

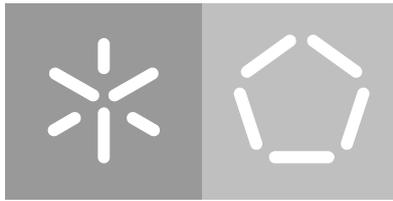
**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

Luís Manuel Cachada Dias

**Gestão de Informação em Contexto Hospitalar**



**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

Luís Manuel Cachada Dias

## **Gestão de Informação em Contexto Hospitalar**

Dissertação de Mestrado

Dissertação de Mestrado em Engenharia Informática

Dissertação orientada por

**Professor Doutor José Machado**

**Professor Doutor António Abelha**

Julho 2021

**Despacho RT - 31/2019 - Anexo 3**  
**Direitos de autor e condições de utilização do trabalho por terceiros**

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositórioUM da Universidade do Minho.

**Licença concedida aos utilizadores deste trabalho:**



**Atribuição**

**CC BY**

*<https://creativecommons.org/licenses/by/4.0/>*

---

## AGRADECIMENTOS

---

Quero agradecer aos orientadores por toda a disponibilidade que demonstraram para o desenvolvimento desta dissertação pois sem eles não seria possível, uma vez que foram eles que lidaram com as minhas questões acerca do projeto e que me conduziram para o caminho correto de forma a ter sucesso no término deste projeto.

Quero também agradecer aos meus pais por todos os sacrifícios que eles fizeram de forma a que eu conseguisse alcançar os meus objetivos. Para além disso quero agradecer por eles me terem conseguido encaminhar sempre pelos caminhos certos de forma a que conseguisse concluir os meus objetivos delineados até ao momento e por nunca terem desistido de me apoiar e de alimentar os meus objetivos. Parte importante deste percurso efetuado também foram os meus avós que sempre mostraram orgulho e confiança em mim e que também contribuíram, à maneira deles, para o meu sucesso e por isso também a eles devo uma palavra de agradecimento.

Quero também agradecer à minha namorada porque sem a companhia dela em alguns momentos mais complicados e sem o seu incentivo nada disto teria sido possível e desta forma tornou-se uma peça importante no apoio e incentivo para conquistar mais um objetivo.

Por último quero agradecer também aos meus amigos e colegas que sempre estiveram perto de mim e que me ajudaram, de forma direta ou indireta, com a sua opinião acerca de opiniões relacionadas com o que desenvolvi e escrevi.

**Despacho RT - 31/2019 - Anexo 4**  
**Declaração de integridade**

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

---

## ABSTRACT

---

Nowadays it appears that more and more information is stored digitally, since it has several advantages in relation to the storage of that information on paper. One of these advantages is the minimization of errors, for example the loss of a paper with important information or when trying to access certain information can be done more quickly and effectively. Due to the advantages brought by digital information storage, health infrastructures increasingly use digital systems that allow the storage of relevant information in relation to all those who participate in its processes, namely the different health professionals.

Despite the advantages listed above in the existing systems in the health area and thus making a very important contribution, these systems also have disadvantages, such as the fact that there is still no system that can be of universal use. The lack of this universal system is essentially due to the fact that the use of medical language is not universal and thus it is possible that each health unit may have one or more subsystems of its own. With the lack of this universal language in the area of health, the [openEHR](#) arises, which allows processes to be created in a global system allowing them to be a system with universal use to be used by the different health units.

In order to continue using this platform, there was a need for a component that would allow communicating to stakeholders in the system which processes they would have to participate in. That said, it was necessary to create a notification system that alerts stakeholders. In order to establish this communication between system and user, a technology based on *Javascript* was used, in this case web sockets, with the aid of Docker technology.

**Keywords:** Hospital Information Systems, openEHR, notifications, *Javascript* communication technologies, web sockets and Docker technology.

---

## RESUMO

---

Nos dias correntes verifica-se que cada vez mais as informações são armazenadas digitalmente, uma vez que acarreta diversas vantagens em relação ao armazenamento dessa informação em papel. Uma dessas vantagens é a minimização de erros, por exemplo a perda de um papel com informação importante ou aquando a tentativa de acesso a uma determinada informação pode ser efetuada de uma forma mais rápida e eficaz. Devido às vantagens acarretadas pelo armazenamento digital da informação as infraestruturas de saúde recorrerem cada vez mais a sistemas digitais que permitam armazenar informações pertinentes relativamente a todos os que participam nos seus processos, nomeadamente os diversos profissionais de saúde.

Apesar das vantagens enumeradas anteriormente nos sistemas existentes na área da saúde e assim ter um contributo muito importante também estes sistemas possuem desvantagens, tais como o facto de ainda não existir um sistema que possa ser de uso universal. A falta deste sistema universal deve-se essencialmente ao facto de o uso da linguagem médica não ser universal e assim é possível que cada unidade de saúde possa ter um ou vários subsistemas próprios. Com a falta desta linguagem universal área da saúde surge o [openEHR](#) que possibilita que sejam criados processos num sistema global possibilitando que sejam um sistema com uso universal a ser utilizado pelas diversas unidades de saúde.

De forma a dar continuidade à utilização desta plataforma surgiu a necessidade de existir uma componente que permitisse comunicar aos intervenientes no sistema quais os processos em que teriam de participar. Dito isto, foi necessário criar um sistema de notificações que alerte os intervenientes. Para que fosse possível estabelecer esta comunicação entre sistema e utilizador foi utilizada uma tecnologia baseada em *JavaScript*, neste caso os web sockets, com o auxílio da tecnologia Docker.

**Palavras-chave:** Sistemas de Informação Hospitalar, openEHR, notificações, tecnologias de comunicação *JavaScript*, web sockets e tecnologia Docker.

---

## CONTEÚDO

---

1	INTRODUÇÃO	1
1.1	Contexto	1
1.2	Motivação	3
1.3	Contribuições e Objetivos	3
1.4	Estrutura do documento	4
2	ESTADO DE ARTE	6
2.1	Sistemas de informação Hospitalar - <i>HIS</i>	6
2.1.1	<i>Regulamento Geral sobre a Proteção de Dados (RGPD)</i>	8
2.1.2	OpenEHR	14
3	METODOLOGIAS E TECNOLOGIAS	17
3.1	Design Science Research Methodology	17
3.2	SCRUM	18
3.2.1	Equipa	20
3.2.2	Artefactos	21
3.2.3	Sprint	22
3.3	Tecnologias de desenvolvimento	23
3.3.1	Comunicações com <i>Javascript</i>	23
3.3.2	Tecnologias de virtualização	28
3.4	Metodologia de avaliação	31
3.4.1	Avaliação de Performance	31
3.4.2	Análise SWOT	32
4	PLATAFORMA DE COMUNICAÇÃO	34
4.1	Identificação do problema	34
4.2	Requisitos Funcionais	35
4.3	Requisitos técnicos	35
4.4	Arquitetura da Solução	36
4.4.1	Ambiente de Desenvolvimento	37
4.4.2	Ambiente de Produção	45
5	RESULTADOS, DISCUSSÃO E AVALIAÇÃO	52
5.1	Avaliação de Performance	52
5.2	Análise SWOT	55
6	CONCLUSÃO E TRABALHO FUTURO	59

---

## LISTA DE FIGURAS

---

Figura 1	Grupos de alterações	9
Figura 2	Arquitetura OpenEHR	14
Figura 3	Relações entre os componentes-chaves de cada modelo no OpenEHR.	15
Figura 4	Metodologia Design Science Research. Adaptado de Peffers 2008	17
Figura 5	<i>framework</i> SCRUM	20
Figura 6	Funcionamento do <i>XHR</i>	23
Figura 7	Funcionamento do Web Messaging / PostMessage	24
Figura 8	Funcionamento dos WebSockets	25
Figura 9	Funcionamento do Server Sent Events	25
Figura 10	Vantagens do desenvolvimento baseado em componentes	28
Figura 11	Grupos de alterações	29
Figura 12	Grupos de alterações	30
Figura 13	Arquitetura da solução	36
Figura 14	Arquitetura da solução em simulação local	38
Figura 15	Página <i>web</i> com formulário de <i>login</i>	39
Figura 16	Página <i>web</i> com formulário de <i>login</i> preenchido	39
Figura 17	Página <i>web</i> com apresentação das notificações	40
Figura 18	Página <i>web</i> com apresentação das notificações, colocando uma em pausa devido ao <i>hover</i> do ponteiro do rato	41
Figura 19	Página <i>web</i> com apresentação das notificações, exceto a que o utilizador efetuou o <i>click</i>	41
Figura 20	Página <i>web</i> com apresentação das notificações, exceto a que o utilizador efetuou o <i>click</i> anteriormente	42
Figura 21	<i>Insert query</i> para a introdução de uma nova mensagem relativo a um utilizador	43
Figura 22	Página <i>web</i> com apresentação das mensagens, incluindo a nova mensagem inserida	43
Figura 23	Páginas <i>web</i> com formulário de autenticação para dois utilizadores	44
Figura 24	Páginas <i>web</i> com apresentação das mensagens atribuídas a cada utilizador	44

Figura 25	Arquitetura da solução em ambiente de produção	45
Figura 26	Página <i>web</i> com fomulário de <i>login</i>	47
Figura 27	Página <i>web</i> com fomulário de <i>login</i> preenchido	48
Figura 28	Página <i>web</i> com apresentação das notificações, colocando uma em pausa devido ao <i>hover</i> do ponteiro do rato	49
Figura 29	Página <i>web</i> com apresentação das notificações	49
Figura 30	Página <i>web</i> com apresentação das notificações, exceto a que o utilizador efetuou o <i>click</i>	50
Figura 31	Página <i>web</i> com apresentação das notificações, incluindo a nova notificação inserida	50
Figura 32	Página <i>web</i> com apresentação das mensagens com os comportamentos definidos perante a autenticação de vários utilizadores	51

---

## LISTA DE TABELAS

---

Tabela 1	Comparação de tecnologias de comunicação em Javascript	27
Tabela 2	Comparação entre resultados obtidos e os parâmetros estabelecidos no capítulo 3	54
Tabela 3	Análise SWOT	56



---

## INTRODUÇÃO

---

Esta dissertação tem como principal propósito descrever o desenvolvimento de uma ferramenta que possibilite a comunicação dos processos destinados aos intervenientes no sistema, sendo que estes processos são criados através do OpenEHR. Esta ferramenta têm um papel fundamental nestes sistemas dado que através dela é possível que sejam criados diversos processos possibilitando a existência de um sistema comum a todas as unidades de saúde. Estes processos podem ser reutilizados diversas vezes o que facilita a utilização do sistema.

Este trabalho foi realizado em parceria no *Departamento de Informática (DI)* da *Universidade do Minho (UM)* no contexto do curso superior de *Mestrado em Engenharia Informática (MEI)*.

De forma a estruturar o que foi realizado ao longo deste documento, este capítulo inicial foca-se em abordar temas relevantes que contribuem para a melhor perceção do que foi abordado ao longo de todo o documento. Desta forma este capítulo encontra-se estruturado da seguinte forma:

- Em que contexto surgiu o trabalho proposto assim como quais foram as motivações que originaram o desenvolvimento do mesmo;
- Quais as principais questões que se pretendem resolver no sistema, dito isto o que se pretende resolver na plataforma com o desenvolvimento desta ferramenta assim como quais os objetivos de forma a responder às questões levantadas;
- A estrutura global do documento.

### 1.1 CONTEXTO

Nos dias atuais existiu a necessidade de se criar sistemas informáticos que permitissem auxiliar determinados profissionais nas suas áreas, nomeadamente os Sistemas de Informação Hospitalares (*HIS*).

Um dos principais propósitos destes sistemas é auxiliar os diversos profissionais na área da saúde. Um exemplo deste auxílio é na tomada de decisão dos profissionais com a apresentação de informações importantes dado que para um dado paciente pode existir

uma grande quantidade de informação, que foi coletada previamente, e com estes sistemas é possível efetuar uma análise mais rápida sobre um possível diagnóstico em caso de urgência [1]. Outro exemplo de auxílio é a tentativa de minimizar erros que existiam na área da saúde até ao surgimento destes sistemas, como por exemplo perder um registo médico ou não ter sido efetuado o registo médico. O surgimento destes sistemas também permitiu que fosse possível que os profissionais de saúde poupassem tempo na inserção dos dados relativos a um tratamento ou a uma consulta médica entre outros. Para além disto e algo que é considerado das funcionalidades mais importantes neste tipo de sistemas é o facto de os profissionais de saúde possuírem a informação dos pacientes centralizada permitindo que possa ser acedida de qualquer local, possuindo normalmente restrições.

De forma a que possam existir estes sistemas existem questões que devem ser aprofundadas e estudadas dado que é uma área onde existem muitas questões interligadas, como é o caso de questões legais entre outras. De forma a tentar perceber melhor estas questões foram analisados sistemas semelhantes que descrevam algumas destas questões. Além disso também foram exploradas questões que poderiam não estar em sistemas estudados, como é o caso do *RGPD*, dado que são questões que se encontram implementadas recentemente.

Após o estudo das questões relacionadas com este tipo de sistemas é apresentada uma plataforma que permite a criação de uma sistema na área da saúde, nomeadamente o OpenEHR [2]. O OpenEHR permite realizar o que de facto se passa neste tipo de sistemas, dito isto, permite efetuar operações sobre registos electrónicos de saúde (*EHR*) particularmente permite gerir, armazenar, recuperar e trocar os dados de saúde através destes registos.

Como para estes sistemas existe também a necessidade da constante troca de informações entre um cliente e o servidor, como verificado anteriormente, também é apresentado uma análise sobre algumas tecnologias existentes que permitem a realização destas comunicações sendo efetuado no final uma análise comparativa de forma a selecionar a que melhor se enquadra no desenvolvimento desta componente neste tipo de sistemas.

Por último de forma a decidir qual poderá ser a melhor arquitetura que poderá existir neste tipo de sistemas também é efetuada uma comparação na parte que diz respeito à integração dado que este tipo de sistemas são sistemas que devem estar disponíveis a quase 100% durante todos os dias do ano dado que são sistemas que tratam e trabalham com informação altamente importante e crítica e sem estes sistemas alguns profissionais de saúde poderão não efetuar o seu trabalho. Assim sendo será analisada qual a melhor metodologia, baseada em um só componente ou baseada em vários através da utilização de subsistemas, de forma a que seja possível lidar da forma mais eficiente possível com o problema descrito anteriormente. Após efetuada esta análise são comparadas as ferramentas mais utilizadas para o tipo de metodologia estudada e que vai de encontro com a necessidade destes sistemas, sendo as duas comparadas no final de forma a poder indicar aquela que à partida oferece a melhor solução.

## 1.2 MOTIVAÇÃO

Um dos principais fatores que contribuíram para a seleção deste tema para esta dissertação foi o facto de existir uma enorme vontade e curiosidade de realizar um projeto informático que se enquadrasse na área da saúde. Esta curiosidade provém do facto de ser uma área sobre a qual existia um grande interesse de perceber como funciona devido a possuir algumas restrições, nomeadamente privacidade de dados e a forma como são tratados. Para além disto o facto de serem utilizadas ferramentas que se encontram em crescimento em termos de utilização contribuíram para o interesse.

Além das questões relacionadas com o projeto o que também contribuiu para a motivação foi o facto de os orientadores me terem apresentado o tema de uma forma que cativou o meu interesse desde o primeiro momento e por se demonstrarem sempre disponíveis para as dúvidas que tinha acerca da seleção do tema.

O que teve também um peso importante para a minha motivação para este tema foi o facto de o mesmo ser desenvolvido através de um centro de investigação permitindo assim que o projeto fosse realizado através de um ambiente profissional dado que foi demonstrado desde o primeiro momento a intenção de integrar o que eu iria realizar.

## 1.3 CONTRIBUIÇÕES E OBJETIVOS

De forma a ser possível responder a algumas das questões levantadas sobre o sistema que utiliza o OpenEHR surgiu a ferramenta proposta. A principal questão levantada é o facto de o OpenEHR permitir que sejam criados processos e sejam atribuídos aos profissionais de saúde mas não possuir um mecanismo que permita comunicar aos utilizadores em questão quais os processos que os mesmos possuem aquando a sua interação com o sistema. Dito isto, se os utilizadores, neste caso os profissionais de saúde, não possuem conhecimento relativo aos processos que possuem e que têm que realizar os mesmos não podem realizar o seu trabalho. Por esta razão, a criação de uma ferramenta semelhante à que se encontra descrita ao longo deste documento passa a ter um papel fundamental nos sistemas da área da saúde que utilizem a ferramenta OpenEHR.

Estando o principal sistema, com a utilização do OpenEHR, a ser desenvolvido na Universidade do Minho, pelo centro de investigação Algoritmi, é então uma das prioridades deste centro que os profissionais de saúde possam ser alertados de forma instantânea e em conformidade com o *RGPD*, acerca dos processos que possuem e desta forma conseguir realizar os seus trabalhos com a maior eficiência possível através da utilização deste sistema.

Tendo em consideração o que foi abordado anteriormente é fulcral obter respostas perante as questões levantadas ao longo desta dissertação, nomeadamente com a utilização do OpenEHR neste sistema implica que exista uma ferramenta que seja capaz de realizar a

comunicação, através da utilização de *Javascript*, entre sistema e os utilizadores sendo que a informação a ser disponibilizada deve ter em consideração restrições impostas pelo *RGPD*.

Outra questão a ter em consideração é o facto de se pretender que as comunicações sejam efetuadas de forma instantânea, dito isto sejam efetuadas em tempo-real.

Para além das anteriores também o facto de ser pretendido a integração desta ferramenta no sistema qual será a melhor maneira de realizar a integração sem comprometer o principal sistema de informação hospitalar.

De forma a realizar a resolução destas questões foi fulcral a definição de objetivos que vão de encontro com cada questão. Desta forma de forma a perceber esses objetivos, os mesmos foram divididos de forma a que fosse facilitada a execução dos mesmos:

- Numa primeira fase foi realizado um estudo sobre sistemas de informação hospitalares já existentes de forma a que fosse possível a compreensão geral destes sistemas. Para além disto também foi realizado um estudo sobre as ferramentas que se encontram a ser utilizadas assim como das ferramentas que se podem utilizar no desenvolvimento da ferramenta de comunicação entre sistema e utilizador. Desta forma é possível averiguar se será possível desenvolver uma ferramenta que vá de encontro ao pretendido.
- Numa segunda fase foi fulcral definir quais os requisitos funcionais que a solução deve cumprir de forma a alcançar o que se pretende. Posto isto foram realizados testes às tecnologias selecionadas e à arquitetura desenhada. Para a realização destes testes foi simulado localmente um ambiente em que esta ferramenta viria a atuar de forma a ser possível averiguar se é possível obter a comunicação de forma instantânea.
- Por último a implementação da solução obtida em ambiente real de forma a que seja possível averiguar se a integração pode ser realizada sem que exista o risco de comprometer todo o sistema.

#### 1.4 ESTRUTURA DO DOCUMENTO

De forma a obter uma melhor compreensão acerca do documento o mesmo encontra-se dividido em 6 capítulos:

- No capítulo 1, é realizada uma introdução ao tema em questão, sendo abordados assuntos relacionados e importantes de forma a obter uma melhor compreensão acerca do projeto. Para além disto é apresentado o contexto, as motivações, quais as questões que a ferramenta desenvolvida pretende resolver assim como quais os objetivos que possui a capacidade de responder aos problemas levantados;

- No segundo capítulo são apresentados os conceitos relevantes para o desenvolvimento, nomeadamente Sistemas de Informação Hospitalar (*HIS*), *Regulamento Geral sobre a Proteção de Dados* (*RGPD*) e o openEHR;
- Relativamente ao terceiro capítulo é apresentada a metodologia de investigação que foi seguida assim como a de avaliação relativamente ao desenvolvimento do projeto e das tecnologias utilizadas, nomeadamente comunicações utilizando *Javascript* e tecnologias de virtualização.
- No capítulo 4 é apresentada a solução desenvolvida assim como a descrição de alguns aspetos considerados relevantes aquando o desenvolvimento da ferramenta, nomeadamente requisitos funcionais e técnicos, as arquiteturas utilizadas quer na simulação local quer na implementação e as funcionalidades da ferramenta.
- No quinto capítulo surge a apresentação dos resultados relativos à implementação da ferramenta em ambiente real tendo sido realizada a sua discussão e avaliação.
- No último capítulo é apresentada a conclusão da dissertação e onde é apresentado trabalho futuro.

---

## ESTADO DE ARTE

---

### 2.1 SISTEMAS DE INFORMAÇÃO HOSPITALAR - *HIS*

Os Sistemas de Informação Hospitalar (*HIS*) baseiam-se num elemento de informação médica que tem como principal objetivo concentrar-se nas necessidades administrativas dos Hospitais. Dito isto, basicamente, é um sistema de informação bastante abrangente e integrado que permite gerir algumas das operações de um hospital, nomeadamente, questões médicas, administrativas entre outras. Este sistema também pode ser denominado por Hospital Management Software (*HMS*) ou Hospital Management System.

Um exemplo de uma funcionalidade que pode estar presente nestes sistemas é a capacidade dos profissionais de saúde conseguirem coordenar os cuidados de saúde de um dado paciente, uma vez que estes sistemas são capazes de apresentar somente informações importantes perante uma grande quantidade de dados de saúde relativos a um paciente [1]. Dito isto, são fornecidas informações de saúde do paciente assim como quais os cuidados e horários que devem ser aplicados, por exemplo. Para além disto ainda podem existir exames com resultados visuais, como por exemplo raio-x, que devem poder ser consultados pelos profissionais sendo que para tal nestes sistemas são utilizadas comunicações internas e externas.

Como se verifica pelo exemplo anterior estes Sistemas de Informação Hospitalar(*HIS*) possuem normalmente um histórico relativo à saúde de um paciente. Assim sendo estes dados devem ser armazenados em um local seguro e deve ser controlado quem pode aceder a esses dados dependendo das circunstâncias em que serão apresentados [3].

Como referido anteriormente neste tipo de sistemas é comum existir uma grande quantidade de informação relativa a dados de cada paciente, dado que sempre que este se desloca a uma unidade de saúde quer seja para uma consulta ou tratamento ou mesmo quando é realizado um novo exame são introduzidos dados no sistema que o profissional de saúde que se encontra com o paciente ache relevantes.

Devido à existência de uma elevada quantidade de informações nestes sistemas surgem diversas dificuldades para quem se encontra utilizar o sistema, nomeadamente os profissionais de saúde. Isto ocorre devido ao facto de existir muita informação, sobre o utente,

e o profissional de saúde poderá demorar imenso tempo a encontrar o que pretende ou até mesmo poderá acabar por não encontrar podendo em alguns casos a obtenção desta informação ser de carácter urgente. Assim sendo surge a dificuldade dos profissionais de saúde encontrarem o que necessitam com facilidade e rapidez. De forma a tentar combater este problema nas diversas áreas de saúde começaram a ser desenvolvidas e utilizadas diversas técnicas desde filtragem de dados, auxílio de diagnóstico, entre outras.

Começando pela filtragem de dados em alguns casos é utilizada uma técnica de filtragem colaborativa, baseado em *deep learning*, de forma a ser possível a realização de uma filtragem sobre toda a informação existente e assim recomendar a informação considerada relevante, possibilitando que um profissional de saúde seja capaz de encontrar de forma mais rápida e com maior facilidade os dados pretendidos para um dado paciente [4].

Indo de encontro com as necessidades de algumas áreas de saúde surgiram mecanismos que possibilitam auxiliar na prevenção e diagnóstico de determinadas doenças. Normalmente estes mecanismos são baseados em *Data Mining* e possuem um papel preponderante porque normalmente são utilizados para doenças que nos seus estágios iniciais podem ser assintomáticas e como tal torna-se difícil a sua deteção [5][6]. Esta técnica, *Data Mining*, também pode ser utilizada para o auxílio na tomada daquilo que poderá ser a melhor decisão. Para tal é importante tentar perceber futuros eventos antes dos mesmos ocorrerem, por exemplo perceber se um utente deverá frequentar um determinado especialista com base num conjunto de dados específicos [7].

Além da utilização de *Data Mining* podem ser utilizadas outras técnicas, baseadas em Redes Neurais Artificiais, por exemplo, dado existirem outras doenças em que o seu diagnóstico pode ser de dificuldade bastante elevada uma vez que os sintomas associados podem referir-se a outras doenças e como tal estes sistemas são vistos como mais valias para o auxílio na deteção destas doenças [8].

Embora existam inúmeras técnicas que poderiam ser integradas nestes sistemas é fulcral a realização de uma avaliação sobre as técnicas que se pretendem integrar porque na área da saúde a qualidade dos dados é extremamente importante dado que os profissionais de saúde realizam determinadas tomadas de decisão baseadas nas técnicas utilizadas e como tal estas técnicas devem fornecer informações que sejam aceites pelos próprios profissionais de saúde mas que acima de tudo sejam confiáveis e seguras [9].

Em suma, a utilização destes sistemas agregando algumas das técnicas existentes permite que sejam vistos como uma mais valia para a área da saúde dado que possuem fácil usabilidade, permitem auxiliar os profissionais de saúde e ainda permite que possam ser eliminados erros que normalmente eram comuns, por exemplo através da utilização de manuscritos.

### 2.1.1 Regulamento Geral sobre a Proteção de Dados (RGPD)

Com a introdução do Regulamento Geral de Proteção de Dados (*RGPD*) (*UE*) 2016/679 [10], que começou a ser implementado a partir de 2018 na União Europeia e no Espaço Económico Europeu (*EEE*), passaram a existir restrições relativas à privacidade e à proteção de dados pessoais.

Devido ao facto de o Regulamento Geral sobre a Proteção de Dados se tratar de um instrumento que tem como principal propósito reforçar os direitos dos titulares dos dados implica a que as Organizações ampliem as suas obrigações em prol da privacidade.

Assim sendo com a introdução deste regulamento passaram a existir novas definições, como “violação de dados pessoais” ou “limitação de tratamento”, assim como o princípio da transparência e o princípio da responsabilidade [11]. Dito isto, é esperado que os dados recolhidos pelas Entidades sejam os mínimos face à finalidade para os quais irão ser utilizados e tratados, sendo que o tratamento será da efetiva responsabilidade pelo responsável para o tratamento.

Com a introdução do *RGPD* foi criado um modelo que procura obrigar as Entidades a ter em consideração as preocupações e os riscos de privacidade dos dados dos utilizadores desde o momento inicial de conceção de um dado projeto e não numa fase mais posterior.

Tendo em conta o facto anterior passam a ser utilizados princípios como o *Privacy by Design* e *Privacy by Default* [12]. Estes princípios defendem que em qualquer sistema digital ou processo de negócio deve ser procurado implementar o mais alto nível de privacidade assegurando assim a proteção dos dados de forma automática (*Privacy by Default*). Caso o titular dos dados não realize nenhuma ação para proteger a privacidade deve ser assegurado pelo sistema essa mesma proteção (*Privacy by Default*).

Devido às alterações impostas pelo *RGPD* é fulcral destacar-se quais as obrigações que se aplicam na área da saúde, nomeadamente nas Entidades integrantes do *SNS* [13]. Assim sendo estas Entidades poderão ter de realizar novas obrigações ou realizar alterações das obrigações já existentes tentando encontrar-se em conformidade com o regime legal em vigor atualmente.

A determinação das medidas a serem utilizadas devem ser desenhadas e implementadas tendo por base os requisitos de cada uma das Entidades do *SNS*. Para além disso também se deve ter em conta quais os modelos de governança e gestão do sistema de informação da saúde (*eSIS*) [14] que se encontram em vigor.

Após a investigação realizada é possível verificar que as alterações com a introdução do *RGPD* podem ser agrupadas sobretudo em 5 grupos diferentes, como se verifica através da seguinte figura:



Figura 1: Grupos de alterações

No primeiro grupo, Administração dos dados, é na fase que defende que todas as Organizações devem possuir um conjunto de medidas de forma a reduzir o risco de incumprimento das regras de privacidade e proteção dos dados pessoais. Assim sendo as medidas a serem implementadas serão as seguintes:

- **Implementação dos conceitos “*privacy by design*” e “*privacy by default*”:** Esta medida defende que as Entidades integrantes do *SNS* devem respeitar em todas as operações o princípio de “*privacy by design*”, que basicamente consiste na preocupação de garantir que a privacidade do utilizador estará sempre assegurada desde a conceção de um novo produto, serviço ou projeto. Outro princípio a ser tido em consideração é o “*privacy by default*” que defende que as Entidades devem respeitar a minimização, pseudominização e transparência. Dito isto, que apenas a quantidade de dados necessária é recolhida, utilizada e conservada de acordo com a tarefa para a qual se destinam. Para além disto também deve ser assegurado qual o tempo que os mesmos serão mantidos.
- **Realização de “*privacy impact assessment*” e consulta prévia à *CNPD*:** O “*privacy impact assessment*” (*PIA*) [15] garante que as Entidades integrantes do *SNS* devem realizar uma avaliação do impacto da realização de operações sobre a proteção de dados pessoais. Resumidamente o que é efetuado através do *PIA* é a avaliação e identificação de riscos associados a uma operação para a proteção dos dados de forma a que seja possível as Entidades integrantes do *SNS* adotarem medidas que permitam minimizar, contornar ou eliminar os riscos identificados. As Entidades integrantes do *SNS* devem consultar a *CNPD* [16] quando for realizado o tratamento dos dados

considerados de risco de forma a que seja possível garantir o cumprimento das disposições do *RGPD*. Tanto a realização de impacto como a consulta à *CNPD* devem ser efetuadas tendo em conta determinados requisitos com a inclusão de informações e ações mínimas que se encontram no *RGPD*.

- **Designação de Data Privacy Officer:** As Entidades integrantes do *SNS* passam a estar obrigadas a designar um encarregado de proteção de dados (“*Data Privacy Officer*”) dado pertencerem ao organismo público. Este encarregado passará a estabelecer contacto com a *CNPD*, titulares dos dados e será o responsável por responder a todas as questões sobre a proteção dos dados pessoais. De entre as várias responsabilidades delineadas no *RGPD* destacam-se sobretudo a monitorização do cumprimento das regras de proteção de dados pessoais, a gestão e registo de toda a documentação relevante e deverá acompanhar os projetos que possam ter impacto na privacidade dos dados.
- **Realização de auditorias de conformidade e adoção de políticas:** De forma a que seja possível as Entidades integrantes do *SNS* demonstrar que se encontram a cumprir as regras do *RGPD* podem incluir medidas que incluam a realização de auditorias, a elaboração e implementação de políticas e procedimentos internos que devem ser realizados por toda a Organização.
- **Registo de atividades de tratamento:** Todas as Entidades do *SNS* encontram-se obrigadas a manter um registo acerca das atividades de tratamento realizadas sob a sua responsabilidade. Para tal devem ser mantidas informações como o tipo de dados tratado, a finalidade, medidas de segurança e prazo de conservação. Caso o tratamento seja efetuado por subcontratantes também estes devem manter um registo de todas as atividades de tratamento realizadas em nome das Entidades integrantes do *SNS*.
- **Maior responsabilização na escolha de entidades externas:** Com o *RGPD* as Entidades passaram a ter uma maior responsabilidade aquando a escolha dos subcontratantes, passando a existir obrigações claras e precisas. Para além disto também os subcontratantes passam a ter uma série de obrigações, inclusive sanções, que somente eram impostas aos responsáveis pelo tratamento. Relativamente à relação entre o responsável e subcontratante continua a ser obrigatório a celebração de um contrato ou documento que regule a relação contratual entre ambos assim como os termos do tratamento. Desta forma o subcontratante passa a assumir mais responsabilidades de forma direta e desta forma possui um maior conjunto de obrigações.

Relativamente ao segundo grupo, Consentimento, indica que o responsável pelo tratamento dos dados deve ser capaz de demonstrar que o titular dos dados deu o seu consentimento de forma livre, específica, informada e explícita. Caso este consentimento

seja dado através de um documento escrito que também possua outros assuntos, a parte do consentimento deve estar assinalada de forma evidente e numa linguagem clara e de fácil acesso. Assim sendo não é permitido o consentimento implícito (por exemplo caso o titular acesse ou navegue num portal/site), dito isto as Entidades devem rever os mecanismos de consentimento, nomeadamente online para a utilização de cookies [17]. Após o consentimento dos dados o titular possui igualmente o direito de retirar o seu consentimento a qualquer momento, dito isto o direito ao esquecimento.

Os dados médicos podem ser denominado por “dados sensíveis” contudo para este tipo de dados não foram introduzidas alterações específicas pelo *RGPD* para o tratamento dos mesmos. Contudo para este conjunto de dados o *RGPD* denomina o tratamento destes dados como tratamento de categorias especiais de dados pessoais. Dito isto, esta denominação é utilizada caso o tratamento for necessário para efeitos de medicina preventiva, para o diagnóstico médico, a prestação de cuidados ou tratamentos de saúde ou de ação social ou a gestão de sistemas e serviços de saúde ou ação social.

Os dados médicos podem ser tratados para os fins enunciados anteriormente, contudo só deve ser efetuado por um profissional ou sob a responsabilidade de um profissional sujeito à obrigação de sigilo profissional ou por outra pessoa que se encontra igualmente sujeita a uma obrigação de confidencialidade.

Sem alterar o que foi abordado anterior o Regulamento permite que os Estados-Membros mantenham ou imponham novas condições, incluindo limitações, no que respeita ao tratamento de dados genéticos, dados biométricos ou dados relativos à saúde [18].

Já o terceiro grupo, Direitos dos titulares, possui essencialmente dois direitos que devem ser respeitados sendo eles o reforço do direito de informação e de acesso dos titulares dos dados e a garantia dos direitos de apagamento dos dados, limitação do tratamento e portabilidade.

No primeiro subgrupo é defendido que devem ser estabelecidos requisitos mais exigentes a aplicar sobre a informação a prestar ao titular. Um exemplo do que foi enumerado anteriormente é a obrigação dos responsáveis pelo tratamento dos dados disponibilizarem mais informações de forma mais transparente e acessível.

Assim de forma a cumprir este direito devem ser definidos e utilizados mecanismos que possibilitem agilizar o exercício dos direitos dos titulares.

Para além disto um novo direito introduzido pelo *RGPD* é a consagração do direito de informação dos titulares dos dados em relação aos destinatários dos dados. Dito isto, o titular dos dados tem o direito de ser informado sobre quem irá tratar os seus dados e/ou a quem serão transmitidos.

Além do frisado anteriormente as Entidades referentes ao *SNS* também devem prestar informação adicional, tal como qual o fundamento jurídico para o tratamento dos dados, o

prazo de conservação dos dados e o direito de apresentação de reclamações às autoridades competentes, nomeadamente a *CNPD*.

Caso os dados recolhidos provenham de outros meios que não recolhidos diretamente com o titular dos mesmos as Entidades integrantes do *SNS* devem ter em consideração que existem obrigações específicas de informação para estes casos em concreto.

Relativamente ao direito do acesso aos dados por parte dos titulares dos mesmos, as Entidades integrantes do *SNS* estão obrigadas a dar uma resposta a um pedido de acesso por parte do titular ou então fornecer-lhe informações relativas às medidas que se encontram a ser utilizadas para o tratamento dos seus dados pessoais sendo que esta resposta deve ser dada sem uma demora injustificada e deve cumprir o prazo máximo de um mês a contar imediatamente após a data de receção do pedido.

Já no segundo subgrupo desta secção é defendido que os utilizadores possuem o direito de limitação do tratamento, portabilidade dos seus dados assim como devem possuir o direito a que os seus dados sejam apagados caso assim ele o pretenda.

Dito isto, existem essencialmente três direitos que defendem estes princípios que são:

- **Direito a ser esquecido:** Defende que após um pedido de eliminação de dados caso se verifiquem todas as condições previstas no *RGPD* as Entidades integrantes do *SNS* devem utilizar mecanismos que garantam que efetivamente todos os dados foram eliminados, incluindo cópias ou reproduções dos mesmos. Este direito poderá não ser concretizado na medida em que o tratamento destes dados esteja a ser realizado em prol do interesse público no domínio da saúde pública, como previsto nos termos do *RGPD* [19].
- **Direito à limitação do tratamento:** Com este direito é previsto que o titular dos dados possa opor-se à eliminação dos seus dados e solicitar a limitação referente ao tratamento dos seus dados. Assim sendo as Entidades integrantes do *SNS* devem ter em consideração que devem comunicar a todas as Entidades terceiras que forneceram os dados do titular quais as limitações do tratamento que foram impostas.
- **Direito à portabilidade dos dados:** Através deste direito o utilizador passa a possuir o direito a receber todos os seus dados pessoais através de um formato estruturado e com uso corrente e que possibilite uma leitura automática. Contudo caso se pretenda transmitir os dados de um responsável para outro este processo deve ser efetuado através de meios automatizados, ou seja os dados devem ser transmitidos através de um sistema eletrónico de forma direta de um responsável para outro. Este processo deve ser efetuado sempre que seja tecnicamente possível.

Relativamente ao quarto grupo, Segurança, verifica-se que é um dos principais temas que com a introdução do *RGPD* em vigor tem uma maior ênfase. Dito isto é de prever que com este regulamento exista uma maior responsabilidade sobre a decisão de medidas a adotar

de forma a que seja possível proteger os dados pessoais, sobretudo contra acessos indevidos. Esta responsabilidade deve então ser dividida quer pelo responsável pelo tratamento dos dados quer pelo subcontratante do mesmo.

Assim sendo algumas das medidas técnicas e organizativas que podem ser tidas em conta de forma a assegurar a segurança tendo em conta os diferentes níveis de risco associadas são:

- **Pseudonimização [20] e a encriptação dos dados pessoais;**
- **A capacidade de assegurar a confidencialidade, integridade, disponibilidade e a resiliência dos sistemas e serviços de tratamento;**
- **No caso de incidente físico ou técnico deve existir a capacidade de restabelecer a disponibilidade e o acesso aos dados pessoais de forma atempada;**
- **Deve existir um processo que possa testar, apreciar e avaliar regularmente a eficácia das medidas técnicas e organizativas de forma a garantir a segurança aquando o tratamento dos dados.**

Caso exista violação dos dados pessoais, denominados *data breaches*, também neste grupo é defendido que deve existir a obrigação de notificar a violação à *CNPD*. Esta notificação pode ser ignorada no caso de a violação praticada não resultar em nenhum risco para os direitos e liberdades por parte dos titulares dos dados, sendo que esta avaliação terá de ser efetuada tendo em conta cada caso.

Desta forma, caso exista a violação de dados pessoais, as Entidades integrantes do *SNS* devem proceder à notificação, sem demora injustificada e sempre que possível até 72 horas após tomado o conhecimento desta violação. Para além disto as Entidades integrantes do *SNS* também devem documentar todas as violações ocorridas e informação acerca das mesmas. Perante esta documentação em conjunto com a *CNPD* será analisado o cumprimento das regras previstas no *RGPD*.

Caso a violação ocorrida possa representar um elevado risco para os direitos e liberdades dos titulares também estes devem ser notificados pelas Entidades integrantes do *SNS*.

Para além da restrição do *RGPD* quanto à segurança de forma a proteger os dados também a diretiva *SRI* se encarrega da proteção e segurança dos dados [21]. Esta diretiva impõe um conjunto de medidas de forma a garantir um elevado nível comum de segurança das redes e da informação nomeadamente no sector da saúde. Deste modo as Entidades que pertençam ao sector da saúde possuem ainda a obrigação de comunicar ocorrências de incidentes que possuam um impacto significativo na segurança das redes e dos sistemas de saúde.

Quanto ao último grupo, sanções, defende que com a introdução deste regulamento foi reforçado sobretudo o poder de fiscalização sobre a proteção dos dados, nomeadamente em Portugal da *CNPD*. Assim sendo caso sejam efetuadas violações de dados ou outras

operações ilegais sobre os dados o *RGPD* estabelece sanções que podem ser aplicadas sobre as Entidades.

As sanções que se encontram atualmente previstas são divididas essencialmente em dois grupos, os casos menos graves e os casos mais graves. Assim sendo para os casos menos graves é prevista uma coima até ao valor de 10 milhões de euros ou 2% do volume de negócios anual. Já no caso das mais graves a coima pode ser até 20 milhões de euros ou 4% do volume de negócios anual. Estes valores são escolhidos tendo em conta aquele que é mais alto [22].

### 2.1.2 OpenEHR

O openEHR [2] é uma especificação “open standard” na área da informática na saúde que descreve a gestão, armazenamento, recuperação e troca de dados de saúde em Registos de Saúde Eletrónicos (Electronic Health Records - *EHR*).

Os dados relacionados com saúde no openEHR são armazenados em “one lifetime” sendo independentes do local de onde provém e são centrados na pessoa em questão.

As especificações no openEHR incluem uma especificação *EHR* Extract [23]. Contudo estas especificações não possuem principal propósito a troca de dados entre sistemas pois para este propósito já existem outras normas, nomeadamente a HL7 e a EN 13606 [24]. Assim sendo as especificações do openEHR incluem informações e modelos de serviço para Registos de Saúde Eletrónicos (*EHR*), dados demográficos e para o fluxo de trabalho clínico.

Relativamente à arquitetura das especificações do openEHR pode ser representada de acordo com a figura seguinte:

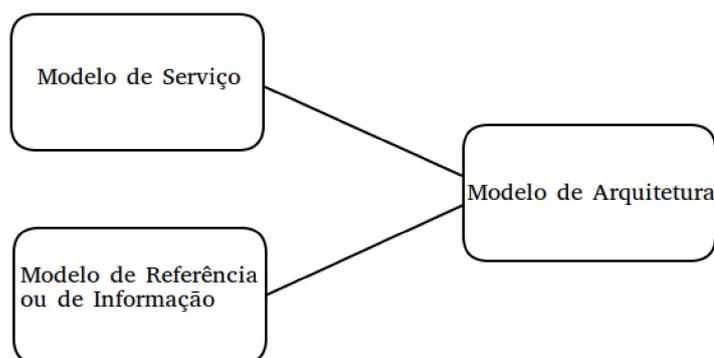


Figura 2: Arquitetura OpenEHR

Efetuada uma análise à Figura 2 verifica-se que a arquitetura do openEHR divide-se essencialmente em 3 elementos chave sendo eles os seguintes:

1. Modelos de serviço/*API's*, que inclui definições básicas de serviços sobre informações na área da saúde, centralizados no openEHR.
2. Modelo de Arquitetura, onde estão contidos os modelos necessários para descrever a semântica dos arquétipos e templates assim como o seu comportamento dentro do openEHR.
3. Modelo de referência ou de informação, onde é efetuada uma divisão informal em domínio, padrões e núcleo com o propósito de, em conjunto com outros pacotes externos, fornecer acesso a recursos de conhecimento, tipos e estruturas de dados, semântica de controlo de versões e suporte para o desenvolvimento de arquétipos.

Em cada um dos modelos enumerados anteriormente existem pacotes chaves se interligam entre eles através de relações. Para estes modelos as dependências organizam-se dos maiores para os menores como é possível verificar através da figura 3.

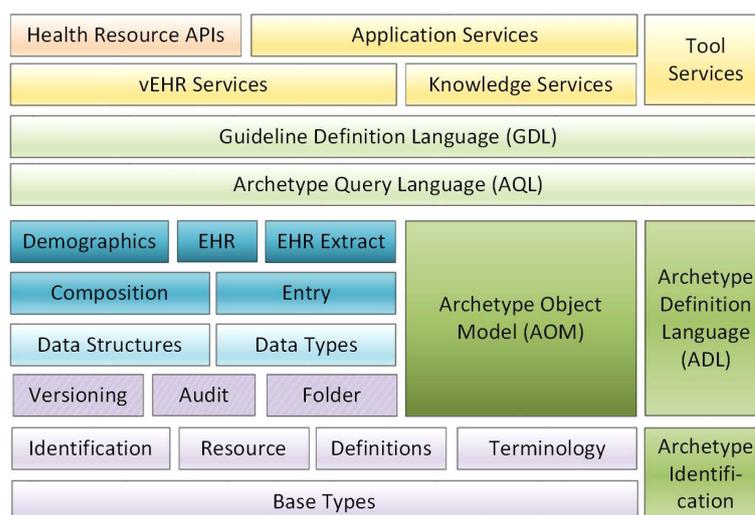


Figura 3: Relações entre os componentes-chaves de cada modelo no OpenEHR.

[25]

O modelo de arquitetura também pode ser dividido em dois elementos-chaves nomeadamente em formalismo arquétipo [26] e linguagem de consulta do arquétipo portátil.

Então depois de decomposta a arquitetura nos principais elementos é possível verificar com mais detalhe o que cada um dos mesmos realiza. Assim sendo verifica-se que o modelo de referência e o formalismo arquétipo permitem que sejam desenvolvidos modelos e arquétipos. Dito isto, permite que sejam criados modelos formais de conteúdo clínico e relacionado constituindo deste modo uma cadeia de padrões de fato própria sendo muito mais numerosos do que as especificações básicas em que foram construídos.

Relativamente à linguagem de consulta, esta permite que sejam efetuadas consultas/-queries com base nos arquétipos criados em vez de serem geradas queries de acordo com os esquemas das base de dados.

Por último nos modelos de serviço é onde é definido o acesso aos principais serviços de backend, sendo também incluído o Serviço *EHR* e o Serviço de Demografia. Para que seja possível realizar os acessos à plataforma são utilizadas *API's REST* baseadas nos arquétipos.

Seguindo a abordagem do openEHR à partida poderia ser possível a partilha de arquétipos o que possibilita que os dados possam ser constantemente manipulados e visualizados através do openEHR independentemente de questões étnicas, organizacionais ou culturais.

Para além disto a utilização desta abordagem permite que os modelos de dados atuais que utilizam *EHR* sejam flexíveis. Dito isto, possibilita que novos arquétipos possam ser definidos de forma a atender a futuras necessidades de manutenção sobre registos clínicos. Contudo atualmente ainda não foi possível atingir um consenso que permita que exista um sistema que possa ser utilizado de forma internacional principalmente devido a questões culturais ou técnicas como é o caso dos termos clínicos SNOMED [27] que devem integrar esse sistema.

---

 METODOLOGIAS E TECNOLOGIAS
 

---

Ao longo deste capítulo são apresentadas as metodologias de investigação e de avaliação que foram utilizadas para o desenvolvimento desta dissertação. Enquanto metodologia de investigação foi utilizado o método *Design Science Research*, enquanto que para a gestão do projeto foi utilizada a metodologia *SCRUM*.

Para além destas metodologias também ao longo deste capítulo são apresentadas as tecnologias que fazem parte da solução assim como a realização de análises comparativas entre as tecnologias.

### 3.1 DESIGN SCIENCE RESEARCH METHODOLOGY

O *Design Science Research* é uma metodologia de pesquisa de informação que abrange princípios, práticas e procedimentos com a intenção de realizar um estudo científico consistente e assim fornecer um modelo nominal para efetuar o estudo científico e um modelo mental para que seja possível apresentar e avaliar o estudo realizado nos sistemas de informação.

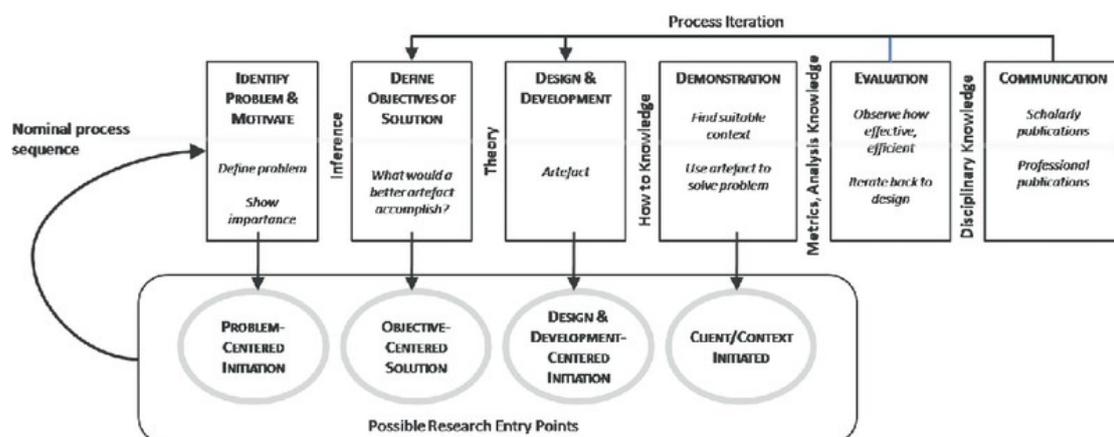


Figura 4: Metodologia Design Science Research. Adaptado de Peffers 2008 [28]

De acordo com a figura 4 verifica-se que esta metodologia se encontra dividida nas 6 secções seguintes:

- **Identificação do problema e da motivação** : Nesta secção deve ser definido o problema que levou a realizar a pesquisa de forma a justificar o valor de uma dada solução. Como se pretende desenvolver um artefacto com base no problema existente é fulcral o enquadramento de forma conceptual de forma a averiguar qual será a complexidade associada. Para esta fase foram identificadas as questões da investigação assim como a motivação para obter as respostas para este projeto.
- **Definir objetivos da solução** : Consiste na identificação dos objetivos de uma solução a partir do problema definido anteriormente sendo que se pretende obter o reconhecimento de que é possível e viável a solução encontrada. De forma a analisar é fulcral o conhecimento do estado atual do problema existente, das soluções que existem para problemas semelhantes assim como a viabilidade das mesmas e a eficácia de cada uma. Tendo isto em vista foi efetuado um levantamento de algumas tecnologias que são capazes de auxiliar na solução do problema.
- **Desenho e desenvolvimento** : Após determinada a solução segue-se o desenvolvimento do artefacto que contribua para o estudo realizado até ao momento. Desta forma após tomadas algumas decisões será fulcral a determinação das funcionalidades que se deseja ver implementadas no artefacto, assim como a arquitetura da solução e por último é que se passará à criação da mesma.
- **Demonstração** : Após o desenvolvimento do artefacto para a resolução de uma ou mais instâncias do problema torna-se uma parte importante a realização de experiências, simulações, casos de uso ou testes.
- **Avaliação** : Após a realização das secções anteriores é possível visualizar e avaliar a qualidade do artefacto desenvolvido na resolução do problema identificado. A qualidade pode ser obtida através da comparação dos resultados obtidos e dos objetivos reais, estabelecidos anteriormente.
- **Comunicação** : De forma a que seja possível criar uma estrutura comum para trabalhos empíricos de investigação a comunicação do problema e a relevância do artefacto desenvolvido assim como a utilidade, rigor e eficácia é relevante para os outros investigadores e público pertinente conseguir compreender todo o trabalho realizado.

### 3.2 SCRUM

Nos dias correntes de forma a ser possível organizar uma equipa para que seja possível realizar um trabalho mais produtivo e com melhor qualidade são utilizadas metodologias

que assim o permitem. Para este caso será utilizada uma metodologia ágil, designadamente o SCRUM, que de acordo com Sutherland [29] é uma abordagem que permite o desenvolvimento de software que possibilita que uma dada equipa selecione a quantidade de trabalho a realizar e que decidam qual a melhor abordagem para o realizar. Dito isto possibilita que não seja necessário realizar descrições completas e detalhadas do que deve ser realizado pois assume-se que a equipa sabe qual a melhor maneira de resolver o problema existente.

De acordo com a própria *framework*, [30], esta metodologia baseia-se no processo empírico, que basicamente utiliza uma abordagem incremental e iterativa de forma a otimizar a previsibilidade e o controlo de risco. Assim de forma a sustentar a implementação do processo empírico existem 3 pontos vitais:

- **Transparência** : De forma a evitar equívocos este ponto define que os aspetos importantes do processo devem ser visíveis para os responsáveis que controlam o resultado obtido. Estes equívocos devem ser evitados com a utilização de uma linguagem ou padrão que seja comum a todos os membros que se encontram envolvidos no processo.
- **Inspeção** : Para que seja possível verificar o progresso realizado em conformidade com os *sprints* delineados os utilizadores desta *framework* devem realizar análises regulares sobre esses artefactos de forma a detetar possíveis variações não desejadas.
- **Adaptação** : O processo ou material em produção poderá ter de ser ajustado no caso de se verificar que um ou mais aspetos do processo se desviaram além dos limites definidos o que poderá resultar num produto inaceitável. Assim que detetado este ajuste deve ser realizado o mais rapidamente possível e de forma cautelosa sem comprometer outros aspetos de forma a minimizar imprevistos adicionais.

Após isto verifica-se que esta *framework* é constituída por uma estrutura baseada em inspecionar o que vai sendo efetuado para o produto e ir adaptando de acordo com o que é realizado. Assim sendo é necessário nomear cargos, artefactos e momentos da seguinte maneira:

- Cargos: Product Owner, Team, SCRUM Master;
- Artefactos: *Product backlog*, *Sprint backlog* e *Burndown chart*;
- Momentos: Planeamento, revisão, retrospectiva e reunião diária do *sprint* a realizar.

## Scrum Process

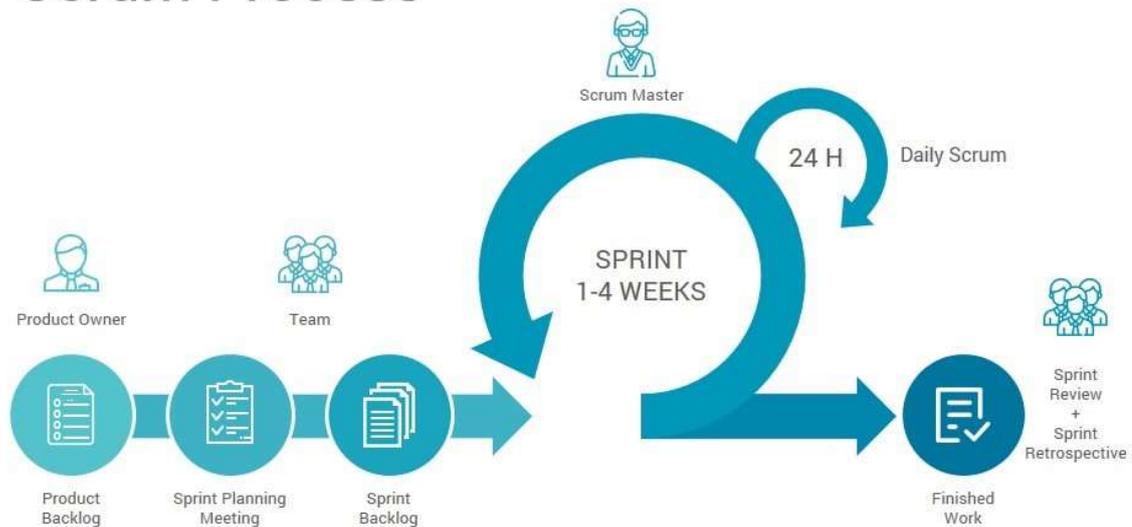


Figura 5: *framework* SCRUM

[31]

De acordo com a figura 5 é possível visualizar de que forma é que os intervenientes enunciados anteriormente se encontram interligados para que seja possível fornecer um produto funcionável com base na utilização de *sprints*.

### 3.2.1 Equipa

De acordo com Sutherland [29], uma equipa que utilize esta metodologia deverá ser auto-organizada e multi-funcional onde não deverá existir um líder de equipa que possua como responsabilidade de atribuir tarefas a uma dada pessoa ou de decidir como essa pessoa irá resolver um dado problema. Desta forma todos os elementos da equipa serão preciosos para que seja possível levar uma ideia à implementação. Este modelo possibilita que se otimize a flexibilidade, criatividade e produtividade. Com esta metodologia passa a existir duas funções específicas dentro da equipa, nomeadamente:

- **SCRUM Master** : Este elemento encontra-se encarregue de auxiliar e promover o SCRUM para a equipa de forma a que os mesmos sejam capazes de conseguir retirar o melhor proveito possível do processo auxiliando estes a entender a teoria, as práticas, as regras e os valores da utilização da metodologia ágil;
- **Product Owner** : Este elemento possui a responsabilidade de gerir o "*Product backlog*" assim como maximizar o valor do produto e do trabalho da Equipa. Este elemento

representa os clientes ou utilizadores ajudando desta forma a orientar a equipa na conceção do produto correto sendo que ainda auxilia no desenvolvimento de software.

### 3.2.2 *Artefactos*

No SCRUM os artefactos correspondem a trabalho ou valor, que podem ser inspecionados e adaptados, que fornecem transparência acerca da informação essencial, dito isto para que todos tenham melhor entendimento acerca dos artefactos existentes. Nesta metodologia os artefactos existentes são os seguintes:

- **Produto** : O próprio produto em si corresponde a um artefacto, neste caso o principal pois espera-se que seja desenvolvido o produto que foi planeado inicialmente sendo que se espera que após a realização de cada *sprint* este possua um estado minimamente útil.
- **Produto Backlog** : Corresponde a uma lista ordenada que se espera que esteja presente no produto. O *Product Owner* deve dar prioridade ao *backlog* para que seja possível que a equipa trabalhe em primeiro lugar sobre as tarefas que são consideradas mais críticas e essenciais. Nesta secção ainda é realizada uma lista com todas as funcionalidades, requisitos e melhorias que correspondem às alterações que devem ser realizadas sobre o produto em entregas futuras.
- **Sprint Backlog** : Corresponde a um conjunto de elementos seleccionados do *Product Backlog* para o *sprint*, sendo também elaborado um plano para a entrega dos mesmos de forma a que seja possível concluir o objetivo delineado para o *sprint* corrente. Para além disto a equipa faz uma previsão acerca das funcionalidades que irão fazer parte do próximo *sprint* assim como o trabalho que deve ser realizado para que seja possível entregar as funcionalidades num estado final.
- **Incremento** : Este campo diz respeito à soma de todos os elementos que foram concluídos durante o *sprint* e que se encontravam presentes no *Product Backlog* somando ainda a estes os valores obtidos em *sprints* anteriores.
- **Outros artefactos** : Além dos artefactos enunciados anteriormente, *Backlogs* e o produto, existem outros artefactos que possuem relevância para a análise do trabalho desenvolvido, que é o caso de gráficos *Burndown*. Este gráfico é realizado aquando cada *sprint* e é comparado com o gráfico esperado para o lançamento do produto sendo assim possível apresentar a quantidade de trabalho que fica a faltar realizar e que pode ser realizado em um ou mais *sprints*. Para além disto também ajuda a verificar se o trabalho planeado irá cumprir o prazo estipulado.

### 3.2.3 *Sprint*

De acordo com o que foi enunciado até ao momento verifica-se que o *sprint* é uma iteração de um dado evento que possui uma duração de entre 1 a 4 semanas sendo que possui um objetivo claro que se pretende atingir, dito isto deve ser definido o que se pretende desenvolver assim como deverá existir um plano que irá acompanhar o desenvolvimento.

- **Planeamento do *Sprint*** : Antes de ocorrer o *sprint* é necessário existir uma reunião com todos os membros da equipa de forma a verificar que todos irão prever quais as funcionalidades que devem ser desenvolvidas durante a ocorrência do *sprint*. Durante esta reunião o *Product Owner* discute qual o objetivo a atingir após o *sprint*, o que corresponde a tarefas do *Product Backlog* que necessitam de ser realizadas. Para a existência desta reunião existem parâmetros que devem ser tidos em consideração, nomeadamente a existência do *Product Backlog*, o que foi incrementado ao produto desde a última vez, o desempenho que se espera da equipa de desenvolvimento durante o *sprint* e o desempenho da equipa no *sprint* anterior.
- **Reunião diária de SCRUM** : Nesta secção é defendido que deve existir uma reunião diária com todos os elementos da equipa, nomeadamente o *SCRUM Master* e o *Product Owner*, sendo que esta reunião não deve ter duração superior a 15 minutos. O propósito da existência desta reunião é no sentido de verificar o progresso em direção a atingir o objetivo do *sprint* delineado anteriormente e de forma a avaliar a capacidade de concluir o trabalho contido no *sprint Backlog*. Para esta reunião cada elemento deve indicar o que realizou no dia anterior, o que irá realizar no dia atual e quais os problemas com que se está a deparar, caso existam. Com estas reuniões é esperado que se melhore a comunicação e para que sejam identificados possíveis problemas para que sejam eliminados o mais rapidamente possível. Este processo é um dos principais contribuintes para o processo de inspeção e adaptação.
- **Revisão do *Sprint*** : De forma a analisar as funcionalidades resultantes do *sprint* é efetuada uma revisão a esse mesmo *sprint* para que seja possível obter *feedback* resultante da colaboração com o *Product Owner* ou qualquer outra parte interessada. A informação obtida desta revisão é crucial pois influenciará o próximo *sprint* dado que podem existir alterações a efetuar sobre as funcionalidades terminadas recentemente, podendo estas alterações resultar na revisão ou adição de funcionalidades existentes no *Product Backlog*.
- **Retrospectiva do *Sprint*** : Efetuada a revisão do *sprint* é efetuada outra reunião com todos os elementos que é conduzida pelo *SCRUM Master*, onde se pretende verificar o que foi realizado durante o *sprint* finalizado e verificar o que poderá ser alterado ou melhorado. O principal objetivo desta reunião é identificar melhorias que podem ser

adotadas pela equipa no próximo *sprint*, sendo que estas melhorias são obtidas a partir da inspeção efetuada acerca do *sprint* anterior.

### 3.3 TECNOLOGIAS DE DESENVOLVIMENTO

#### 3.3.1 Comunicações com Javascript

Nestes sistemas é muito comum existir uma constante comunicação entre o cliente e o servidor e possivelmente entre documentos. De forma a que seja possível existir esta comunicação existem algumas tecnologias que o permitem realizar. Assim sendo algumas das tecnologias que serão analisadas e estudadas de forma mais profunda de forma a que seja possível tirar conclusões relativas à utilização das mesmas são as seguintes: [32]

##### 1. XHR e XHR 2 com CORS

A utilização do *XHR* [33] pode ser realizada de forma síncrona ou assíncrona. Esta *API* suporta solicitações síncronas, de forma propositada, de forma a que seja bloqueada a execução do código até que seja recebido um retorno da chamada invocada.

Esta *API* também possibilita uma função que permite indicar o estado da transação que se encontra a decorrer.

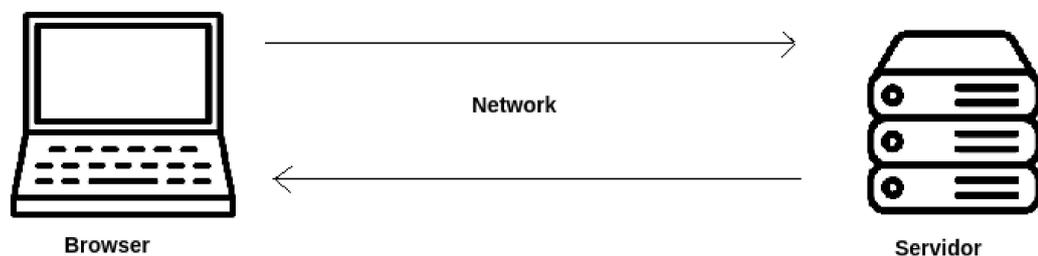


Figura 6: Funcionamento do *XHR*

Analisando a figura 6 é possível apresentar um exemplo do funcionamento desta tecnologia possibilitando uma melhor compreensão sobre a tecnologia abordada. Desta forma, tudo começa quando existe um evento no **Browser** que é despoletado, por exemplo através de um clique num botão. Realizada esta operação é enviado um pedido *HTTP* através da utilização do objeto *XMLHttpRequest* (*XHR*). Quando este pedido chega ao **Servidor** é processado e é criada uma resposta que é novamente enviada para o **browser**, sendo que é neste que a resposta recebida é tratada e transformada em Javascript.

Relativamente ao *XHR* 2 esta possui como elemento diferenciador do *XHR* o suporte ao Cross-Origin Resource Sharing (*CORS*). Com este elemento é possível fazer pedidos *XHR* entre domínios desde que o servidor ao qual se pretende aceder assim o permita.

## 2. Web Messaging / PostMessage

Esta *API* é uma das mais antigas que ainda se encontra em utilização e permite que seja possível comunicar entre domínios. Dito isto esta *API* permite que sejam enviadas, por exemplo, strings entre dois domínios que são mutuamente confiáveis [34][35].

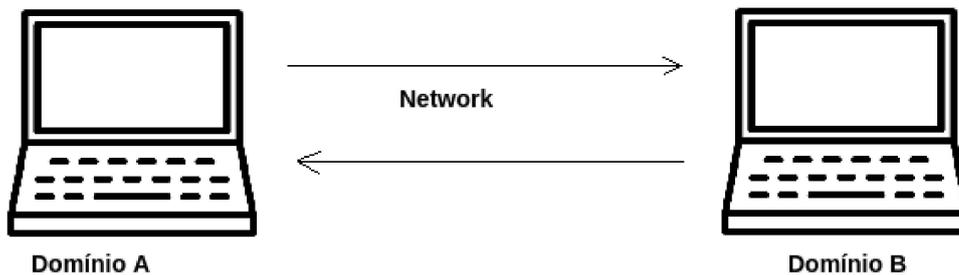


Figura 7: Funcionamento do Web Messaging / PostMessage

Abordando a explicação anterior de forma mais prática, através da figura 7, verifica-se que existem dois domínios que pretendem efetuar uma comunicação. Para que seja possível criar esta comunicação ela tem de ser segura e de origem cruzada. Dito isto, ambos os domínios devem estar localizadas no mesmo protocolo, por exemplo em *https*<sup>1</sup> e na mesma porta.

Após as restrições anteriores serem cumpridas é que é então possível estabelecer comunicação entre domínios, normalmente através de um evento que foi despoletado.

## 3. WebSockets

Os WebSockets<sup>2</sup> são uma tecnologia que permite obter de forma nativa mensagens do servidor no browser em tempo real. Assim sendo esta *API* pode ser utilizada para enviar e receber mensagens de um servidor, caso seja utilizado um socket bidirecional.

Através da figura 8 é possível ter uma melhor perceção do que é efetuado pelos WebSockets. Assim sendo o que se verifica é que inicialmente é definido um socket que irá servir para realizar a comunicação entre o **Cliente** e o **Servidor**. Definido o socket, o mesmo pode ser utilizado para comunicar num só sentido(unidirecional) ou em

<sup>1</sup> <https://en.wikipedia.org/wiki/HTTPS>

<sup>2</sup> <https://pt.wikipedia.org/wiki/WebSocket>

ambos(bidirecional). Neste caso encontra-se representado um socket bidirecional que permite que sejam trocados dados quer do **Cliente** para o **Servidor** quer do **Servidor** para o **Cliente**.

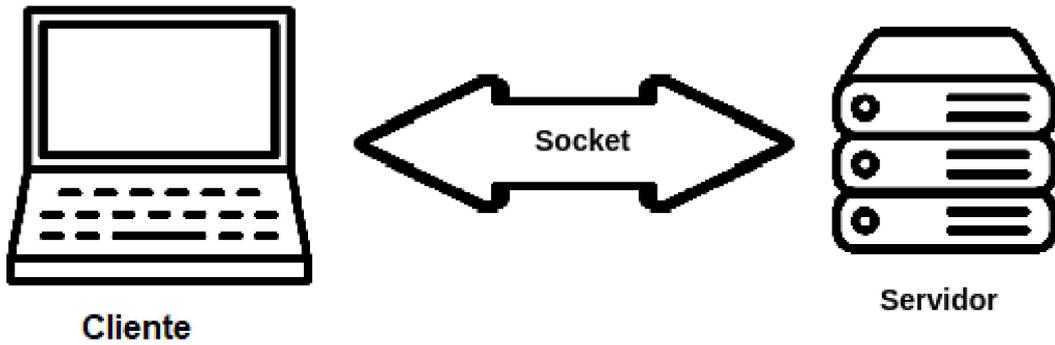


Figura 8: Funcionamento dos WebSockets

Para além disto esta *API* também possibilita que seja possível ir sobre outros domínios sem que seja restringido pela política da mesma origem [36].

#### 4. Server Sent Events

O Server Sent Events tal como o próprio nome sugere é uma *API* utilizada para que se possam enviar eventos do servidor para o cliente. Assim sendo o cliente só consegue “ouvir” as mensagens de um servidor e não consegue enviar mensagens [37].

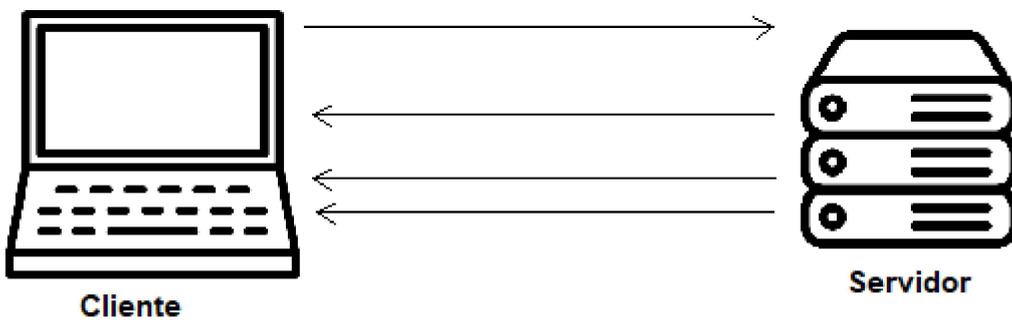


Figura 9: Funcionamento do Server Sent Events

Analisando o funcionamento do Server Sent Events exposto na figura 9 o que se verifica é que inicialmente um **Cliente** efetua o pedido por uma página *web*, através de um pedido regular *HTTP*. Posteriormente o **Cliente** recebe a página requerida e é executado um código Javascript que irá estabelecer uma ligação com o **Servidor**. Assim que se encontra estabelecida esta ligação o **Servidor** envia um ou vários eventos para o **Cliente** de forma a ser-lhe proporcionadas novas informações.

#### ANÁLISE COMPARATIVA

De forma a determinar qual poderá ser a *API* que melhor se poderá enquadrar no desenvolvimento de um sistema de informação hospitalar é relevante realizar uma comparação entre as *API*'s abordadas anteriormente e verificar se as mesmas cumprem alguns dos requisitos que são impostos neste tipo de sistemas.

Como à partida se pretende que sejam enviados dados do servidor para o sistema, uma vez que existe a possibilidade de serem requeridos dados em determinadas situações. Algumas situações foram analisadas e encontram-se na tabela 1 de forma a auxiliar na análise daquela que é a tecnologia que melhor se enquadra na solução pretendida.

No caso de se pretender realizar um sistema baseado em notificações, que normalmente se baseiam em operações em tempo-real, a utilização de *XHR* ou *XHR2* com *CORS* não deverá ser a melhor metodologia uma vez que para que se possa obter as mensagens da parte do servidor será necessário realizar constantemente pedidos ao servidor, o que não é uma boa prática dado que poderá ocorrer sobrecarga por parte do servidor.

Outro factor a ter em consideração é a segurança existente na conexão e na troca de mensagens com informações. Assim sendo tanto os WebSockets como o Server Sent Events realizam a troca de informações através de *HTTP*, sendo que os WebSockets utilizam somente o *HTTP* para estabelecer a conexão entre cliente e servidor e depois utiliza *TCP* para enviar dados. Já o Server Sent Events utiliza fluxos regulares de octetos *HTTP*, sendo que são limitados a aproximadamente 6 conexões *HTTP* em simultâneo por servidor. Com a utilização desta arquitetura possibilita que os balanceadores de carga e os proxies entendam de imediato qual a forma de envio de dados do servidor para os clientes.

O grande fator diferenciador entre os WebSockets e o Server Sent Events é que os WebSockets possuem um mecanismo que possibilita que sejam detetados quais os clientes que se desconectam enquanto que o Server Sent Events só determina quais são os clientes desconectados quando uma mensagem é enviada [38].

	Troca de dados	Segurança existente na conexão	Mecanismos de desconexão automáticos
<i>XHR / XHR 2 CORS</i>	Realiza novo pedido a cada troca de dados	<i>CORS</i>	Não possui
Web Messaging/ PostMessage	Limite de uma mensagem por segundo	Domínios com mesmo protocolo e porta	Não possui
WebSockets	Uma conexão para troca de várias mensagens	<i>HTTP e TCP</i>	Deteta desconexões
Server Sent Events	Utiliza eventos	Fluxos regulares de octetos <i>HTTP</i>	Não possui

Tabela 1: Comparação de tecnologias de comunicação em Javascript

### 3.3.2 Tecnologias de virtualização

Para o desenvolvimento e execução de uma plataforma a mesma pode ser feita de várias formas, nomeadamente através de um só componente ou dividida por vários subsistemas.

Através do desenvolvimento de vários subsistemas é possibilitado que todos em conjunto formem o sistema pretendido. Dito isto os subsistemas trabalham todos de forma agregada para que o sistema possa fornecer uma ou várias funcionalidades do sistema geral assegurando assim que os subsistemas funcionam em conjunto como se fosse um só sistema [39].

Ao contrário desta metodologia existe a metodologia de desenvolvimento baseada num só componente em que todas as funcionalidades são implementadas no mesmo componente e não distribuídas por vários como acontece no uso de desenvolvimento de vários subsistemas.

De forma a perceber as vantagens e desvantagens de cada um foi analisada a seguinte tabela, de acordo com o seguinte artigo [40].

Advantages	Descriptions
Faster development	Since a new application can be assembled through existing components, development time will be reduced
Easier to maintain	Since a component can be managed individually, the maintenance becomes easy
Quality	The quality is warranted since the components are sold after thorough tests
Easier to create application variants and upgrades	Since each function is implemented individually, corresponding components are changed or upgraded separately
Lower overall development costs	Overall development costs of the information systems could be reduced

Figura 10: Vantagens do desenvolvimento baseado em componentes [40]

Como se averigua através da tabela anterior as vantagens associadas ao desenvolvimento de um sistema baseado em vários componentes são:

- **O rápido desenvolvimento** que permite a reutilização de outros componentes que já se encontrem desenvolvidos e que se enquadrem no novo sistema;
- **A facilidade de manutenção** onde como cada componente se encontra individualizado sem depender de outros torna-se mais fácil efetuar a manutenção desse mesmo componente;
- **A criação e atualização de variantes do sistema** em que a realização se torna mais fácil através do uso de componentes dado que como se encontram individualizados os componentes as alterações e atualizações são realizadas de forma independente;
- **O custo geral de desenvolvimento** tornam-se menores aquando o uso de desenvolvimento baseado em componentes dado que com o uso da reutilização de componentes

e com a facilidade de alterações e atualizações enunciadas anteriormente é despendido menos tempo no desenvolvimento e manutenção e desta forma o custo geral do desenvolvimento de um sistema irá ser menor;

- **A qualidade** que é assegurada aquando a colocação do sistema no mercado onde devem ter sido realizados testes completos sobre os mesmos.

Depois de efetuada a análise anterior é possível verificar que a utilização da metodologia baseada em subsistemas, que por conseguinte irão possuir vários componentes, é a melhor abordagem quando comparada com a abordagem de desenvolvimento através do uso de um só componente.

De forma a que seja possível criar, integrar e correr aplicações é possível a utilização de ferramentas que possibilitam que tudo isto seja realizado.

- **Docker**

O Docker [41] é uma dessas ferramentas que utiliza containers para que seja possível realizar isso mesmo. Dito isto, os containers [42] são unidades standard de software que encapsulam código e todas as suas dependências. O Docker permite correr vários containers na mesma máquina para que seja possível correr toda a aplicação de forma rápida e confiável em ambientes de computação diferentes partilhando o mesmo kernel do sistema operativo. Apesar de todos os containers correrem a aplicação global cada um corre como um processo isolado o que permite a criação de subsistemas de forma a que se algum desses subsistemas falhe a aplicação continue a funcionar sem esse container específico.

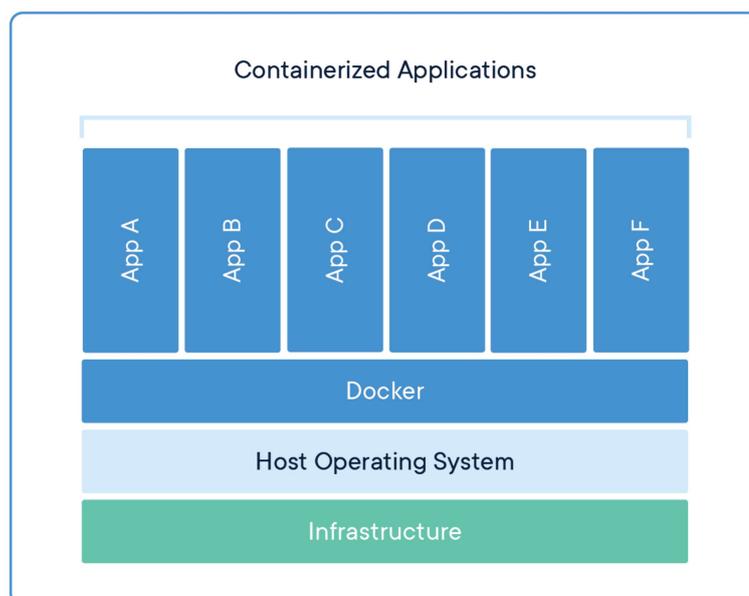


Figura 11: Grupos de alterações

[42]

- Máquinas Virtuais

Uma máquina virtual [43] é uma abstração de hardware físico. Dito isto permite tornar um server em vários para que estes corram em uma só máquina. Cada uma destas máquinas virtuais que forem criadas irão conter cópias do sistema operativo e da aplicação a correr assim como das livrarias dependentes entre outros elementos necessários o que irá fazer com que sejam utilizados dezenas de *Gigabytes* o que irá tornar a aplicação mais lenta. Para além disso outro factor a ter em conta é o facto de quando efetuado o arranque das máquinas virtuais o mesmo pode ser lento ou com o tempo poderá tornar-se lento.

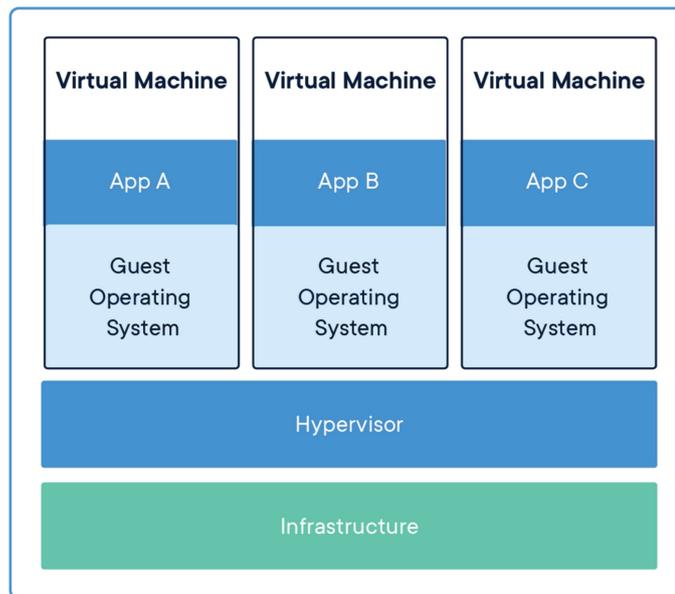


Figura 12: Grupos de alterações  
[42]

#### ANÁLISE COMPARATIVA

Após analisadas estas duas ferramentas e indo de encontro ao pretendido, ou seja a execução de vários subsistemas de forma isolada e de forma a que todos em conjunto contribuam para o sistema geral, verifica-se que a ferramenta mais adequada para tal é o Docker. Isto é possibilitado dado que através do uso dos containers é possibilitado que cada um dos subsistemas corra isoladamente com as suas dependências. Esta independência entre subsistemas permite que no caso de existir um que falhe por qualquer razão específica o sistema geral em si continuará a ser executado com a exceção das funcionalidades proporcionadas pelo subsistema que se encontra em falha. Para além disto permite que possa ser efetuada manutenção no sistema indo de subsistema em subsistema o que permite que não tenha que se parar todo o sistema para a realização dessa mesma manutenção.

Outra razão que diferencia o uso do docker das máquinas virtuais é o facto de este não utilizar muitos recursos, nomeadamente memória, que no caso das máquinas virtuais as torna computacionalmente mais pesadas, dado que utilizarão mais recursos da máquina em que se encontrem alocadas, e como tal irão correr os subsistemas de forma mais lenta. Para além disto como analisado no estudo das máquinas virtuais foi abordado que o arranque destas poderá ser bastante lento, o que de certa forma prejudica o sistema geral em si dado que estes sistemas necessitam de informação em tempo real, de forma constante e rápida.

### 3.4 METODOLOGIA DE AVALIAÇÃO

Ao longo desta secção serão abordadas quais as metodologias que irão ser utilizadas de forma a que seja possível avaliar a solução desenvolvida. Com esta avaliação espera-se conseguir perceber se os problemas indicados no capítulo introdutório são resolvidos através da solução desenvolvida, capítulo 1.

#### 3.4.1 Avaliação de Performance

De forma a avaliar a ferramenta, que permite a comunicação entre sistema e utilizadores, foram definidos alguns parâmetros das ações que devem ser realizadas quer no ambiente de desenvolvimento, que neste caso corresponde à simulação local, quer no ambiente de produção, que diz respeito à solução implementada no sistema real.

##### *Ambiente de Desenvolvimento*

No que diz respeito ao ambiente de desenvolvimento pretende-se que seja avaliado se a solução desenhada consegue comunicar ao utilizador informações acerca dos processos que lhe foram atribuídos. Assim sendo foram definidos os seguintes parâmetros de forma a facilitar a avaliação:

- A simulação deve conseguir enviar somente as mensagens relativas aos processos atribuídos ao profissional de saúde que se encontra a utilizar o sistema;
- As mensagens devem reaparecer enquanto o utilizador não confirmar que as visualizou;
- A solução deve procurar novas mensagens a cada 30 segundos de forma a que seja possível o utilizador ser notificado de forma mais rápida acerca de novas mensagens que possam surgir aquando a utilização do sistema.

### *Ambiente de Produção*

Relativamente ao ambiente de produção pretende-se que seja avaliado outras vertentes da solução desenvolvida, nomeadamente situações que não foram testadas no ambiente de desenvolvimento. Dito isto testar a solução para situações em que vários utilizadores se encontram autenticados no sistema e para casos onde possa ocorrer um pico na quantidade de mensagens relativas aos processos que necessitem de ser comunicadas a vários utilizadores. Com estes parâmetros é possível averiguar como se comporta a solução e se vai de encontro com as expectativas. De forma a aprofundar estes parâmetros foram divididos da seguinte forma:

- A solução deverá conseguir comunicar com vários utilizadores ao mesmo tempo transmitindo para cada um as respetivas mensagens de processos que lhes foram atribuídos;
- A solução deverá conseguir apresentar todas as mensagens a um dado utilizador apesar de o mesmo poder ter um número avultado de mensagens relativos a processos que lhe foram atribuídos;
- Apesar de existir um pico no número de utilizadores e portanto no número de comunicações que devem ser realizadas a solução deve ser capaz de conseguir apresentar as mensagens de acordo com o que foi testado no ambiente de desenvolvimento.

#### 3.4.2 *Análise SWOT*

De forma a ser possível analisar um projeto existe uma análise *Strengths, Weaknesses, Opportunities, Threats* (SWOT) que basicamente consiste num método capaz de avaliar as **forças (Strengths)**, **fraquezas (Weaknesses)**, **oportunidades (Opportunities)** e **ameaças (Threats)** associadas a esse mesmo projeto. Dito isto este método consiste em indicar qual o principal objetivo do projeto e posteriormente identificar fatores que possam ser favoráveis ou desfavoráveis quer através de fatores internos quer de externos [44].

De forma a que seja possível identificar alguns aspetos mencionados anteriormente pode-se definir um conjunto de questões que devem ser respondidas e que auxiliarão à identificação dos aspetos relevantes. Para o caso das **forças** existem as seguintes questões:

- O que é que o sistema faz de melhor e o que é que o diferencia?

Relativamente às **fraquezas** a questão colocada é:

- O que pode ser melhorado?

Quanto às **oportunidades** as questões colocadas são:

- Quais são as **oportunidades** existentes que podem ser utilizadas no sistema?
- Quais são as **oportunidades** que podem ser retiradas a partir das **forças**?

Por último as questões abordadas para as **ameaças** são:

- Quais as **ameaças** que podem surgir através das **fraquezas** identificadas no sistema?
- Quais as **ameaças** que podem fazer com que a solução não seja a adequada?

Através da utilização deste método é possível chegar a conclusões importantes uma vez que se realiza uma avaliação sobre a solução desenvolvida. Uma conclusão a retirar é se de facto todos os aspetos da solução estão bem realizados ou se podem ser melhorados. Para além disto também é possível averiguar se existem problemas que possam surgir e desta forma é possível resolver antes dos mesmos surgirem. Por último também é possível obter aspetos que possam vir a ser relevantes no caso de se querer dar continuidade ao projeto e através desta avaliação é possível identificar aspetos a incorporar em versões futuras.

---

## PLATAFORMA DE COMUNICAÇÃO

---

Ao longo deste capítulo será abordado e explicado o problema que levou ao desenvolvimento da solução realizada. Para além disso também serão apresentados os requisitos funcionais e técnicos que foram levantados e sobre os quais a solução teve de se guiar. Também será apresentada a arquitetura quer da solução em ambiente de desenvolvimento quer em ambiente de produção.

### 4.1 IDENTIFICAÇÃO DO PROBLEMA

Atualmente verifica-se que cada vez existe maior afluência a sistemas informáticos tornando-os desta forma uma parte integrante nas mais diversificadas áreas, desde restauração até saúde. Dito isto, verifica-se que para a realização de um dado trabalho existe a necessidade da existência de um sistema informático de forma a facilitar a realização desse mesmo trabalho. Nomeadamente na área da saúde onde se incluem as unidades de saúde como centros de saúde, clínicas e hospitais tem-se verificado que existe sempre uma grande quantidade de informação a circular quer relativamente a dados pessoais sobre pacientes quer sobre processos atribuídos a profissionais de saúde, com o auxílio do openEHR.

Devido ao facto de na área da saúde não existir a possibilidade de aceder aos registos de saúde relativos a um dado paciente em diferentes unidades de saúde surgiu o openEHR. Com este surgimento passou a ser permitido a criação de um *Registo de saúde eletrónico (Electronic Health Record) (EHR)* relativo a um dado paciente podendo-se aceder a esses registos a partir de qualquer sistema que integre esta plataforma. Para além disto, o openEHR, também permite auxiliar no fluxo do trabalho clínico a ser realizado pelos profissionais de saúde, nomeadamente é possível criar processos a atribuir aos profissionais de saúde com informações relevantes para a realização do seu trabalho. Contudo o openEHR não disponibiliza um mecanismo que possibilite alertar os profissionais de saúde relativamente aos processos que lhes foram atribuídos.

Além disto também estão interligadas outras questões relativamente aos dados que podem ser apresentados. Dito isto, surgiu em 2018 uma nova lei, *RGPD*, com o propósito de defender os titulares dos dados tanto clínicos como pessoais e como neste tipo de sistemas o que

mais circula são estes dados denominados sensíveis é necessário verificar quais os dados que podem ser apresentados e quais as suas condições, sendo que estes dados só devem ser apresentados quando necessários e com o consentimento do titular dos mesmos.

Como os sistemas informáticos na área da saúde são indispensáveis para que os profissionais de saúde consigam realizar o seu trabalho é fulcral que estes sistemas estejam disponíveis para utilização o máximo de tempo possível de forma a minimizar ao máximo problemas que possam surgir aquando a indisponibilidade do mesmo. Um exemplo, é aquando a tentativa de aceder a informação relativa a um paciente para diagnosticar o que se poderá passar com a saúde do mesmo e no caso do sistema estar indisponível o profissional de saúde também poderá não conseguir realizar o seu trabalho.

#### 4.2 REQUISITOS FUNCIONAIS

Com a identificação dos requisitos funcionais espera-se conseguir levantar funcionalidades relevantes para a solução final. Dito isto, foram levantadas as funcionalidades que se espera que sejam implementadas na ferramenta, sendo que também é identificado o comportamento que se espera que a funcionalidade consiga realizar. Desta forma foram realizados os seguintes requisitos:

- A ferramenta deve conter um sistema de autenticação, de forma a que seja possível identificar o profissional de saúde;
- Após o *login*, devem ser apresentadas notificações com as informações do processos atribuídos ao utilizador em questão;
- As notificações a serem apresentadas devem estar disponíveis para visualização durante pelo menos 30 segundos;
- As notificações devem ser marcadas como visualizadas;
- Notificações que não tenham sido marcadas como visualizadas devem surgir novamente;
- A ferramenta deve procurar a existência de novas notificações a cada 60 segundos, no mínimo.

#### 4.3 REQUISITOS TÉCNICOS

Ao contrário dos requisitos funcionais os requisitos técnicos procuram focar-se no sistema e não apenas em funcionalidades que devem ser desempenhadas. Assim sendo, são definidos os requisitos que se espera que o sistema deva conseguir cumprir, tais como desempenho,

fiabilidade ou usabilidade. A definição destes requisitos é fulcral pois tem como propósito auxiliar na decisão da arquitetura a seguir assim como no cumprimento dos objetivos para a solução final. Seguindo os critérios anteriores foram definidos os seguintes requisitos técnicos:

- O utilizador, profissional de saúde, deverá conseguir utilizar a solução final com eficiência ao fim de 3 minutos de uso;
- O utilizador deve efetuar o *login* no sistema de forma a utilizar a solução final;
- O utilizador deverá conseguir visualizar a informação a ser-lhe disponibilizada de forma imediata após a realização do *login*;
- As mensagens a serem apresentadas ao utilizador devem ser somente criadas pelo sistema a partir do openEHR.

#### 4.4 ARQUITETURA DA SOLUÇÃO

De forma a tentar solucionar os problemas expostos no primeiro capítulo foi proposta a arquitetura da figura 13 que tem como principal propósito o uso das tecnologias estudadas no terceiro capítulo, dito isto onde se pretende utilizar comunicação em *Javascript*, mais concretamente *Websockets*<sup>1</sup>, e *Docker*<sup>2</sup>.

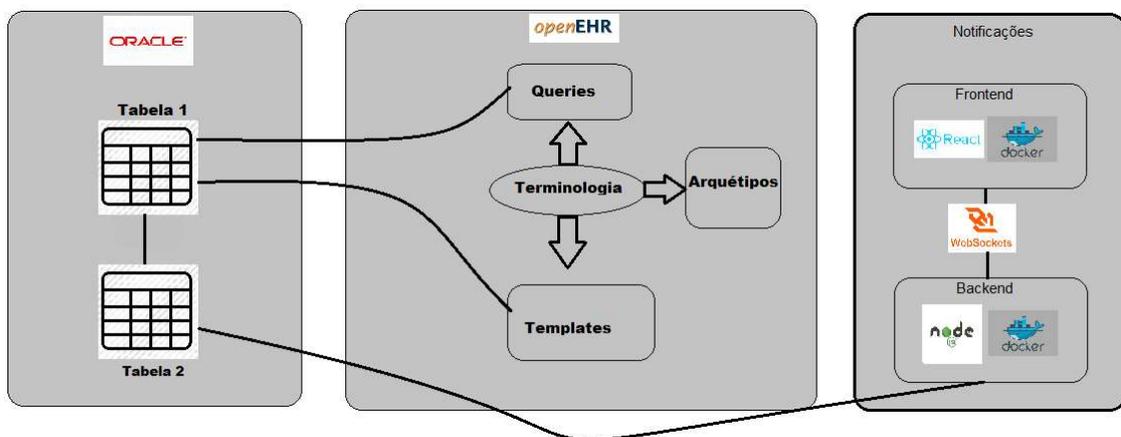


Figura 13: Arquitetura da solução

Aprofundando a análise sobre a figura anterior verifica-se que inicialmente é construída, por pessoas especializadas, uma biblioteca de *data points* relevantes na área da saúde, definidos em qualquer linguagem, que se denomina por **arquétipos**.

<sup>1</sup> <https://pt.wikipedia.org/wiki/WebSocket>

<sup>2</sup> <https://www.docker.com/>

Após definidos os **arquétipos** é nos **templates** que é possível transformar os *data points* em *data sets* que são usados localmente para casos de uso.

Relativamente às **queries** o que se passa neste passo é que o mesmo acede aos **arquétipos** de forma a realizar consultas que normalmente devolve dados.

Após estes processos o passo denominado **queries** interliga-se ao **openEHR data** de forma a que seja possível retornar os dados pretendidos através da execução deste passo. Também o passo denominado **templates** se interliga com o **openEHR data** e através desta ligação é permitida a inserção e criação de novos *data sets*.

Posto isto é a partir do **OpenEHR data** que é possível começar a utilizar o openEHR como um serviço a incorporar num sistema de saúde e desta forma é utilizada uma base de dados em Oracle<sup>1</sup> onde se pretende armazenar os dados relativos a cada pessoa, quer pessoais quer clínicos. Para além destes dados também se pretende armazenar dados que possam auxiliar na utilização do sistema geral, como por exemplo os dados de autenticação dos utilizadores, as mensagens relativas aos processos associados, entre outras.

Após tudo isto estar construído é possível passar para a fase que permite a apresentação das mensagens no sistema geral para cada utilizador, relativos aos processos atribuídos. Basicamente a solução encontrada passa por dividir em duas partes: **Frontend** e **Backend**. Começando pelo **backend** foi utilizado NodeJS<sup>2</sup> para que fosse possível conectar à base de dados e realizar operações de consulta sobre a mesma, nomeadamente acerca das mensagens que existiam para o utilizador autenticado.

No que diz respeito ao **frontend** foi utilizado ReactJS<sup>3</sup> que possibilita a apresentação visual das mensagens relativas a cada utilizador.

De forma a que fosse possível interligar o **Frontend** e **Backend** em termos de comunicação foram utilizados **Websockets**<sup>4</sup>, sendo que um *socket* é atribuído a um utilizador quando o mesmo começa a utilizar a plataforma, ou seja quando se encontra na página inicial/*login*.

Para além disto tanto o **Frontend** como o **Backend** foram encapsulados em *containers* através da utilização do Docker<sup>5</sup> para que a integração dos mesmo seja realizada de forma mais simples e fácil.

#### 4.4.1 Ambiente de Desenvolvimento

Para que se conseguisse implementar uma solução eficaz foram realizados testes sobre as ferramentas utilizadas, nomeadamente foi realizada uma simulação local do ambiente em que se iria pretender integrar a solução desenvolvida. Desta forma utilizou-se uma base de

<sup>1</sup> <https://www.oracle.com/>

<sup>2</sup> <https://nodejs.org/en/>

<sup>3</sup> <https://reactjs.org/>

<sup>4</sup> <https://pt.wikipedia.org/wiki/WebSocket>

<sup>5</sup> <https://www.docker.com/>

dados - MySQL<sup>1</sup> - em que foram utilizadas duas tabelas, NodeJS<sup>2</sup> de forma a conseguir tratar de questões como aceder às informações pretendidas que com o auxílio de Websockets<sup>3</sup> permite transmitir as informações a serem apresentadas através da utilização do ReactJS<sup>4</sup>. Em todas as tecnologias foi utilizado Docker<sup>5</sup> de forma a que seja possível encapsular as mesmas em *containers*, de forma a facilitar a integração destas tecnologias.

Para que fosse possível a realização dos testes sobre as tecnologias selecionadas foi utilizado um ficheiro que continha uma script de forma a popular a base de dados, tendo como principal objetivo simular a atribuição de processos pelo openEHR.

Expondo o anterior na figura 14 encontra-se a arquitetura utilizada para o desenvolvimento da simulação exposta.

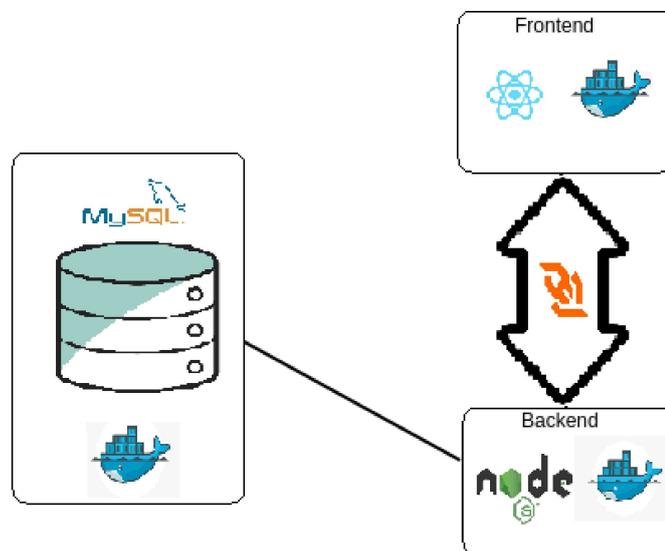


Figura 14: Arquitetura da solução em simulação local

Começando pelo **backend**, que se encontra representado na figura 14, é neste local onde se realizam operações fulcrais para o correto funcionamento da solução. É através deste processo onde se realiza a conexão à base de dados em que foi utilizado um mecanismo, *pool*<sup>6</sup>, que permite armazenar um conjunto de conexões à base de dados, neste caso limitado a 10. Este mecanismo permite a reutilização das conexões, para a realização de operações na base de dados, quando estas se encontram disponíveis ou então é efetuada uma espera até que uma fique disponível para ser utilizada enquanto que se este mecanismo não tivesse sido utilizado teria de se criar uma nova conexão sempre que se pretendesse ter mais do que

1 <https://www.mysql.com/>

2 <https://nodejs.org/en/>

3 <https://pt.wikipedia.org/wiki/WebSocket>

4 <https://reactjs.org/>

5 <https://www.docker.com/>

6 [https://en.wikipedia.org/wiki/Pool\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Pool_(computer_science))

uma conexão a ser utilizada em simultâneo para a realização de diversas operações sobre a base de dados.

Após o processo anterior o próximo processo a ser realizado é quando o utilizador entra na página web disponibilizada. No momento em que é realizada esta operação é atribuído, através da utilização dos Websockets<sup>1</sup>, um *socket* ao utilizador. Os *sockets* atribuídos são únicos para cada utilizador e, neste caso, tem um papel preponderante pois será através do mesmo que será estabelecida uma conexão entre o **frontend** e o **backend** e desta forma serão efetuadas trocas de informação entre ambos os processos.

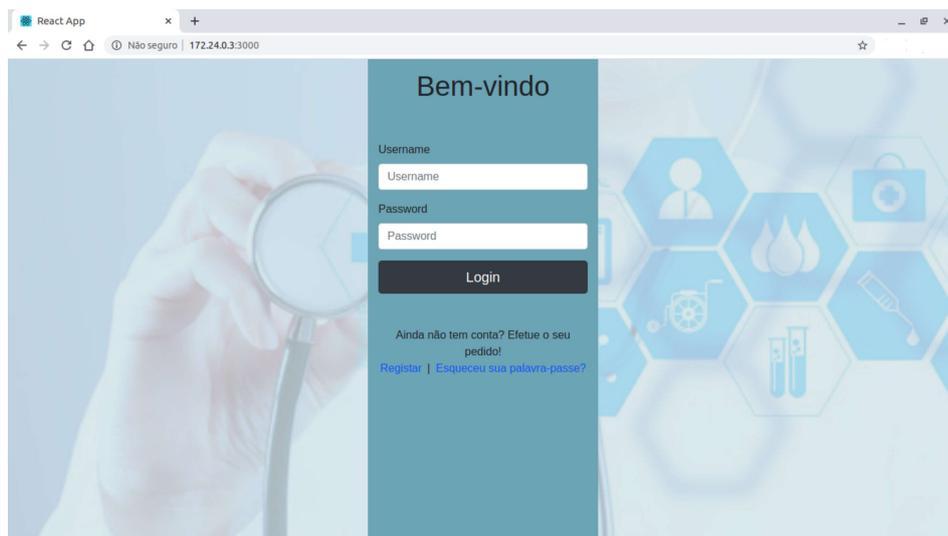


Figura 15: Página *web* com formulário de *login*



Figura 16: Página *web* com formulário de *login* preenchido

<sup>1</sup> <https://pt.wikipedia.org/wiki/WebSocket>

Na mesma página web, após ser atribuído o respetivo *socket* ao utilizador, é apresentado um formulário de *login*, apresentado na figura 15, e após o preenchimento de forma a tentar a autenticação, figura 16, os dados são enviados através do *socket* atribuído ao utilizador em questão para o **backend** onde é verificado, através da *pool*<sup>1</sup> e dos dados introduzidos, se o utilizador se encontra registado na plataforma. Em caso negativo no **frontend** o utilizador volta à página de *login*, enquanto que em caso afirmativo o utilizador é encaminhado para outra página onde serão apresentadas as notificações do utilizador.

A obtenção das notificações tem início no **backend** onde é utilizada novamente a *pool*<sup>1</sup> de forma a realizar a procura de mensagens relativas aos processos atribuídos ao utilizador. Assim que são obtidas da base de dados as mesmas são reencaminhadas para o **frontend** através do *socket* do respetivo utilizador. De forma a automatizar o máximo possível este processo este passo é repetido a cada 60 segundos possuindo como restrição o facto de só procurar por mensagens que ainda não tenham sido visualizadas pelo utilizador.

Recebendo as mensagens provenientes do **backend** é possível no **frontend** realizar a apresentação das mesmas de forma visual ao utilizador, figura 17.

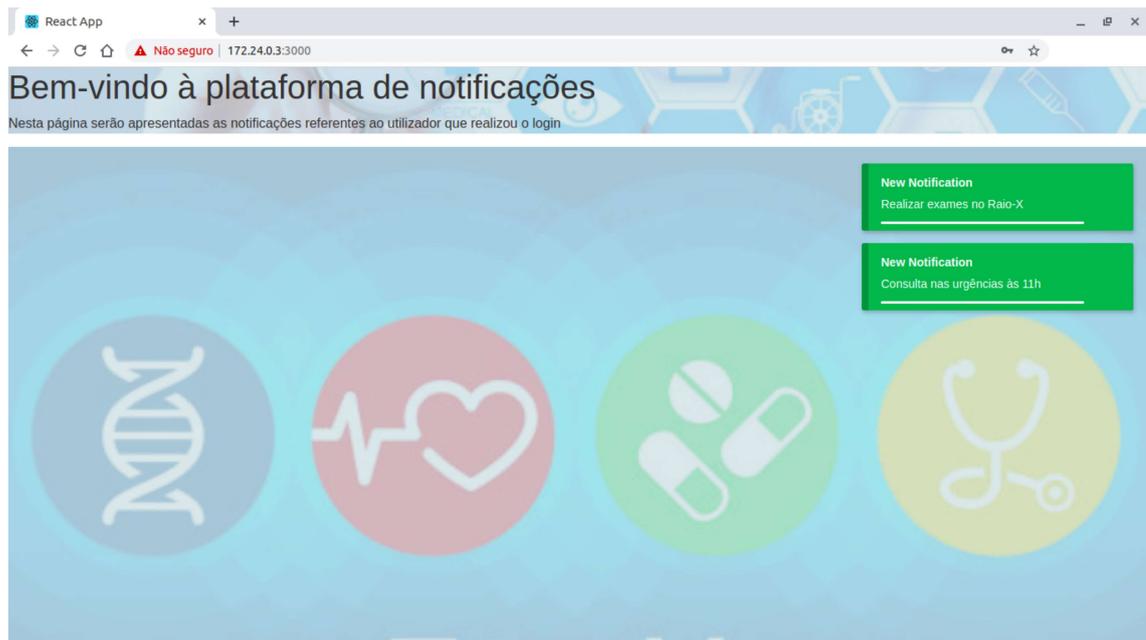


Figura 17: Página *web* com apresentação das notificações

Desta forma foi utilizada uma biblioteca do ReactJS<sup>2</sup> onde se definiu um conjunto de restrições para a apresentação das notificações para este caso de testes. Algumas dessas restrições consistem no tempo disponível para a visualização, definido em 30 segundos, na adição de uma pausa sobre o tempo definido anteriormente quando se coloca o ponteiro do rato sobre a notificação, figura 18, dado ser de interesse do utilizador que o tempo de

<sup>1</sup> [https://en.wikipedia.org/wiki/Pool\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Pool_(computer_science))

<sup>2</sup> <https://reactjs.org/>

visualização da notificação demore mais tempo e por último quando o utilizador clica sobre a notificação é considerado que o mesmo visualizou a notificação e portanto o estado da mesma muda pelo que não voltará a ser apresentada, figura 19.

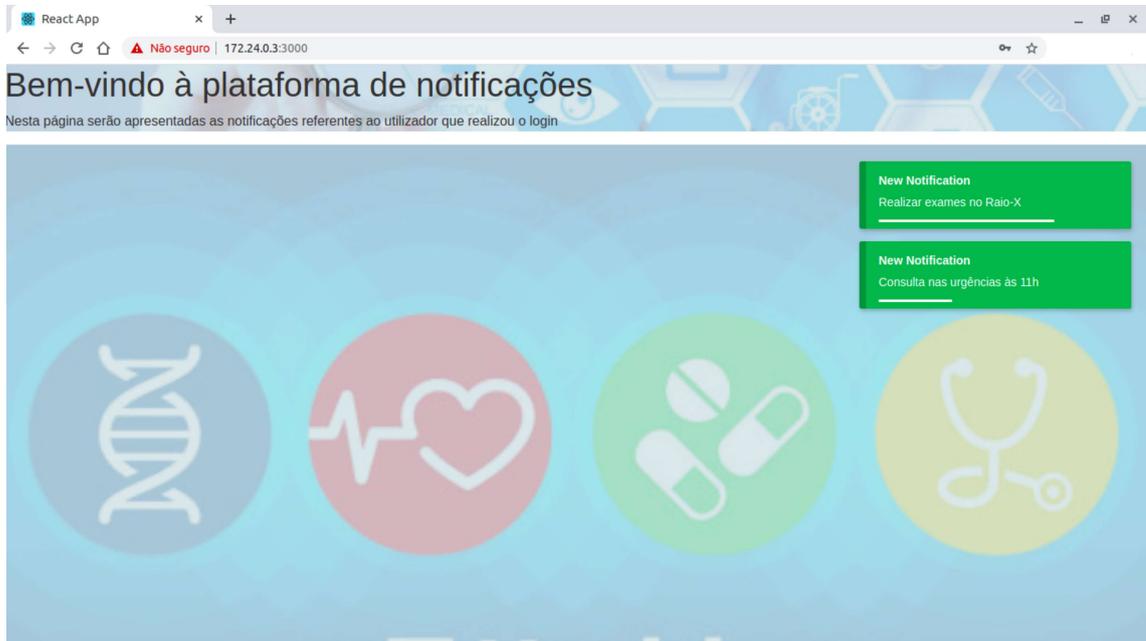


Figura 18: Página *web* com apresentação das notificações, colocando uma em pausa devido ao *hover* do ponteiro do rato

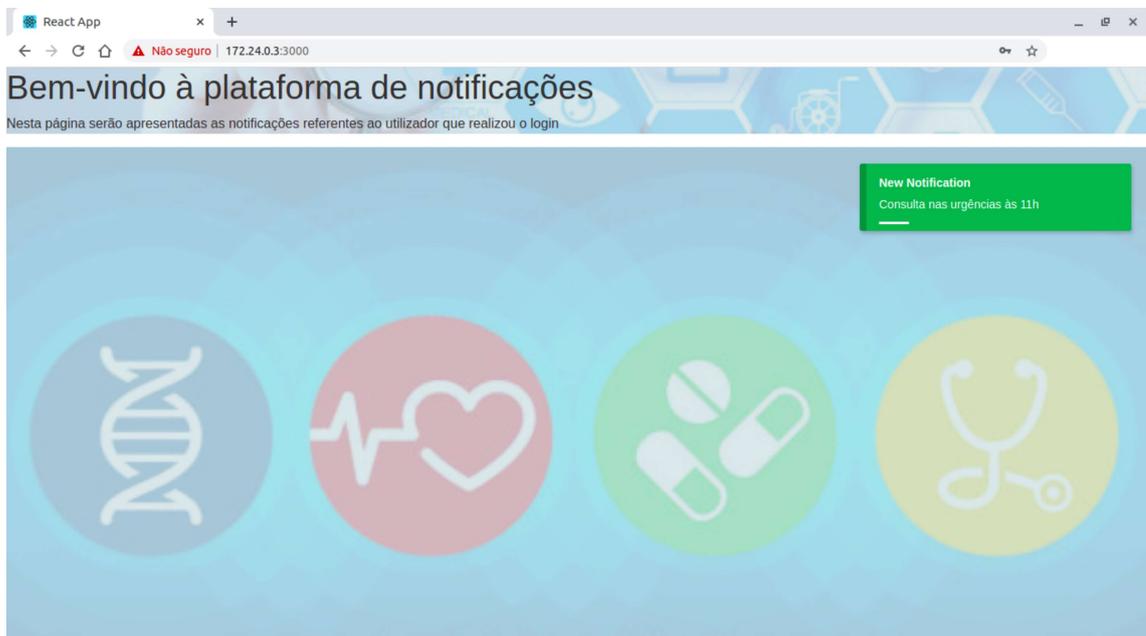


Figura 19: Página *web* com apresentação das notificações, exceto a que o utilizador efetuou o *click*

Desta forma é esperado que quando a solução efetuar novamente a procura por novas mensagens, para o utilizador autenticado, a mensagem onde foi efetuado o clique não volte a ser apresentada, como se verifica pela figura 20.

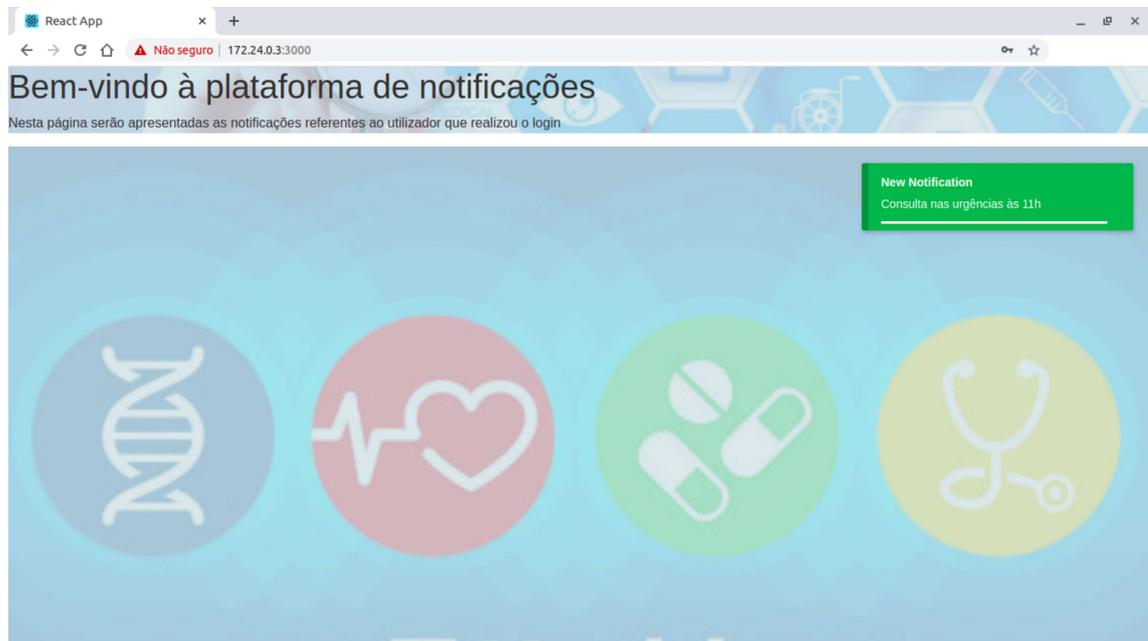


Figura 20: Página *web* com apresentação das notificações, exceto a que o utilizador efetuou o *click* anteriormente

Posto isto, de forma a testar se a solução desenvolvida resolve os problemas identificados foi utilizada uma *workbench* que permitiu simular a inserção de uma nova mensagem a ser apresentada a um utilizador, figura 21. Com a realização desta operação espera-se que a solução apresente a nova mensagem aquando a procura de novas mensagens, a cada 60 segundos, para o utilizador que se encontra autenticado. Como se verifica pela figura 22 a mensagem introduzida é de facto apresentada ao utilizador.

Para além dos testes realizados anteriormente também foi simulada a tentativa de autenticar dois utilizadores ao mesmo tempo de forma a testar se a solução tem o mesmo comportamento aquando um único utilizador, figura 23.

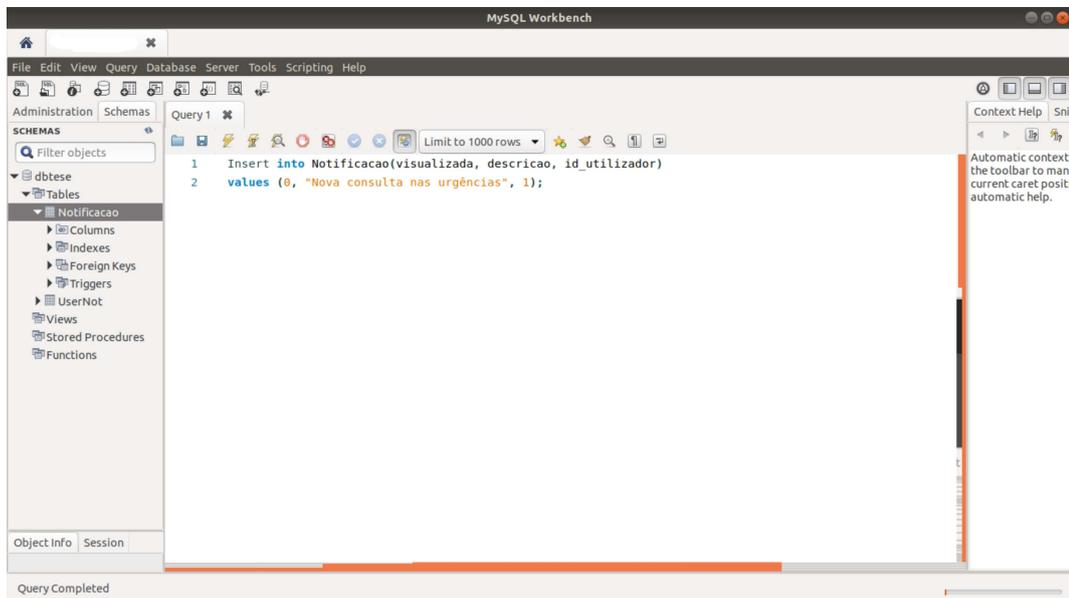


Figura 21: *Insert query* para a introdução de uma nova mensagem relativo a um utilizador

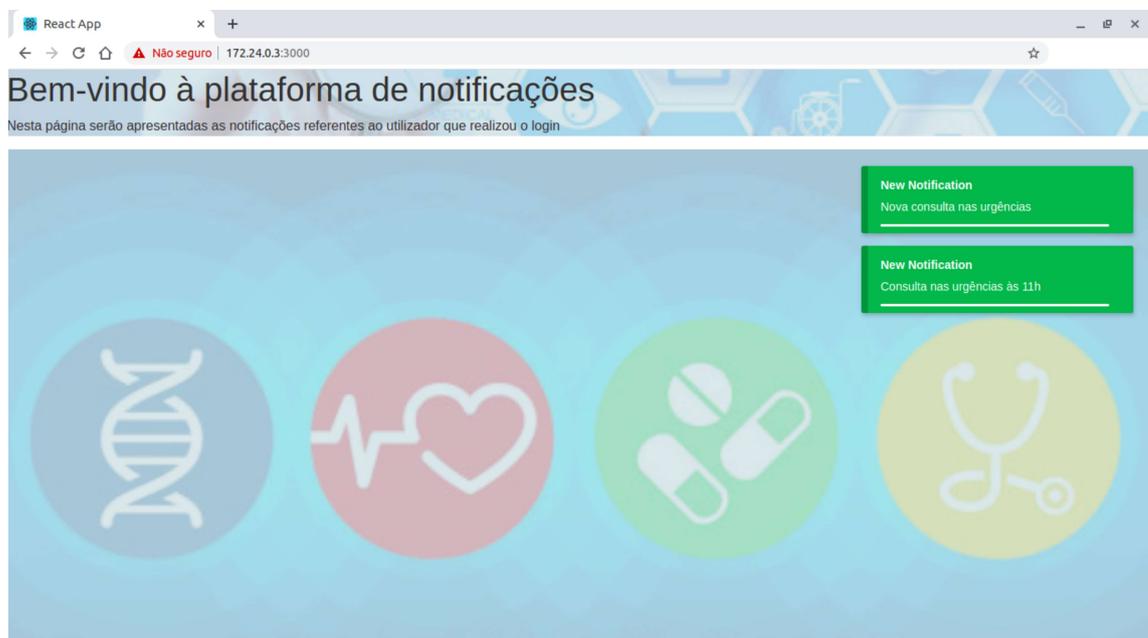


Figura 22: Página *web* com apresentação das mensagens, incluindo a nova mensagem inserida



Figura 23: Páginas *web* com formulário de autenticação para dois utilizadores

Como se visualiza pela figura 24 cada utilizador recebeu as mensagens que lhe foram atribuídas e não as mensagens atribuídas aos utilizadores autenticados.



Figura 24: Páginas *web* com apresentação das mensagens atribuídas a cada utilizador

#### 4.4.2 Ambiente de Produção

Assim que terminada a simulação relativa à implementação da solução localmente foi realizada uma implementação de modo a permitir a integração da solução desenvolvida num ambiente real. A solução atualmente não foi integrada no sistema dado que o mesmo ainda se encontra em desenvolvimento. Contudo nesta solução desenvolvida a mesma foi adaptada o máximo possível, incluindo algumas das tecnologias que se encontram em utilização, de forma a facilitar a integração quando o mesmo se encontrar desenvolvido de forma a permitir esta inclusão.

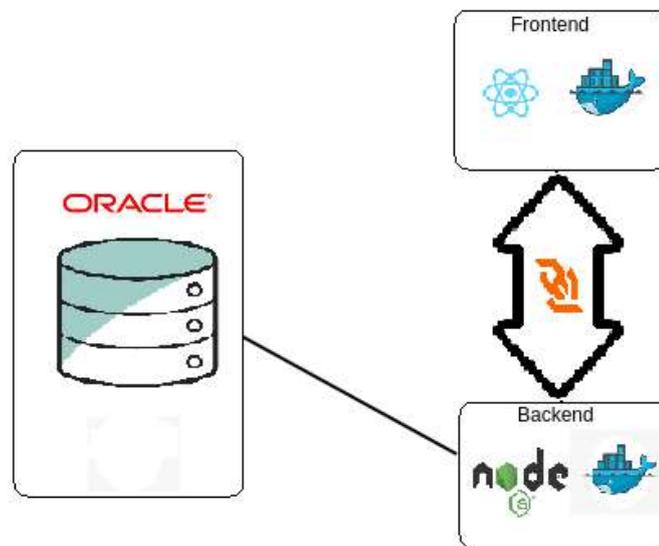


Figura 25: Arquitetura da solução em ambiente de produção

Analisando a figura 25, que corresponde à arquitetura utilizada para a solução desenvolvida num ambiente real, verifica-se que tal como na solução desenvolvida localmente esta se divide essencialmente em **backend** e **frontend**. Para a parte de **frontend** e Websockets<sup>1</sup> a solução manteve-se igual dado que a principal diferença se encontrava no **backend**, de forma a ser possível realizar a integração desta solução. A principal alteração que foi identificada e enumerada anteriormente tem a ver com o fato de ser pretendido utilizar uma das tecnologias que já se encontra em utilização no sistema em desenvolvimento, nomeadamente a tecnologia utilizada na base de dados que trata do armazenamento/consulta da informação. A inclusão desta restrição de uso da tecnologia em utilização por parte do sistema aquando o desenvolvimento da solução teve o propósito de facilitar a integração da solução no sistema quando assim for oportuno. Analisando mais cuidadosamente verifica-se que na solução

<sup>1</sup> <https://pt.wikipedia.org/wiki/WebSocket>

local foi utilizada uma base de dados MySQL<sup>1</sup> enquanto que para esta solução se pretendia utilizar uma base de dados Oracle<sup>2</sup>. Este processo teve por base a utilização de uma base de dados que se encontra alocada numa máquina presente no *Departamento de Informática (DI)* da *Universidade do Minho (UM)* dado ser a que se encontra em utilização por parte do sistema em desenvolvimento.

Seguindo o raciocínio anterior e sabendo que as alterações a serem efetuadas estariam presentes na parte do **backend** foi realizado o mesmo procedimento aquando a implementação da solução no ambiente de desenvolvimento (4.4.1), onde a conexão à base de dados passa pela utilização do mecanismo *pool*<sup>3</sup>. Tal como anteriormente também nesta solução foram limitadas a 10 o número de conexões à base de dados armazenadas por cada *pool*<sup>3</sup>. A escolha deste mecanismo teve como principal propósito o facto de ser possível reutilizar as ligações sempre que estas não se encontrem a ser utilizadas e em caso de não existirem conexões disponíveis é realizada uma espera até que exista uma que fique disponível e pronta a ser reutilizada. Este mecanismo permite que não seja necessário efetuar uma nova conexão sempre que se pretende aceder à base de dados para a realização de qualquer operação sobre a mesma, permitindo a realização de operações em simultâneo através das diversas conexões existentes.

De forma a seguir a arquitetura, figura 25, e colocar as tecnologias em *containers* através da utilização do Docker<sup>4</sup> existiu a necessidade de realizar a criação de uma imagem, em relação à simulação local, que permitisse ao NodeJS<sup>5</sup> criar uma conexão à base de dados através da utilização da *string* de conexão. Como se tratava de uma base de dados em Oracle<sup>2</sup> teve de ser utilizada no Docker<sup>4</sup> uma imagem baseada em *oraclelinux*<sup>6</sup>. Através da utilização desta imagem e de forma a ser possível conectar à base de dados através da utilização do NodeJS<sup>5</sup> foi necessário proceder à instalação do próprio NodeJS<sup>5</sup>. Após esta instalação na imagem utilizada também foi necessário proceder à instalação de outros componentes necessários de forma a ser possível realizar a conexão à base de dados e realizar as demais operações. Assim sendo foi necessário proceder à instalação do *Oracle Instant Client*<sup>7</sup>, que permite criar e correr aplicação que se conectam remotamente ou localmente sobre base de dados Oracle<sup>4</sup> e instalação do *node-oracledb*<sup>8</sup> que permite ao NodeJS<sup>5</sup> a realização de conexões sobre base de dados Oracle<sup>2</sup> para ser possível a realização de operações sobre a base de dados.

Depois de obtida a imagem com acesso à base de dados a correr no Docker<sup>4</sup> foi possível passar para o **backend** onde como foi utilizado um *driver* diferente para a realização de

1 <https://www.mysql.com/>

2 <https://www.oracle.com/>

3 [https://en.wikipedia.org/wiki/Pool\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Pool_(computer_science))

4 <https://www.docker.com/>

5 <https://nodejs.org/en/>

6 [https://hub.docker.com/\\_/oraclelinux](https://hub.docker.com/_/oraclelinux)

7 <https://www.oracle.com/pt/database/technologies/instant-client.html>

8 <https://www.npmjs.com/package/oracledb>

operações sobre as base de dados utilizadas quer localmente quer em ambiente de produção tiveram de ser feitas alterações na forma como era efetuada a conexão e na forma como se realizava as operações sobre a base de dados.

Efetuada as alterações anteriores o restante processo é semelhante ao processo da simulação realizada em ambiente de desenvolvimento, desde a atribuição e utilização do *socket* para efetuar a comunicação e assim trocar as mensagens entre **backend** e **frontend**. Assim o processo inicia com a conexão à base de dados através da utilização do *pool*<sup>1</sup>. Após estabelecida a conexão é atribuído um *socket* ao utilizador através da utilização dos *Websockets*<sup>2</sup> quando o utilizador entra na página web disponibilizada. De realçar que será através deste *socket* que irão ser efetuadas as comunicações entre **backend** e **frontend**, neste caso é o mecanismo encarregue pelo envio das mensagens que são obtidas da base de dados no **backend** para o **frontend** onde serão apresentadas visualmente ao utilizador.



Figura 26: Página *web* com formulário de *login*

<sup>1</sup> [https://en.wikipedia.org/wiki/Pool\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Pool_(computer_science))

<sup>2</sup> <https://pt.wikipedia.org/wiki/WebSocket>

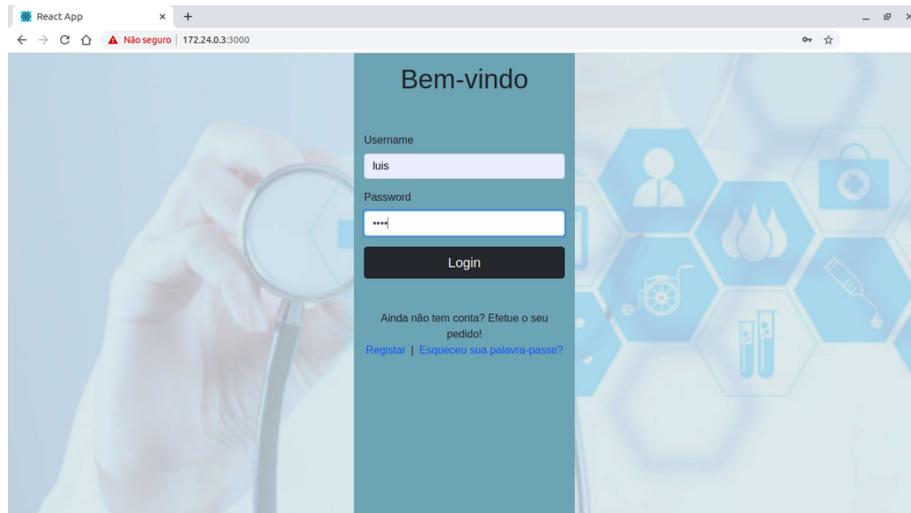


Figura 27: Página *web* com formulário de *login* preenchido

Nesta primeira página em que é atribuído o *socket* ao utilizador é também apresentado um formulário de *login*, figura 26, onde o utilizador após o seu preenchimento, figura 27, tenta efetuar a autenticação na plataforma. A transferência dos dados de forma a verificar a autenticação do utilizador é feita através do *socket* atribuído anteriormente, dito isto são enviadas as credenciais para o **backend** que verifica se o utilizador se encontra registado na plataforma ou não. Esta verificação é efetuada através da utilização da *pool*<sup>1</sup> e em caso de o utilizador se encontrar registado é efetuada a autenticação e portanto é reencaminhado para a página onde lhe serão apresentadas as notificações.

De forma a serem apresentadas as notificações o processo é iniciado no **backend** com a utilização da *pool*<sup>1</sup> onde é realizada a procura por mensagens, que ainda não foram lidas, relativas à atribuição de processos aos utilizadores em questão. Após obtidas as mensagens as mesmas são enviadas pelo *socket* para o **frontend** onde são apresentadas ao utilizador e tal como na solução local este processo é realizado a cada 60 segundos apresentando somente mensagens que ainda não tenham sido marcadas como lidas pelo utilizador.

De forma a apresentar ao utilizador as mensagens no **frontend** foi utilizada uma biblioteca do ReactJS<sup>1</sup> com algumas restrições, nomeadamente o tempo disponível para visualização, fixado em 30 segundos, adição de pausa no tempo referido anteriormente aquando a colocação do ponteiro do rato sobre a mensagem, figura 28, e adição de um evento capaz de marcar uma mensagem como lida quando o utilizador clica sobre a mesma.

Desta forma espera-se que quando for efetuado o processo de procura e apresentação de mensagens a um utilizador autenticado, a cada 60 segundos, se verifique que não é apresentada a mensagem que o utilizador marcou como visualizada, como se verifica pela

<sup>1</sup> <https://reactjs.org/>

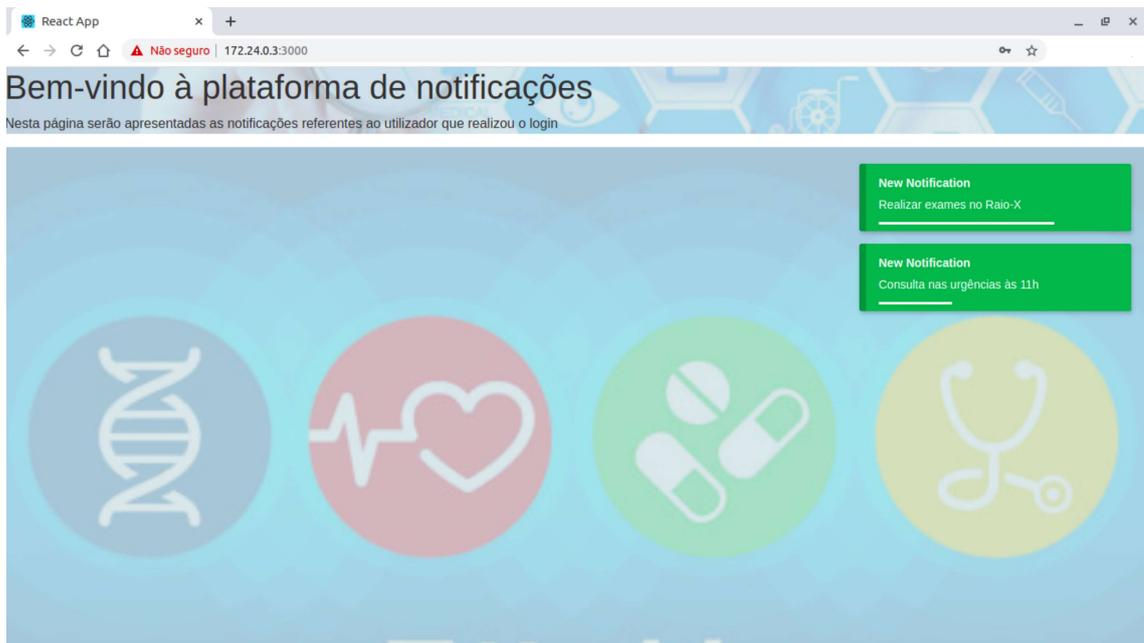


Figura 28: Página *web* com apresentação das notificações, colocando uma em pausa devido ao *hover* do ponteiro do rato

figura 29 onde o utilizador marcou uma das notificações como visualizada e a mesma já não é apresentada na figura 30.

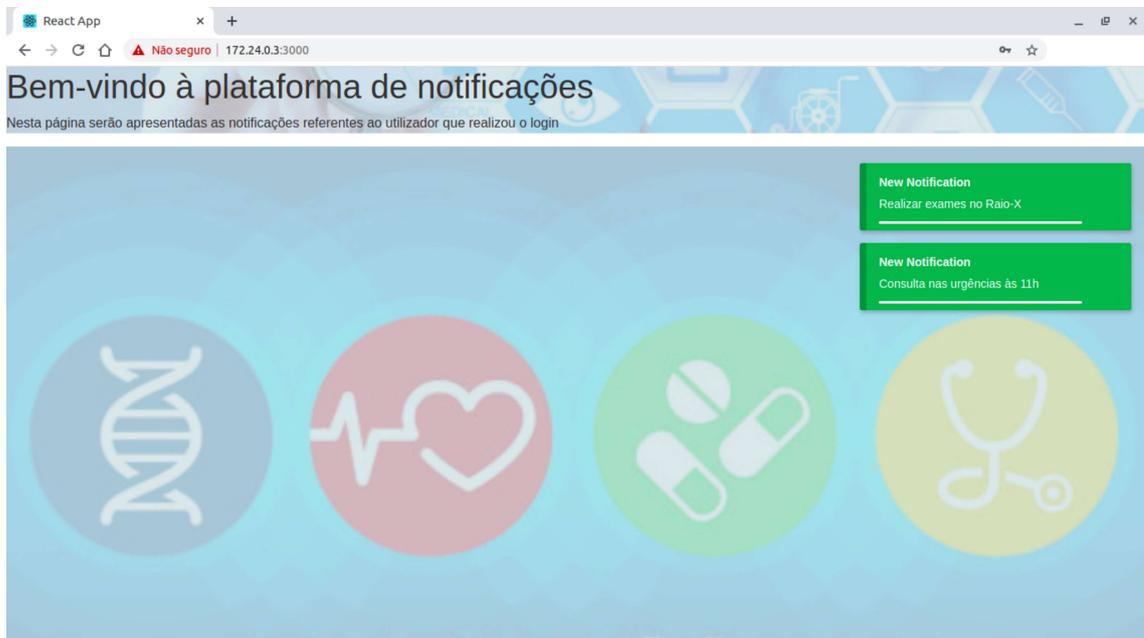


Figura 29: Página *web* com apresentação das notificações

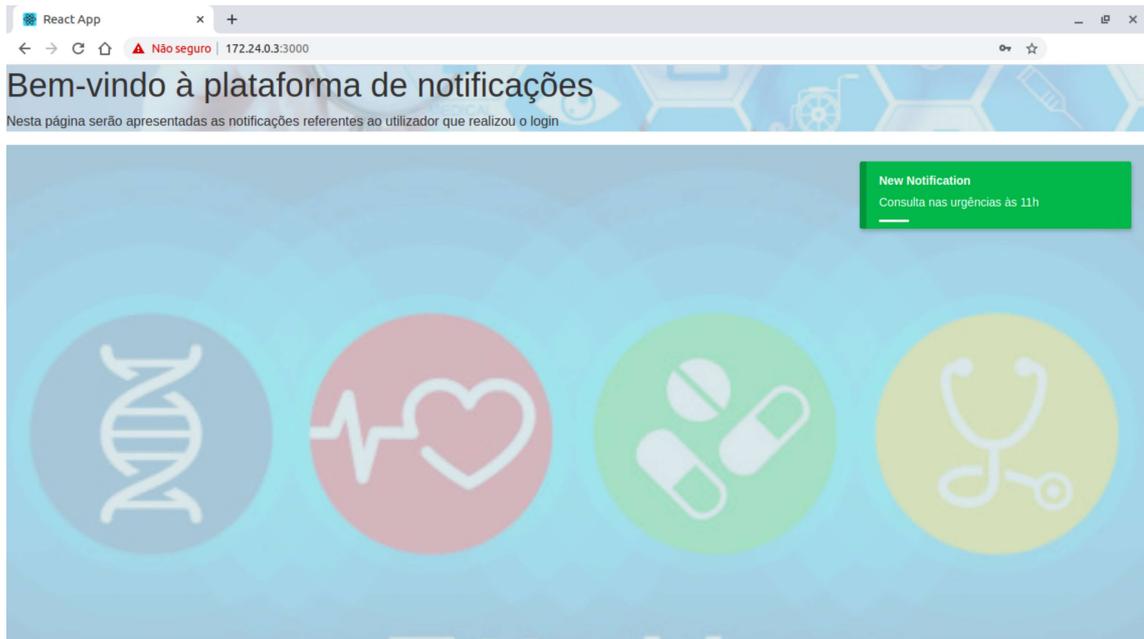


Figura 30: Página *web* com apresentação das notificações, exceto a que o utilizador efetuou o *click*

Após se verificar, que tal como na solução local, também nesta solução todos estes processos estavam a ser realizadas foram adicionadas, sobre a base de dados em utilização, novas mensagens de forma a simular a atribuição de processos e assim verificar se as mesmas eram apresentadas aos utilizadores. Dito isto espera-se que quando é efetuado o processo de procura e apresentação das mensagens, não lidas, também sejam apresentadas as mensagens que foram adicionadas. Através das figuras 30 e 31, que corresponde à apresentação de

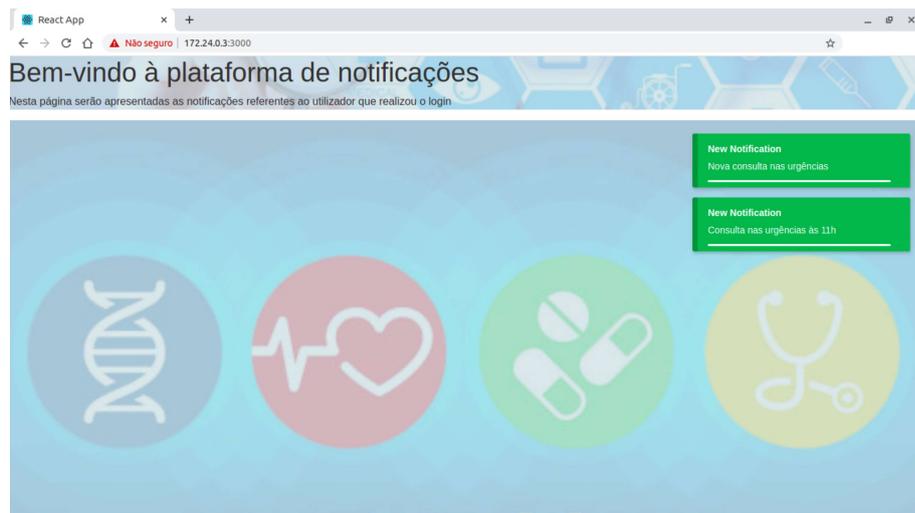


Figura 31: Página *web* com apresentação das notificações, incluindo a nova notificação inserida

mensagens para o mesmo utilizador, verifica-se que são incluídas as novas mensagens que foram inseridas anteriormente.

Além dos testes anteriores serem importantes para verificar a viabilidade da solução desenvolvida perante as questões levantadas foi fulcral a realização de mais um teste que teve por base perceber se a solução seria capaz de lidar com uma questão com que terá de lidar no dia-a-dia durante a sua utilização. Dito isto, foi realizado um teste com o propósito de verificar o comportamento perante um número significativo de utilizadores autenticados e a utilizar a solução. Desta forma conectaram-se diversos utilizadores, aproximadamente 30, à solução e tentou-se verificar se a solução continuava a desempenhar o que foi proposto e testado até ao momento desde a apresentação de mensagens corretas para cada utilizador e se as mensagens só apareciam durante 30 segundos até ser efetuada a procura pelas mensagens marcadas como não lidas a cada 60 segundos.

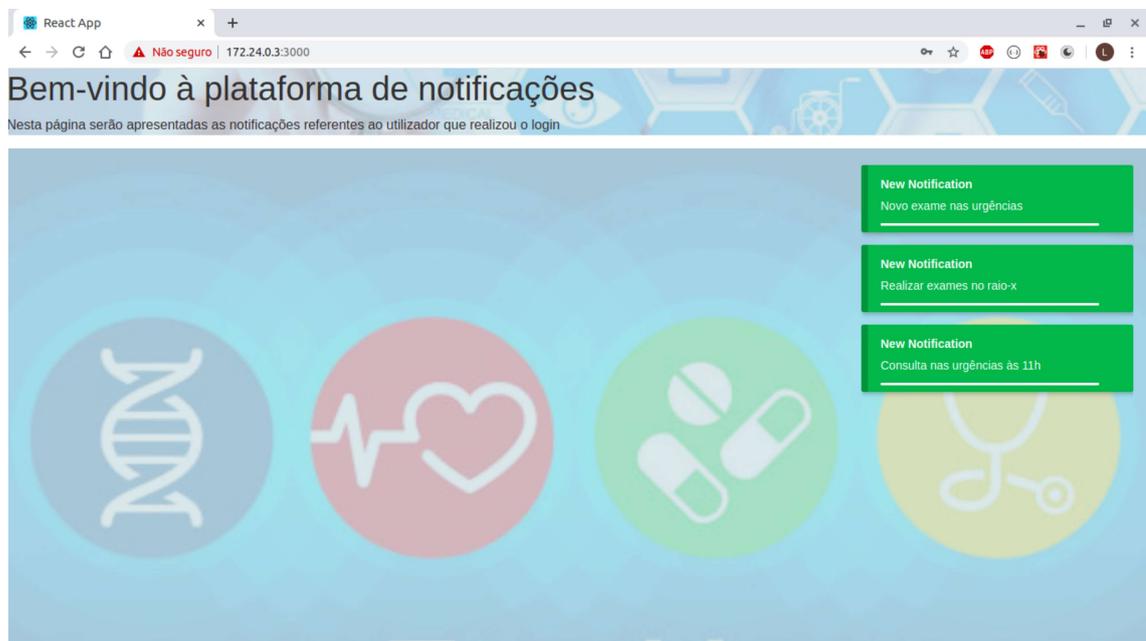


Figura 32: Página web com apresentação das mensagens com os comportamentos definidos perante a autenticação de vários utilizadores

Através da figura 32 verifica-se que efetivamente o sistema conseguiu resistir a este teste e efetuar o seu normal comportamento perante um maior número de utilizadores autenticados na solução.

---

## RESULTADOS, DISCUSSÃO E AVALIAÇÃO

---

Ao longo deste capítulo irá ser efetuada uma avaliação sobre a solução desenvolvida tendo em consideração os métodos mencionados anteriormente, no capítulo 3. Para além disto também serão discutidos os resultados obtidos assim como avaliados os testes realizados, quer tenham sido implementados localmente quer em ambiente real.

### 5.1 AVALIAÇÃO DE PERFORMANCE

Relativamente à solução desenvolvida em ambiente de desenvolvimento foram realizadas simulações com o intuito de averiguar o comportamento da solução perante os parâmetros definidos no capítulo 3. Com a realização da primeira simulação foi avaliado se era possível enviar para cada profissional somente as mensagens relativas aos processos que lhe foram atribuídos. Posto isto foi realizada a segunda simulação onde se verificou que sempre que o utilizador entrava na plataforma e não confirmava a visualização das mensagens atribuídas quando o mesmo voltava a entrar novamente na plataforma as mensagens voltavam a aparecer até que este marcasse como lida cada mensagem. Após marcar cada mensagem como lida a mesma já não aparecia ao utilizador quando o mesmo voltava a entrar na plataforma. Para além disto também foi visível que após a autenticação do utilizador na plataforma eram não só apresentadas todas as mensagens marcadas como não lidas assim como novas mensagens que foram criadas. Para a terceira simulação foi efetuada a autenticação com um utilizador e verificou-se que após 60 segundos voltavam a ser apresentadas todas as mensagens que o utilizador não marcou como lidas assim como mensagens que tinham sido introduzidas durante este intervalo de tempo. Também se verificou que todo este processo de apresentação de mensagens acontecia a cada 60 segundos que ocorria o anterior.

Para o ambiente de produção foi fulcral realizar a avaliação do desempenho da solução desenvolvida quando a mesma se depara com situações ou dificuldades que possam ocorrer no quotidiano e que não foram simuladas no ambiente desenvolvimento. Como à partida existirão diversos utilizadores autenticados em simultâneo na plataforma foi realizada uma primeira simulação onde se verificou que para vários utilizadores autenticados cada um

recebe somente as suas mensagens relativas a processos atribuídos. Já com a segunda simulação foi analisada a possibilidade de existir um número avultado de mensagens a serem apresentadas a um utilizador e portanto foi simulado este cenário onde se verificou que todas foram apresentadas ao utilizador. A terceira e última simulação efetuada teve como objetivo averiguar o comportamento da solução quando existia um maior número de utilizadores autenticados e verificou-se que o sistema era capaz de efetuar as trocas de informações corretamente e respeitando tudo o que foi testado até ao momento, incluindo o tempo de procura por novas mensagens.

Objetivos	Simulação 1	Simulação 2	Simulação 3
Enviar somente as mensagens relativas aos processos atribuídos ao profissional de saúde autenticado	Introdução de dois utilizadores e de mensagens para cada um. Foram enviadas e apresentadas as mensagens para o utilizador que se encontrava autenticado.		
Mensagens reaparecem enquanto o utilizador não confirmar que visualizou	Mensagens apareciam sempre que utilizador se autenticava		
Procurar novas mensagens a cada 60 segundos	Introdução de novas mensagens enquanto o utilizador se encontra autenticado. Apresentação de novas mensagens e de mensagens não lidas a cada 60 segundos.		

Objetivos	Simulação 1	Simulação 2	Simulação 3
Comunicar com vários utilizadores e enviar as mensagens de processos atribuídos para cada utilizador	Introdução de dois utilizadores na base de dados e das respetivas mensagens. Foram enviadas e apresentadas as respetivas mensagens para dois utilizadores que se encontravam autenticados em simultâneo.		
Apresentar todas as mensagens a um utilizador apesar de um número elevado	Foi introduzido um elevado número de mensagens para um dado utilizador. Após autenticação todas as mensagens foram apresentadas para esse utilizador.		
Elevado número de utilizadores autenticados e apresentação de mensagens sem falhas	Introduzidos diversos utilizadores na base de dados assim como a introdução de mensagens para cada um. Autenticação de diversos utilizadores em simultâneo e apresentação das respetivas mensagens a cada 60 segundos		

Tabela 2: Comparação entre resultados obtidos e os parâmetros estabelecidos no capítulo 3

De forma a ser possível obter uma outra perspetiva sobre os resultados obtidos aquando a realização das simulações, quer no ambiente de desenvolvimento quer no ambiente de produção, os mesmos encontram-se representados na tabela 2. Através desta tabela é possível averiguar que os resultados obtidos vão de encontro com os objetivos delineados uma vez que foram cumpridos.

Outra conclusão que foi possível reter após a realização das primeiras três simulações, em ambiente de desenvolvimento, é que a solução desenvolvida seria viável uma vez que tratava do problema encontrado, neste caso ser possível apresentar as mensagens dos processos atribuídos ao respetivo profissional.

Com o segundo conjunto das três simulações realizadas, em ambiente de produção, verificou-se que a solução foi capaz de ultrapassar problemas que poderiam surgir no quotidiano com a utilização de todo o sistema.

Para além das conclusões anteriores é possível verificar que a arquitetura utilizada para a implementação desta solução é viável uma vez que a utilização de componentes permite que se integre facilmente esta solução no sistema, assim como permite que seja possível realizar a alteração das tecnologias a serem utilizadas possam ser alteradas de forma bastante simples. Para além disto também permite que em caso de algum problema inesperado somente seja afetada a componente em que ocorreu o erro e não todo o sistema. Um exemplo do que foi descrito anteriormente foi a alteração da base de dados a ser utilizada onde somente o componente **backend** teve de ser alterado sem afetar os outros componentes.

## 5.2 ANÁLISE SWOT

Finalizada a avaliação da performance da solução desenvolvida, na secção anterior, foi também realizada uma análise *SWOT*. Esta análise foi realizada com o propósito de analisar na solução quais os aspetos negativos e positivos que a mesma poderia ter, quer estes fossem internos ou externos.

	Fatores Positivos	Fatores Negativos
<b>Fatores Internos</b>	<b>Forças:</b> <ul style="list-style-type: none"> <li>• Apresentação de mensagens filtradas como não lidas</li> <li>• Apresentação de um elevado número de mensagens</li> <li>• Apresentação das respetivas mensagens para cada profissional de saúde</li> <li>• Apresentação da mensagens em quase tempo-real (Possibilidade de alteração de tempo de procura por mensagens não lidas)</li> <li>• Apresentação das mensagens de acordo com o <i>RGPD</i></li> </ul>	<b>Fraquezas:</b> <ul style="list-style-type: none"> <li>• Ausência de uma funcionalidade que permita visualizar as mensagens não lidas até as mesmas surgirem novamente</li> <li>• Ausência de uma base de dados para efetuar <i>backup</i> dos dados de forma a prevenir eventuais falhas que possam surgir</li> </ul>
<b>Fatores Externos</b>	<b>Oportunidades:</b> <ul style="list-style-type: none"> <li>• Possibilidade de alteração da tecnologia que efetua a comunicação e que transmite as mensagens a serem apresentadas.</li> <li>• Ausência de funcionalidade para apresentação de mensagens já lidas</li> <li>• Inclusão de funcionalidade para priorizar apresentação de mensagens com carácter prioritário/urgente</li> </ul>	<b>Ameaças:</b> <ul style="list-style-type: none"> <li>• Falha na realização da comunicação entre as tecnologias utilizadas e por conseguinte não apresentação de mensagens</li> <li>• Alteração do <i>RGPD</i> afetando a apresentação das mensagens</li> </ul>

Tabela 3: Análise SWOT

Da realização da análise resultou a tabela 3 onde se verifica que a solução desenvolvida possui diversas forças relativamente à apresentação de mensagens onde poderá existir a necessidade de ser apresentado um elevado número de mensagens, que são somente apresentadas ao respetivo utilizador. Desta forma é dada a possibilidade de apresentação de um elevado número de mensagens para cada utilizador sendo que somente são apresentadas as mensagens que o utilizador ainda não marcou como lida de forma a facilitar consulta das

mensagens a serem apresentadas de um dado profissional de saúde. Para além disto, de forma a que o utilizador possa receber as mensagens quase em tempo-real à medida que as mesmas são geradas de processos que lhe são atribuídos existe uma funcionalidade que procura estas novas mensagens e apresenta essas novas mensagens e as mensagens que o utilizador ainda não marcou como lidas. Para este tempo de procura existe a possibilidade de alteração do mesmo podendo tornar assim essa procura mais rápida ou mais lenta. Sabendo que os dados que circulam neste sistema são dados pessoais e críticos existiu a preocupação de verificar que as mensagens a serem transmitidas aos profissionais se encontram de acordo com o *RGPD*.

Em oposição às forças existentes na solução desenvolvida, enumeradas anteriormente, também foram denotadas fraquezas, nomeadamente a ausência de uma base de dados que permita efetuar *backup* da base de dados em utilização de forma a que no caso de esta última falhar existir uma forma de recuperar os dados e de ser possível continuar a utilizar o sistema mas com a base de dados de *backup*, enquanto é resolvido o problema que ocorreu. A outra fraqueza existente na solução é ausência de uma funcionalidade que permita ao utilizador visualizar mensagens não lidas. Isto ocorre devido ao facto de as mensagens possuírem um determinado tempo para serem apresentadas e depois desaparecerem fazendo com que o utilizador tenha de esperar novamente pela procura e apresentação das mensagens não lidas para as conseguir visualizar.

Passando para os fatores externos obteve-se oportunidades e ameaças. Relativamente às oportunidades basicamente é onde são indicados pontos sobre os quais se pode tirar proveito para explorar em trabalhos futuros. Dito isto, relativamente à tecnologia que efetua a comunicação de forma a transmitir as mensagens para serem apresentadas, Websockets<sup>1</sup>, existe a possibilidade de a mesma ser alterada na eventualidade de surgir uma tecnologia que seja diferenciadora da existente e que vá de acordo com possíveis alterações que possam surgir. A alteração desta tecnologia não afeta a arquitetura utilizada apenas é necessário somente a alteração da parte que trata desta comunicação. Além desta oportunidade também existe a possibilidade de ser incluída uma funcionalidade que permita apresentar as mensagens que o utilizador já visualizou. A última oportunidade detetada na solução é a inclusão de uma funcionalidade que permita priorizar mensagens que futuramente tenham um carácter urgente ou prioritário. Dito isto apresentar em primeiro lugar estas mensagens de carácter urgente ou prioritário, quando existente, e depois serem apresentadas as restantes mensagens. Relativamente às ameaças que podem surgir de fatores externos e fazer com que seja necessário a alteração da solução desenvolvida encontra-se uma possível alteração que possa existir no *RGPD* fazendo que seja necessário a alteração da mensagem antes da mesma ser apresentada. Outra ameaça detetada é a existência de falhas de comunicação entre as diversas tecnologias que poderá afetar a apresentação das mensagens. Um exemplo desta

---

<sup>1</sup> <https://pt.wikipedia.org/wiki/WebSocket>

falha poderá ser no acesso e realização de operações sobre a base de dados que poderá fazer com que novas mensagens não sejam apresentadas assim como podem ser apresentadas mensagens marcadas como lidas mas em que o estado das mesmas não foi alterado na base de dados e portanto são apresentadas. Outro exemplo é a ocorrência de uma falha na atribuição do Websocket<sup>1</sup> que desta forma não irá apresentar as mensagens do respetivo utilizador.

---

<sup>1</sup> <https://pt.wikipedia.org/wiki/WebSocket>

---

## CONCLUSÃO E TRABALHO FUTURO

---

Este trabalho está enquadrado numa das áreas mais importantes que existem atualmente, a área da saúde. Esta é uma área onde se lida com dados pessoais e dados sensíveis e devido à entrada em vigor do *RGPD* existem alguns cuidados que se deve ter em consideração, nomeadamente no acesso e apresentação dos dados pessoais quer seja de pacientes ou de profissionais de saúde. Para além disto verifica-se um crescimento na adesão a sistemas que permitem o armazenamento digital da informação. Esta adesão deve-se ao facto de os sistemas nesta área terem um papel preponderante pois permite rentabilizar o trabalho dos profissionais e minimizar erros associados à utilização de papel para o armazenamento dos dados. Apesar das vantagens associadas à utilização deste método, este ainda não é o melhor dado que também acarreta desvantagens, nomeadamente o facto de não existir um sistema que seja universal para uso de todas as unidades de saúde. Este problema resulta do facto de não haver uma linguagem médica universal fazendo com que cada unidade de saúde procure por um subsistema próprio. De forma a combater este problema surge o *openEHR* que possibilita a criação de processos a ser utilizados num sistema global. Com a utilização destes processos é possível criar uma linguagem universal e assim criar um sistema de uso universal. Com a criação de um sistema baseado em *openEHR* existe a necessidade de apresentar aos utilizadores mensagens que indiquem aos profissionais de saúde os processos que lhes foram designados. A apresentação destas mensagens deve ser realizada quase em tempo-real ou até mesmo em tempo-real dado que após a atribuição dos processos é fulcral o conhecimento por parte dos utilizadores que os mesmos lhes foram atribuídos. De forma a ser possível resolver esta questão foi criada uma solução com uma funcionalidade que permite apresentar ao utilizador mensagens que este possua relativas aos processos que lhe foram atribuídos para que este possa visualizar e saber que os mesmos se encontram atribuídos a si.

Para que fosse possível o desenvolvimento desta solução foram analisados alguns conceitos, nomeadamente foi estudado o funcionamento dos Sistemas de Informação Hospitalar (*HIS*) e algumas das técnicas que se encontram associadas a estes sistemas de forma a auxiliar os profissionais de saúde. Também foram aprofundados conhecimentos relativos ao *openEHR* que permitiram uma melhor compreensão acerca do sistema global em que

a solução se iria enquadrar. Para além disto também foram aprofundados conhecimentos sobre várias tecnologias que poderiam ser relevantes para o desenvolvimento da solução assim como conhecimentos relevantes que se enquadravam no desenvolvimento da mesma.

Posteriormente foram identificadas quais as metodologias de investigação a utilizar para o desenvolvimento da solução assim como as metodologias de avaliação a serem utilizadas. Também nesta fase foram levantadas questões e traçados objetivos sobre os quais a solução desenvolvida seria capaz de responder. Tendo isto em conta foi definida uma arquitetura para o desenvolvimento da solução. De forma a perceber a eficácia da solução perante as questões e objetivos levantados foram realizados diversos testes. Com a realização destes testes localmente foi possível averiguar que a solução desenvolvida ia ao encontro das necessidades atuais do que era pretendido e que desta forma era possível o envio das mensagens dos processos atribuídos a cada profissional de saúde através da utilização de ferramentas de comunicação.

Terminados os testes anteriores, realizados localmente, foi possível concluir que a solução poderia ser implementada num ambiente de produção mas para tal existiu a necessidade de alteração da arquitetura que foi utilizada para o desenvolvimento dessa solução. O principal foco desta alteração encontrava-se na base de dados a ser utilizada que teve de ser alterada para Oracle<sup>1</sup>. Conforme foi alterada a base de dados também o **backend** teve de sofrer alterações dado que é onde se efetua operações sobre a base de dados. Efetuada esta alteração o resto da arquitetura manteve-se onde foi utilizado Docker<sup>2</sup> para encapsular as tecnologias utilizadas, sendo elas o NodeJS<sup>3</sup> que é responsável por todo o **backend** da aplicação, ou seja pela realização de consultas sobre a base de dados, o ReactJS<sup>4</sup> que é responsável pelo **frontend** da solução dado que é a parte responsável pela apresentação das mensagens para o utilizador e os Websockets<sup>5</sup> que efetua a comunicação entre o NodeJS<sup>3</sup> e o ReactJS<sup>4</sup> transmitindo as mensagens da base de dados para a apresentação ao utilizador. De forma a preparar a solução desenvolvida a ser integrada futuramente no sistema geral que tem por base a utilização do openEHR foi fulcral a realização de alterações de forma a facilitar a integração. Dito isto, foi necessário proceder à alteração de uma base de dados MySQL<sup>6</sup> para Oracle<sup>4</sup>. Esta mudança teve de ser efetuada dado que é a base de dados que se encontra a ser utilizada para o desenvolvimento do sistema geral e será a mesma aquando a implementação do sistema desenvolvido em ambiente real. De modo a ser possível a incorporação e utilização desta base de dados foi necessário realizar alterações, nomeadamente no **backend**, uma vez que teve de ser alterado o modo como se realiza a conexão e a conexão à base de dados. É importante frisar que relativamente a esta

---

1 <https://www.oracle.com/>

2 <https://www.docker.com/>

3 <https://nodejs.org/en/>

4 <https://reactjs.org/>

5 <https://pt.wikipedia.org/wiki/WebSocket>

6 <https://www.mysql.com/>

solução existiram algumas restrições devido à base de dados se encontrar num servidor no *Departamento de Informática da Universidade do Minho*. Desta forma foi necessária a utilização de uma *Virtual Private Network (VPN)*<sup>1</sup> para se conseguir aceder à mesma. Contudo a principal restrição verificada foi após a realização da alteração indicada anteriormente em que de forma a respeitar a arquitetura definida deveria encapsular todas as tecnologias através da utilização de docker. Contudo como o acesso à base de dados foi realizado através da *string* de conexão dificultou o processo dado o docker estar a correr com os endereços IP<sup>2</sup> localmente e para a utilização desta base de dados foi fulcral encontrar uma solução que permitisse utilizar a *string* de conexão. A solução encontrada passou pela utilização de um *container docker* que tivesse como imagem o *oraclelinux* de forma a que fosse possível a instalação do *Oracle Instant Client Oracle*<sup>3</sup> da Oracle<sup>4</sup>. Após isto foi efetuada a instalação do NodeJS<sup>5</sup> e do *node-oracledb*<sup>6</sup> que possibilitou a utilização deste *container* para **backend**. Após ter o *container* a correr foi necessário proceder às alterações no NodeJS<sup>5</sup> de forma a criar a conexão e realizar as operações realizadas na simulação local mas adaptando à nova base de dados em utilização.

Após todas as alterações enumeradas anteriormente foi possível seguir a primeira arquitetura definida tendo em conta a alteração do local aonde estavam a ser armazenadas as informações, neste caso em concreto as mensagens, e onde era fulcral aceder às mesmas de forma a ser possível apresentar ao utilizador as mensagens aquando a utilização do sistema.

De salientar que apesar de ser um dos temas mais abordados atualmente o estudo do *RGPD* enquadra-se perfeitamente na solução implementada uma vez que nos sistemas da área da saúde circulam dados pessoais e portanto sensíveis e deve-se ter um maior cuidado aquando a apresentação dos mesmos a terceiros, existindo um conjunto de regras que devem ser cumpridas antes de ser possível efetuar essa apresentação. Dito isto e como são trocadas mensagens entre sistema e utilizadores não é possível definir o que é transmitido aquando a atribuição de um processo, podendo ser transmitidos dados pessoais, quer de utentes ou de profissionais de saúde, ou não. Desta forma o *RGPD* é peça integrante e fulcral do sistema e daí a troca de mensagens ser realizada através de comunicação.

A comunicação é considerada uma funcionalidade fulcral no sistema pois permite que o sistema esteja constantemente em comunicação com o utilizador e indique ao mesmo os processos que lhe foram atribuídos de forma a que este possa decidir que processos pretende realizar ou visualizar, sendo que na área da saúde, apesar de todos os processos

---

1 [https://pt.wikipedia.org/wiki/Rede\\_privada\\_virtual](https://pt.wikipedia.org/wiki/Rede_privada_virtual)

2 [https://pt.wikipedia.org/wiki/Endereço\\_IP](https://pt.wikipedia.org/wiki/Endereço_IP)

3 <https://www.oracle.com/pt/database/technologies/instant-client.html>

4 <https://www.oracle.com/>

5 <https://nodejs.org/en/>

6 <https://www.npmjs.com/package/oracledb>

serem importantes, poderão existir alguns que requeiram uma maior atenção e a realização de ações mais rápidas por parte do profissional de saúde em questão.

Apesar das características iniciais pretendidas se encontrarem resolvidas e implementadas na solução desenvolvida existem alguns trabalhos futuros a serem realizados sobre a mesma. Dito isto, ainda existem funcionalidades que podem ser adicionadas à solução de forma a fornecer ao utilizador uma melhor experiência aquando a utilização do sistema, nomeadamente serem apresentadas mensagens que o utilizador já tenha visto e marcado como visto e também a utilização de um mecanismo que permita priorizar mensagens que necessitem da atenção do utilizador de forma mais rápida e urgente.

Esta solução apesar de concluída ainda não se encontra implementada no restante sistema dado que o mesmo ainda se encontra em desenvolvimento. Contudo a base de dados em Oracle<sup>1</sup> que se encontra a ser utilizada é a mesma que está a ser utilizada para o desenvolvimento do sistema onde se enquadra a solução e que à partida será a base de dados a utilizar aquando a conclusão do sistema. Espera-se que esta solução seja capaz de apresentar as mensagens dos processos atribuídos para cada um dos diferentes profissionais de saúde que se encontrem a utilizar a plataforma.

---

<sup>1</sup> <https://www.oracle.com/>

---

## BIBLIOGRAFIA

---

- [1] Bárbara Martins, Diana Ferreira, Cristiana Neto, António Abelha and José Machado. Data Mining for Cardiovascular Disease Prediction. *Journal of Medical Systems*, 45, 2021.
- [2] OpenEHR. [https://www.openehr.org/about/what\\_is\\_openehr](https://www.openehr.org/about/what_is_openehr).
- [3] Yousef Mehdipour, Hamideh Zerehkafi. Hospital information system (his):at a glance. [https://www.researchgate.net/publication/329029643\\_Hospital\\_Information\\_System\\_HISAt\\_a\\_Glance](https://www.researchgate.net/publication/329029643_Hospital_Information_System_HISAt_a_Glance), 2013.
- [4] Diana Ferreira, Sofia Silva, António Abelha and José Machado. Recommendation System Using Autoencoders. *Applied Sciences*, 10(16), 5510, 2020.
- [5] Ana Pinto, Diana Ferreira, Cristiana Neto, António Abelha and José Machado. Data Mining to Predict Early Stage Chronic Kidney Disease. *Procedia Computer Science*, 177, 2020.
- [6] Cristiana Neto, Maria Brito, Vítor Lopes, Hugo Peixoto, António Abelha and José Machado. Application of Data Mining for the Prediction of Mortality and Occurrence of Complications for Gastric Cancer Patients. *Entropy*, 12, 1163, 2019.
- [7] Rita Reis, Hugo Peixoto, José Machado and António Abelha. Machine Learning in Nutritional Follow-up Research. *Open Computer Science*, 7(1), 2017.
- [8] Luciana Cardoso, Fernando Marins, Ricardo Magalhães, Nivaldo Marins, Tiago Oliveira, Henrique Vicente, Antonio Abelha, José Machado and José Neves. Abstract Computation in Schizophrenia Detection through Artificial Neural Network based Systems. *The Scientific World Journal*, 2015, Article ID 467178, 2015.
- [9] Júlio Duarte, Maria Salazar, César Quintas, Manuel Santos, José Neves, António Abelha and José Machado. Data Quality Evaluation of Electronic Health Records in the Hospital Admission Process. *9th IEEE/ACIS International Conference on Computer and Information Science*, ICIS 2010.
- [10] Regulamento geral sobre a proteção de dados (RGPD). [https://pt.wikipedia.org/wiki/Regulamento\\_Geral\\_sobre\\_a\\_Prote%C3%A7%C3%A3o\\_de\\_Dados](https://pt.wikipedia.org/wiki/Regulamento_Geral_sobre_a_Prote%C3%A7%C3%A3o_de_Dados).
- [11] Serviços Partilhados do Ministério da Saúde. Privacidade da Informação no setor da Saúde. "https://ciberseguranca.spms.min-saude.pt/wp-content/uploads/2018/03/Guia-Privacidade-SMPS\_RGPD\_digital\_20.03.172-v.2.pdf", 2017.

- [12] *Privacy by Default e Privacy by Design*. <https://www.rbconsulting.pt/2018/09/10/rgpd-privacy-by-design-desenvolvimento-de-software>.
- [13] Serviço nacional de saúde (SNS). <https://www.sns.gov.pt/>.
- [14] Estrutura de governança e gestão do eSIS. <https://enesis.spms.min-saude.pt/2017/06/26/estruturas-de-governacao-e-gestao-do-esis/>.
- [15] *Privacy Impact Assessment* (PIA). [https://en.wikipedia.org/wiki/Privacy\\_Impact\\_Assessment](https://en.wikipedia.org/wiki/Privacy_Impact_Assessment).
- [16] Comissão nacional de proteção de dados (CNPd). <https://www.cnpd.pt/bin/cnpd/acnpd.htm>.
- [17] Espera-se também que a Diretiva e-Privacy e, por conseguinte, a lei n.º 41/2004, de 18 de agosto (alterada pela lei n.º 46/2012, de 29 de agosto), venha a sofrer alterações de forma a ser compatibilizada com o RGPD. <https://dre.pt/pesquisa/-/search/174793/details/maximized>.
- [18] O n.º 4 do artigo 9.º do Regulamento permite que os Estados-Membros mantenham ou imponham “novas condições, incluindo limitações, no que respeita ao tratamento de dados genéticos, dados biométricos ou dados relativos à saúde.  
. <http://www.privacy-regulation.eu/pt/9.htm>.
- [19] Por motivos de interesse público no domínio da saúde pública, nos termos do artigo 9.º, n.º 2, alíneas h) e i), bem como do artigo 9.º, n.º 3; <http://www.privacy-regulation.eu/pt/17.htm>.
- [20] Tratamento de dados pessoais de forma a que deixem de poder ser atribuídos a um titular de dados específico sem recorrer a informações suplementares, desde que essas informações suplementares sejam mantidas separadamente e sujeitas a medidas técnicas e organizativas para assegurar que os dados pessoais não possam ser atribuídos a uma pessoa singular identificada ou identificável. <http://www.openlimits.pt/pt/thinking-ahead-blog/glossario-rgpd-regulamento-europeu-protacao-dados/?all=1>.
- [21] Diretiva RSI/NIS. <https://www.cncs.gov.pt/transposicao-da-diretiva-nissri/>.
- [22] Presidente da Comissão dos Assuntos Constitucionais, Direitos, Liberdades e Garantias. Os valores das sanções encontram-se no prómio dos números 4 e 5 do artigo 83.º do RGPD. <http://app.parlamento.pt/webutils/docs/doc.pdf?path=6148523063446f764c324679626d56304c334e706447567a4c31684a53556c4d5a5763765130394e4c7a4644fich=cef7a328-6d4e-4e59-b044-b50f92527a31.pdf&Inline=true>, 2018.
- [23] EHR Extract. [https://specifications.openehr.org/releases/RM/latest/ehr\\_extract.html](https://specifications.openehr.org/releases/RM/latest/ehr_extract.html).

- [24] Peter Schloeffel, Thomas Beale, George Hayworth, Sam Heard, Heather Leslie. The relationship between cen 13606, hl7, and openehr. [https://www.researchgate.net/publication/228372222\\_The\\_relationship\\_between\\_CEN\\_13606\\_HL7\\_and\\_openEHR](https://www.researchgate.net/publication/228372222_The_relationship_between_CEN_13606_HL7_and_openEHR), 2006.
- [25] Arquitetura do openEHR. [https://specifications.openehr.org/releases/BASE/Release-1.0.3/architecture\\_overview.html](https://specifications.openehr.org/releases/BASE/Release-1.0.3/architecture_overview.html).
- [26] In the field of informatics, an archetype is a formal re-usable model of a domain concept. [https://en.wikipedia.org/wiki/Archetype\\_\(information\\_science\)](https://en.wikipedia.org/wiki/Archetype_(information_science)).
- [27] SNOMED. [https://en.wikipedia.org/wiki/SNOMED\\_CT](https://en.wikipedia.org/wiki/SNOMED_CT).
- [28] Carl Lawrence, Tuure Tuunanen, and Michael Myers. Extending design science research methodology for a multicultural world. *IFIP Advances in Information and Communication Technology*, 2010.
- [29] Jeff Sutherland and Ken Schwaber. Nut, bolts, and origins of an agile framework. *The Scrum Papers*, 2007.
- [30] Scrum. [www.scrumguides.org](http://www.scrumguides.org).
- [31] Framework scrum. <https://blog.4geeks.io/2019/11/scrum-methodology.html>.
- [32] Methods of communication. <http://html5doctor.com/methods-of-communication/>.
- [33] Xmlhttprequest. <https://en.wikipedia.org/wiki/XMLHttpRequest>.
- [34] Post message. <https://developer.mozilla.org/pt-PT/docs/Web/API/Window/postMessage>.
- [35] Post message sender and receiver. <https://davidwalsh.name/window-postmessage>.
- [36] Web sockets. <https://blog.stanko.io/do-you-really-need-websockets-343aed40aa9b>.
- [37] Server-sent events. <https://king.host/blog/2018/03/server-sent-events-o-que-e-e-como-usar/>.
- [38] Comparação entre sse e web sockets. <https://www.smashingmagazine.com/2018/02/sse-websockets-data-flow-http2/>.
- [39] Gilkey Herbert T (1960). *New Air Heating Methods*. Washington, 1960, November 1959.
- [40] Cheonl-Han Kim Chankwon Park Hyeong-Cheol Woo Han-Il Jeong, Chang-Soo Lee. Design of a software component bank for distribution. [https://www.researchgate.net/publication/220588371\\_Design\\_of\\_a\\_Software\\_Component\\_Bank\\_for\\_Distribution](https://www.researchgate.net/publication/220588371_Design_of_a_Software_Component_Bank_for_Distribution), 2001.
- [41] Docker. <https://www.docker.com/>.

- [42] Container. <https://www.docker.com/resources/what-container>.
- [43] Virtual machine. [https://minds.wisconsin.edu/bitstream/handle/1793/11154/file\\_1.pdf?sequence=1&isAllowed=y](https://minds.wisconsin.edu/bitstream/handle/1793/11154/file_1.pdf?sequence=1&isAllowed=y).
- [44] Pandya M. Cloud computing for libraries: A swot analysis. 2012.