    console.log('Web server listening at: %s', baseUrl);
    if (app.get('loopback-component-explorer')) {
      var explorerPath = app.get('loopback-component-explorer').mountPath;
      console.log('Browse your REST API at %s%s', baseUrl, explorerPath);
    }
  });
};

```

Exemplo 2 - Código necessário para aceder a interface gráfica da API.

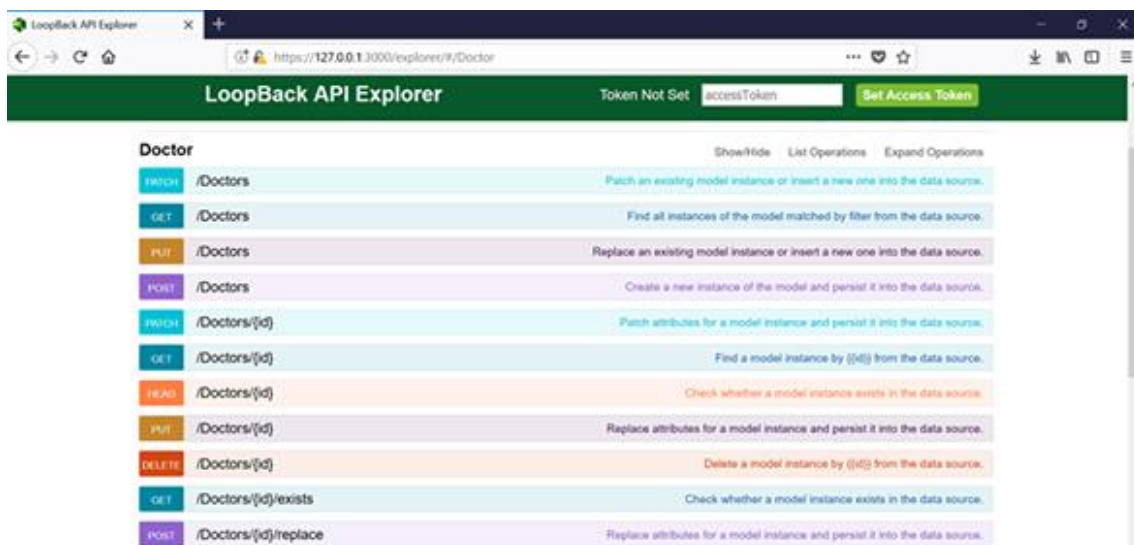


Figura 5 - Página de teste da API acedida através do URL <http://localhost:8080/explorer>

3.2. Base de Dados

3.2.1 MongoDB

Uma vez que o foco do nosso projeto é a partilha de informação, foi necessário encontrar a melhor forma de armazenamento destas informações (dados da ficha do cliente e dados pessoais). Atualmente quando se fala em armazenamento de informação em base de dados a grande pergunta que surge é : NoSQL^[17] ou Base de Dados Relacional^[18]? A resposta a essa pergunta é normalmente um assunto sobre o qual há muitas opiniões, porém não entraremos em detalhes, a nossa escolha foi uma base de dados NoSQL , a MongoDB^[19]. A razão principal para a escolha foi a fácil integração com o Loopback, o investimento que quem desenvolve esta BD está a realizar na área da segurança e pelos mecanismos de consulta e recursos de indexação muito poderosos que facilitam e agilizam a execução de várias consultas otimizadas diferentes, pois um dos objetivos futuros é fornecer dados estatísticos que poderão ser usados por outras aplicações. A MongoDB suporta consultas ricas, distinguindo-se de outras bases de dados documentais que dificultam consultas complexas.

3.2.2. Cifra dos dados

A segurança dos dados do utente é uma parte importante neste projeto. Uma vez que os dados serão partilhados é necessário garantir a segurança desses dados, tanto no armazenamento, como no transporte dos mesmos. No transporte usamos o protocolo HTTPS^[21] que garante um canal seguro para a transição dos dados. Quanto à segurança no armazenamento os dados são cifrados antes de serem armazenados.

A cifra dos dados do utente é feita usando o tipo de cifra simétrica, usamos duas chaves, a Data Encryption Key (DEK) que é a chave usada para cifrar os dados do utente, e a Key Encryption Key (KEK) que é um identificador que juntamente com um URI (a ser explicado mais a frente) permite obter a chave para decifrar a DEK, uma vez que a DEK se encontra igualmente cifrada.

Depois de ser decifrada, a DEK é usada como chave para cifrar e decifrar os dados do utente, esta cifra e decifra é realizada utilizando o módulo cryptoJson.

O processo de cifra e decifra da DEK é explicada detalhadamente na próxima secção.

3.3 Microsoft Azure

Microsoft Azure é um serviço de *cloud* criado pela Microsoft com diversos propósitos, entre eles o construir e alojar aplicações.

3.3.1 Key Vault

Azure Key Vault é um serviço de *cloud* que funciona como um local de armazenamento seguro de chaves.

O Key Vault fornece funções de cifra e decifra. Usamos a função de cifra do Key Vault para cifrar a DEK, a cifra é feita passando a DEK e o KEK à função de cifra, o Key Vault internamente adiciona um URI(armazenado nas variáveis de ambientes) à KEK formando assim o caminho completo para a chave (que se encontra armazenada no Azure Key Vault) que será usada para cifrar a DEK.

Quando um utente ou um profissional autorizado acede aos dados do utente, os dados são retornados seguindo os seguintes passos:

1º Decifrar a DEK:

Como já foi mencionado usamos a KEK como uma parte do caminho para o local onde se encontra a chave que será usada para decifrar a DEK, sendo a KEK um identificador parcial da chave armazenada no serviço Key Vault, precisamos também do Uri ou localização (guardada nas variáveis de ambiente) do Vault que juntamente com a KEK formam o identificador da chave que decifrá a DEK.

A seguir, usamos a função de decifra do Key Vault para decifrar a DEK.

2º Decifrar os dados:

Uma vez que já temos a DEK decifrada, usamos novamente o módulo npm cryptojson para decifrar os dados do utente.

Este processo é utilizado de igual modo para os dados pessoais de outros utilizadores.

3.3.2 Storage Service

Como foi referido nos capítulos anteriores, na ficha médica do utente constam fotografias de sinais. Essas fotografias juntamente com outros dados do utente (referidos anteriormente) também são partilhadas com os profissionais autorizados. Por questões de segurança decidimos desassociar os dados do utente das fotografias dos sinais, assim as fotografias não são armazenadas na MongoDB como os outros dados da ficha do utente.

Para o armazenamento de dados, texto ou dados binários, o Microsoft Azure, fornece o *Storage service*. Usamos o *Storage service* para armazenar as fotos dos sinais do utente na *cloud*.

Dentro dum *Storage service* existem *Containers*, estes por sua vez contem vários *blobs*, que são uma coleção de dados binários.

Quando um médico cria uma ficha do utente podem ser adicionadas fotografias dos sinais. Armazenamos as fotografias da ficha do utente, em *containers*, um *container* por cada ficha do utente. Assim sempre que é adicionado uma fotografia à ficha do utente, a mesma é guardada num *container*.

O armazenamento da foto na *cloud* é feita em dois passos:

1º Criação do *container*

Quando é adicionada a primeira foto numa ficha, usamos o módulo *azure storage* para criar um *container*, passando uma chave parcial (gerada pelo módulo npm key generator) e esta chave parcial é armazenada na ficha do utente.

2º Upload da foto

Depois da criação do *container* usamos a função *upload* do modelo *container*, fornecido pela framework Loopback, para fazer o *upload* da foto para *cloud*, passando-lhe a chave parcial.

Quando adicionamos uma foto a uma ficha que já contém fotos, não é preciso criar um *container*, basta passar à função *upload* a chave parcial que foi armazenada na ficha.

Para ambos os passos acima é adicionada à chave parcial o uri para a conta (azureStorageAccount) e o *storage accessKey* para autenticação.

3.4 Client-Side

Como já foi mencionado na seção 4.1, a componente cliente foi construída em JavaScript. É cada vez mais raro ver aplicações clientes serem construídas sem recorrer às várias *frameworks* de JavaScript que vão surgindo cada vez com maior frequência. A procura de melhor desempenho e simplicidade no modo como se desenvolvem as aplicações são alguns dos motivos que levam os desenvolvedores à procura dessa *frameworks*. A nossa componente cliente foi desenvolvida sem recorrer a *frameworks*. Usamos o modelo JavaScript Html e CSS. Uma decisão tomada com o objetivo de poupar a necessidade do estudo das *frameworks* existentes.

3.4.1 Captura de Foto Através dum Dermatoscópio.

O dermatoscópio é um aparelho médico, utilizado para diagnósticos e exames de pele. O dermatoscópio é formado por uma fonte de luz que pode ser de LED ou lâmpada de halogéneo, e uma lente (dubleto acromático) que proporciona aumento (*zoom*) da imagem visualizada.

As imagens de sinais dos utentes podem ser capturadas via dermatoscópio. Essas imagens são interpretadas de acordo com os critérios e sinais dermatológicos, os quais necessitam ser reconhecidos e entendidos pelo médico dermatologista para um diagnóstico preciso.

A informação fornecida pelo dermatoscópio é um método auxiliar de verificação da existência de lesões pigmentadas na pele, sendo útil para avaliação e prevenção de pré-operatórios em clínicas e hospitais. A figura 8 mostra o dispositivo que foi utilizado como “dermatoscópio” neste projeto, mas que não é um dermatoscópio profissional, apenas uma câmara de vídeo com uma lente que permite realizar *zoom* até 100x e passar essa informação ao computador via interface USB. Tem a vantagem de custar apenas cerca de 20€ contra as centenas de euros de um dermatoscópio “digno do nome”. Serve, no entanto, para ilustrar a utilização de um verdadeiro dermatoscópio.

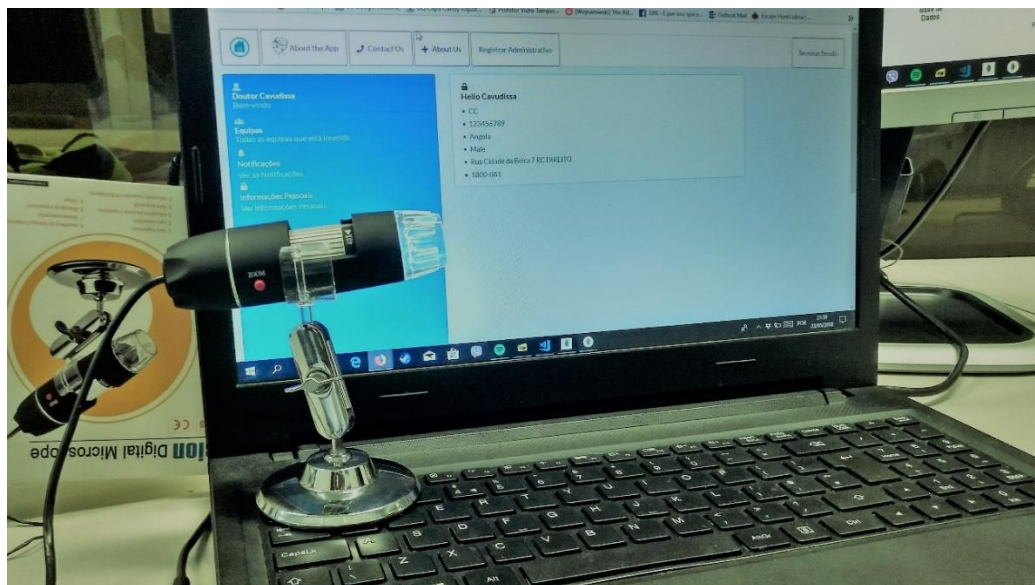


Figura 6 - Substituto de um dermatoscópio – pode ser ligado ao computador e permite através da sua camera e da API MediaDevice capturar imagens ampliadas (até 100x) de sinais.

4. Arquitetura

Devido a complexidade do projeto, assim como para facilitar a futura manutenção do código, optámos por desenvolver o projeto utilizando uma arquitetura em camadas como mostra a figura 7. A arquitetura em camadas pode ser definida como um processo de decomposição de sistemas complexos em camadas para facilitar a compreensão do mesmo, como também facilitar a manutenção do sistema. Utilizando a arquitetura em camadas é possível compreender uma única camada coerentemente como um todo, sem a necessidade de muito conhecimento das outras camadas sendo também é possível substituir as camadas por implementações alternativas dos mesmos serviços básicos.

Os componentes da aplicação encontram-se organizados em camadas, que são:

- Camada de acesso aos dados
- Camada de negócio
- Camada de apresentação.

Neste capítulo, vamos descrever mais detalhadamente cada camada e como elas interagem.

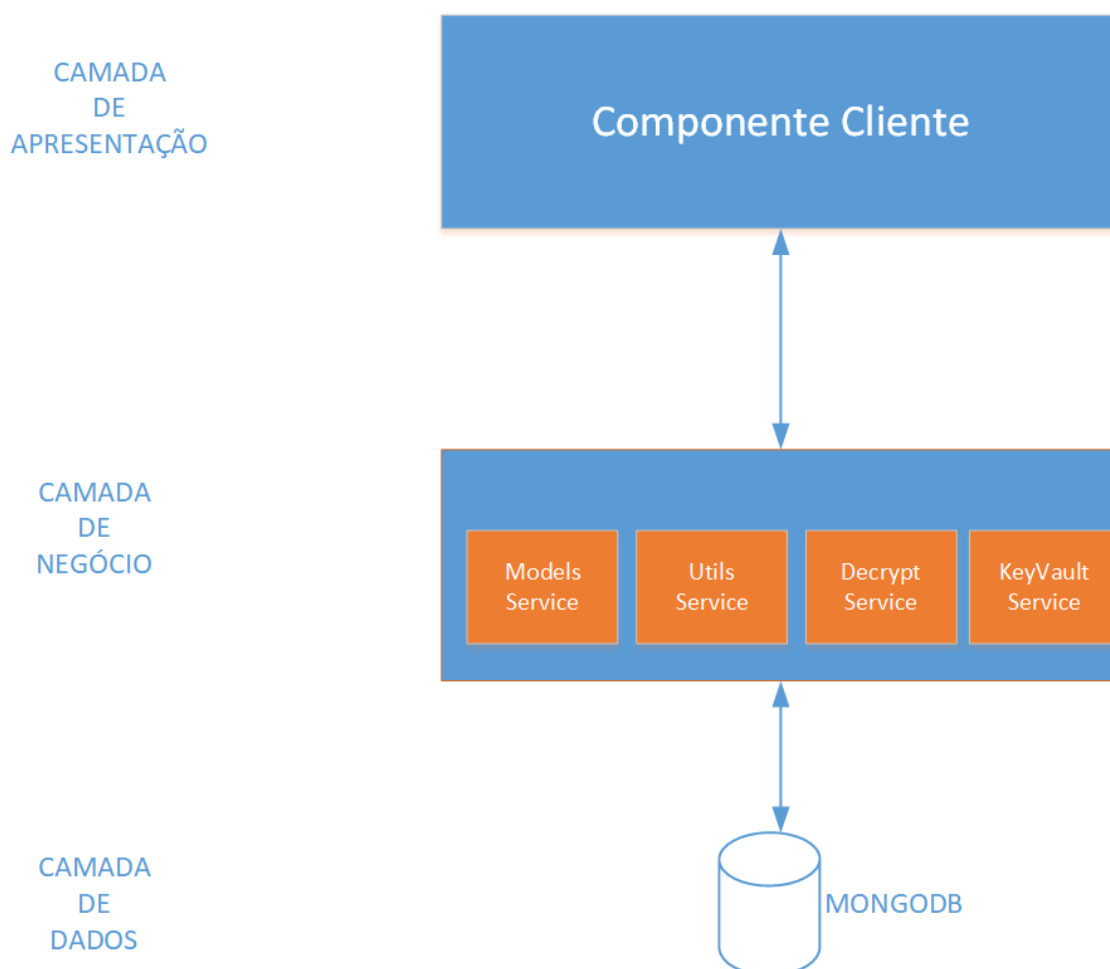


Figura 7 – Arquitetura do Sistema

4.1 Camada de apresentação

A camada de apresentação é responsável pela apresentação e interação do utilizador com a aplicação. Uma vez que o foco da aplicação é a partilha de informação, a recolha de informação é crucial. A mesma é feita através de formulários HTML que são apresentados ao utilizador pela componente cliente. Os formulários contêm informações sobre fichas de utentes que são enviadas para a componente servidora.

Uma vez que nos focamos em recolher a máxima quantidade de informação possível de modo a que os profissionais da saúde autorizados tenham sempre as informações que precisam, temos um conjunto de formulários HTML que foram construídos usando o Handlebars tirando partido da simplicidade na criação de *views* através de reutilização de *templates* (partial).

4.2. Camada de negócio

Nesta camada fazemos a ponte entre a camada de apresentação e a camada de acesso a dados, a qual será apresentada a seguir. Os formulários preenchidos na camada de apresentação são recebidos e verificados antes de serem enviados para a camada de acesso a dados, da mesma maneira, quando chegam dados da camada de acesso a dados, antes de serem enviados para a camada de apresentação, esses dados são decifrados e só depois apresentados. É também nesta camada onde tratamos das restrições e acesso aos dados do utente que será descrita a seguir.

4.2.1 Cifra dos Dados

Os dados recolhidos na camada de apresentação através de formulários e questionários, são cifrados antes de serem armazenados na camada de acesso aos dados, assim como os dados provenientes da camada de acesso de dados são decifrados antes de serem apresentados na camada de apresentação de dados. O processo de cifra e decifra foi apresentado nas seções 3.2. e 3.3 do capítulo 3.

4.2.2 Armazenamento das imagens

As imagens dos sinais que podem constar na ficha médica do utente, não são armazenadas na MongoDB, mas sim no serviço de *cloud* Microsoft Azure, a seção 3.3.2 do capítulo 3 explica de forma detalhada o processo de armazenamento.

4.2.3 Registo e Autenticação

Mesmo a um utilizador registado é necessário restringir o acesso do mesmo a certas informações. São exemplo o caso de um administrativo da unidade hospitalar que não deve ter acesso à ficha médica do utente, tendo somente acesso à informação pessoal do utente, e o caso do enfermeiro que não pode criar uma ficha médica. Para esses casos usamos a abordagem *role-based access control* (RBAC).

Quando é registado um utilizador, é-lhe atribuído um *role* que permitirá restringir o acesso deste a ações.

Ao fazer *login* é atribuído um *token* ao utilizador através do qual é possível determinar o seu *role* de modo a garantir a restrição de ações não permitidas ao mesmo. Aquando do *logout* é destruído o *token* de acesso que lhe foi atribuído anteriormente.

A *framework* Loopback fornece um conjunto de modelos embutidos que derivam de *PersistedModel* que permitem realizar as operações de CRUD.

Para o utilizador particular (Utente, Doutor ou enfermeiro) usamos o modelo User da *framework* Loopback. O modelo User também deriva de PersistedModel, porém tem mais 4 *endpoints* que permitem fazer *login*, *logout*, confirmar o *email* e mudar a *password*.

Criamos o objeto *user* que tem como base o User da *framework* Loopback, o objeto *user* tem relação com todos os utilizadores particulares. Esta relação permite aos utilizadores particulares aceder aos 4 *endpoints* de User (*login*, *logout*, confirmar o *email* e mudar a *password*).

Quando um administrativo regista um utilizador particular são criados dois objetos, um *user* apenas com *email* e *password*, outro (cifrado) que representa o utilizador a ser criado (utente, doutor ou enfermeiro), é estabelecida uma relação os dois objetos.

4.3 Camada de Acesso Aos Dados

4.3.1. Modelo de Dados

O modelo de dados, apresentado resumidamente na figura 8, é constituído por 13 entidades, das quais Administrative que representa uma unidade hospitalar, uma unidade hospitalar tem associada várias equipas Team, por sua vez as equipas podem ter vários médicos Doctor e enfermeiros Nurse. Quando uma unidade hospitalar envia um pedido para que uma equipa possa ter acesso as fichas do utente patient, caso o utente aceite o pedido de acesso à sua ficha, essa informação é registada em Shared-Permission, criando assim uma relação entre o utente e a equipa. O utente tem uma relação com a sua ficha médica Medical-Record-Group e os seus questionários, por conseguinte cada elemento da equipa terá acesso a essa informação através do utente.

O grupo de ficha médica Medical-Record-Group é um conjunto de fichas médicas agrupadas por patologia, essas fichas médicas estão divididas em duas partes:

Medical-Record - Contém informações sobre exame físico geral. O exame é feito em um ambiente iluminado (de preferência com iluminação natural), com temperatura confortável para o paciente, com o mínimo de roupa possível ou sem roupas íntimas, meias e sapatos, óculos e maquiagem essas avaliação começam a distância de 1-2 metros.

Detailed-Medical-Record - Contém informações mais detalhadas de avaliações que são recolhidas de análises feitas a uma distância de 20-30cm , informações sobre o tipo de lesão cutânea , a localização a textura, descrição da imagem dos sinais. Essas descrições estão associadas a fotografias que estão alojadas no *storage service* do Microsoft Azure. Essas informações dão a possibilidade de o médico realizar um diagnostico provisório.

Os questionários do utente estão armazenados em 4 documentos distintos.

Medical-History - Contém informações sobre o historial médico do utente, de entre elas a estrutura física do utente, tipo sanguíneo alergias, histórico de cirurgias e doenças clínicas, vacinas.

Family-Medical-History - Contém informações relacionada aos parentes mais próximos do utente, históricas doenças hereditárias.

Social-History – Contém informação sobre a os fatores sociais que podem estar na origem de problemas de saúde do utente, informações sobre o consumo de bebidas alcoólicas, atividades desportivas praticadas, higiene, hobbies entre outras.

Sexual-History – Contém informações sobre o histórico sexual do utente e doenças transmissíveis sexualmente.

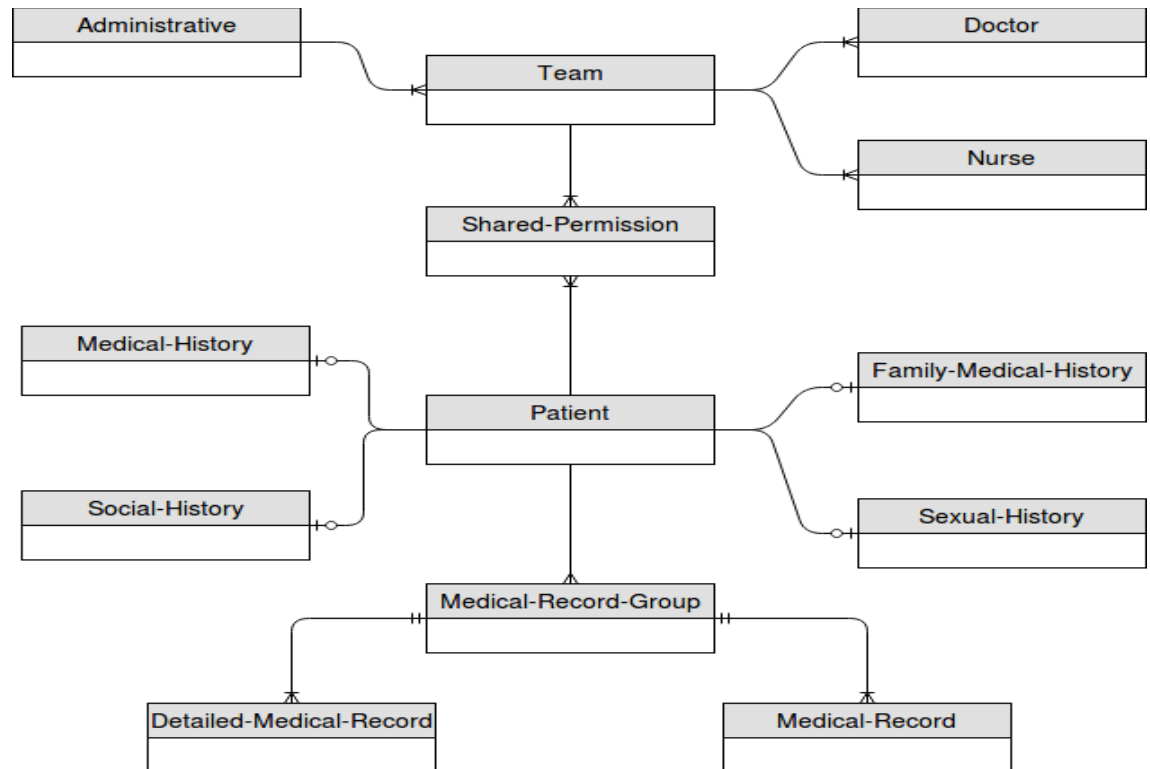


Figura 8 - Representação gráfica do modelo de dados.

5. Notas finais

Nos capítulos anteriores, abordamos detalhadamente os passos seguidos para a realização do projeto que resulta numa aplicação Web que permite a partilha de fichas médica de utente entre médicos dermatologistas.

Na fase preparatória do projeto, fizemos uma análise das necessidades dos futuros utilizadores e os projetos já existente no âmbito da partilha de informação. A realização do projeto só foi bem sucedida devido as várias pesquisas realizadas antes da execução, tivemos consulta um médico dermatologista e com alguns enfermeiros, também fizemos pesquisas sobre regulamentos de proteção de dados, essas pesquisas ajudaram-nos a desenvolver um produto que facilitará o trabalho de muitos profissionais da saúde.

Na execução do projeto tivemos o cuidado de desenvolver a aplicação de forma modular, onde cada modulo poderá ser substituído futuramente por outros, com o objetivo de futuramente estender o projeto de modos a suportar aplicações para dispositivos moveis, criar módulos que permitam outras formas de autenticação, estender a aplicação para outras áreas da saúde.

A aplicação desenvolvida tem como vantagem a privacidade no tratamento dos dados constantes na ficha do utente de dermatologia com especial consideração nos requisitos de segurança relativos à privacidade e integridade dos dados do utente. Porém a desvantagem reside no tempo que poderá levar até conseguirmos a confiança das unidades hospitalares pois sabemos que não será fácil convencer as unidades hospitalares a usarem uma plataforma independente para registar os médicos e utentes, cremos, no entanto, que a necessidade de uma plataforma de partilha de informação falará mais alto.

Referências

Ordem	Link
[1]	http://spms.min-saude.pt/wp-content/uploads/2017/03/Guia-Privacidade-SMPS_RGPD_digital_20.03.172-v.2.pdf
[2]	https://www.sns.gov.pt/
[3]	https://www.deco.proteste.pt/tecnologia/tablets-computadores/noticias/protecao-de-dados-o-que-precisa-saber-sobre-o-novo-regulamento-europeu
[4]	https://github.com/mertyildiran/Dermatron
[5]	http://github.com/
[6]	https://openmrs.org/
[7]	https://openmrs.org/https://github.com/dermatologist/openmrs-module-skinhelpdesk
[8]	https://docs.microsoft.com/en-us/aspnet/index
[9]	https://www.java.com/pt_BR/
[10]	https://nodejs.org/en/
[11]	https://www.javascript.com/
[12]	https://medium.freecodecamp.org/what-is-an-api-in-english-please-b880a3214a82
[13]	https://www.restapitutorial.com/
[14]	http://loopback.io/doc/
[15]	https://expressjs.com/
[16]	https://www.restapitutorial.com/
[17]	https://www.mongodb.com/nosql-explained
[18]	https://www.codecademy.com/articles/what-is-rdbms-sql
[19]	https://www.mongodb.com
[20]	https://en.wikipedia.org/wiki/Document-oriented_database
[21]	https://www.instantssl.com/ssl-certificate-products/https.html

Bibliografia

As informações sobre o conteúdo das fichas de Utente, foram recolhidas com ajuda de profissionais da saúde e dos livros seguintes:

F.Guerra Rodrigo e al., Dermatologia - Fichero Clinico (inclui CD-ROM) - 4.ª edição, Fundação Calouste Gulbenkian, junho de 2010, ISBN: 9789723113167

Kail, Robert V (2011), Children and Their Development (6th Edition) (Mydevelopmentlab Series), Englewood Cliffs, N.J: Prentice Hall, ISBN: 0205034942

Klaus Wolff, Richard Allen Johnson, Arturo Saavedra, Ellen K. Roh, Fitzpatrick's Color Atlas, 8th Ed, – 16 Mar 2017, McGrawHill Education, isbn: 9781259642197

