Projeto de Mestrado em

GESTÃO DE SISTEMAS DE INFORMAÇÃO MÉDICA

APLICAÇÕES DE TELEMEDICINA USANDO WEBRTC

Joaquim José Abrantes Pinto Bernardes Barranca

Leiria, Março de 2016

Projeto de Mestrado em

GESTÃO DE SISTEMAS DE INFORMAÇÃO MÉDICA

APLICAÇÕES DE TELEMEDICINA USANDO WEBRTC

Joaquim José Abrantes Pinto Bernardes Barranca

Projeto de Mestrado realizado sob a orientação da Doutora
Catarina Helena Branco Simões Silva e do Doutor Mário João Gonçalves Antunes,
docentes da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Agradecimentos

Ao Doutor Rui Pedro Charters Lopes Rijo, Coordenador do Curso de Mestrado em Gestão de Sistemas de Informação Médica pelas palavras de incentivo.

Aos orientadores Doutora Catarina Helena Branco Simões Silva e Doutor Mário João Gonçalves Antunes pela orientação e motivação.

Ao Instituto Politécnico de Leiria e à Escola Superior de Tecnologia e Gestão de Leiria, pelas excelentes condições que sempre colocaram ao dispor para a realização deste trabalho.

À minha família, em especial à minha esposa Leonor e nossos filhos Bernardo e Gustavo pelos muitos e longos momentos de privação.

Aos colegas de mestrado pelos momentos vividos.

A todos aqueles que não estando aqui discriminados, contribuíram para que este trabalho fosse uma realidade.

Esta página foi intencionalmente deixada em branco.

Resumo

O uso das tecnologias de informação e comunicação em Portugal na área médica tem tido um grande aumento nas últimas décadas. Tal pode constatar-se a vários níveis, como sejam a implementação crescente e em larga escala de sistemas de informação em hospitais e centros de saúde, o desenvolvimento de aplicações para auxiliar a análise dos principais meios complementares de diagnóstico, a receita eletrónica e o registo de paciente eletrónico, apenas para citar alguns exemplos.

A procura crescente de profissionais de saúde de várias especialidades tem tido um aumento considerável nos últimos anos. Por um lado, a população está cada vez mais envelhecida, por via do aumento da esperança média de vida. Por outro lado tem aumentado a preocupação com a saúde e o bem estar próprio dos cidadãos, levando-os a recorrer mais vezes e a mais especialidades do que no passado.

O êxodo da população do interior do país para os grandes centros no litoral, complementado pelas políticas orçamentais restritivas na área da saúde, tem acentuado as diferenças de prestação de cuidados de saúde a toda a população do país, de forma equitativa e eficaz. Para tal tem ainda contribuído o emagrecimento dos orçamentos dos hospitais e a pressão para que estes cumpram as metas de produtividade definidas, com custos cada vez mais reduzidos.

Um dos contributos das tecnologias de informação para mitigar o afastamento entre o paciente e os profissionais de saúde, consiste na implementação de soluções de consulta "à distância", com a utilização de vídeo e voz, através de aplicações de telemedicina.

Ao nível da teleconsulta e da telemedicina têm existido alguns avanços significativos, sendo possível encontrar alguns casos de sucesso na utilização destes meios para facilitar o acesso generalizado de toda a população a cuidados médicos de saúde. Constata-se contudo que as aplicações usadas são geralmente proprietárias, carecem de instalação de software

específico, muitas vezes proprietário e por vezes com custos para as entidades que disponibilizam o serviço. Por exemplo, a utilização de uma ligação por *Skype* para uma teleconsulta obriga a que a aplicação esteja instalada em ambos os computadores (médico e paciente).

Nesta dissertação apresenta-se uma solução de telemedicina baseada na Application Programming Interface (API) Web Real-Time Comunication (WebRTC), que permite o envio de voz e imagem entre dois *browsers* usando os protocolos de comunicação na *Web*. Além do vídeo e da voz foram integrados na aplicação duas funcionalidades particularmente interessantes numa teleconsulta: envio bidirecional de ficheiros (por exemplo, ficheiro PDF com o resultado das últimas análise que o paciente realizou) e desenho num "quadro branco", permitindo ao paciente ou ao médico ilustrarem de forma livre algum aspeto associado à consulta em causa.

A aplicação utiliza exclusivamente componentes de software *opensource* e apenas necessita que ambos os computadores tenham instalado um *browser* de acesso à *Web* que suporte a comunicação por WebRTC, como o *Google Chrome* ou o *Firefox*. Pretende-se desta forma facilitar o acesso aos serviços de telemedicina evitando a instalação e configuração de software específico, bem como reduzir os custos através de soluções *opensource* com licença General Public License (GPL) e isenta de custos.

Foram realizados alguns testes de aceitação da solução, em ambiente hospitalar. Genericamente, pretendeu-se validar o funcionamento da API WebRTC, aferir sobre a aceitação das funcionalidades implementadas e identificar obstáculos técnicos à sua implementação na rede de um hospital ou centro de saúde. Embora tenham sido identificados alguns problemas na comunicação, resultantes maioritariamente do tipo de configurações da rede em que os computadores estavam instalados, os resultados globais obtidos são bastante promissores, dando-nos boas perspetivas quanto à sua implementação em ambiente de produção.

Palavras-chave: WebRTC, Telemedicina, Teleconsulta, Comunicação em tempo real

Abstract

In Portugal, the use of Information and Communication Technologies (ICT) within the medical area has experienced a large increase in recent decades. This can be observed at various levels, such as the increasing implementation in large-scale of the information systems in Hospitals and Health Centers, the applications development to support the analysis of the main complementary means of diagnosis, the electronic prescription and also the electronic patient registration, just to mention a few examples.

The growing demand for health professionals of different medical specialties has had a considerable increase in the last years. On one hand, the population is growing older due to a longer average life expectancy. On the other hand, there has been a noticeable increasing of the citizens concern for their own health and welfare, leading them to more frequent use of further medical specialties than in the past.

The exodus of the population from the interior to the large cities on the coast, complemented with the restrictive fiscal policies in health sector, have emphasized differences in health care provision to the entire country population, in a fair and effective manner. Furthermore, the hospitals budgets are being squeezed more tightly, along with the intense pressure to meet productivity targets with a lower cost-effective manner.

One of the Information and Communication Technologies contributions to mitigate the gap between the patient and health professionals, is the implementation of distance medical health care solutions, through the use of voice and video via telemedicine applications.

Teleconsultation and telemedicine has suffered significant advances, and it can be found some success stories in using these means to facilitate widespread population access to medical health care. However used applications are often proprietary, requires specific software

installation and in many cases with high cost. For example, if skype is used for a teleconference call between a healthcare professional and a patient, it is necessary that Skype is installed in both computers.

It is presented in this dissertation a telemedicine application based on the Web Real-Time Communication (WebRTC) Application Programming Interface (API), which permits sending voice, image and data between two browsers using communication protocols over the WEB. Beside audio and video, it were applied others particularly interesting functionalities to teleconsultation like the exchange of files, for example a PDF file with the last medical analyses results that the pacient did and a whiteboard like functionality that enables sharing and discussing the exams.

This application only uses open source software components and only requires that both computers have installed a Web browser that support WebRTC, like Google Chrome or Mozilla Firefox. The application's intended is to facilitate access to telemedicine services, avoiding specific software installation and configuration, as well as reduce costs through open source applications with General Public License (GPL) and costless.

The applications has been tested in hospital environment. It's supposed to validate WebRTC API and check on the implemented features acceptance and identify technical barriers to implementation on a hospital or health center networks. Although it has been identifying some communications issues most of which are related with network configurations, the overall results are very promising, giving us good reasons to intensify architecture testing, and subsequent implementation in a production environment.

Keywords: WebRTC, Telemedicine, ehealth, real time communication

Lista de Figuras

Lista de códigos

Esta página foi intencionalmente deixada em branco.

Acrónimos

ACES Agrupamento de Centros de Saúde

ACSS Administração Central do Sistema de Saúde

AES Advanced Encryption Standard

ANPC Autoridade Nacional Proteção Civil

API Application Programming Interface

ATA American Telemedicine Association

AVCia Acidente Vascular Cerebral isquêmico agudo

CCTV Close Circuit TeleVision

CPAP Continuous Positive Airway Pressure

CSP Cuidados de Saúde Primários

CSS Cascading Style Sheets

CTH Consulta a Tempo e Horas

CU-RTC-WEB Customizable, Ubiquitous Real Time Communication over the WEB

CVP Cruz Vermelha Portuguesa

DGS Direção Geral da Saúde

DPOC Doença Pulmonar Obstrutiva Crónica

DTLS Datagram Transport Layer Security

ECG EletroCardioGrama

EHR Electronic Health Record

ESTG Escola Superior de Tecnologia e Gestão

FTP File Transfer Protocol

GB Giga Bytes

GPL General Public License

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

HTTP/2 Hypertext Transfer Protocol Version 2

HTTPS HyperText Transfer Protocol Secure

IANA Internet Assigned Numbers Authority

ICE Interactive Connectivity Establishment

IETF Internet Engineering Task Force

IIS Internet Information Services

INEM Instituto Nacional de Emergência Médica

IP Internet Protocol

IPL Instituto Politécnico de Leiria

JSEP JavaScript Session Establishment Protocol

Kbps Kbits por segundo

LDAP Lightweight Directory Access Protocol

MAC Message Authentication Code

MB Mega Bytes

NAPT Network Address Port Translation

NAPT-T Network Address Port Translation - Traversal

NASA National Aeronautics and Space Administration

NAT Network Address Translation

NAT-T Network Address Translation - Traversal

OCDE Organização para a Cooperação e Desenvolvimento Económico

OMS Organização Mundial de Saúde

ORTC Object Real-Time Communications

OSI Open Systems Interconnection

P2P Peer to Peer

PALOP Países Africanos de Língua Oficial Portuguesa

PAT Port Adress Translation

PATIB Passaporte Telemedicina Ibero-brasileiro

PDS Plataforma de Dados da Saúde

RAM Random Access Memory

RCE Registos Clínicos Eletrónicos

RFC Request For Comments

RIS Rede Informática da Saúde

ROI Return On Investment

RTCP RTP Control Protocol

RTP Real-time Transport Protocol

SAVPF Secure Audio Video Profile with Feedback

SCTP Stream Control Transmission Protocol

SDK Software Development Kit

SDP Session Description Protocol

SIEM Sistema Integrado de Emergência Médica

SIP Session Initiation Protocol

SMTP Simple Mail Transfer Protocol

SNS Serviço Nacional de Saúde

SPMS Serviços Partilhados do Ministério da Saúde

SRTP Secure Real-time Transport Protocol

SSD Solid State Drive

STARPAHC Space Technology Applied to Rural Papago Advanced Health Care

STUN Session Traversal Utilities for NAT

SWOT Strengths Weaknesses Opportunities Threats

TCP Transmission Control Protocol

TCP/IP Transmission Control Protocol / Internet Protocol

TIC Tecnologias de Informação e Comunicação

TLS Transport Layer Security

TURN Traversal Using Relay NAT

UCSP Unidade de Cuidados de Saúde Personalizados

UDP User Datagram Protocol

URL Uniform Resource Locator

VoIP Voice over IP

W3C World Wide Web Consortium

WebRTC Web Real-Time Comunication

WHATWG Web Hypertext Application Technology Working Group

WS Web Socket

WSS Web Socket Secure

WWW World Wide Web

Esta página foi intencionalmente deixada em branco.

Conteúdo

Capítulo 1

Introdução

Este trabalho foi realizado no âmbito da tese do Mestrado em Gestão de Sistemas de Informação Médica, na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria em associação com a Faculdade de Medicina da Universidade do Porto subordinada ao tema Aplicações de Telemedicina usando WebRTC.

Telemedicina é um termo híbrido que resulta da junção do prefixo "tele"do grego que significa "distante"ou "à distância"com o termo "medicina". Ou seja, significa literalmente medicina à distância, incluindo um conjunto alargado de conceitos e soluções que, através de sistemas áudio, vídeo e outras tecnologias de processamento de informação e comunicação, permitem prestar serviços na área da medicina.

O trabalho desenvolvido insere-se assim na área da telemedicina. Foca-se na investigação aplicada em tecnologias de informação e comunicação, com vista ao desenvolvimento de aplicações para telemedicina.

1.1 Enquadramento e Motivação

De uma forma global, governos e instituições defendem que o futuro da prestação de cuidados de saúde deverá passar pela telemedicina. Concretamente, a Comissão Europeia preconiza o desenvolvimento da telemedicina como parte integrante do seu **Plano de Acção de e-Saúde 2012-2020**, pela sua integração na sua linha de ação "Dotar os Europeus de acesso

online seguro aos seus dados médicos e atingir a implementação alargada da telemedicina"[1]. De acordo com a agenda da Comissão Europeia, esta ação visa desenvolver mecanismos para dotar os cidadãos europeus com acessos seguros aos seus dados de saúde em 2015 e atingir, até 2020, a implementação generalizada de serviços de telemedicina[Commission, 2012].

As motivações para esta necessidade, e mesmo urgência, são bastante heterogéneas. Se, por um lado, a sustentabilidade dos sistemas de saúde está em risco e a telemedicina pode ser encarada como uma forma de racionalizar custos, numa altura em que tal é visto como incontornável, por outro lado o aumento da exigência dos pacientes, a disseminação crescente de doenças crónicas e o impacto do envelhecimento da população são fatores decisivos de avaliação da qualidade dos serviços de saúde em que a telemedicina pode desempenhar um papel fundamental.

Adicionalmente, os recursos humanos e financeiros escasseiam de uma forma global, colocando pressão suplementar na sustentabilidade dos sistemas de saúde. Esta pressão tem levado à procura de novos modelos de cuidados de saúde e envolvimento de todos os *players*, que são essenciais para garantir a sustentabilidade de uma sociedade mais saudável.

Nas sociedades ocidentais, o envelhecimento da população e a concentração de especialistas de cuidados de saúde nos hospitais centrais, leva a que, racionalmente, se desenvolvam mecanismos de comunicação rápidos e eficazes, de forma a que os prestadores em hospitais remotos, em centros de cuidados de saúde primários, na rede de cuidados continuados e outras instituições presentes nos sistemas de saúde, consigam obter segundas opiniões de casos de saúde mais graves ou menos comuns, além de prestarem do mesmo modo cuidados médicos.

Com este enquadramento e motivação, este trabalho visa explorar as tecnologias que permitam suportar o avanço da telemedicina, impondo por um lado o mínimo de restrições possível, mas por outro tentando proporcionar o maior número de serviços a prestadores e populações.

[1]No original: *Action 75 - Give Europeans secure online access to their medical health data and achieve widespread telemedicine deployment*.

1.2 Objetivos

No âmbito deste projeto foi delineado um conjunto de objetivos que se enumeram de seguida:

- Análise do estado da arte na área da telemedicina: o objetivo específico deste ponto passa por adquirir conhecimentos base nesta área, caracterizar a situação atual, nomeadamente em Portugal e identificar os maiores desafios neste momento, de forma a que se consiga definir uma estratégia de investigação por um lado implementável e, por outro, útil no contexto atual;
- Análise da tecnologia WebRTC: sendo uma tecnologia recente, ainda em evolução e pouco explorada a nível da telemedicina, este ponto pretende definir as capacidades e limitações do WebRTC, de forma a que possam ser melhor exploradas e/ou evitadas se possível na solução a propor;
- Solução de telemedicina baseada em WebRTC: o ponto primordial deste projeto, ou seja, a definição de uma arquitetura de comunicação e aplicacional que permita a sua implementação numa solução de telemedicina;
- Protótipo com a implementação da solução proposta: o ponto central deste trabalho, que inclui a implementação da solução ao nível da infraestrutura de comunicação, da API do WebRTC e das interfaces dos utilizadores (prestadores de serviços e pacientes).
- Testes em ambientes reais que realizem a prova de conceito da solução implementada.

1.3 Metodologia e estrutura do documento

De acordo com os objetivos definidos para o projeto, a primeira fase consistiu em realizar uma análise das formas de telemedicina. O Capítulo 2 apresenta os aspetos mais relevantes da telemedicina que servem de base à aplicação deste trabalho, incluindo a sua história, o estado atual e os principais desafios.

De seguida, no Capítulo 3, investigou-se com profundidade a tecnologia WebRTC, que suportará a solução a propor no capítulo seguinte. Neste capítulo expõe-se os conceitos necessários para a compreensão do trabalho apresentado ao longo do documento, nomeadamente a tecnologia WebRTC, as vantagens e desvantagens associadas, bem como as questões relacionadas com a segurança, a privacidade e o funcionamento do WebRTC.

Com base na recolha e análise das informações anteriores, no Capítulo 4 a solução proposta é especificada sendo apresentada a sua arquitetura. São ainda estabelecidas as várias características e funcionalidades, bem como a implementação do sistema. A solução implementada é analisada de forma detalhada e é apresentada a estrutura de implementação. Vários detalhes da implementação são expostos e documentados de acordo com os requisitos e necessidades do sistema, nomeadamente exemplos de código com funções mais importantes. Neste capítulo estão ainda incluídos alguns dos testes levados a cabo como forma de prova de conceito da solução proposta.

No capítulo final, apresentamos as principais conclusões do trabalho, bem como propostas de linhas de trabalho futuro.

Capítulo 2

Telemedicina

Neste capítulo serão abordados os aspetos mais relevantes da telemedicina que servem de base a este trabalho. Serão descritos os conceitos e definições fundamentais relacionadas com a telemedicina, bem como a sua história até ao momento. Apresentam-se ainda os principais desafios relacionados com a implementação deste tipo de aplicação, de forma a enquadrar o trabalho realizado.

2.1 Introdução

Existem várias definições de telemedicina propostas por diversos autores e entidades, não diferindo muito umas das outras. Segundo a Organização Mundial de Saúde (OMS) a telemedicina, envolve a prestação de cuidados de saúde, usando as Tecnologias de Informação e Comunicação (TIC) especialmente onde a distância é uma barreira a esses cuidados [Organization-WHO, 2011].

A telemedicina tem como principal objetivo a prestação de cuidados clínicos, de forma a melhorar a saúde dos pacientes, através da interligação dos utilizadores que estejam geograficamente separados. Consiste, assim, na transferência de informações médicas através de diversos meios como correio eletrónico, telefone, telemóvel e videoconferência. Podem ser utilizados vários meios físicos disponíveis, como a palavra escrita ou falada, sondas, digitalizadores de imagem ou versões eletrónicas de instrumentos correntes. A telemedicina pode ser tão simples como dois profissionais de saúde discutirem um caso ao telefone, ou tão sofisticada como uma cirurgia levada a cabo através de um robot controlado remotamente entre dois pontos geograficamente distantes.

2.2 Tipos de telemedicina

A telemedicina é praticada tendo por base dois conceitos fundamentais, nomeadamente *real time* (síncrona) e *store and forward* (assíncrona) conforme ilustrado nas Figuras 2.1 e 2.2 [Freitas, 2005] [Gravenstein et al., 1974]. Estes dois conceitos diferem fundamentalmente na altura em que a informação é transmitida e na interação entre os envolvidos, seja entre profissionais de saúde, ou entre paciente e profissional de saúde.

A telemedicina em tempo real, ou síncrona, apresentada na Figura 2.1, assume que todas as partes envolvidas na comunicação estão presentes simultaneamente e que o canal de comunicação está disponível para todos em cada momento. Pressupõe interatividade entre as partes e é utilizada, por exemplo, quando é necessária uma consulta em tempo real em que o médico e o paciente têm de se ver mutuamente.

Atualmente a videoconferência, independentemente do nível de tecnologia envolvida, é a forma mais comum de efetuar telemedicina em tempo real. Existem diversos dispositivos que se podem adicionar à videoconferência de forma a obter um sistema mais completo. Por exemplo, com a adição de um telestereoscópio, o médico poderá analisar remotamente e em tempo real o ouvido de um paciente. Este conceito tem a vantagem de ser interativo e assim a resposta ter efeitos imediatos.

O conceito abordado neste trabalho insere-se na telemedicina síncrona, ou por outras palavras, telemedicina em tempo real, uma vez que há interação em tempo real entre os utilizadores do sistema.
A Figura 2.1 representa o fluxo do conceito da telemedicina em *real time*, onde é percetível o sincronismo entre os atores da comunicação. Neste exemplo, os exames são enviados ao centro de análises onde os seus resultados são examinados e discutidos em tempo real entre os centros de análise e exames.

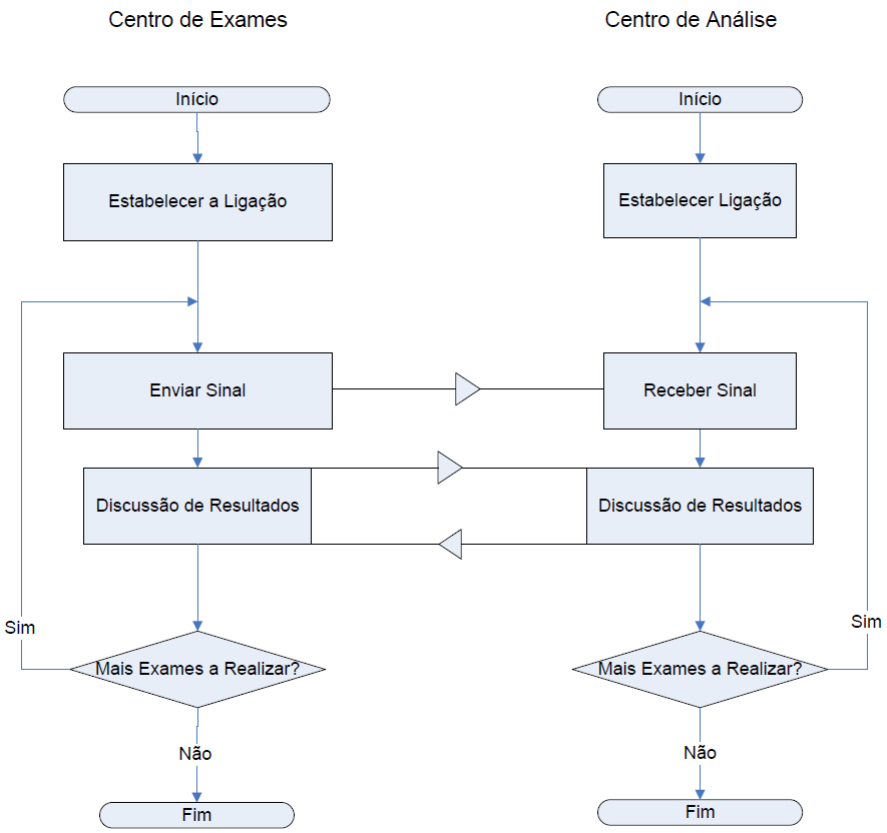

Figura 2.1: Telemedicina Real Time (síncrona)

Na telemedicina do tipo *Store and forward*, ou assíncrona, ilustrada na Figura 2.2, a informação após ser gerada (por exemplo uma radiografia), é guardada num repositório e só mais tarde acedida pelo médico ou por outro profissional. Com este conceito não há a obrigatoriedade da presença simultânea dos intervenientes. O paciente e o profissional de saúde podem, por isso, não estar presentes simultaneamente. É utilizado tipicamente em cenários de não emergência e essencialmente para consulta de meios auxiliares de diagnóstico. A dermatologia e radiologia são especialidades que recorrem normalmente a este conceito sendo efetuado o exame que posteriormente é consultado, de forma a auxiliar o diagnóstico.

Na Figura 2.2, é representada a sequência de ações levadas a cabo no tipo de telemedicina *store and forward*. É possível observar que há uma janela temporal desde o momento em que se envia o exame para o centro de análises até que este transmite os resultados à origem.

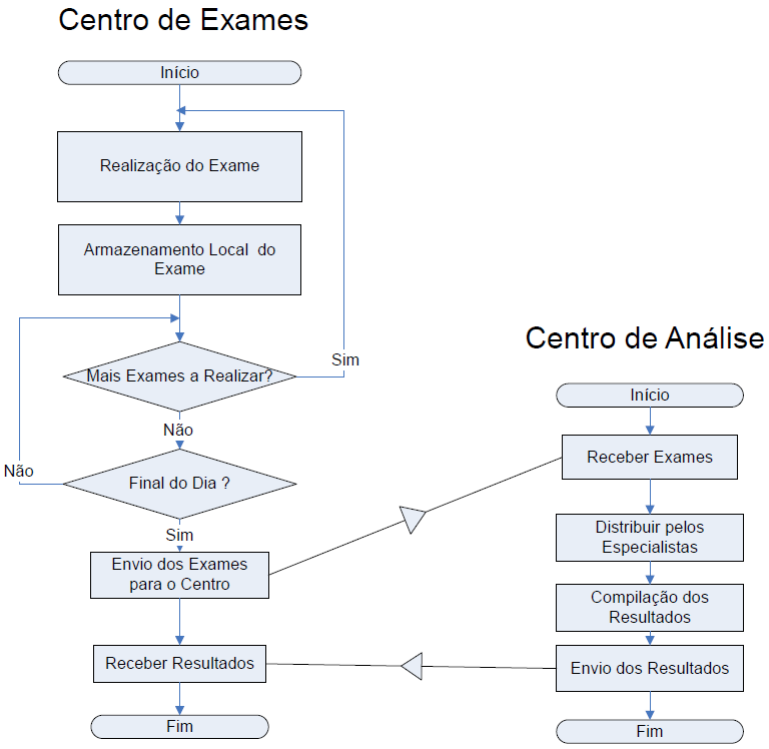

Figura 2.2: Telemedicina Store and Forward (assíncrona)

Um sistema mais completo poderá incluir os dois tipos de telemedicina. Por exemplo, com a recente inclusão dos registos clínicos eletrónicos, qualquer médico tem a possibilidade de consultar exames realizados anteriormente e os seus resultados, além de conseguir verificar as consultas realizadas em todo o Serviço Nacional de Saúde (SNS). Este sistema integrado tem a vantagem de necessitar de requisitos tecnológicos menores e de menor exigência de largura de banda, sendo por isso mais económico e de simples implementação.

2.3 História da telemedicina

Em Abril de 1924, a capa da revista *Radio News* (Figura 2.3) representava uma visão futurista, para a época, da telemedicina. Esta imagem, caricaturando a prática de então, conseguiu antever a realidade atual, identificando os componentes de uma consulta de telemedicina em tempo real, nomeadamente o médico distante do utente, efetuando o diagnóstico e emitindo a prescrição.

Figura 2.3: Visão da telemedicina em 1924

Pese embora os maiores desenvolvimentos da telemedicina terem ocorrido indiscutivelmente nas ultimas décadas, o conceito é bastante mais antigo. É, por exemplo, conhecido de há muitos anos o uso de sinais de fumo e sinais sonoros de sinos e troncos de árvore para transmitir informações em situações de emergência.

Tal como a conhecemos hoje, a telemedicina é facilmente referenciada até aos anos 70 e 80 do século XX, fundamentalmente devido à evolução dos computadores pessoais. Mas foi essencialmente nos anos 90, com a massificação do uso das telecomunicações e da Internet, que se assistiu ao seu maior desenvolvimento e crescimento. Como consequência do desenvolvimento, registe-se o aparecimento de aplicações de telemedicina como a teleradiologia, a telepatologia e a teledermatologia.

Os maiores avanços da telemedicina são marcados fundamentalmente por importantes marcos no desenvolvimento das tecnologias usadas para a comunicação das informações médicas [Ferrer-Roca and Sosa-Iudicissa, 1998], como se descreve de seguida.

A telemedicina pré-eletrónica era efetuada em épocas remotas, como já foi referido anterior-

mente, com recurso a sinais de fumo para comunicar um surto de peste numa povoação, ou ainda no método de troca de informações médicas via correio postal, como prescrições ou indicações terapêuticas.

Um dos grandes avanços na telemedicina deu-se quando surgiu a eletrónica. Nessa altura, as novas formas de comunicação deram suporte a novos procedimentos na telemedicina. Assim, a telemedicina teve duas fases distintas. Uma com comunicações analógicas, através da utilização do telégrafo, telefone e rádio; a outra baseada em comunicações digitais, aquela em que nos encontramos atualmente. A distinção destas duas fases não está claramente definida, uma vez que os equipamentos que as suportam, originalmente analógicos, são idênticos e alguns foram convertidos para digitais, mantendo a sua forma e funcionalidade, como por exemplo o telefone.

O telégrafo, invenção do século XVIII, foi usado para transmitir informações médicas de que são exemplo os raios-X. Na guerra civil Americana, entre 1861 e 1865, [Zundel, 1996] foi novamente usado o telégrafo para transmitir listas de mortos e feridos e para acionar o fornecimento de medicamentos.

Com o aparecimento do telefone em finais do século XIX, este foi adotado desde cedo pelos profissionais de saúde para uso em aplicações de telemedicina. Usado não só para as normais comunicações de voz, foi ainda utilizado para transmitir sons amplificados de tele-estetoscópio de e para áreas remotas. Numa fase mais recente, as linhas telefónicas passaram a suportar o uso de modems para a transmissão de eletrocardiogramas.

Ainda nos finais do século XIX surgiram as comunicações via rádio frequência, inicialmente usadas de forma idêntica ao telégrafo, através de código Morse e posteriormente através de voz. O potencial deste tipo de comunicações foi aproveitado e, em 1920, a maior parte dos navios utilizava o rádio para promover consultas entre os navios e profissionais de saúde em terra. Foi também com este tipo de comunicações que os passageiros de aviões eram assistidos quando não existiam médicos a bordo. Na altura da primeira grande guerra mundial, as comunicações via rádio foram usadas para transmitir informações médicas em áreas remotas. Na guerra da Coreia e do Vietname eram já amplamente usadas as comunicações via rádio para envio de equipas médicas.

Na década de 50 do século XX, com o início dos circuitos fechados de televisão, a telemedicina era efetuada com recurso a vídeo e som sincronizados, idêntico ao que acontece atualmente, com recurso a um *link* bidirecional de *Close Circuit TeleVision (CCTV)*. Médicos especialistas do Instituto de Fisiatria do Nebraska [Ferrer-Roca and Sosa-Iudicissa, 1998] efetuavam consultas com clínicos gerais em áreas remotas.

Não tão longe na história a National Aeronautics and Space Administration (NASA) desenvolveu um sistema para monitorizar os efeitos fisiológicos da ausência de gravidade nos astronautas [Bashshur R., 1977]. Esse sistema permitiu que médicos controlassem com sucesso, a partir da Terra, a frequência cardíaca e respiratória, a pressão arterial e a temperatura dos astronautas no espaço, através de sistemas de telemetria durante as viagens espaciais nas décadas de 60 e 70 do século XX.

Sabe-se ainda que a NASA durante esse período lançou vários projetos de telemedicina. Um deles foi o Space Technology Applied to Rural Papago Advanced Health Care (STARPAHC) [Fuchs, 1979] em funcionamento de 1972 até 1975, cujo objetivo era fornecer cuidados de saúde aos índios nativos de uma reserva isolada no Arizona. Eram efetuados eletrocardiogramas e raios-X localmente na reserva por paramédicos índios e transmitidos a especialistas de um hospital de saúde pública remoto.

Mais recentemente, a telemedicina tem sido usada em países como a Austrália, Canadá, Espanha e Suécia especificamente em áreas rurais e remotas, de forma a colmatar a falta de médicos e melhorar assim o acesso a cuidados de saúde [Peña-López et al., 2010].

2.4 A telemedicina em Portugal

Portugal, comparado com outros países, não é diferente no que toca à telemedicina. A telemedicina em Portugal sofreu também de "pilotismo", ou seja, surgiram vários projetos que apenas existiram enquanto houve o suporte das entidades que os subsidiavam, acabando por desaparecer assim que o suporte era retirado.
A telemedicina, como a conhecemos hoje, surgiu em Portugal (com outro nível de tecnologia) há mais de 15 anos, inicialmente com poucos pontos de contacto foram aumentando ao longo destes anos. A sua utilização, numa primeira fase baixa, deve-se fundamentalmente a três fatores;

1. Redes e infraestruturas de telecomunicações com baixo débito;

2. Fraco poder computacional face às necessidades;

3. Baixo nível de literacia dos *players* e da sociedade em geral relacionado com as TIC

Apesar destes problemas, em Portugal é conhecido o caso de sucesso denominado Saúde24. Este serviço teve início em 2008, envolve utentes e unidades prestadoras de cuidados de saúde e pode ser contactado em casos de urgência médica. A Saúde24 funciona numa plataforma multi-canal através de telefone, fax, correio eletrónico e internet e presta um serviço ao nível da triagem, aconselhamento e encaminhamento, de assistência em saúde pública e de informação geral de saúde, através de dois centros de atendimento em Lisboa e no Porto [1].

Outro exemplo positivo é o do Serviço de Cardiologia Pediátrica do Hospital Pediátrico de Coimbra, um dos pioneiros da telemedicina em Portugal, onde há consultas de pediatria recorrendo à telemedicina há vários anos, ultrapassando as 10 mil consultas anuais. Seguindo este exemplo, neste momento há diversos hospitais e Centros de Saúde onde é possível recorrer a consultas pediátricas de telemedicina. Nos Países Africanos de Língua Oficial Portuguesa (PALOP) a contar com esta consulta de telemedicina de cardiologia pediátrica e fetal, estão as cidades de Luanda e Benguela em Angola, Cidade da Praia e Mindelo em Cabo Verde e, outras cidades no Brasil e Espanha com acesso a esta rede de cuidados.

Existem muitos outros exemplos em Portugal de telemedicina com telemonitorização e de teleconsulta. Por exemplo o Projeto AIRMED para monitorização remota de doentes cardíacos do Hospital de Santa Marta em Lisboa, ou a teleconsulta da Dor Crónica do Hospital do Espírito Santo em Évora [DGS, 2012].

Podem ainda ser referidos projetos internacionais entre Portugal e Espanha, como o Projeto CALENO, de telemedicina em Castela e Leão no Nordeste Transmontano da ARS do Norte[2], cujo objetivo principal consiste em implementar uma rede de telemedicina na Região de Trás-os-Montes e Alto Douro e Castela e Leão nas áreas de imagiologia, dermatologia, psiquiatria, alcoologia, de forma a rentabilizar equipamentos e recursos humanos existentes, bem como melhorar a acessibilidade dos utentes aos serviços de saúde.

[1] http://www.saude24.pt

[2] http://portal.arsnorte.min-saude.pt/portal/page/portal/ARSNorte/Relações%20Internacionais/Cooperacao/Projetos

Ainda da ARS do Norte, o projeto GAMITE[3], que envolve os distritos de Viana do Castelo e Braga e a Galiza numa rede de telemedicina nas áreas de radiologia, dermatologia, oftalmologia, psiquiatria, alcoologia, gastrenterologia e cardiologia.

O projeto INTERREG, um programa de iniciativa da Comunidade Europeia, de telemedicina entre o Alentejo, o Algarve e a Andaluzia onde as autoridades de saúde dessas três regiões participaram no desenvolvimento de programas de saúde em rede telemática Alentejo-Algarve-Andaluzia, para melhorar a qualidade dos cuidados sanitários recebidos por habitantes das três regiões e aumentar, desta forma, o acesso da população aos serviços de saúde.

Durante a realização deste trabalho, os Serviços Partilhados do Ministério da Saúde (SPMS), a entidade que, entre outras responsabilidades, tem a seu cargo o desenvolvimento de software para o SNS, está numa fase de desenvolvimento e teste da plataforma *PDSLive*, um projeto a nível nacional que irá integrar a Plataforma de Dados da Saúde (PDS) e que pretende ser um veículo para democratizar a telemedicina em Portugal.

2.5 Usos da telemedicina

De acordo com o Portal do cidadão,[4] o canal de comunicação entre a Administração Pública e os cidadãos portugueses, a telemedicina diminui o esforço dos utentes e dos serviços de saúde, diminuindo o recurso à urgência hospitalar. Reduz ainda gastos com exames e transporte de doentes, assim como as desigualdades no acesso aos serviços de saúde. É possível ainda aumentar a satisfação dos pacientes, promover o desenvolvimento de recursos de saúde integrados e melhorar a comunicação entre profissionais. Por outro lado, poderão surgir questões sociais e legais associadas, como a despersonalização do serviço, perdendo-se a relação entre médico e paciente, diferentes processos para uma mesma consulta onde se pode incluir o reembolso da consulta, além de questões físicas como a incapacidade de realizar todos os procedimentos no mesmo momento. São exemplos concretos a palpação ou ainda a incapacidade de fazer a teleconsulta no caso de má visão ou audição.

A telemedicina pode ser praticada recorrendo a diversas formas, conforme o utilizador seja emissor ou recetor e, tendo em conta o tipo de telemedicina, síncrona ou assíncrona.

[3] http://portal.arsnorte.min-saude.pt/portal/page/portal/ARSNorte/Relações%20Internacionais/Cooperacao/Projetos
[4] www.portaldocidadao.pt

Os processos podem ter sofrido melhoramentos desde 1998, a tecnologia envolvida com certeza que evoluiu, poderá existir outro grupo de utilizadores ou mesmo outras formas de telemedicina, mas na Figura 2.4 estão explanados de forma simples as principais formas envolvidos na telemedicina: diagnóstico, consulta remota e teleducação, assim como os utilizadores de cada forma.

Como se pode observar na Figura 2.4, há formas de telemedicina mais orientadas a um tipo específico de utilizadores, enquanto outras, mais abrangentes, envolvem mais intervenientes. Por exemplo no ponto 4, a teleducação está vocacionada para a formação e, dessa forma, os seus utilizadores são na sua esmagadora maioria estudantes e instituições de ensino. No ponto 1, estão incluídas as formas mais comuns de telemedicina onde um grupo bem definido de utilizadores, sendo médicos e outros profissionais de saúde e instituições de saúde, nomeadamente hospitais. O ponto 5 está praticamente todo reservado ao Sistema Integrado de Emergência Médica (SIEM), um conjunto de entidades onde estão incluídos serviços conhecidos como o Instituto Nacional de Emergência Médica (INEM) na ajuda às vitimas de acidentes e desastre, a Autoridade Nacional Proteção Civil (ANPC) na prevenção e reação a acidentes graves, catástrofes e desastres, o serviço Saúde 24 com triagem telefónica, aconselhamento terapêutico e encaminhamento em situação de doença e ainda a Cruz Vermelha Portuguesa (CVP) que apoia não só idosos, mas pessoas que estão expostas a situações que ameaçam a sua sobrevivência com dignidade, para além de terem uma participação ativa em situações de emergência e socorro.

Forma de telemedicina	Utilizadores
Todas as formas de medicina à distância: Teleconsultas, telepatologia, teleradiologia, Telepsiquiatria, teledermatologia, telecardiologia	Médicos Profissionais de saúde Instituições de saúde Pacientes
Entre instituições, Registos clínicos e de pacientes, Sistemas de registos clínicos eletrónicos, Bases de dados clínicas acessíveis remotamente	Profissionais de saúde Instituições de saúde Clínicas médicas Investigadores
Comunidades de Saúde Pública e de redes de informação para a saúde	Governo Epidemiologistas Profissionais de saúde pública Farmácias Clínicas
Teleducação e aplicações multimédia para profissionais de saúde, Bases de dados de investigação e de pacientes Serviços de Internet	Universidades Associações Investigadores Médicos Profissionais de saúde Pacientes
Telemonitorização, Serviços de telecuidados, Triagens telefónicas, Serviços de emergência e cuidados domiciliarios remotos	Idosos Doentes crónicos Vítimas de desastres Vitimas de acidentes Telenfermagem Utilizadores e profissionais de centros médicos de atendimento telefónico

Figura 2.4: Usos da telemedicina por utilizadores
[Ferrer-Roca and Sosa-Iudicissa, 1998]

2.6 Estado atual da telemedicina

Nesta secção identificam-se alguns casos de telemedicina em Portugal, Alemanha, Estados Unidos da América e Canadá. Estes casos evidenciam vantagens claras no uso da telemedicina, seja pelo lado mais economicista das instituições que prestam os cuidados, pelo lado dos profissionais de saúde ou ainda o lado dos pacientes e utentes do serviço de saúde.

Existe, atualmente em Portugal, um projeto piloto financiado pela Administração Central do Sistema de Saúde (ACSS), em parceria numa primeira fase apenas com alguns hospitais do país, que pretende usar a telemonitorização de forma a acompanhar doentes portadores de Doença Pulmonar Obstrutiva Crónica (DPOC). Os doentes são monitorizados nos seus domicílios, onde são colocados dispositivos médicos que recolhem diferentes parâmetros, que são posteriormente analisados pelas equipas de Pneumonologia dos Hospitais de referência. Pretende-se, assim, diminuir o agravamento da situação clínica dos pacientes e consequentemente evitar novos internamentos. Os doentes domiciliários realizam os testes que permitem a emissão e receção dos parâmetros pretendidos, através da PDS.

Outro projeto a decorrer em Portugal é o telerastreio dermatológico, que está a contribuir para a diminuição das listas de espera. Este projeto de telerastreio foi iniciado no nordeste transmontano no início de 2015 e está a revelar bons resultados na redução das listas de espera para consultas da especialidade. São usados equipamentos de captura de imagens para fotografar a lesão do doente e, seguidamente, o médico procede ao envio da fotografia para especialistas de centros hospitalares, onde é efetuado o diagnóstico ou marcada uma teleconsulta ou uma consulta presencial. O sucesso deste projeto na região conduziu ao seu alargamento progressivo ao restante território nacional, estando já implementado nos hospitais de Gaia, Viana do Castelo e Setúbal e no Centro Hospitalar e Universitário de Coimbra. Irá ser estendido brevemente à região centro, designadamente aos distritos de Viseu, Guarda, Covilhã, Leiria e Castelo Branco.

Em julho de 2015, no decorrer das II Jornadas Luso-Brasileiras de Telemedicina e Telesaúde, os SPMS assinaram um acordo de cooperação a nível técnico e científico com a Universidade Estadual do Amazonas. Este acordo prevê a promoção de atividades no âmbito da telemedicina, o desenvolvimento de informação e formação nas áreas das TIC e o desenvolvimento do Passaporte Telemedicina Ibero-brasileiro (PATIB).

A Direção Geral da Saúde (DGS) tem produzido vários documentos em forma de normas e *guidelines*, no sentido de padronizar cada vez mais a telemedicina em Portugal e de a alinhar com a estratégia de implementação Europeia. Esses documentos são o resultado de grupos de trabalho de especialistas em várias áreas, nomeadamente na área da dermatologia, radiologia, patologia, qualidade na saúde e comunicação. Exemplos dos documentos são as normas 005/2014 onde são focados aspetos da consulta no âmbito da teledermatologia, a norma 004/2015 da consulta de telepatologia, a norma 005/2015 da consulta de teleradiologia, a norma 010/2015 com o modelo de funcionamento das teleconsultas, estando neste momento a ultimar a norma de telemonitorização da DPOC.

Segundo a Agenda digital para a Europa, [Commission, 2014] fazer parte de um programa de telemedicina pode aumentar a probabilidade de sobrevivência de doentes cardíacos em mais de 70%. Esta foi uma das principais conclusões de um estudo do Instituto de Medicina Comunitária da Universidade de Greifswald na Alemanha, na sequência da análise do programa de telemedicina da seguradora de saúde AOK Nordost. A análise científica que foi realizada com o objetivo de desvendar a real eficácia do programa, foi conduzida através de um paralelismo entre quase dois mil participantes e um grupo de controlo com quase quatro mil participantes. Para além do aumento da probabilidade de sobrevivência, a investigação concluiu que integrar o programa *AOK - Curaplan Heart Plus* tem vantagens a nível económico, uma vez que as poupanças conseguidas superam os custos do programa. Um dos objetivos deste programa é conhecer os sinais de um ataque cardíaco iminente para que a intervenção seja feita o mais rapidamente possível e para que os pacientes possam ser estabilizados, evitando o tratamento hospitalar. O conceito, vencedor do prémio *MSD Health Prize 2014*, assenta no aconselhamento e prestação de cuidados individuais de saúde por telefone, assim como na monitorização dos sinais e sintomas de ataques cardíacos.

Nos Estados Unidos da América, onde não existe o conceito de serviço público de saúde idêntico ao que conhecemos na Europa, há preocupações a outros níveis. Uma delas é a forma de reembolso aos utentes quando recorrem a uma teleconsulta bem como a própria faturação dos serviços efetuados por esta via. No estudo [North et al., 2014] levado a cabo por clínicos especialistas em cuidados de saúde primários e por especialistas pediatras do Centro Hospitalar Mayo, uma das maiores clínicas sem fins lucrativos a nível mundial, sediada nos Estados Unidos da América, revela que ainda existem barreiras para o funcionamento da telemedicina, designadamente relacionadas com o licenciamento, faturação e reembolso dos atos praticados por esta via. Outra barreira crítica, em algumas situações, é a qualidade das

avaliações e das amostras de pequenos auxiliares de diagnóstico executados pelos próprios utentes de telemedicina, como por exemplo o batimento cardíaco, pressão sanguínea entre muitos outros possíveis.

No caso prático apresentado em [William et al., 2014], é comparado o uso em detrimento do não uso da telemedicina na identificação de AVC's e posterior administração de um ativador do plasminogênio tecidual, um anti-coagulante em casos de AVC. A qualidade das avaliações, sendo uma das desvantagens identificada no centro hospitalar Mayo, não se coloca, uma vez que são efetuadas por clínicos profissionais. Neste caso, compararam-se as taxas de tratamento de Acidente Vascular Cerebral isquêmico agudo (AVCia). Foi analisado o uso da telemedicina de forma a obter diagnósticos exatos e atempados, fornecidos pelos hospitais centrais e, consequente tratamento dos doentes com esse diagnóstico em hospitais periféricos ou sem especialistas na área. É demonstrado o aumento da eficácia superior a 50% no uso de anti-coagulantes em caso de AVCia, ou seja, apenas em casos precocemente detetados este tratamento tem eficácia, obtendo assim ganhos significativos tanto para os pacientes, com o diagnóstico rápido e posterior tratamento, como para os serviços de saúde em ambos os hospitais que, juntamente com uma taxa de produtividade elevada, consegue manter a distribuição de recursos médicos pelos hospitais, aumentando também a sua eficiência.

Em lares de idosos nos Estados Unidos da América, [Grabowski et al., 2014] foi testado o uso de serviços baseados na telemedicina, de forma a diminuir hospitalizações desnecessárias, reduzindo dessa forma custos evitáveis. Os investigadores que implementaram estes serviços concluíram que, com o uso deste serviço, foi possível atingirem poupanças superiores aos custos da implementação do serviço de telemedicina.

No caso descrito em [Mendelson et al., 2014], a desvantagem identificada no centro hospitalar Mayo, a qualidade das avaliações, também é ultrapassada já que são realizadas automaticamente pelos equipamentos usados no tratamento. Neste caso, a telemedicina é usada essencialmente na monitorização da pressão sanguínea no uso do *Continuous Positive Airway Pressure (CPAP)*, avaliando também a adaptação ao uso do CPAP além de dados acerca da qualidade e quantidade do sono. Usando um *smartphone* para comunicação destes dados, os pacientes obtêm relatórios com a pressão sanguínea e com a evolução do risco cardiovascular. Após a utilização do sistema durante 16 semanas, concluiu-se que houve uma melhoria do sono dos pacientes e consequente aumento da qualidade de vida.

Ainda outro exemplo é a rede de telemedicina de Ontário no Canadá, um serviço de telemedicina em desenvolvimento há mais de uma década, onde foi realizado um estudo [Qiang and Marras, 2015] cujo objetivo era conhecer as mais-valias do uso da telemedicina em pacientes com a doença de Parkinson, perceber a perspetiva dos pacientes e sua satisfação no uso desse serviço.
O estudo revelou uma redução do tempo e do custo, fundamentalmente com o transporte dos pacientes. Revelou ainda que o maior foco de insatisfação deveu-se à inexperiência dos enfermeiros para lidar com o sistema de telemedicina e talvez por essa razão, os pacientes valorizariam um sistema misto de atendimento.

Embora haja ainda questões a definir ao nível da gestão de processos, através dos resultados dos casos descritos anteriormente, podemos concluir que há vantagens claras na utilização da telemedicina. Essas vantagens são mais evidentes em ambientes remotos ou rurais ou ainda em ambientes com baixa densidade populacional.

Com o uso da telemedicina é possível o aumento da equidade no acesso aos cuidados de saúde, realizar rastreios e diagnósticos precoces remotamente e de uma forma atempada, monitorizar o estado de saúde da população, melhorar a organização dos cuidados prestados e fomentar a eficiência e qualidade da prestação desses cuidados.

2.7 Desafios da telemedicina

Existem vários desafios para a implementação da telemedicina dependendo do *player* em questão; ao nível financeiro com o Return On Investment (ROI) ou com os custo de aquisição e de manutenção do sistema por parte de alguns prestadores; ao nível tecnológico com a desadequação dos sistemas informáticos às particularidades das instituições; ao nível da qualidade de serviço com a satisfação dos utentes com a possível falta de aptidão para o manuseamento das tecnologias envolvidas. Ainda ao nível da qualidade de serviço, dada a relação nova com o profissional de saúde pela distância que se encontra e ainda desafios na adesão dos profissionais com resistência à mudança e dificuldade de adaptação a novos procedimentos.

Um dos grandes entraves ao desenvolvimento da telemedicina é transversal a todos os *players* e está relacionado com as tecnologias envolvidas. Neste tópico, o maior desafio está

relacionado com a comunicação [Hjelm, 2005], ou seja com a transferência da informação desde a sua origem a um destino remoto. As ligações existentes em localizações remotas, a maior parte delas rurais, não possuíam conetividade por cabo ou fibra, inviabilizando a maior parte das aplicações de telemedicina uma vez que não disponibiliza uma largura de banda adequada ao uso das aplicações. A este nível, as barreiras estão progressivamente a desaparecer uma vez que todas as unidades de saúde do SNS estão a ser dotadas de ligações fiáveis e redundantes de ultima geração com grandes potencialidades de largura de banda. Ao nível da aquisição e salvaguarda da informação o cenário é estável, uma vez que as tecnologias envolvidas tem sofrido evoluções de forma a responder às necessidades, existem soluções de armazenamento de grande capacidade assim como câmaras de alta definição cada vez mais acessíveis.

O desenvolvimento das TIC, de novos serviços e equipamentos levam a que a relação custo/benefício seja cada vez mais favorável, potenciando assim a sua utilização. O aumento do nível de confiança dos sistemas também ajuda à generalização do seu uso. A constante evolução das TIC permite aos sistemas de telemedicina uma oportunidade de renovação e evolução praticamente permanente.
A facilidade com que pacientes e prestadores de cuidados de saúde se relacionam com a tecnologia é cada vez maior e constitui um fator preponderante para o sucesso da telemedicina.

Um desafio emergente para a utilização massiva da telemedicina é a interoperabilidade entre os sistemas utilizados. Qualquer que seja o sistema em que assente um dos extremos da comunicação da telemedicina, o outro extremo tem obrigatoriamente que ser igual, ou no mínimo, reconhecer o tipo de comunicação de forma a perceber e processar a informação que é passada. Imagine-se por exemplo um português a tentar falar com um japonês, não conhecendo nenhum dos intervenientes a língua do outro. É o principio básico da comunicação, que os intervenientes reconheçam o tipo, forma e conteúdo da comunicação. Este é o desafio que o *WebRTC* tenta resolver. Além de adicionar segurança, uma vez que cifra toda a comunicação, alta qualidade de vídeo, simplicidade no uso e sem necessidade de instalação de aplicações adicionais. Conforme veremos no Capítulo 3, a interoperabilidade com outros sistemas de comunicação e acessibilidade através de um dispositivo simples com um navegador web e a custo zero, constituem as maiores vantagens do WebRTC para encarar os desafios propostos pela telemedicina.

A American Telemedicine Association (ATA) publicou recentemente um relatório onde apresenta as diferentes regras de serviços de telemedicina nos 50 estados dos Estados Unidos da América bem como o seu nível de implementação. Segundo esse estudo, o pagamento/reembolso aliado a uma rede de serviços de telemedicina são os principais desafios identificados [Thomas and Capistrant, 2014]. Neste momento e, segundo o estudo, utilizadores e prestadores de cuidados de saúde encontram-se com o que chamam de "manta de retalhos"de requisitos arbitrários de seguros de saúde e fluxos de pagamento/reembolso, impossibilitando assim o sucesso do serviço de telemedicina. Seria importante, além de haver um consenso entre os *players*, entidades prestadoras de serviços de telemedicina, reguladores dos serviços e estado, a definição de regras para a prestação de cuidados saúde através da telemedicina.

A segurança da informação e privacidade dos utilizadores dos sistemas de *e-Health* está a ser amplamente debatida desde há uns anos a esta parte. Esta é uma questão fulcral ao qual a OMS tem contribuído com publicações do observatório global para o *e-Health*. O volume 5[5] deste observatório, debate temas como legislação, jurisprudência, ética, privacidade e confidencialidade em torno do *e-Health* e dos Registos Clínicos Eletrónicos (RCE) e propõe-se a responder à questão: *Estará a legislação da privacidade no coração dos RCE? Uma vez que a privacidade da relação médico-paciente está no coração dos bons cuidados de saúde e, que os RCE estão no centro das boas práticas do e-Health*.

A OMS concluiu que, globalmente, existem bons níveis de direitos básicos em relação à privacidade da informação; que a privacidade dos RCE não está amplamente prevista na lei embora esteja prevista nos códigos de conduta dos profissionais; foram adotadas iniciativas de forma a criar legislação para a privacidade dos RCE e ainda que existem leis e regulamentos que tendem a ser reativos e que poucos países usam a legislação para promover o uso dos RCE.

2.8 Conclusão

A telemedicina sempre esteve e sempre estará intimamente ligada às tecnologias de informação e comunicação. Independentemente do tipo, forma ou utilizador de telemedicina a comunicação é uma parte fulcral do processo.

[5]http://www.who.int/goe/publications/ehealth_series_vol5/en

Os avanços tecnológicos, nomeadamente na área das TIC, potenciam e facilitam o uso da telemedicina. Esta é assim, uma renovada forma para estreitar o relacionamento entre profissionais de saúde e utentes, com cada vez mais capacidades para modificar e revolucionar a prestação de cuidados de saúde. A relação entre médico e utente está paulatinamente a sofrer mutações sobretudo na forma de comunicação. Com o avanço tecnológico, novos dispositivos poderão fazer parte integrante de um sistema de telemedicina, aumentando assim o espectro e a capacidade de diagnóstico.

A telemedicina, como a conhecemos hoje, está numa fase de maturação elevada e com potencial de desenvolvimento nos próximos anos.

A telemedicina irá revolucionar a forma como profissionais, instituições de saúde e utentes se relacionam de maneira a melhorar os cuidados de saúde, para dessa forma atingir o que todos anseiam, saúde a todos os níveis.

Capítulo 3

WebRTC - Real Time Comunication

3.1 Introdução

O WebRTC é uma definição de API que suporta ligações entre *browsers* para aplicações de transmissão de voz, vídeo, *chat* e partilha de ficheiros. De código aberto e para a *web*, permite a comunicação ponto-a-ponto, de áudio, vídeo e dados em tempo real.
Pela primeira vez, os *browsers* podem interagir diretamente entre si para troca de conteúdos multimédia, em tempo real. O WebRTC está a revolucionar a forma de comunicar na *web* em tempo real, permitindo enviar vídeo, voz e outro tipos de dados diretamente, através de um *browser* sem a necessidade de *plugins*, *downloads* ou instalações de aplicações, usando comunicação ponto-a-ponto, ou seja, sem intervenção de servidores intermediários.

O WebRTC permite ainda interligar sistemas legados de comunicação, como o telefone, ou o Voice over IP (VoIP). Torna-se, assim, possível integrar tecnologias existentes com o WebRTC, servindo de motor de interoperabilidade, por forma a trazer todas as potencialidades da *web* para o mundo das telecomunicações. Por exemplo, usar um computador para efetuar uma chamada de voz para um qualquer número da rede fixa de comunicações, ou ainda efetuar uma ligação para um sistema VoIP.

Nem todas as especificações do WebRTC estão concluídas. Algumas delas estão ainda em processo de desenvolvimento pelo grupo World Wide Web Consortium (W3C) [1]ao nível da API e pelo Internet Engineering Task Force (IETF) [2] ao nível do protocolo, além de estarem

[1]https://www.w3.org/TR/webrtc/
[2]https://tools.ietf.org/wg/rtcweb/

também a ser discutidas por toda a comunidade *web*, por forma a que todos os detalhes sejam exaustivamente verificados e otimizados.

Os *codecs* utilizados no áudio e vídeo são um exemplo de especificações ainda em discussão. Um *codec*, de uma forma simples, é um componente usado para comprimir e descomprimir vídeo e áudio. Existem *codecs* sem perda e com perda de qualidade, resultando respetivamente em maior ou menor quantidade de dados a transferir. O facto de estarem em discussão, não impede nem tão pouco compromete o seu funcionamento, uma vez que se trata maioritariamente de otimizações.

O IETF é uma comunidade internacional aberta e multidisciplinar que propõe e desenvolve normas para a Internet, fundamentalmente as que envolvem a pilha protocolar Transmission Control Protocol / Internet Protocol (TCP/IP). Nesta organização incluem-se técnicos, agências, fabricantes, fornecedores e investigadores voluntários. O IETF está organizado em grandes grupos de trabalho e de discussão onde, em cada grupo, são discutidas e analisadas questões específicas de determinados tópicos. Cada grupo tem um ou mais responsáveis que definem regras, prazos e objetivos a atingir. É da responsabilidade do IETF definir e especificar os protocolos, por forma a ser possível a comunicação entre os participantes, usando áudio, vídeo e outros dados da forma mais direta possível.
Esta comunidade cria Request For Comments (RFC) que são publicações que contêm as especificações pormenorizadas dos protocolos de comunicação, documentos das especificações da Internet, procedimentos e eventos. Pretende ser um espaço aberto de discussão, funcionando de forma a recolher propostas de melhoria e que antecede a padronização desse protocolo. A RFC 2026, documenta e padroniza a própria criação de *RFC's*.

3.2 Conceitos gerais

Esta secção pretende dar a conhecer alguns conceitos fundamentais necessários para a compreensão do trabalho apresentado. São apresentados, sobretudo, protocolos de comunicação e conceitos no âmbito das redes de computador.

3.2.1 Protocolos de comunicação

A Internet assenta numa pilha de protocolos de comunicação denominada TCP/IP, por forma a que a comunicação flua sem problemas.

Conforme é possível ver na figura 3.1, existem 4 camadas, sendo que cada uma delas tem funções específicas e bem identificadas. Seguidamente, indica-se de forma sucinta as principais funções de cada camada.
A camada de aplicação é utilizada para receber e enviar informações das aplicações para a camada de rede, a camada de transporte verifica a integridade dos dados e divide em pacotes, a camada de internet coloca os dados em datagramas IP, que contêm informações de endereço de destino e de origem utilizadas para reencaminhar os datagramas entre redes. Finalmente, a camada de rede especifica detalhes sobre a forma como os dados são enviados fisicamente através da rede.

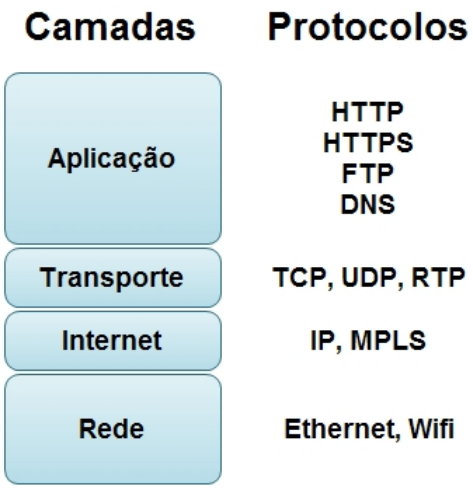

Figura 3.1: Pilha protocolar TCP/IP

Existem diversos protocolos divididos em camadas, dependendo cada um do seu propósito. A camada que interessa agora explorar é a de transporte e, dentro dela, os protocolos que carecem de uma breve explicação mas, sobretudo diferenciação são o User Datagram Protocol (UDP) e Transmission Control Protocol (TCP).

Uma das principais diferenças é que o TCP é um protocolo orientado à ligação e fiável, i.e. apresenta uma série de mecanismos que garantem que a informação chega de certeza ao destino, como a sequenciação de segmentos, controle de fluxos e confirmação de entrega. O UDP, ao contrário do TCP, não é fiável porque não usa esses mecanismos, sendo por isso mais rápido na entrega dos dados.

O UDP é principalmente usado em comunicações multimédia, uma vez que não realiza algumas tarefas como verificações dos pacotes. Aproveitando esse facto, o UDP é muito mais rápido e dessa forma indicado para o conteúdo multimédia, principalmente o conteúdo em tempo real. Facilmente se pode fazer um exercício exageradamente simples para se conseguir verificar as diferenças.
Imagine-se que o utilizador A, está numa rede de alto débito e pretende passar a mensagem em tempo real "`hoje está um dia bonito`" ao utilizador B que está numa rede de baixo débito. Tendo por base que toda a informação flui pela rede em pequenos pacotes, esta mensagem poderia seguir desta forma `'hoj' 'e est' 'á' 'um' 'dia bonito'`.
O TCP numera sequencialmente os 5 pacotes e envia-os ao utilizador B. Como a rede é de baixo débito, o pacote nº 2 `'e est'` não chega ao destino e o sistema do utilizador B solicita o envio de um novo pacote 2. O pedido chega ao sistema do utilizador A e é enviado um novo pacote, com a informação que faltava para "montar"todo o puzzle, sendo entregue ao utilizador B. No mesmo cenário, o UDP não controla os pacotes que chegam, ou seja, simplesmente envia os pacotes.
Esta pequena diferença entre estes dois protocolos, faz com que o utilizador B não receba a informação em tempo útil, interrompendo a comunicação até o "puzzle"estar todo montado no caso do TCP.
No caso do UDP, o utilizador receberia a mensagem `'hoj' 'á' 'um' 'dia bonito'` de uma forma rápida. Ou seja, os pacotes não chegariam todos, mas a mensagem era percetível.
No caso de aplicações em tempo real, é mais importante que os pacotes cheguem com uma cadência certa, podendo faltar alguns, do que estar demasiado tempo à espera de um determinado pacote, comprometendo assim a comunicação.

3.2.2 Comunicação Cliente/Servidor

O modelo de comunicação cliente/servidor é um modelo no qual o processamento da informação é dividido em dois processos distintos. Tipicamente, o processo da lógica de negócio e o processo da apresentação. O processo da lógica de negócio é responsável pela manutenção da informação e reside no servidor, o processo da apresentação é responsável pela aquisição da informação e reside no cliente. O cliente estabelece comunicação com o servidor, envia o seu pedido de dados e o servidor recebe, processa, responde ao pedido e termina a comunicação.

Neste modelo, é no servidor que são colocados os serviços mais pesados a nível computacional, uma vez que são máquinas com mais recursos, ficando os clientes libertos para outras operações. Por norma, também os servidores executam sistemas operativos específicos para os serviços que irão disponibilizar, requerem também por vezes *hardware* próprio para esta função. A figura 3.2 representa os passos necessários da comunicação típica cliente/servidor, através de um protocolo de transporte orientado à ligação designadamente o TCP.

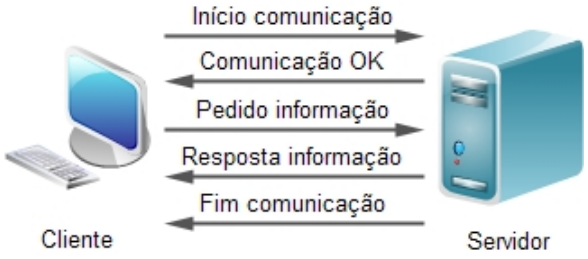

Figura 3.2: Modelo de computação Cliente Servidor

Antes de iniciar a troca de informação propriamente dita, no caso do protocolo TCP, que como foi referido acima, é um protocolo orientado à ligação, ou seja é fiável, é necessário que haja um mecanismo que garanta a existência de conetividade suficiente para transferir a informação. Esse mecanismo é conhecido por *three-way handshake*, porque o estabelecimento da comunicação entre duas aplicações é efetuado em "três passos", envolvendo outras tantas mensagens. Conforme se pode ver na Figura 3.3, o cliente envia um pacote *SYNchronize* para o servidor que responde com um *SYNchronize-ACKnowledgement* que por sua vez, é respondido com um pacote *ACK*. Só após esta troca de mensagens a comunicação é estabelecida. Como foi referido, o protocolo UDP não é fiável porque não utiliza este mecanismo, mas também por isso é mais rápido e não tem tanta latência, como o protocolo TCP.

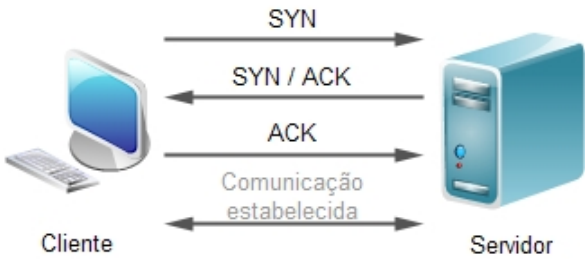

Figura 3.3: Processo three-way handshake

3.2.3 P2P

A arquitetura Peer to Peer (P2P) ou "ponto-a-ponto"consiste num tipo de rede diferente da cliente/servidor, uma vez que qualquer ponto pode assumir o papel de servidor ou cliente, ou ainda os dois simultaneamente, conforme se pode ver na Figura 3.4. Esta arquitetura de rede tornou-se popular sobretudo pela facilidade e rapidez na transferência de ficheiros áudio e vídeo. Cada computador representa um nó e é responsável pela partilha de recursos na rede. A transferência de um recurso pode ser dividida por todos os nós que partilham esse recurso, ou seja, um nó pode transferir um ficheiro de dezenas ou mesmo centenas de locais (nós) diferentes.

Esta forma paralela de transferência de dados resulta, na prática, num tempo de transferência inferior em relação à comunicação cliente/servidor.

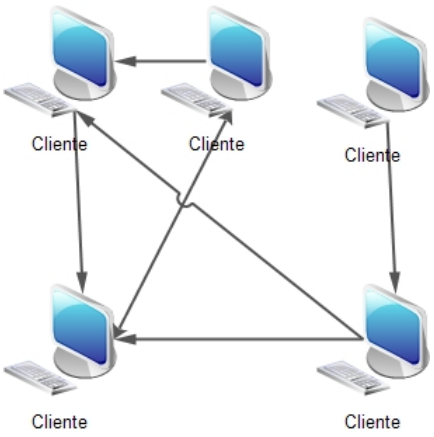

Figura 3.4: Arquitetura ponto-a-ponto

3.2.4 TLS

O objetivo do protocolo Transport Layer Security (TLS) é fornecer privacidade e integridade na comunicação entre duas aplicações [Dierks and Rescorla, 2008]. Este protocolo é usado para encapsular os dados de outros protocolos de níveis mais altos. No *handshake* deste protocolo, o servidor e cliente autenticam-se um perante o outro, negoceiam o algoritmo de cifra e trocam chaves desse algoritmo antes de serem transmitidos quaisquer dados.

Uma grande vantagem do protocolo TLS é o facto de ser independente do protocolo da aplicação, podendo ser usado de forma transparente pelos protocolos de níveis mais altos

[Dierks and Allen, 1999]. É usado por outros protocolos bastante conhecidos, como por exemplo o HyperText Transfer Protocol (HTTP), o Web Socket (WS) File Transfer Protocol (FTP), Lightweight Directory Access Protocol (LDAP) ou ainda o Simple Mail Transfer Protocol (SMTP).

Como se pode ver na Figura 3.5, o protocolo TLS divide-se em duas camadas, o protocolo de registo e os protocolos de *handshaking*.

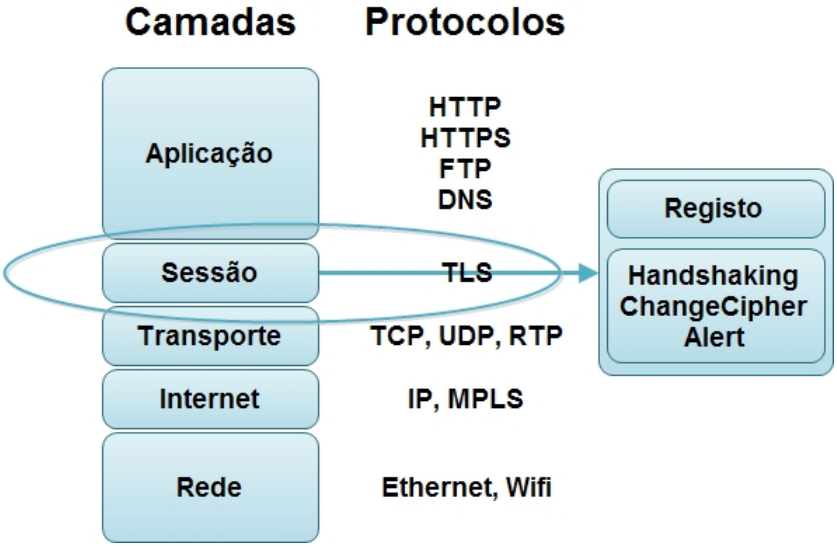

Figura 3.5: Transport Layer Security

O protocolo de registo fornece segurança à ligação adicionando privacidade e integridade.
Para fornecer privacidade, este protocolo usa a cifra simétrica como por exemplo Advanced Encryption Standard (AES) com chaves geradas a cada nova ligação, baseadas num segredo negociado pelo protocolo *handshake*.
Para fornecer integridade, este protocolo inclui um mecanismo de verificação chamado *keyed Message Authentication Code (MAC)*. Um algoritmo MAC usa a chave previamente negociada e a mensagem para gerar o MAC. É assim garantida a integridade e a autenticidade da mensagem, permitindo aos recetores (que também possuem a chave secreta) detetar quaisquer alterações, à mensagem como pode ser visto na Figura 3.6.

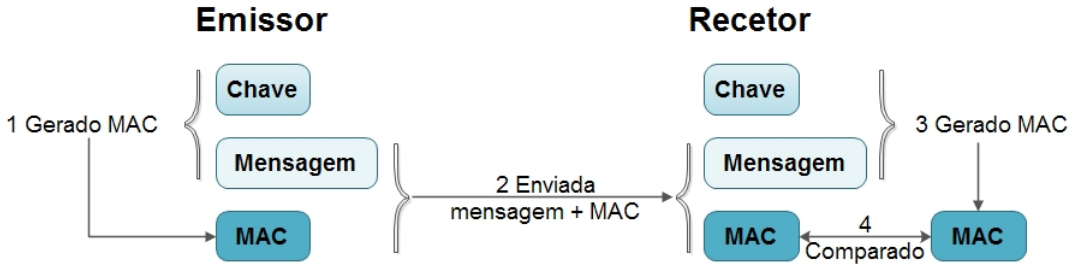

Figura 3.6: Algoritmo MAC

3.2.5 HTTP

O HTTP para além de um protocolo de comunicação da camada de aplicação, é a base para a comunicação na World Wide Web (WWW) e, juntamente com o HyperText Markup Language (HTML), constitui um dos principais padrões para o funcionamento da WWW.
O HTTP 1.1 está definido na RFC 2616, é um protocolo para a transferência de hipertexto, ou de uma forma mais simples, de recursos incluídos nas páginas da Internet.
O protocolo funciona através do estabelecimento de um pedido de um recurso a um servidor *web*, por parte de um cliente. Este pedido usa normalmente o método *GET*, um dos oito métodos que estão definidos no protocolo. Esse recurso, normalmente um ficheiro HTML ou outro conteúdo, é então enviado de volta para o utilizador que o solicitou e apresentado no *browser*.
Este protocolo, à semelhança da maioria dos protocolos de rede, utiliza o modelo cliente-servidor ou seja, um cliente (normalmente um *browser*) estabelece uma ligação a uma aplicação servidor (p.e. o Apache ou Internet Information Services (IIS)) e solicita um recurso. Após a resposta a ligação estabelecida é terminada [Fielding et al., 1999].
O código abaixo exemplifica um pedido da página index.html ao *site* da Escola Superior de Tecnologia e Gestão (ESTG) do Instituto Politécnico de Leiria (IPL)

```
GET /index.html HTTP/1.1
Host: www.estg.ipleiria.pt
```

O protocolo HTTP é inseguro para transportar dados uma vez que são transportados "em claro", sem estarem cifrados. Caso existam ataques à rede de forma a escutar o tráfego que por ela circula, esse poderá ser visto e inclusivamente modificado antes de chegar ao seu destino. De forma a resolver este grave problema de segurança, surgiu a versão segura deste protocolo. O HyperText Transfer Protocol Secure (HTTPS) usa uma camada adicional de

criptografia utilizando TLS para proteger os dados. Dessa forma o protocolo HTTPS evita que os dados sejam percetíveis no caso de um ataque de escuta.

O IETF propôs, em Maio de 2015, uma nova versão para o protocolo HTTP. O Hypertext Transfer Protocol Version 2 (HTTP/2) definido na RFC 5740, garante um uso mais eficiente dos recursos de rede uma vez que permite múltiplos pedidos na mesma ligação TCP, além de implementar o conceito de *push*, ou seja o envio de dados sem terem sido explicitamente solicitados e ainda comprimir o cabeçalho dos pacotes.

3.2.6 Websockets

O protocolo WS, à semelhança do HTTP, é um protocolo da camada de aplicação, especificado na RFC 6455. Pode afirmar-se que é um passo em frente em relação ao HTTP, no que diz respeito a comunicações em tempo real. Como foi visto anteriormente, o HTTP estabelece uma comunicação cliente/servidor típica, ou seja, normalmente o cliente faz o pedido HTTP, o servidor responde ao pedido e fecha a comunicação. Ora este funcionamento não é o mais eficiente em comunicações em tempo real, uma vez que teriam que existir constantes ligações ao servidor de forma a ter a informação atualizada.
Para uma melhor compreensão exemplifica-se um exercício simples. Um cliente faz um pedido a um servidor que disponibiliza o valor de um ação no mercado de valores, obtém o valor atualizado num dado momento e, até aqui tudo funciona da melhor forma. Mas agora imagine-se que o valor, entretanto, se alterou. O cliente só terá conhecimento do novo valor depois de fazer novo pedido ao servidor. Ou seja, teria que estar constantemente a atualizar a página ou o componente da página, criando novos pedidos ao servidor.
Existem mecanismos que são usados para que este procedimento seja contornado, nomeadamente o *polling*, *long-polling* e o *streaming*. Estas são algumas das técnicas usadas para obter informação em tempo real e, simultaneamente, minimizar os pedidos HTTP.

Com um WS não só a comunicação se mantém ativa para receber novos dados, como ainda essa comunicação é *full-duplex*. Ou seja, com uma só ligação comunica-se de facto em tempo real e ainda nos dois sentidos. Este comportamento já era realizado com o HTTP simples, no entanto teriam que existir muitas mais ligações de forma a emular a comunicação bidirecional em tempo real.

Uma outra limitação do HTTP é que o seu paradigma é o de "puxar". O cliente faz um pedido ou "puxa"os dados de um determinado servidor, que não pode simplesmente enviar esses dados sem que o cliente os solicite. Ou seja, usando o exemplo dado anteriormente, mesmo que existissem novos dados, o servidor não poderia enviá-los, devendo ser o cliente a solicitá-los repetidamente, criando novas ligações.
Ao contrário do HTTP que termina a ligação após o envio dos dados solicitados, embora a porta se mantenha aberto para novas ligações, um WS que está ligado a um servidor mantém-se ligado para que haja comunicação de dados. Desta forma, a informação pode ser "empurrada"para os clientes em tempo real.

Este protocolo traz vantagens tanto para o cliente como para o servidor. Do lado do cliente, não são necessárias ligações permanentes ao servidor para se manter atualizado, poupando assim recursos internos da máquina e da rede. Do lado do servidor, o esforço computacional e também de rede é significativamente menor uma vez que, dessa forma, processa menos ligações.

Como se pode verificar na Figura 3.7 e Código 1, a comunicação com um *Websocket* é efetuada inicialmente com HTTP e, só depois é enviado um pedido de *upgrade* para o servidor WS. De forma a simplificar a Figura 3.7, as etapas necessárias à ligação a um WS foram agrupadas. O *handshake* inicial representa o processo *three-way handshake* do protocolo TCP, visto na Figura 3.3. O *upgrade* para WS é realizado de forma idêntica ao *handshake* do TCP, mas envolve troca de mais alguns dados, como por exemplo a versão do WS em uso e uma chave gerada aleatoriamente, de forma a garantir que o canal estabelecido é único e não um outro já em uso. Após a troca de todos estes dados, a ligação WS é estabelecida e criado um canal bidirecional, onde o cliente e servidor poderão comunicar simultaneamente [Fette and Melnikov, 2011].
O protocolo WS recorre, assim como o HTTP, ao protocolo TLS para disponibilizar a sua versão segura. O funcionamento da versão segura, o *Websocketsecure*, como utiliza o TLS e como já foi referido anteriormente, é transparente para a camada de transporte. Ou seja, após o *handshake* onde são trocadas informações necessárias para a cifra e autenticação do cliente e servidor, funciona da mesma forma que o WS, com a diferença que todos os dados que circulam são cifrados.

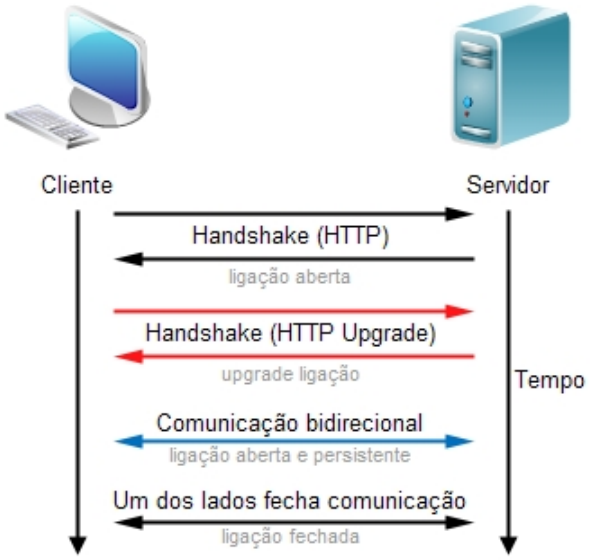

Figura 3.7: Websockets

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

Código 1: Pedido de upgrade de HTTP para websocket enviado ao servidor

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: chat
```

Código 2: Resposta ao pedido de upgrade enviada ao cliente

3.2.7 HTML

HTML é uma linguagem de marcação com mais de 20 anos de existência, usada para criar documentos para a *web* que implementa o conceito de hipertexto. O HTML define a estrutura

e o *layout* de um documento através do uso de etiquetas e atributos. A estrutura de um documento HTML segue regras recomendadas pelo W3C, onde estão definidos todos os requisitos para o seu desenvolvimento.

A correta estrutura de um documento HTML deve começar por declarar o tipo de documento com a etiqueta <!DOCTYPE html>, seguidamente deverá conter etiquetas básicas como <HTML> <HEAD> e terminar com </HTML> </HEAD>, as informações que se desejam incluir no documento devem estar entre a etiqueta <BODY> e </BODY>. Estes são exemplos simples das inúmeras etiquetas que o HTML suporta para formatar e mostrar o conteúdo das páginas web.

O código da Figura 3.8 é um exemplo simples que demonstra a utilização de algumas etiquetas como a <link>, que permite a importação de ficheiros (neste caso CSS) que descrevem como os elementos HTML devem ser mostrados, a etiqueta <meta> que permite adicionar metadados acerca da página, autor ou outras (informação que não é mostrada aos utilizadores) e a etiqueta que é uma divisão ou secção de um documento HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <!--Import materialize.css-->
    <link type="text/css" rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.97.
    0/css/materialize.min.css"  media="screen,projection"/>
    <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
    <link rel="stylesheet" href="/stylesheets/jquery.minicolors.css" type="text/css" media="screen"/>
    <link rel="stylesheet" href="/stylesheets/conference_style.css" type="text/css" media="screen"/>
    <!--Let browser know website is optimized for mobile-->
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  </head>

  <body>
    <div class="main">
      <div class="sidebar" id="sidebar">
        <div class="topbar blue valign-wrapper">
          <h5 class="grey-text text-lighten-5 valign center" style="width: 100%;" id="my-uid"><%=userid%></h5>
        </div>

        <div class="menu">
          <div class="contacts">
            <ul class="collection">
            </ul>
          </div>
        </div>

        <div class="myCam">
          <video class="responsive-video" muted autoplay id="myCam"></video>
        </div>
      </div>
    </div>
  </body>
</html>
```

Figura 3.8: Código HTML

O código HTML é uma linguagem de marcação *client-side*, ou seja o código é interpretado e executado no *browser* do cliente, ao contrário de linguagens como PHP, C#, Java, etc. que são executadas num servidor e só depois enviadas para o cliente em forma de HTML.

O WebRTC usa HTML5, a última versão desta linguagem, em desenvolvimento pelo grupo Web Hypertext Application Technology Working Group (WHATWG) e o consórcio W3C desde 2008.

O HTML5 trouxe novos recursos, sobretudo ao nível da sintaxe, sendo facilmente integrado de forma nativa os conteúdos multimédia como vídeo e áudio. Esta arquitetura *web* também pode conter embebido na linguagem HTML código *JavaScript*, que vai interagir com os *browsers*, de forma a correr código do lado do cliente. O WebRTC, usa também *JavaScript server-side* ou seja, código que será interpretado por um servidor *JavaScript*.

3.2.8 JavaScript

JavaScript é uma linguagem de programação interpretada, orientada a objetos, amplamente usada em ambiente *web*. Inicialmente criada para correr código nos *browsers* para interagir com o cliente, sem necessidade de contactar qualquer servidor. Neste momento é também possível correr código *JavaScript* no lado do servidor, por exemplo com um servidor *nodejs*[3]. Sendo a linguagem mais usada para os *browsers* correrem código, serve por exemplo para verificar informação introduzida num formulário, ou seja, um campo que irá receber o NIF de uma pessoa só deverá aceitar 9 caracteres numéricos. Com *JavaScript* é possível verificar essa informação no lado do cliente, no momento em que está a ser inserido. Como pode ser observado na Figura 3.9, o código *JavaScript* pode ser embebido no código HTML com as etiqueta `<script>`, ou pode estar num ficheiro separado do código HTML e ser importado à semelhança dos ficheiros Cascading Style Sheets (CSS) com o código `<script src="ficheiro.js»`.

[3]https://nodejs.org/en/

```
<!DOCTYPE html>
<html lang="pt-BR">
    <head>
        <meta charset="UTF-8" />
        <title>WebRTC Rules</title>
        <script>
            window.onload = function () {
                document.getElementById("btn").addEventListener("click", function () {
                    alert("Bem-vindo ao WebRTC!");
                }, false);
            };
        </script>
    </head>
    <body>
        <noscript>O seu browser não suporta JavaScript.</noscript>
        <button id="btn">Clique aqui</button>
    </body>
</html>
```

Figura 3.9: Código JavaScript

3.2.9 NAT

O objetivo do Network Address Translation (NAT) é permitir que, com um único endereço Internet Protocol (IP), seja possível aceder à internet através de vários dispositivos, partilhando assim o mesmo IP público [Srisuresh and Egevang, 2001]. O NAT está descrito na RFC 3022 e tem diversas vantagens, mas, a principal assim como inicial função é economizar os endereços IPv4 públicos, uma vez que são recursos escassos, embora existam alguns tipos de NAT que não o procedam dessa forma.

O endereçamento IP não é ilimitado e com a massificação das TIC, a Internet Assigned Numbers Authority (IANA), entidade responsável pela coordenação global dos endereços da internet, corria o risco de esgotar o endereçamento, impossibilitando o acesso de novos dispositivos à rede. De forma a limitar a distribuição dos IP's públicos foram desenvolvidas algumas soluções, entre elas o NAT, embora a implementação mais simples do NAT não economizar endereços.

Há diversos tipos de NAT, mas todos eles se incluem em duas categorias, estático e dinâmico. O NAT estático é configurado manualmente de forma a manter o mapeamento de IP's e está, normalmente, associado ao tráfego de entrada (sentido *untrusted, trusted*). O mapeamento do NAT dinâmico é criado automaticamente pelo router quando é iniciada a comunicação e está associado ao tráfego de saída (sentido *trusted, untrusted*).

Qualquer pacote IP contém o endereço IP de origem, a porta de origem, o endereço IP de destino e a porta de destino. Todos os tipos de NAT criam os mapeamentos usando estes valores.
O objetivo desta secção é demonstrar os tipos de NAT e identificar o que implica com a comunicação mais direta do WebRTC.

O mapeamento de endereços IP e portas através do NAT pode ser elaborado usando uma das seguintes formas:

1. **NAT Básico (estático)**
   Este tipo de NAT mapeia endereços IP locais internos para endereços IP globais internos. Desta forma, um endereço IP local interno, irá ter sempre o mesmo endereço IP global interno. Este tipo de NAT é útil quando existem dispositivos na rede local interna que precisam de ser acedidos pela internet, através de um IP único.

2. **NAT Básico (dinâmico)**
   O NAT básico, idêntico ao anterior, mapeia endereços IP locais internos para uma gama de endereços IP globais internos mas de forma dinâmica. Desta forma, os endereços locais internos podem ser diferentes dos endereços globais internos, ao contrário do definido no NAT básico estático.

3. **Network Address Port Translation (NAPT) ou Port Adress Translation (PAT)**
   No caso do NAPT, ao invés de mapear endereços IP, mapeia números de portas dos endereços locais internos.

Network Address Translation - Traversal (NAT-T) é o termo que define a técnica que estabelece e mantém ligações ponto-a-ponto, atravessando equipamentos configurados com NAT, como pode ser visto na Figura 3.10[4]. Na Figura 3.11[5] pode ser analisado o mesmo conceito mas aplicado ao NAPT, ou seja o Network Address Port Translation - Traversal (NAPT-T).

[4]http://www.cisco.com
[5]http://www.cisco.com

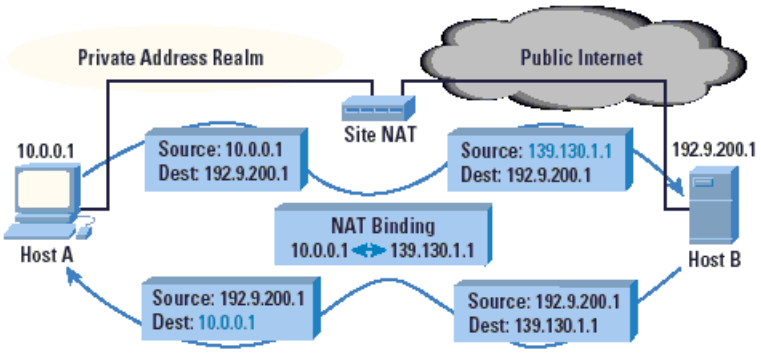

Figura 3.10: NAT Traversal

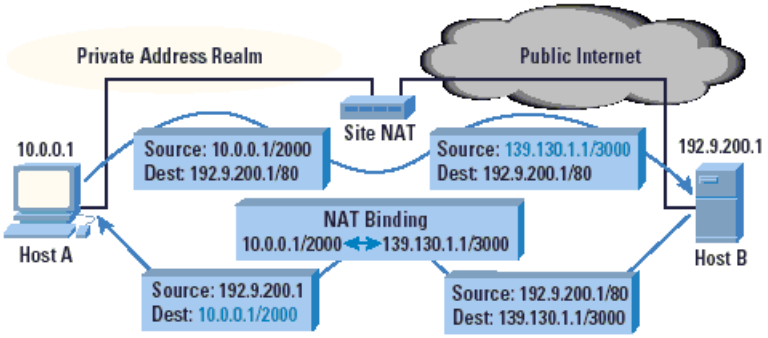

Figura 3.11: NAPT Traversal

Além das características descritas, a tradução/mapeamento de endereços, o NAT também pode aplicar restrições de modo a elevar o nível de segurança. Essas restrições têm por base os endereços IP internos e externos, assim como o número das portas internas e externas.

As restrições de endereços IP e portas existentes são as seguintes:

1. ***Full cone* NAT (estático)**
   O *Full cone NAT* como pode ser visto na Figura 3.12[6], também conhecido como "um para um"é o menos restritivo de todos. A comunicação apenas terá que respeitar a porta para ter sucesso. Mapeia um IP e uma porta externa para um IP e uma porta interna. Aceita comunicações de qualquer IP externo.

[6]http://www.cisco.com

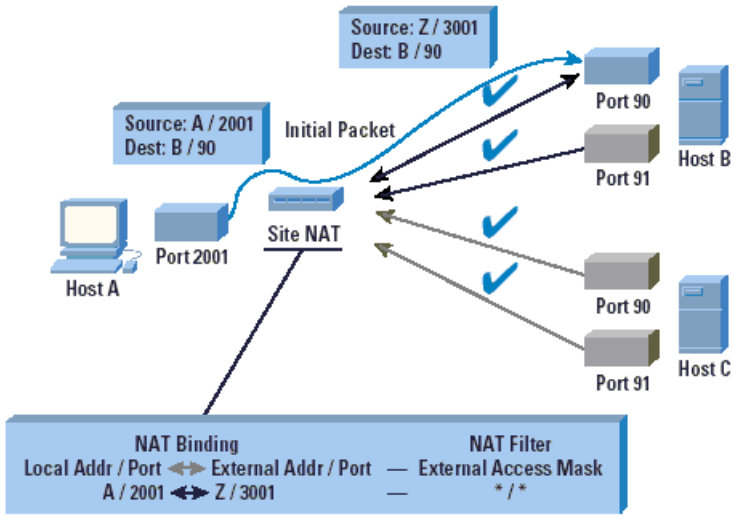

Figura 3.12: Full cone NAT

2. ***Restricted cone* NAT (dinâmico)**

   O funcionamento do *Restricted cone NAT*, que pode visto na Figura 3.13[7], é idêntico ao *Full cone NAT*. Aplica restrições apenas ao nível do endereço IP. A comunicação tem que ser iniciada pelo IP interno e só depois poderá receber pacotes do IP externo. Numa situação inversa, a comunicação não é bem sucedida.

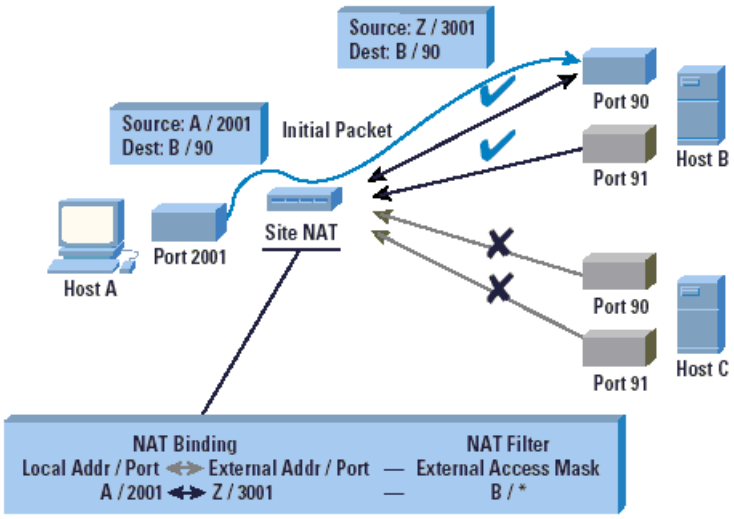

Figura 3.13: Restricted Cone NAT

[7]http://www.cisco.com

3. ***Port restricted cone* NAT (dinâmico)**

   O *Port restricted cone NAT*, ilustrado na Figura 3.14[8] adiciona mais uma camada de proteção. O funcionamento é idêntico ao *Restricted cone NAT* mas aplica restrições também ao nível das portas. Ou seja, a comunicação só é bem sucedida se tanto o endereço IP como a porta igualarem aos do *host* interno.

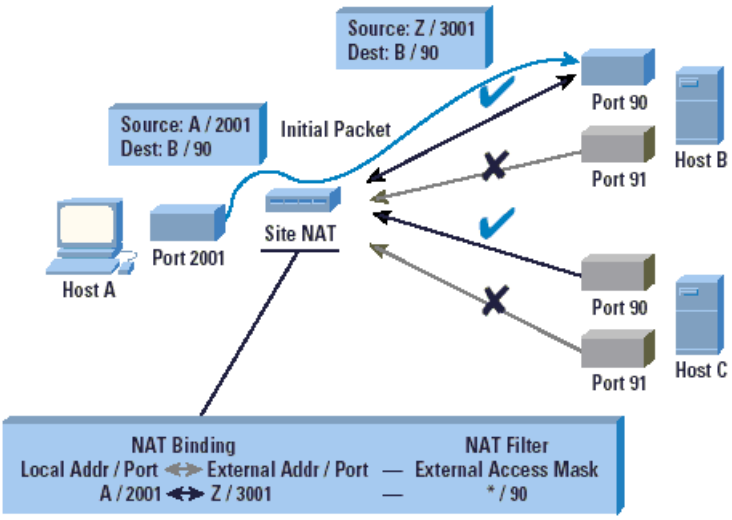

Figura 3.14: Port Restricted Cone NAT

4. ***Symetric* NAT (dinâmico)**

   O *Symetric NAT* representado na Figura 3.15[9], aplica as restrições de forma idêntica ao *Port restricted cone NAT*, mas aplica a tradução de forma diferente. Os tipos de NAT referidos anteriormente não alteram o número da porta de comunicação ao efetuar a tradução, mantêm a mesma porta usando a característica de *port preservation*. O NAT simétrico mapeia as ligações para uma porta gerada aleatoriamente, mantendo esta característica a cada nova ligação. Num exemplo simples e rápido, uma ligação com o IP 192.168.10.1 e porta 45678 é mapeada para o IP externo 10.10.10.1 e porta 56045 (gerada aleatoriamente). Quando o *host* de destino desta comunicação responder, a comunicação terá que ser efetuada através da porta 56045, mantendo os requisitos adicionais do *Port restricted cone NAT*.

[8] http://www.cisco.com
[9] http://www.cisco.com

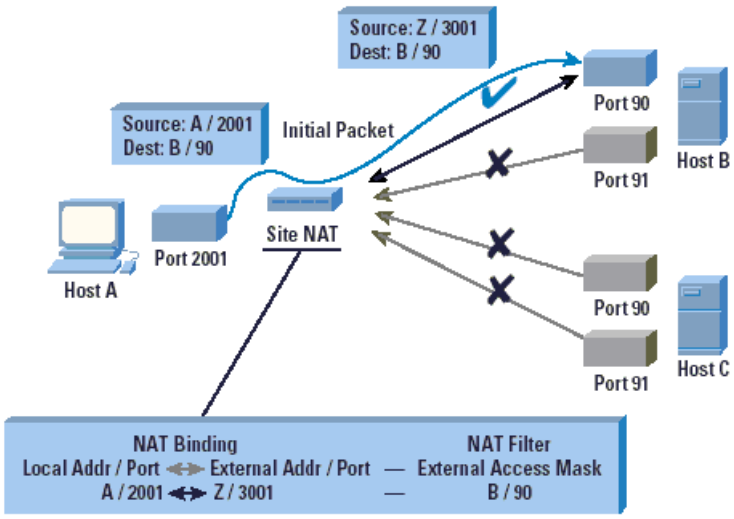

Figura 3.15: Symmetric NAT

3.3 A tecnologia WebRTC

Neste secção apresentam-se os componentes de um sistema WebRTC. A Figura 3.16 descreve o funcionamento típico do WebRTC. Nestes elementos incluem-se servidores *web*, *browsers* que poderão ser executados em vários sistemas operativos e em vários dispositivos como computadores, *smartphones* ou *tablets*. Podem ainda ser ligados ao WebRTC, através de *gateways*, outros sistemas como o telefone ou o VoIP através do protocolo Session Initiation Protocol (SIP) ou Jingle que são protocolos de sinalização.

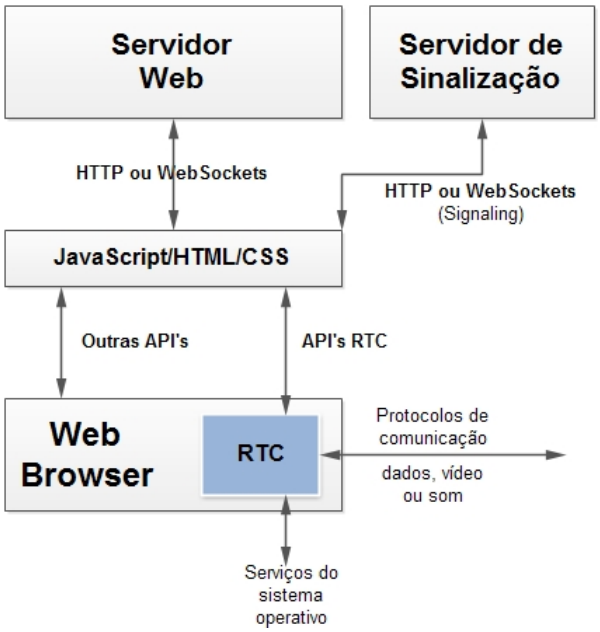

Figura 3.16: Elementos do WebRTC

3.3.1 API

Seguidamente apresentam-se as três *API* principais do *WebRTC*: *MediaStream*, *PeerConnection* e *DataChannel*.
Na Figura 3.17 é possível compreender as tarefas principais destas *API*.

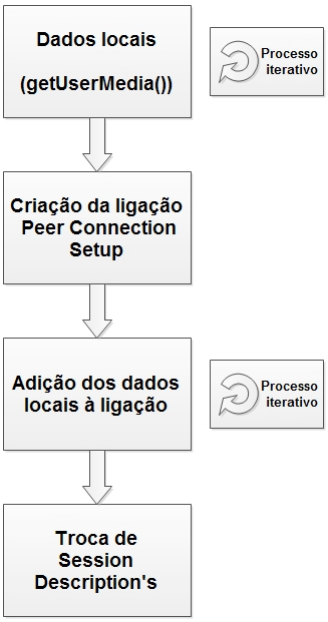

Figura 3.17: Ligação WebRTC do ponto de vista da API

MediaStream

A API *MediaStream* tem dois componentes principais que são o *MediaStreamTrack* e o *MediaStream*. O objeto *MediaStreamTrack* representa os dados de um tipo específico, capturado por um dos dispositivos como o microfone ou a câmara de vídeo. O objeto *MediaStream* é usado para agrupar os vários objetos *MediaStreamTrack* num só *MediaStream*, que pode ser enviado a um *peer* remoto ou reproduzido localmente.

A Figura 3.18 representa abstratamente um fluxo sincronizado de informação local ou remota a ser enviado pelo *PeerConnection*. Pode conter múltiplas camadas de dados como áudio ou vídeo. Esta função acede diretamente aos dispositivos, de forma a criar uma fonte de informação para posterior tratamento ou envio a outro ponto. O WebRTC define o método *getUserMedia()* para obter uma ou mais *MediaStream*.

Sempre que é necessário aceder localmente a um qualquer elemento de input, por motivos de segurança o utilizador tem que autorizar esse acesso.

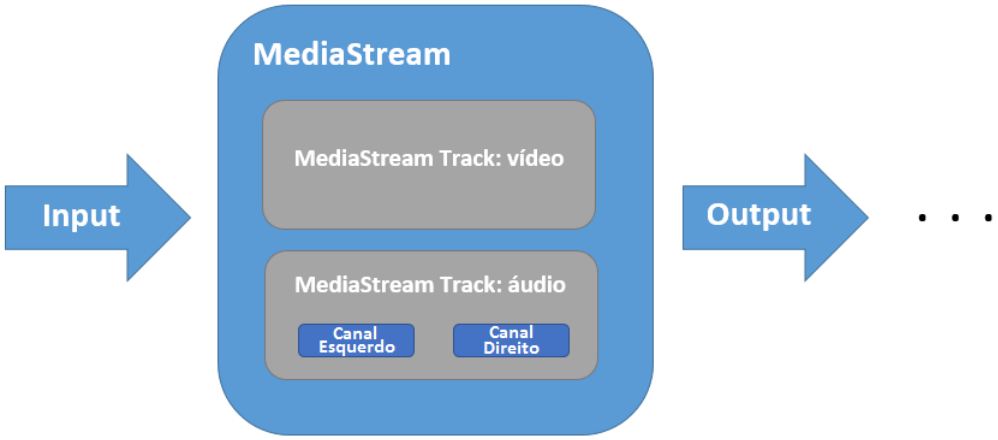

Figura 3.18: Representação abstrata do MediaStream

PeerConnection

A API *PeerConnection* é o objecto principal do sistema WebRTC, já que um dos passos mais importantes é o estabelecimento da ligação ponto-a-ponto. Esta API cria a ligação aos *peers* e adiciona o conteúdo multimédia obtido através da API *MediaStream*. É criado um canal de forma a ligar os dois *browsers*, permitindo a comunicação de áudio e vídeo com gestão da largura de banda e de segurança. Gera o pedido e a resposta de *Session Description*.
O SDP, será detalhado na Subsecção 3.3.2.

DataChannel

A API *datachannel* representa um canal de dados bidirecional entre os dois pontos. O objetivo desta API é criar o canal através da ligação efetuada com a API *PeerConnection*, de forma a enviar os dados obtidos através da API *mediastream*. Esta permite a ligação entre os pontos para trocarem informações e dados de uma forma segura, com baixa latência e alta velocidade. A API *datachannel* fornece controlo de rotas e gestão de congestão com o Stream Control Transmission Protocol (SCTP) e segurança com o Datagram Transport Layer Security (DTLS), um protocolo na camada de transporte,

3.3.2 Protocolo de descrição de sessão

O SDP foi especificado pela primeira vez em 1998, na RFC 2327 e tem uma revisão na RFC 4566. É um protocolo usado para trocar e negociar as propriedades e atributos multimédia, de ligação, de segurança entre outros metadados, nas ligações que requerem esse tipo de

funcionalidade como chamadas VoIP, *streaming* de vídeo e no WebRTC. O SDP apenas transporta informação de sessão, não transportando os dados multimédia em si. Como pode ser observado na Figura 3.19, uma sessão de SDP contem por exemplo o nome da sessão e o seu propósito, o tempo em que a sessão está ativa, o tipo de dados e a informação necessária para os receber como endereços IP, portas, formatos, informação sobre a largura de banda, etc [Handley et al., 2006].

```
Session description
   v=  (protocol version)
   o=  (originator and session identifier)
   s=  (session name)
   i=* (session information)
   u=* (URI of description)
   e=* (email address)
   p=* (phone number)
   c=* (connection information -- not required if included in
        all media)
   b=* (zero or more bandwidth information lines)
   One or more time descriptions ("t=" and "r=" lines; see below)
   z=* (time zone adjustments)
   k=* (encryption key)
   a=* (zero or more session attribute lines)
   Zero or more media descriptions

Time description
   t=  (time the session is active)
   r=* (zero or more repeat times)

Media description, if present
   m=  (media name and transport address)
   i=* (media title)
   c=* (connection information -- optional if included at
        session level)
   b=* (zero or more bandwidth information lines)
   k=* (encryption key)
   a=* (zero or more media attribute lines)
```

Figura 3.19: Tipo de informação no SDP

Na Figura 3.20 pode observar-se alguns dos dados que serão transmitidos para estabelecer a sessão.

Time	Event
03/02/2016, 22:34:58	▼ setRemoteDescription type: offer, sdp: v=0 o=- 2473604334433976214 2 IN IP4 127.0.0.1 s=- t=0 0 a=msid-semantic: WMS m=application 9 DTLS/SCTP 5000 c=IN IP4 0.0.0.0 a=ice-ufrag:ppNDddxGILiOV01d a=ice-pwd:Bk/iFabhWIuYnH4hgWZeFR/3 a=fingerprint:sha-256 9E:65:D7:6B:8B:AB:3A:4F:E7:B3:A1:72:6F:21:FF:76:FD:E4:02:8F a=setup:actpass a=mid:data a=sctpmap:5000 webrtc-datachannel 1024

Figura 3.20: Session description protocol

3.3.3 Interactive Connectivity Establishment

A *framework* Interactive Connectivity Establishment (ICE) permite que dispositivos protegidos por NAT ou *firewalls* consigam comunicar diretamente ponto-a-ponto, em tempo real. Esta *framework* é a junção dos serviços Session Traversal Utilities for NAT (STUN) e Traversal Using Relay NAT (TURN), com alguns melhoramentos que seguidamente serão explanados.

O WebRTC recorre ao STUN como primeira opção, sendo o primeiro serviço que utiliza caso os clientes não conheçam o IP e a porta da rede externa por onde comunicam com o exterior. Como foi referido anteriormente, os clientes precisam trocar várias informações antes que possam comunicar diretamente entre si. Uma dessas informações é a dupla IP e porta da rede externa por onde o cliente possa ser contactado, informação essa que o cliente por norma desconhece. O STUN fornece essa informação, ou seja, o cliente estabelece uma comunicação a um servidor STUN e, na resposta, recebe essa informação para assim a poder trocar com o outro *peer*.

Existem algumas redes com configurações de NAT ou com *firewalls* que impedem o comportamento descrito anteriormente. O que acontece nestes casos é que, a cada nova ligação que o cliente faça, obtém um IP e porta externa diferente da sua última ligação, impedindo assim que a informação que lhe chega via servidor STUN seja atualizada, falhando a ligação. Nestes casos, o WebRTC recorre a um servidor TURN, que irá servir de *relay* para transmitir os dados.

Cada vez que um utilizador se liga ao WebRTC para comunicar com outro, o processo de descoberta de IP e porta é indispensável e requerem várias tentativas que podem demorar alguns segundos. Como o WebRTC escolhe sempre a ligação mais eficiente, testa sempre todas as ligações até que encontre um IP e porta a que consiga aceder, ou seja, primeiro tenta usar o IP interno, depois o IP do NAT e, por fim o IP do *relay*. Para diminuir o tempo de espera na procura dos detalhes da ligação mais eficiente, o WebRTC usa o ICE de forma a reunir todas as ligações candidatas.

A primeira ligação candidata é a da interface local, chamada de *Host candidate*, seguidamente, o ICE recolhe a ligação do NAT, chamada de *Server Reflexive Address* e, por último,

a ligação do TURN chamada *Relayed candidates*. Após ter recolhido todas as ligações candidatas, o ICE prioriza-as seguindo uma fórmula que combina o tipo de candidato, o número e tipo de ligações IP e ainda um identificador.

3.3.4 Sinalização

A sinalização é um processo usado no WebRTC para detetar os *peers*, os seus detalhes de rede e capacidades de áudio e vídeo. É usado para trocar mensagens de controlo de sessão com o protocolo SDP e trocar informações de rede como os candidatos ICE. O protocolo de sinalização está definido na RFC 4566.
De forma a evitar a redundância e a maximizar a compatibilidade com as tecnologias existentes como o VoIP e o telefone, o processo de sinalização não está definido na norma WebRTC. Esse processo está ainda a ser discutido e desenvolvido no *draft* JavaScript Session Establishment Protocol (JSEP).

A sinalização não é um processo exclusivo do WebRTC, existem outras tecnologias de comunicação ponto-a-ponto que usam estes processos, como é o caso do VoIP.
Para haver comunicação é necessário que os clientes troquem informações, nomeadamente informação de rede, largura de banda disponível, chaves de cifras, entre outras.

Os utilizadores que estejam protegidos por NAT ou *firewalls*, veem o seu IP e/ou porta ser alterado, impossibilitando a troca de quaisquer informações ponto-a-ponto, uma vez que os utilizadores não conhecem essa informação. De forma a resolver esse comportamento, o WebRTC usa um mecanismo que permite aos clientes conhecerem o seu IP e porta externos.

Os clientes contactam um servidor que esteja ligado diretamente à Internet para que dessa forma esse servidor fique a conhecer o IP e porta externos. Como foi indicado anteriormente o WebRTC usa a API *PeerConnection* por forma a criar a ligação entre os clientes e a comunicar áudio e vídeo. Esta API é responsável por duas importantes tarefas: reunir as características de hardware e software dos intervenientes da comunicação; e perceber quais as potenciais rotas para assegurar a comunicação seja efetuada. Depois de obter essas informações são calculadas as mais eficientes e trocadas com os clientes através do serviço de sinalização.
A Figura 3.21 ilustra a comunicação ponto-a-ponto no ambiente mais simples, ou seja, num

ambiente onde os participantes da comunicação conhecem a informação um do outro e conseguem contactar-se usando essa informação.

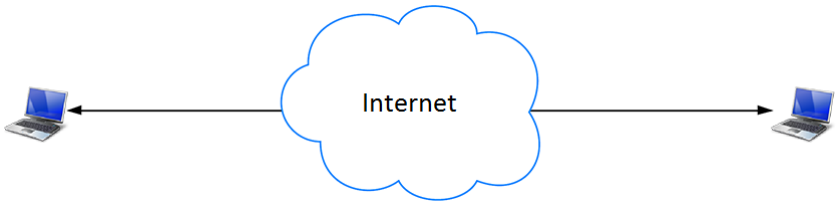

Figura 3.21: Ponto-a-ponto na mesma rede

Adaptado de: http://www.html5rocks.com/en/tutorials/webrtc/infrastructure/p2p.png

Como as redes não estão construídas e configuradas da mesma forma, existem alguns casos onde é necessário um maior nível de proteção, usando para isso, configurações NAT e *firewalls* mais apertadas ao nível da segurança.

3.3.5 Media Connections

Uma das barreiras à utilização maciça das comunicações VoIP, idênticas na forma de ligação ao WebRTC, são as restrições naturais das *firewals* e NAT's. Este tipo de equipamento tem um papel crucial na segurança das redes internas, escondendo e protegendo equipamentos internos e prevenindo, assim, expor máquinas e dados às redes externas. Neste caso, são apenas mostrados os serviços que se pretendam acessíveis a partir de redes externas.

No entanto, dadas as suas característica intrínsecas este tipo de equipamento bloqueia a comunicação ponto-a-ponto e impede que a comunicação seja iniciada do exterior, uma vez que não são conhecidos os endereços internos. Para ultrapassar este problema, o WebRTC usa uma estratégia de NAT *traversal* simples e embutida com a norma ICE que faz uso do protocolo STUN.

- STUN - Session Traversal Utilities for NAT
  O protocolo STUN, definido na RFC 5389, permite verificar se o equipamento está protegido por um NAT e, caso esteja, determina o seu IP público e o tipo de NAT que está configurado, de forma a ajudar a estabelecer a comunicação ponto-a-ponto. O serviço está normalmente ligado diretamente à internet. Este é o primeiro passo quando um dos pontos está protegido por um NAT. O cliente estabelece uma ligação

ao servidor STUN e este responde com o seu IP e porta externos, de forma a que o cliente envie essa informação ao outro extremo da comunicação. O processo de estabelecer uma comunicação ponto-a-ponto entre duas partes que estejam protegidas por NAT é chamado de *hole punching*.

A Figura 3.22[10], representa a comunicação ponto-a-ponto usando um servidor STUN

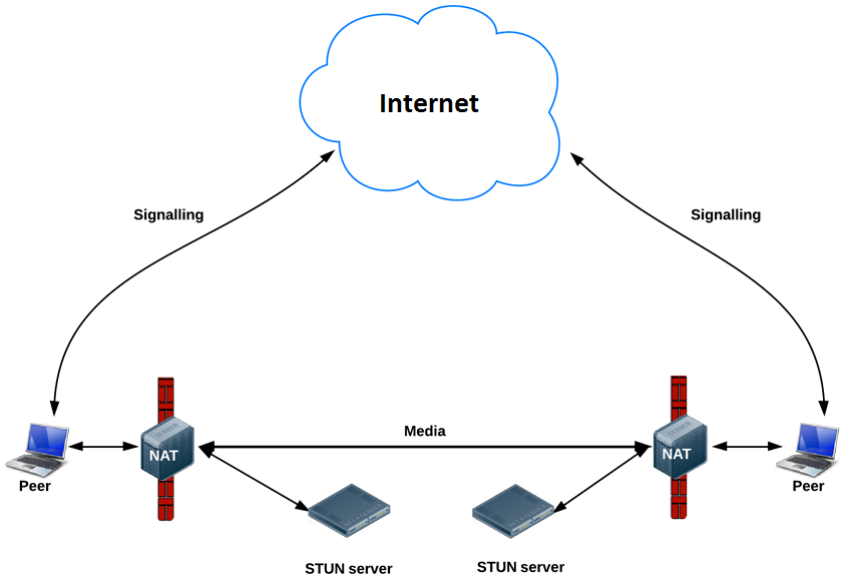

Figura 3.22: Ponto a aponto com servidor STUN

Num cenário ideal para a comunicação ponto-a-ponto, todos os pontos estariam acessíveis diretamente ou, quando muito, seria necessário este serviço para os apresentar. Felizmente, esse é o caso de 90 a 95% das redes. Para os restantes casos, o WebRTC usa o TURN.

- TURN - Traversal Using Relay NAT

  Embora o STUN seja eficaz na maior parte das configurações de NAT nos routers de consumo, por si só não consegue ser eficaz em redes corporativas onde, por norma, são aplicadas políticas de segurança mais restritivas, impedindo assim o funcionamento do STUN. Para contornar esta questão, o WebRTC usa a extensão TURN do protocolo STUN.

  O TURN está definido na RFC 5766 e é uma extensão para o protocolo STUN que facilita o NAT *traversal* quando um ou mais pontos estão protegidos por NAT. Na

[10] Adaptado de: http://www.html5rocks.com/en/tutorials/webrtc/infrastructure/stun.png

prática o STUN serve de intermediário na comunicação, deixando essa de ser verdadeiramente ponto-a-ponto, uma vez que a comunicação terá que ser canalizada por este serviço. A Figura 3.23[11] representa a comunicação usando um servidor TURN, que desta forma não é ponto-a-ponto, embora seja em tempo real.

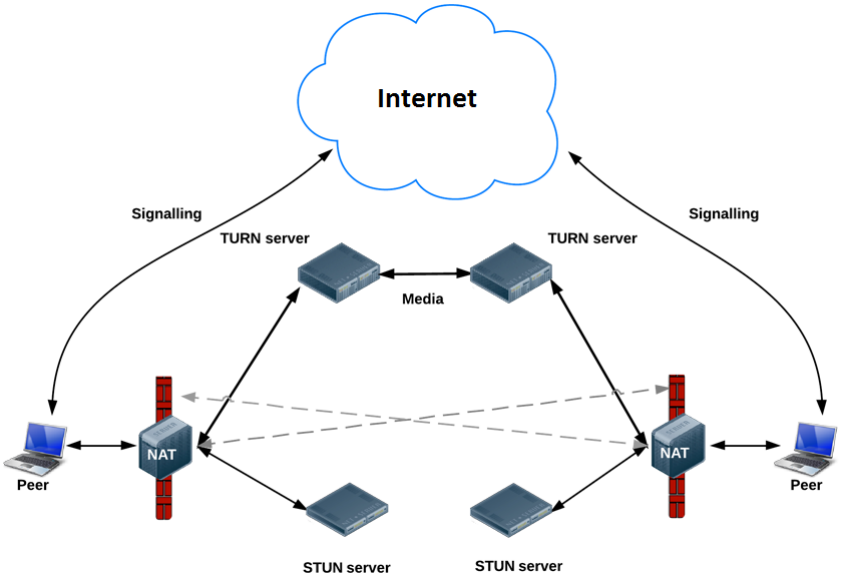

Figura 3.23: Comunicação com um servidor TURN

O WebRTC não foi desenhado para lidar com pormenores de *networking* como o *handshake* (processo pelo qual duas máquinas afirmam uma à outra que se reconhecem e estão prontas para iniciar a comunicação), o *hole punching*, ou simplesmente a falha da comunicação. Se o STUN falhar, é necessário iniciar uma nova sessão para tentar a comunicação com o TURN e estas tentativas demoram tempo, adicionando latência a um processo que se pretende que seja *real time*. Para minimizar essa latência, o WebRTC usa o *standard* ICE.

- ICE Interactive Connectivity Establishment
  A norma ICE está definida na RFC 5245 com atualizações na RFC 6336, é usado pelo NAT traversal nos pedidos de protocolos de resposta e oferta. Na prática, este serviço verifica a oferta de caminhos possíveis, ordena-os de forma a ser escolhido o mais direto e eficiente e entrega essa informação aos potenciais comunicadores, sendo por exemplo o TURN preterido em relação ao STUN.

[11] Adaptado de: http://www.html5rocks.com/en/tutorials/webrtc/infrastructure/turn.png

- Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (Trickle ICE) O *Trickle ICE*, em fase de *draft*, está proposto para *standard* pelo grupo IETF. O *Trickle ICE* permite reduzir, ainda mais, o tempo para o estabelecimento da ligação. O *Trickle ICE* tenta a ligação a cada candidato novo que recebe, ao invés do ICE, que só irá tentar a ligação quando receber todos os candidatos.

3.4 Vantagens e desvantagens

O WebRTC é uma norma de código aberto sendo atualmente, a única solução que possibilita a comunicação de áudio, vídeo e dados em tempo real, diretamente entre dois ou mais *browsers*. O utilizador não necessita instalar software adicional, sendo por isso de fácil e rápida utilização.

As comunicações são cifradas por forma a não ser possível ler ou modificar o conteúdo, quer seja vídeo, áudio ou dados. É independente de plataformas e de equipamentos, uma vez que é executado em *browsers*. As ligações são fiáveis e diretas entre os navegadores, retirando também carga dos servidores. Os diversos tipos de dados e ligações são enviados através de uma única ligação, poupando tempo no estabelecimento das ligações através do SDP. O uso de protocolos como RTP Control Protocol (RTCP) e Secure Audio Video Profile with Feedback (SAVPF), permite o *feedback* das condições de rede por forma a que a sessão se adapte eficazmente. Usa diversas API's para suporte a múltiplos tipos de dados e dispositivos emissores, negociando o tamanho e formato de cada tipo, conseguindo assim o uso eficiente da largura de banda disponível. Consegue fornecer interoperabilidade com sistemas legados como o telefone ou VoIP, através de protocolos *standard* como o SIP.

A maior desvantagem do WebRTC, neste momento, é a falta de compatibilidade nativa com alguns *browsers*. Essa desvantagem irá desvanecer-se assim que o *browser* Internet Explorer suporte o WebRTC, embora existam *plugins* disponíveis para que este navegador possa suportar, desde já, o WebRTC. Outra desvantagem que será diluída a curto prazo é ausência de definição de algumas especificações, uma vez que estão ainda em discussão nos grupos de trabalho e na comunidade *web*.

A Figura 3.24 ilustra uma análise Strengths Weaknesses Opportunities Threats (SWOT) ao WebRTC.

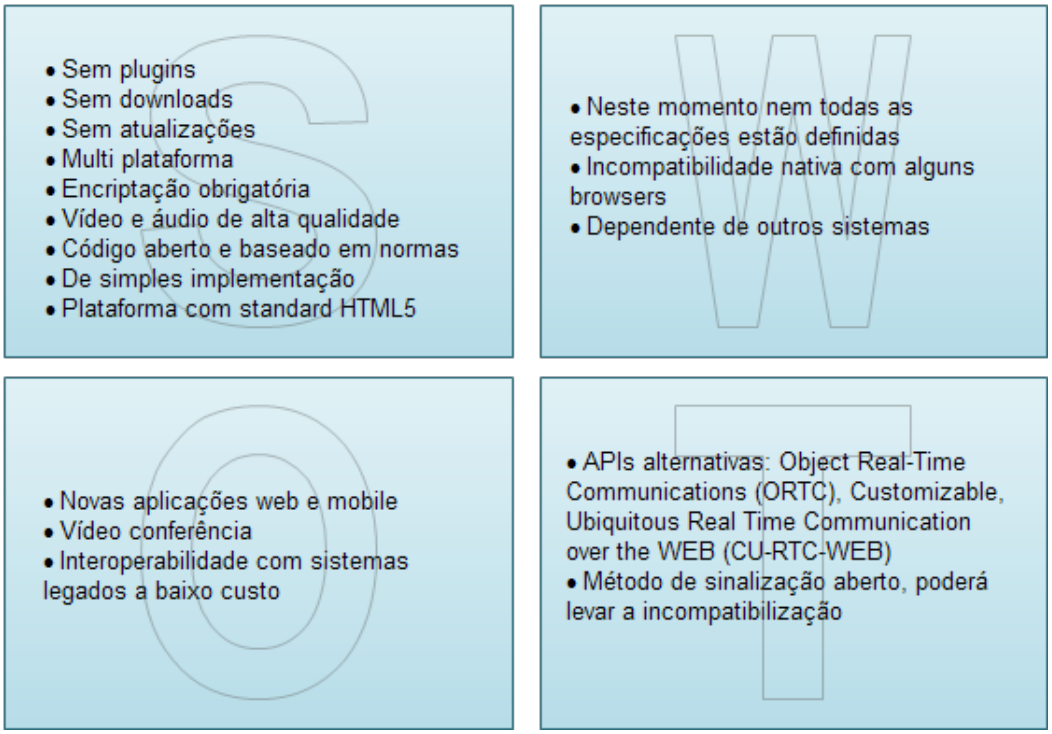

Figura 3.24: Análise SWOT ao WebRTC

3.5 Segurança e privacidade

Sempre que há transferência de dados na Internet, a segurança dos dados é uma questão fulcral mas, por vezes, negligenciada. Na área da saúde essa segurança é primordial. No caso do WebRTC, uma vez que acede à câmara e ao microfone em tempo real, dada a natureza dos dados, a segurança é especialmente importante. Na perspetiva Do lado aplicacional, o WebRTC está embebido no *browser*, logo irá receber atualizações de segurança quando este for atualizado dificultando assim, que *plugins*, vírus ou *malware* consigam aceder e comprometer o código. Em todos os componentes do WebRTC, incluindo a sinalização, a cifragem é obrigatória.

A segurança dos dados durante a comunicação está prevista com a utilização de dois protocolos, o DTLS e Secure Real-time Transport Protocol (SRTP).
O protocolo DTLS definido na RFC 6347, foi desenhado para prevenir escutas não autorizadas, manipulação, alteração e falsificação de mensagens transferidas pelo protocolo UDP.

A troca inicial de informações, conhecida neste caso como *DTLS handshake*, nada mais é do que um *upgrade* ao conhecido *3 way handshake* comum nas comunicações, adicionado de troca de certificados e troca de especificações da chave usada para cifrar a comunicação. Na altura da elaboração deste trabalho, o processo conhecido como *DTLS handshake* era assente em certificados *self-signed*, e, dessa forma não é possível serem usados para autenticar inequivocamente, os clientes uma vez que não existe uma cadeia de confiança para verificar e atestar os certificados. O grupo de trabalho W3C está a preparar um mecanismo para resolver esta questão, de forma a que seja possível verificar a autenticidade dos certificados.
O SRTP, definido na RFC 3711, fornece segurança através de encriptação, autenticação e integridade. O SRTP é usado na comunicação de voz, vídeo e dados enquanto o DTLS é usado na sinalização.

3.6 Funcionamento

De seguida, descreve-se o cenário típico de uma ligação WebRTC.

1. Criação de um objecto *MediaStream* com a captura de vídeo e ou som através da api *MediaStream*
2. Obter a *URL* contendo a *MediaStream* local
3. Executar localmente o conteúdo multimédia obtido anteriormente
4. Criação da ligação entre os interlocutores com a api *PeerConnection*
5. Adicionar o conteúdo multimédia à ligação criada anteriormente
6. Enviar, receber e processar as informações de sessão com o SDP
7. Obter a *URL* contendo a *MediaStream* remota
8. Executar localmente o conteúdo multimédia obtido anteriormente

Este cenário não inclui o processo de descoberta e ligação dos pontos conhecido por sinalização, uma vez que é deixado ao critério do programador a implementação a usar. O processo de sinalização será descrito no próximo capítulo.

Na Figura 3.25 pode-se observar o modelo triangular que é o modelo mais comum do WebRTC. Neste modelo ambos os clientes executam a mesma aplicação *web* tendo sido acedida ao mesmo servidor *web* [Salvatore Loreto, 2014].

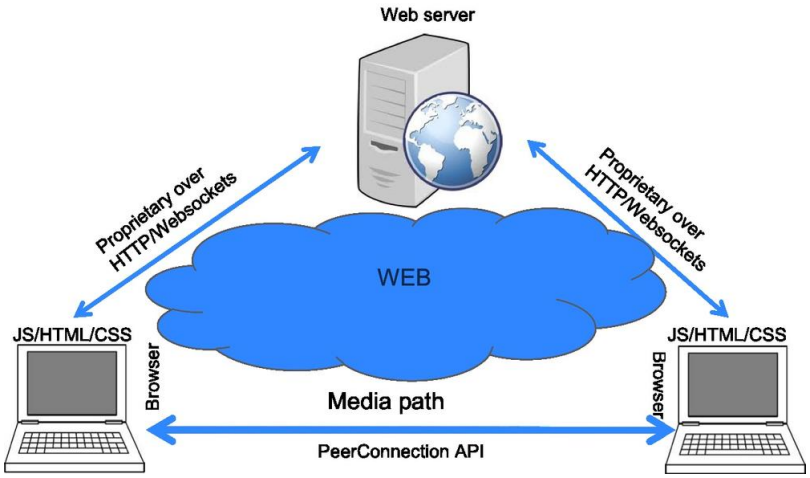

Figura 3.25: Modelo triangular

O modelo WebRTC trapezoide, Figura 3.26 é um modelo genérico que é inspirado no modelo do protocolo SIP definido na RFC3261. Neste caso os clientes executam uma aplicação *web* acedida em diferentes servidores *web* [Salvatore Loreto, 2014].

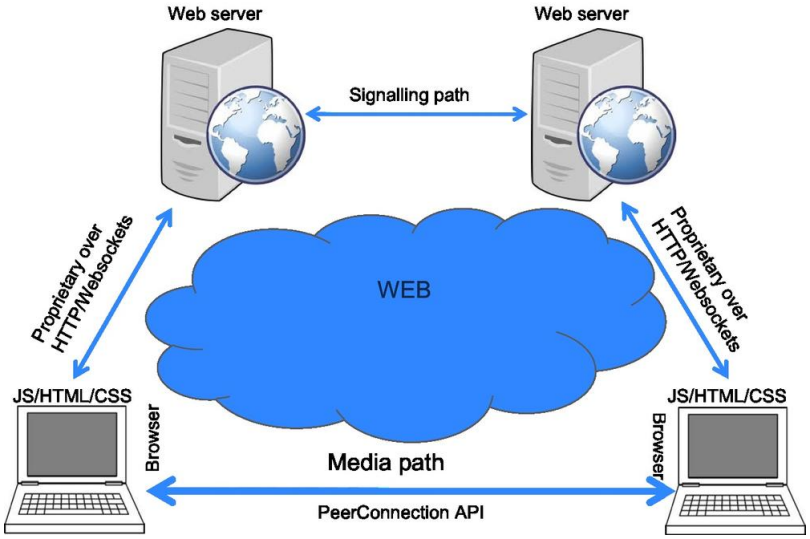

Figura 3.26: Modelo trapezoide

3.7 Conclusão

A tecnologia WebRTC é apresentada como uma solução gratuita para comunicações em tempo real de áudio, vídeo e dados genéricos aplicando algoritmos de segurança obrigatórios necessitando apenas de um navegador de Internet compatível. Estas características por si só são suficientes para que o WebRTC seja reconhecido como uma mais valia. Considerando a possibilidade da integração com o telefone fixo, assim como o VoIP, a adicionando ainda os equipamentos moveis como *smartphones* e *tablets* pode afirmar-se que o potencial do WebRTC é elevado.

Esta página foi intencionalmente deixada em branco.

Capítulo 4

Solução proposta

Este capítulo descreve a solução de telemedicina implementada no âmbito deste trabalho. A solução apresentada tem por base as informações recolhidas, junto de médicos, sobre as funcionalidades fundamentais de uma aplicação de telemedicina. A análise dos requisitos e funcionalidades da aplicação teve em conta não só as características de aplicações comerciais com capacidade de transmissão de vídeo e áudio, usadas em telemedicina, mas também as potencialidades nativas do WebRTC.
Neste capítulo são ainda descritos os resultados preliminares obtidos com o teste do protótipo desenvolvido ao longo desta tese.

4.1 Arquitetura da aplicação

A aplicação desenvolvida é web, assenta no paradigma de comunicação cliente/servidor através do protocolo HTTP. O cliente liga-se a um servidor *web* via HTTP através de um navegador de internet (browser) e solicita um recurso, neste caso uma página HTML com *JavaScript* e CSS. O servidor *web* responde com o recurso solicitado e termina a ligação. Após o acesso à aplicação por parte do cliente, o paradigma passa a ser ponto a ponto, que é precisamente um dos objetivos do WebRTC.

Um dos principais desafios no estabelecimento de uma ligação ponto-a-ponto é a adequada parametrização de firewalls e dos routers. Embora o tráfego do tipo web seja autorizado em todas as redes, os subprotocolos transportados por aquele ou algumas gamas de portas poderão ser negados, o que implicará a não conclusão da ligação. O estabelecimento de uma comunicação através de um router por NAT torna invisível, para o exterior, o endereço IP

correspondente à interface da rede interna. Nesse caso, a ligação ponto-a-ponto deixa de ser possível, já que o estabelecimento da comunicação é feito sempre através do endereço externo (público). A comunicação ponto-a-ponto não será possível e assim o WebRTC estabelecerá uma ligação cliente / servidor entre o browser web e o servidor TURN, que servirá de intermediário na comunicação.

Como se pode ver na Figura 4.1, a solução proposta neste trabalho é constituída por dois servidores aplicacionais; o servidor web e o de sinalização. Embora estes dois servidores aplicacionais pudessem estar alojados na mesma máquina, optou-se por um alojamento em duas máquinas distintas, tentando assim otimizar o desempenho no funcionamento dos serviços disponibilizados, WebRTC e *signaling*. O servidor web é responsável pelo atendimento das ligações HTTP efetuadas entre o *browser* e a aplicação de WebRTC. Quanto ao servidor de sinalização, implementa os mecanismos necessários para o estabelecimento da ligação ponto-a-ponto entre os pontos da ligação, ou seja, serve de facilitador e negociador de todos os detalhes da ligação.

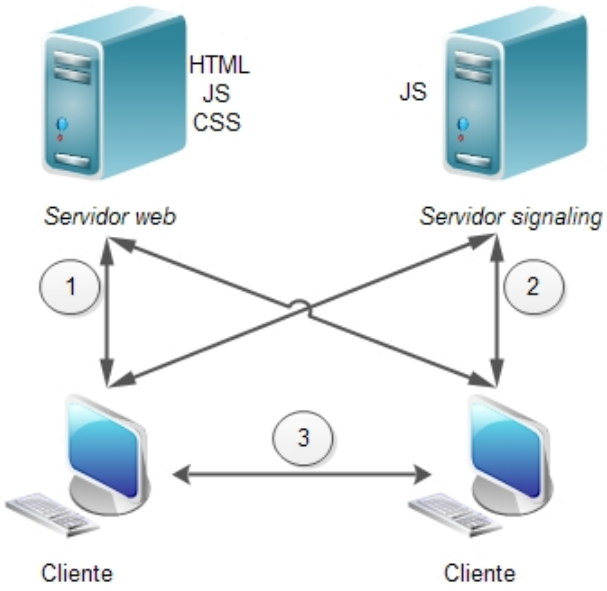

Figura 4.1: Arquitetura servidor *signaling*

A Figura 4.1 ilustra o modo de ligação mais simples, ou seja, para que a comunicação ponto-a-ponto se realize não é necessário outro serviço. O cliente começa por estabelecer uma ligação HTTP ao servidor *Web* (ponto 1) na Figura 4.1 e obtém a página da aplicação com código HTML, *JavaScript* e CSS. Seguidamente, a aplicação estabelece uma ligação

Websocket, ao servidor *Node* que tem alojado o serviço de *signaling* (ponto 2) de forma a trocar informações de sessão entre os participantes.
Caso estejam reunidas as condições para que essa comunicação se faça, os clientes iniciam a comunicação entre si ponto-a-ponto (ponto 3).
Na maioria dos casos o cliente não conhece o seu endereço IP externo porque está numa rede com um serviço de NAT, nestes casos, o WebRTC faz uso de um serviço que possa fornecer o endereço IP externo.

Conforme se ilustra na Figura 4.2, o cliente estabelece uma ligação a um servidor STUN, responsável por inspecionar os pacotes da ligação, de forma a verificar o IP de origem. Este é fornecido ao cliente que, por sua vez, transmite ao servidor responsável pela sinalização (servidor de signalig) para realizar a negociação.

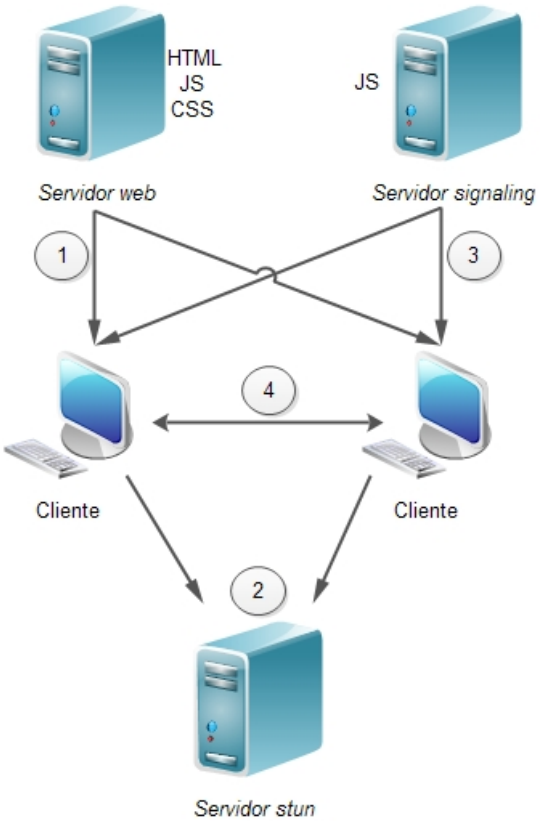

Figura 4.2: Arquitetura Servidor STUN

Na Figura 4.3 podem ser observados os eventos ocorridos no processo de ligação através do protocolo SDP, nomeadamente a receção dos candidatos ICE que pode ser vista com mais

detalhe com a visualização do conteúdo de uma frame na Figura 4.4; o método *setlocaldescription*, que define os detalhes da descrição local associada à ligação, como por exemplo o *codec* a ser usado.

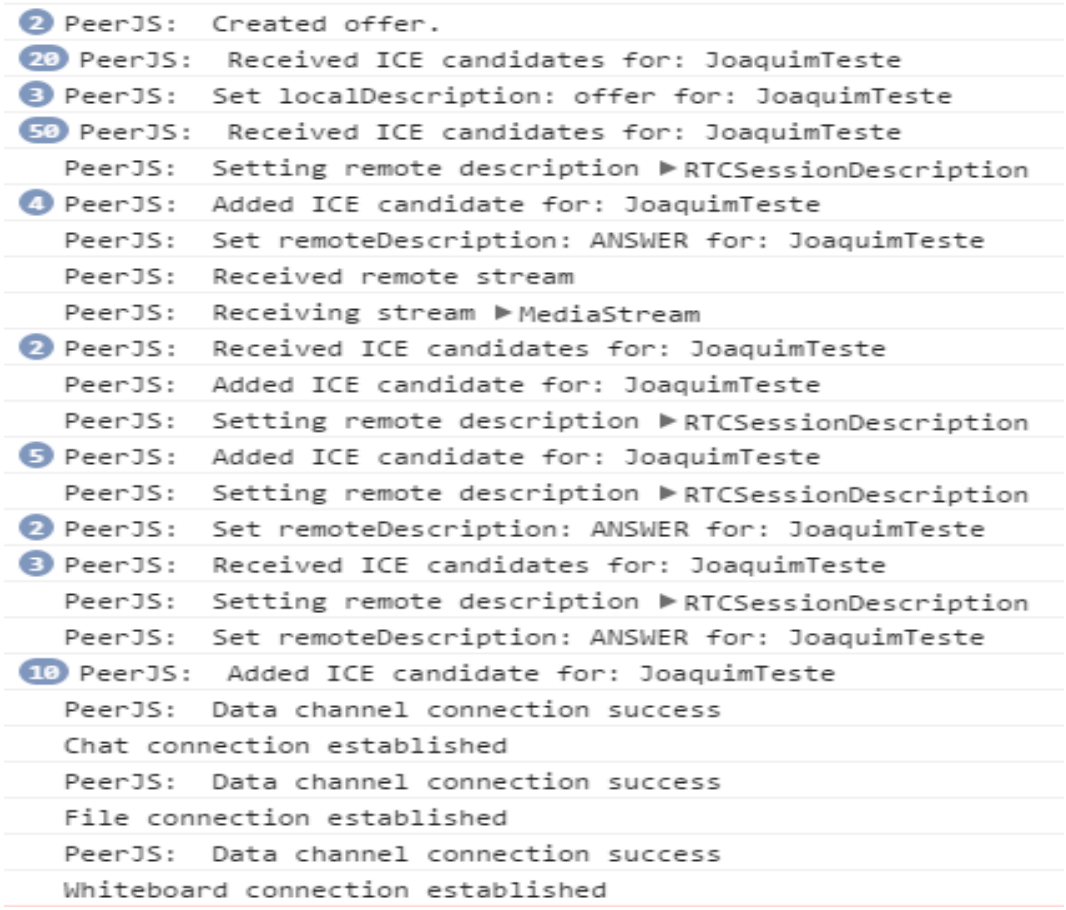

Figura 4.3: Eventos no estabelecimento da ligação

Figura 4.4: Mensagens SDP

O Código 3 representa a informação contida num pacote SDP a ser enviada para um dos pontos da comunicação. Neste caso, contém informações como o tipo de mensagem, a identificação do candidato, o endereço IP e porta, o protocolo de transporte, entre outras, sendo o destinatário o ponto identificado como "JoaquimTeste".

```
{"type":"CANDIDATE","payload":{"candidate":{"candidate":"candidate:744559770 1 udp
2122260223 192.168.227.163649 typ hostgeneration 0","sdpMid":"audio",
"sdpMLineIndex":},"type":"media","connectionId":"mc_tahzsrig4ysexw29"}
,"dst":"JoaquimTeste"}
```

Código 3: Conteúdo pacote *SDP*

4.2 Requisitos

Nesta secção são apresentados os requisitos principais da aplicação desenvolvida, quer do lado do cliente, quer do lado do servidor. Além dos requisitos do cliente e do servidor, ainda existe um requisito bastante importante sem o qual não é possível a comunicação. Trata-se da obrigatoriedade dos servidores ICE conseguirem descobrir o IP externo dos dois candidatos.

4.2.1 Cliente

O requisito mais importante no cliente é um navegador de internet que suporte o WebRTC. Além do navegador de internet, é conveniente uma *webcam* e um microfone, para que se consiga a captura de vídeo e áudio, respetivamente.
Os navegadores de internet, que na altura da elaboração deste trabalho, suportam nativamente o WebRTC, são o *Google Chrome* e o *Mozilla Firefox*. Existem *plugins* para o *Internet Explorer* e para o *Safari*, de forma a que possam igualmente suportar o WebRTC.

Para a utilização em dispositivos móveis existem Software Development Kit (SDK) que podem ser usados para criar aplicações de forma a usarem o WebRTC, além de alguns *browsers Android* suportarem o WebRTC. Nenhum *browser iOS* suporta neste momento o WebRTC.
Além dos *browsers*, existem também SDK para desenvolvimento de aplicações para *desktop* e sistemas *embedded*, não necessitando dessa forma de *browsers* para usar o WebRTC.

4.2.2 Servidor

Do lado do servidor é necessário configurar um servidor *web*, de forma a alojar a aplicação à qual os clientes se ligam. Paralelamente, é necessário um servidor de sinalização de forma a ser possível negociar os detalhes da ligação. Neste trabalho ambos os servidores executam o sistema operativo Linux Ubuntu 14.04 x64, têm instalado 512 Mega Bytes (MB) de Random Access Memory (RAM) e um disco Solid State Drive (SSD) de 20Giga Bytes (GB). Estão alojados em Londres num *datacenter* pertencente a uma empresa fornecedora de servidores virtuais em *cloud*. Para finalizar a arquitetura é necessário ainda servidores para executar os serviços ICE, STUN e TURN. Neste trabalho foram usados servidores ICE públicos como poderá ser visto mais abaixo, no Código 4.

Numa fase final do trabalho, foi instalado e configurado um servidor TURN na máquina onde o servidor de sinalização está a ser executado. Os resultados foram idênticos aos dos servidores ICE públicos a nível de tempo de resposta. Essa instalação serviu para demonstrar que seria possível executar um sistema WebRTC totalmente interno.
O servidor que foi instalado foi o *reTurn*, um servidor *open-source* que é uma implementação das normas das RFC do STUN RFC5389 e TURN RFC5766.

4.3 Aplicação

Nesta secção descreve-se a interface gráfica, *web-based*, disponibilizada pela solução desenvolvida no âmbito deste trabalho.
A aplicação está disponível através do endereço `https://joaquimbarranca.eu`. Este endereço foi criado no âmbito deste projeto e é de caráter temporário, podendo por isso ficar indisponível.

A Figura 4.5 ilustra a página inicial da aplicação onde é solicitado ao utilizador um identificador para a sua sessão. O identificador terá de ter no mínimo 10 caracteres e será tornado visível pelo servidor a todos os utilizadores ligados em cada momento.

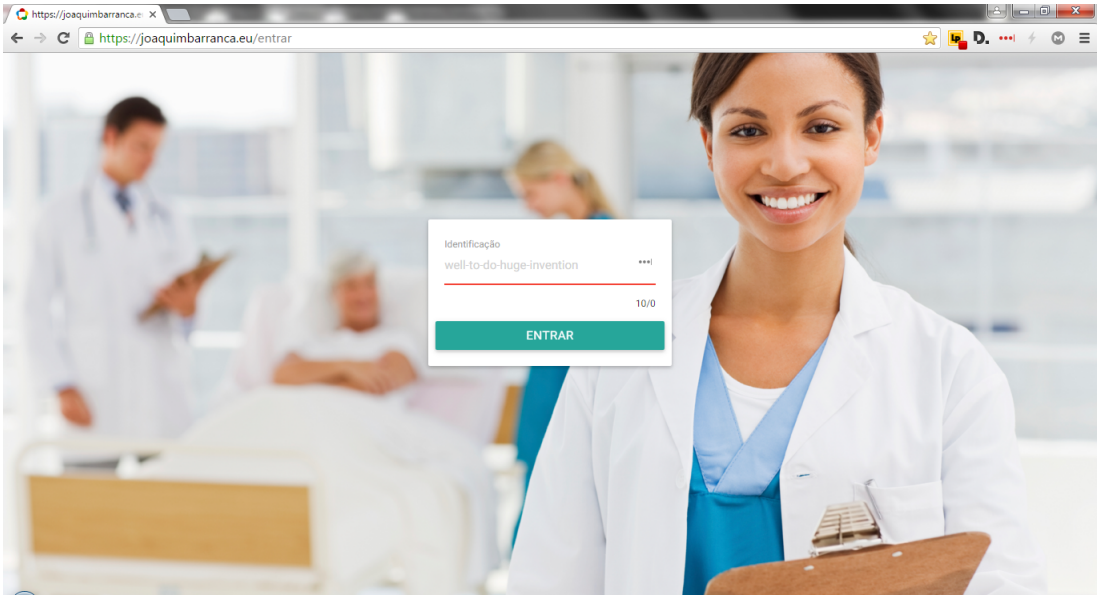

Figura 4.5: Página inicial da aplicação

A Figura 4.6 ilustra a página principal da aplicação. Esta página está dividida em duas colunas que se descrevem de seguida.
A coluna do lado esquerdo do ecrã mostra os utilizadores que estão disponíveis para iniciar uma ligação WebRTC. É utilizado o identificador introduzido na página anterior para identificar o utilizador. Do lado direito de cada identificador é apresentado um botão para efetuar a ligação WebRTC com esse utilizador. Ainda, nesta coluna, na parte inferior, é mostrado o *feed* do vídeo local.

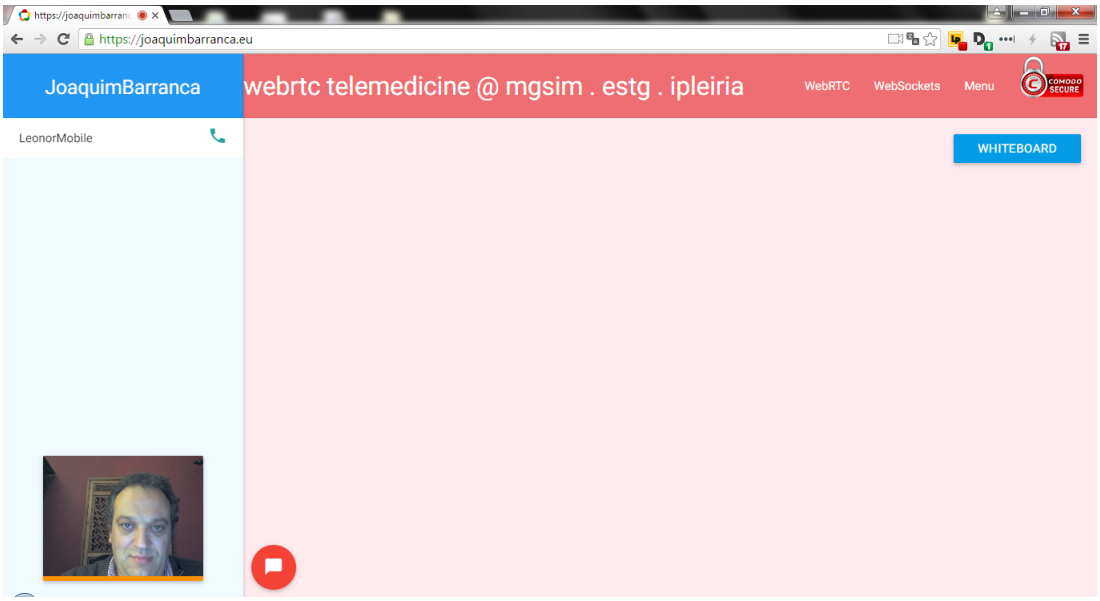

Figura 4.6: Página principal da aplicação

Após ter sido iniciada uma ligação WebRTC entre dois utilizadores, o vídeo remoto é apresentado na coluna do lado direito, bem como três botões, como pode ser visto na Figura 4.7.

O botão *whiteboard*, no canto superior direito, dá acesso à zona do *whiteboard* representado na Figura 4.8. O *whiteboard* é uma zona colaborativa, onde é possível partilhar o que se escreve ou desenha sobre ele, ou ainda enviar uma imagem, por exemplo um raio X e, sobre essa imagem, escrever notas ou ainda destacar determinada zona com hipotético problema.

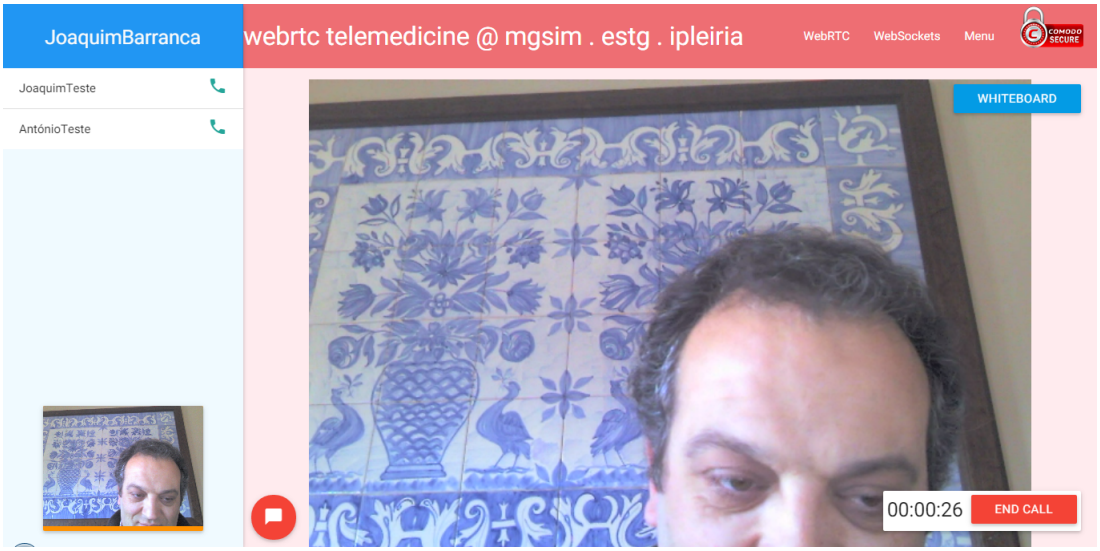

Figura 4.7: Zona de vídeo

Figura 4.8: Zona de whiteboard

Para se ter acesso à zona do *chat*, é necessário clicar no botão na zona inferior esquerda. Surge, então, a zona de *chat* onde para além do envio de mensagens, típico de uma aplicação de *chat*, também é possível o envio de ficheiros, tal como a Figura 4.9 exemplifica.

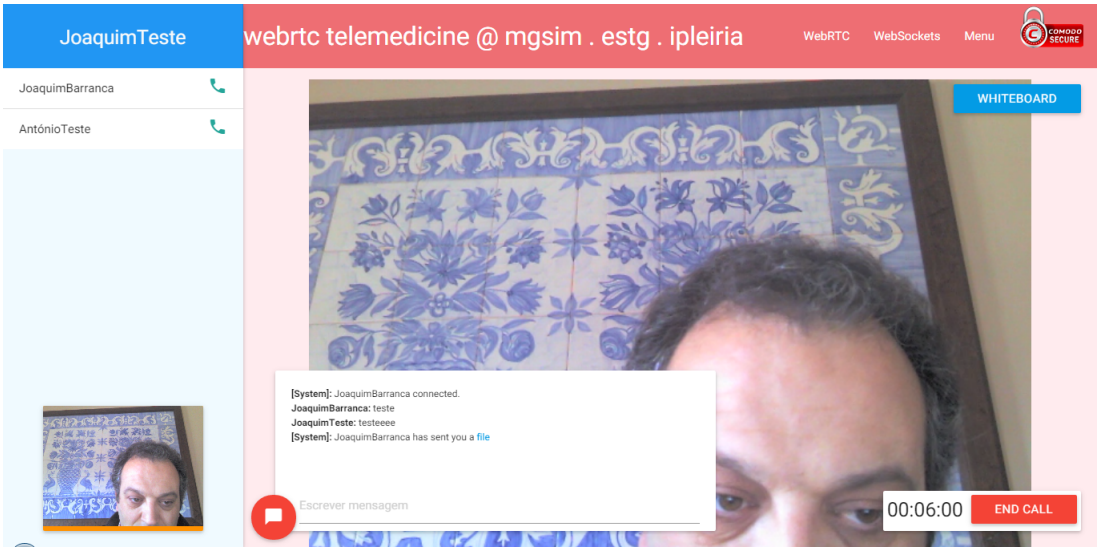

Figura 4.9: Zona de envio de ficheiros e chat

4.4 Principais módulos e APIs

Como já foi referido, a aplicação está dividida por vários módulos, nomeadamente um servidor de sessão que identifica os utilizadores que estão ligados, um *whiteboard*, uma área de *chat* e uma outra área onde é possível o envio de vídeo e áudio. Seguidamente mostram-se os principais algoritmos.

Para o serviço de sinalização optou-se pela API PeerJS uma vez que está bem documentada e é possível a implementação de um servidor próprio. O serviço de sinalização é muito importante para este sistema como já foi indicado anteriormente.

O servidor de sessão foi construído com recurso a um servidor `socket.io`. Esta escolha foi relativamente fácil, pois o `socket.io` é executado num servidor *node*, assim como a aplicação desenvolvida. Por fim para o *whiteboard*, foi usada a *API* `paper.js`. Esta *API* está vocacionada para trabalhar com imagens.

4.4.1 PeerJS

No Código 4 pode ver-se o construtor *peer*, que recebe 2 objetos por parâmetro, o identificador do peer e os *ICE servers*. O identificador peer é criado pelo utilizador quando insere a sua

identificação na página de entrada da aplicação. Os *ICE servers* são inseridos diretamente no código, estando alguns listados neste código.

```
peer = new Peer(myUID, {
      host: 'joaquimbarranca.com', port: 443, debug: 3, secure: true,
        config: {'iceServers':[
          {url:"turn:numb.viagenie.ca:3478",username:
                  "xxxxxxx@my.ipleiria.pt",credential:"xxxxxxx"},
          {url:'stun:stun01.sipphone.com'},
          {url:'stun:stun.l.google.com:19302'},
                 {url:'stun:stun.voxgratia.org'},
          {url:'stun:stun.xten.com'},
          {url:'turn:numb.viagenie.ca', credential: 'muazkh',
                  username: 'webrtc@live.com'},
          {url:'turn:192.158.29.39:3478?transport=udp',
                  credential: 'JZEOEt2V3Qb0y27GRntt2u2PAYA=',
                  username: '28224511:1379330808'},
          {url:'turn:192.158.29.39:3478?transport=tcp',
                  credential: 'JZEOEt2V3Qb0y27GRntt2u2PAYA=',
                  username: '28224511:1379330808'}
      ]}
    });
```

Código 4: Construtor peer

O Código 5 é, juntamente com o Código 12, dos mais importantes no WebRTC. O Código 5 cria um objeto do tipo *RTCPeerConnection*, adiciona-lhe a fonte de dados (*Stream*) e cria a oferta com a descrição das configurações locais para ser enviada ao outro nó.

```
setLocalDescription
var pc = new RTCPeerConnection();
pc.addStream(Stream);
pc.createOffer(function(desc) {
  pc.setLocalDescription(desc, function() {
  });
});
```

Código 5: Criação da oferta (*SDP*)

O Código 6 é usado para iniciar a comunicação com o nó remoto. Recebe o identificador do destinatário e chama o evento *peer.call* para iniciar a chamada. É também adicionado a *Stream* de vídeo local
O Código 7 termina a ligação de todos os componentes.

```
  var callUserID = function(userid){
    console.log("Calling", userid);
    call = peer.call(userid, window.localStream);
    createConnections2peer(userid);
    callAux(call);
  }
```

Código 6: Estabelecimento da ligação com o utilizador remoto

```
function closeCall(){
    console.log("Closing call");
    closeLocalStream();
    window.existingCall = null;
    $('#video').removeAttr("src");
    terminateCallTimer();
    call.close();
    fileConnection.close();
    messageConnection.close();
  }
```

Código 7: Encerramento da ligação

4.4.2 Sessão

O Código 8 mostra que, quando é detetado um novo utilizador, este é adicionado à lista. Verifica se o *userid* não é o próprio, para não o adicionar novamente à lista. É guardado o identificador do utilizador no atributo *data-uid* para que, ao ligar, se faça a chamada para o utilizador correto através da função *callUserID*, que foi vista anteriormente.

```
socket.on('user joined', function(userid) {
      if (userid != myUID) {$(".contacts .collection").append("<li
      class='collection-item' data-uid='" + userid + "'>" + userid
      + "<a href='#!' class= 'secondary-content'><i class=
      'material-icons'>phone</i></a></li>");
        $(".contacts .collection li a").last().click(function () {
          var userid = $(this).parent().attr("data-uid");
          callUserID(userid);
        });
      }
    });
```

Código 8: Login

Quando o utilizador sai de sessão, é verificado o seu *userid* sendo removido da lista, tal como se representa no Código 9.

```
socket.on('user left', function(userid) {
  var li = \$(".contacts .collection li[data-uid="+userid+"]")
  li.slideToggle(400, function(){
    li.remove();
  });
});
```

Código 9: Sai de sessão

4.4.3 Chat/Envio ficheiros

O Código 10 recebe o nome do remetente e a mensagem e adiciona ao *chat*.

```
var addNewMessageToChat = function(from, message){
  var chat = \$('.chat-box');
  chat.append("<div class='message'><p class=''><span class='from'>"
  +from+"</span>"+message+"</p></div>");
  chat.animate({"scrollTop": chat[0].scrollHeight}, "slow");
}
```

Código 10: Adiciona mensagem ao *chat*

A zona de *chat*, para além dessa função, serve também para a partilha de ficheiros.
O Código 11 mostra como foi realizado o processamento dos ficheiros que forem inseridos na zona de *chat*.
O ciclo existente no código permite receber mais do que um ficheiro ao mesmo tempo na zona de *chat*, para posterior envio.

```
function processDrop(e){
  console.log("file dropped");
  e = e.originalEvent;
  var files = e.target.files || e.dataTransfer.files;
  for (var i = 0, f; f = files[i]; i++) {
    addNewMessageToChat("[System]", 'You sent a file');
    fileConnection.send(f);
  }
}
```

Código 11: Envio de ficheiro no *chat*

4.4.4 Áudio/Vídeo

Um dos momentos chave do WebRTC é a captura do sinal áudio e vídeo. O Código 12 ilustra a função *initializeLocalStream* que inicia uma *stream* local que, posteriormente, irá ser

enviada ao *peer* de destino. Essa *stream* contém o áudio e vídeo capturado pela função *getUserMedia*. É adicionada essa *stream* à fonte do controlo local *myCam*, que é um elemento novo do HTML5 existente na página *web* principal. Como já foi referido, por questões de segurança, é necessário autorização do utilizador para aceder à câmara e microfone, que será controlada pelo navegador. O Código 13, idêntico ao Código 12, recebe a *stream* de dados

```
navigator.getUserMedia = navigator.getUserMedia ||
        navigator.webkitGetUserMedia ||
        navigator.mozGetUserMedia;
function initializeLocalStream(){
  navigator.getUserMedia({audio: true, video: true}, function(stream){
    $('#myCam').prop('src', URL.createObjectURL(stream));
    $(".myCam").slideDown();
    window.localStream = stream;
  }, function(error){ console.error(error); });
}
```

Código 12: Captura de som e de vídeo

enviada e adiciona-a à fonte de dados do controlo vídeo. Neste caso, representa o vídeo remoto, ao contrário do Código 12 que apresenta a *stream* do vídeo local.

```
 call.on('stream', function(stream){
    \$('#video').prop('src', URL.createObjectURL(stream));
  });
```

Código 13: Receção chamada

4.5 Testes e análise de resultados

Nesta secção são descritos os principais testes realizados com a solução desenvolvida. Foram efetuados testes à usabilidade e à robustez do protótipo, assim como testes de ligação em vários cenários de rede e com vários equipamentos cliente.
Nos testes não houve preocupação em testar o desempenho, a ocupação da largura de banda ou ainda questões relacionadas com segurança. Foram realizados testes às funcionalidades da aplicação em ambientes tão heterogéneos quanto possível, de forma a cobrir o máximo de realidades a nível de configurações de redes e *hardware* dos clientes.
Foram também efetuados testes em clientes com *hardware* obsoleto como, por exemplo processador lento, pouca memória e em redes de baixo débito. Foram ainda realizados testes em clientes com *hardware* de ultima geração em redes de alto débito, em clientes localizados no interior do país e no litoral. Foram também realizados testes em clientes em redes empresariais mais protegidas e clientes em redes residenciais, por norma menos protegidas.
Realizaram-se ainda testes em clientes com dispositivos de captura de imagem externos, como por exemplo um microscópio digital.
De seguida, apresentam-se os detalhes de alguns dos testes efetuados e seus resultados.

4.5.1 Teste 1

Este teste realizou-se entre uma unidade de saúde do interior do país, a Unidade de Cuidados de Saúde Personalizados (UCSP) de Castanheira de Pera e a sede do Agrupamento de Centros de Saúde (ACES) Pinhal Litoral em Leiria. Neste teste participaram um cliente no interior (um médico especialista de medicina geral e familiar) e outro cliente no litoral (o autor do trabalho), clientes com *hardware* díspares assim como em redes com débitos bastante diferentes, conforme detalhe visto na Figura 4.10 mas estando os dois clientes na Rede Informática da Saúde (RIS).

No entanto não foi possível a ligação dos clientes, uma vez que ambos falharam na obtenção do seu IP externo. Esta questão está relacionada com a configuração das redes, ou seja, nenhum dos servidores ICE conseguiu identificar o IP externo dos clientes.

	Cliente A	Cliente B
Tipo de rede:	Empresarial (Sincrona)	Empresarial (Sincrona)
Velocidade de rede:	100mbps	10mbps
Hardware do cliente:		
Processador:	Core i3	Pentium 4
Memória RAM:	4GB	1GB
Camara:	integrada	Web Cam
Som:	integrado	Sem som

Figura 4.10: Ambiente teste 1

4.5.2 Teste 2

A Figura 4.11 descreve o ambiente do teste realizado entre um cliente residencial e o Hospital de Santo André do Centro Hospitalar de Leiria.
Este teste obteve praticamente o mesmo resultado que o anterior, ou seja, os servidores ICE que estão definidos na aplicação foram ineficazes na obtenção do IP de somente um dos clientes. O outro cliente esteve em condições de iniciar a comunicação WebRTC.
A Figura 4.12 representa os candidatos da ligação WebRTC do cliente residencial. Tal como se destaca, apenas os últimos dois candidatos seriam viáveis neste teste, uma vez que todos os outros são IP da rede interna, que não são passíveis de ser acedidos da rede externa. Este tipo de IP é chamado de *server refexive address*, ou seja, é o IP externo do NAT fornecido por um servidor TURN ou STUN.

	Cliente A	Cliente B
Tipo de rede:	Residencial (Assincrona)	Empresarial (Sincrona)
Velocidade de rede:	30mbps	100mbps
Hardware do cliente:		
Processador:	Core i3	Core i7
Memória RAM:	4GB	8GB
Camara:	integrada	integrada
Som:	integrado	integrado

Figura 4.11: Ambiente teste 2

Time	Component Type	Foundation	Protocol	Address	Port	Priority
Not captured	1 host	744559770	udp	192.168.227.1	60062	126 | 32542 | 255
Not captured	1 host	4124027416	udp	192.168.138.1	60063	126 | 32286 | 255
Not captured	1 host	3515430769	udp	2001:8a0:f253:4901:5408:f1ab:d91b:7730	60064	126 | 32040 | 255
Not captured	1 host	3581307221	udp	2001:8a0:f253:4901:8569:6599:d6b9:b237	60065	126 | 31784 | 255
Not captured	1 host	2085243720	udp	192.168.1.70	60066	126 | 31518 | 255
Not captured	2 host	744559770	udp	192.168.227.1	60067	126 | 32542 | 254
Not captured	2 host	4124027416	udp	192.168.138.1	60068	126 | 32286 | 254
Not captured	2 host	3515430769	udp	2001:8a0:f253:4901:5408:f1ab:d91b:7730	60069	126 | 32040 | 254
Not captured	2 host	3581307221	udp	2001:8a0:f253:4901:8569:6599:d6b9:b237	60070	126 | 31784 | 254
Not captured	2 host	2085243720	udp	192.168.1.70	60071	126 | 31518 | 254
Not captured	1 host	1659037802	tcp	192.168.227.1	0	90 | 32542 | 255
Not captured	1 host	3142342376	tcp	192.168.138.1	0	90 | 32286 | 255
Not captured	1 host	2668284801	tcp	2001:8a0:f253:4901:5408:f1ab:d91b:7730	0	90 | 32040 | 255
Not captured	1 host	2616563109	tcp	2001:8a0:f253:4901:8569:6599:d6b9:b237	0	90 | 31784 | 255
Not captured	1 host	852080568	tcp	192.168.1.70	0	90 | 31518 | 255
Not captured	2 host	1659037802	tcp	192.168.227.1	0	90 | 32542 | 254
Not captured	2 host	3142342376	tcp	192.168.138.1	0	90 | 32286 | 254
Not captured	2 host	2668284801	tcp	2001:8a0:f253:4901:5408:f1ab:d91b:7730	0	90 | 32040 | 254
Not captured	2 host	2616563109	tcp	2001:8a0:f253:4901:8569:6599:d6b9:b237	0	90 | 31784 | 254
Not captured	2 host	852080568	tcp	192.168.1.70	0	90 | 31518 | 254
Not captured	1 srflx	2321167004	udp	2.81.192.150	64992	100 | 31518 | 255
Not captured	2 srflx	2321167004	udp	2.81.192.150	53085	100 | 31518 | 254

Figura 4.12: Candidatos ICE

A Figura 4.13 representa os candidatos da ligação WebRTC do cliente que estava localizado no Hospital de Santo André, ou seja na RIS, tal como o teste efetuado com a UCSP de Castanheira de Pera e a sede do ACES Pinhal Litoral em Leiria. Como pode ser observado, os candidatos da ligação deste cliente tem todos IP internos, não sendo possível assim concluir a ligação WebRTC.

Time	Component Type	Foundation	Protocol	Address	Port	Priority
Not captured	1 host	3660539835	udp	192.168.102.238	57680	126 | 32542 | 255
Not captured	2 host	3660539835	udp	192.168.102.238	57681	126 | 32542 | 254
Not captured	1 host	2494339915	tcp	192.168.102.238	0	90 | 32542 | 255
Not captured	2 host	2494339915	tcp	192.168.102.238	0	90 | 32542 | 254

Figura 4.13: Candidatos ICE

4.5.3 Teste 3

A Figura 4.14 ilustra o teste efetuado com dois clientes residenciais, com os detalhes presentes na Figura 4.15. Este teste foi realizado com um dispositivo de captura de ima-

gem externo de modo a ser testada essa funcionalidade. Foi usado um microscópio digital que foi ligado ao computador através de uma porta USB, ficando disponível na aplicação para ser usado como fonte do sinal vídeo a ser enviado ao cliente remoto. Este teste teve como objetivo demonstrar a possibilidade de ligar aparelhos de captura de imagem externos, exemplificando assim que seria possível ligar diferentes tipos de aparelhos de meios complementares de diagnóstico de uma forma simples e rápida.

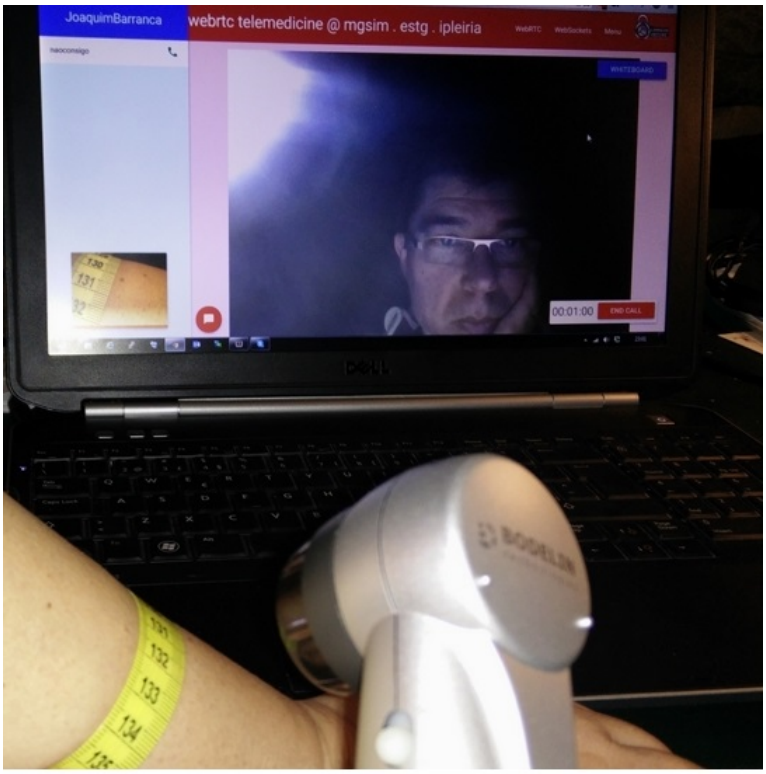

Figura 4.14: Teste 3

	Cliente A	Cliente B
Tipo de rede:	Residencial (Assincrona)	Residencial (Assincrona)
Velocidade de rede:	30mbps	100mbps
Hardware do cliente:		
Processador:	Core i3	Core i7
Memória RAM:	2GB	4GB
Camara:	Externa (microscópio)	Externa (Web Cam)
Som:	integrado	integrado

Figura 4.15: Ambiente teste 3

4.5.4 Análise de resultados

A implementação de uma aplicação pressupõe testes intensivos e abrangentes para que todas as funcionalidades sejam exaustivamente colocadas à prova de forma a que erros menos óbvios possam ser corrigidos. Os testes servem também para verificar e medir o desempenho dos vários componentes.

No que respeita à aplicação apresentada os teste efetuados concentraram-se na exploração dos componentes e sobretudo na ligação ponto-a-ponto dos clientes.
Alguns dos testes tiveram o resultado esperado, tanto a nível dos componentes como da ligação. Outros testes demonstraram que o estabelecimento da ligação ponto-a-ponto falhava. O facto dos servidores ICE testados serem ineficazes em algumas redes foi a principal conclusão a que se chegou. Este e outros testes serão referenciados para trabalho futuro que se apresenta no próximo capítulo.

4.6 Monitorização do WebRTC

O Google Chrome usa a funcionalidade *webrtc-internals*, acessível através do endereço `chrome://webrtc-internals/`, de modo a conseguir monitorizar diversos parâmetros no WebRTC.
Esta capacidade de geração de relatórios sobre o funcionamento do WebRTC é nativa ao navegador *Google Chrome*. Permite a recolha e análise de várias características da ligação, por exemplo o número de bits e pacotes enviados por segundo e os pacotes que não chegaram ao seu destino. Na Figura 4.16 podem observar-se alguns dos parâmetros disponíveis para monitorização, designadamente os dados recebidos pelo canal de áudio.
Na Figura 4.17 ilustra a análise efetuada à recolha de alguns dados enviados pelo canal de vídeo.

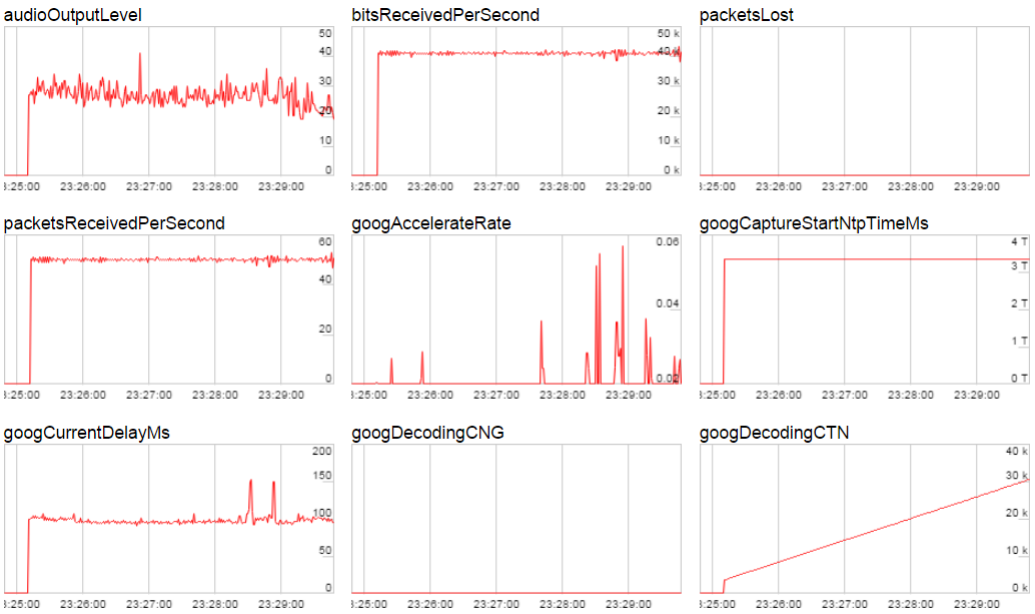

Figura 4.16: Webrtc Internals: dados recebidos através do canal de áudio

Figura 4.17: Webrtc Internals: dados enviados através do canal de vídeo

Além dos gráficos que dão uma perceção rápida dos dados a analisar, é possível verificar ainda a informação contida nos eventos do WebRTC. Pode-se verificar por exemplo a informação contida nos pacotes SDP e os candidatos ICE, que já foram vistos anteriormente, mas

do lado da consola.

Figura 4.18: Webrtc Internals: mensagens dos pacotes SDP

4.7 Conclusão

A implementação de qualquer aplicação para utilização em diferentes ambientes e com múltiplas características representa sempre um desafio.

A necessidade de estabilidade interna, de uma interface simples e útil são os principais requisitos deste tipo de sistemas, sendo que por norma é necessário iterar a arquitetura do sistema até conseguir uma solução estável que materialize todos os objetivos definidos.

O desenvolvimento desta aplicação passou também por diversas etapas de implementação. Numa primeira fase foi implementado um servidor web local e a aplicação desenvolvida continha apenas os requisitos mínimos face ao objetivo final. Apenas as funcionalidades de vídeo e áudio estavam implementadas. Nesta fase ainda era possível a implementação do WebRTC sem mecanismos de segurança, ou seja, o acesso podia ser realizado via HTTP e a sinalização podia também ser feita sem segurança via *websockets*. As funcionalidades foram sendo implementadas e surgiu a necessidade de alojar a aplicação num servidor web dedicado. Migrou-se então a aplicação para um servidor web dedicado mantendo o servidor de sinalização nos servidores do fornecedor da API e mantendo também a interface original.

A fase seguinte passou por investigar soluções para implementação dos restantes requisitos. Sensivelmente durante este período, o WebRTC sofreu uma evolução e deixou de permitir a comunicação insegura, forçando toda a comunicação apenas com segurança via HTTPS. Foi então necessário adaptar a aplicação para o novo paradigma. Nesta fase ainda com um certificado auto assinado apenas.

Os servidores da solução de sinalização elegida, não disponibilizavam o acesso seguro, assim foi necessário encontrar uma alternativa que fosse de encontro aos requisitos de segurança. Uma vez que a aplicação continha um grande esforço em termos de horas de implementação, foi optado por manter a mesma solução de sinalização e implementando segurança conforme imposição do WebRTC que até então era apenas uma sugestão.

Como a solução de sinalização permitia a implementação em servidores próprios para disponibilizar o acesso com segurança, foi implementado num servidor numa empresa de aluguer de espaço em *datacenters*. Após esta fase, iniciou-se a implementação dos restantes requisitos. No final de todos os requisitos implementados, faltava apenas desenhar uma interface mais simples e intuitiva. Essa necessidade foi também impulsionadora para alterar o servidor web e o seu alojamento. Foi então implementado um segundo servidor na mesma empresa que alojava o servidor de sinalização. Com esta implementação foi necessário criar dois certificados emitidos por uma entidade certificadora e com isso, foi necessário criar um domínio para cada um dos certificados. No final deste processo, foi ainda implementado um servidor TURN num dos servidores disponíveis para que todo o sistema WebRTC pudesse ser executado em servidores próprios.

Capítulo 5

Conclusões e trabalho futuro

O trabalho realizado no âmbito da presente dissertação teve como principais objetivos a análise da tecnologia WebRTC de forma a explorar as capacidades para o desenvolvimento de uma solução de telemedicina assente nesta tecnologia.

Houve necessidade de perceber quais as funcionalidades necessárias a uma aplicação de telemedicina. Nesse sentido foram ouvidos alguns médicos e a solução proposta implementa as funcionalidades consideradas mandatórias por esses profissionais de saúde para uma aplicação deste tipo, nomeadamente a troca de vídeo e voz, um *whiteboard* para discussão de resultados, um *chat* e transferência de ficheiros.

De seguida apresentam-se algumas conclusões ao trabalho desenvolvido. Em primeiro lugar, constata-se que os utilizadores devem ser preparados para usar um sistema de telemedicina, de forma a não ser um obstáculo à implementação de um sistema de telemedicina.

Quanto aos testes realizados, permitem concluir que o WebRTC tem características que se adequam, de uma forma natural, à implementação de um sistema de telemedicina e mais concretamente a sistemas de teleconsulta. O WebRTC mesmo sendo recente e com especificações ainda por definir, apresenta grandes potencialidades para disponibilizar, de forma gratuita e segura, um sistema de telemedicina.

Um destaque especial para o papel fundamental dos servidores ICE. Basta que a rede de um dos clientes esteja configurada de forma a impedir que o servidor STUN não consiga

capturar o seu IP externo, para que a comunicação WebRTC ponto-a-ponto seja inviabilizada, passando a comunicação WebRTC a ser feita através de um intermediário.

A configuração dos servidores ICE, presente neste trabalho, não permitiu testar a comunicação com um servidor TURN, ou seja, haver um servidor intermediário por onde a comunicação fosse facilitada quando não fosse possível a comunicação mais direta ponto-a-ponto.

Identificam-se as seguintes atividades como tópicos para trabalho futuro:

- Embora a solução apresentada tenha sido testada e esteja a funcionar, não deixa de ser um protótipo, o que significa que o seu aperfeiçoamento pode e deve ser realizado. Por exemplo, verificar o motivo da falha de comunicação entre o *browser* dos clientes e os servidores ICE. Também a ineficácia dos servidores TURN nessas mesmas redes deverá ser alvo de investigação.
- O WebRTC permite a ligação com outras tecnologias, nomeadamente o telefone e o VoIP. Uma forma de evoluir o protótipo poderá passar por implementar a ligação com uma ou mais destas tecnologias.
- Implementar a solução desenvolvida em ambiente hospitalar e aferir sobre a sua aceitação em ambiente de teleconsulta.
- Testar a utilização da aplicação com outro tipo de equipamento externo ligado via USB e aferir sobre a sua efetiva utilidade em contexto de teleconsulta.

Bibliografia

[Bashshur R., 1977] Bashshur R., L. J. (1977). Assessment of telemedicine: results of the initial experience. *Aviat. Space Environ. Med.*, 48:65–70.

[Commission, 2012] Commission, E. (2012). Putting patients in the driving seat a digital future for healthcare.

[Commission, 2014] Commission, E. (2014). Better survival chances for heart patients with telemedicine.

[DGS, 2012] DGS (2012). Plano nacional de saude 2012-2016 3.1. eixo estrategico - cidadania em saude. `http://pns.dgs.pt/pns-versao-completa/`.

[Dierks and Allen, 1999] Dierks, T. and Allen, C. (1999). The tls protocol version 1.0. RFC 2246, RFC Editor. `http://www.rfc-editor.org/rfc/rfc2246.txt`.

[Dierks and Rescorla, 2008] Dierks, T. and Rescorla, E. (2008). The transport layer security (tls) protocol version 1.2. RFC 5246, RFC Editor. `http://www.rfc-editor.org/rfc/rfc5246.txt`.

[Ferrer-Roca and Sosa-Iudicissa, 1998] Ferrer-Roca, O. and Sosa-Iudicissa, M. C. (1998). *Handbook of telemedicine*, volume 54. IOS press.

[Fette and Melnikov, 2011] Fette, I. and Melnikov, A. (2011). The websocket protocol. RFC 6455, RFC Editor. `http://www.rfc-editor.org/rfc/rfc6455.txt`.

[Fielding et al., 1999] Fielding, R. T., Gettys, J., Mogul, J. C., Nielsen, H. F., Masinter, L., Leach, P. J., and Berners-Lee, T. (1999). Hypertext transfer protocol – http/1.1. RFC 2616, RFC Editor. `http://www.rfc-editor.org/rfc/rfc2616.txt`.

[Freitas, 2005] Freitas, A. (2005). Telemedicina.

[Fuchs, 1979] Fuchs, M. (1979). Provider attitudes toward starpahc: a telemedicine project on the papago reservation. *Medical care*, 17(1):59–68.

[Grabowski et al., 2014] Grabowski, D. et al. (2014). Telemedicine reduces hospitalisations in nursing homes. *PharmacoEconomics & Outcomes News*, 696:7–15.

[Gravenstein et al., 1974] Gravenstein, J., BERZINA-MOETTUS, L., PAO, Y.-H., et al. (1974). Laser mediated telemedicine in anesthesia. *Anesthesia & Analgesia*, 53(4):605–609.

[Handley et al., 2006] Handley, M., Jacobson, V., and Perkins, C. (2006). Sdp: Session description protocol. RFC 4566, RFC Editor. `http://www.rfc-editor.org/rfc/rfc4566.txt`.

[Hjelm, 2005] Hjelm, N. (2005). Benefits and drawbacks of telemedicine. *Journal of telemedicine and telecare*, 11(2):60–70.

[Mendelson et al., 2014] Mendelson, M., Vivodtzev, I., Tamisier, R., Laplaud, D., Domingos, S. D., Baguet, J., Moreau, L., Koltes, C., Pepin, L., Chavez, G., et al. (2014). Continuous positive airway pressure (cpap) supported by telemedicine improves sleepiness and quality of life but not blood pressure in high cardiovascular risk obstructive sleep apnea (osa): A randomized, controlled trial. *Am J Respir Crit Care Med*, 189:A5626.

[North et al., 2014] North, F., Crane, S. J., Takahashi, P. Y., Ward, W. J., Tulledge-Scheitel, S. M., Ytterberg, K., Tangalos, E. G., and Stroebel, R. J. (2014). Telemedicine barriers associated with regional quality measures. *Telemedicine and e-Health*, 20(2):179–181.

[Organization-WHO, 2011] Organization-WHO, W. H. (2011). Atlas ehealth country profiles. *based on the findings of the second global survey on eHealth*.

[Peña-López et al., 2010] Peña-López, I. et al. (2010). Improving health sector efficiency: The role of information and communication technologies.

[Qiang and Marras, 2015] Qiang, J. K. and Marras, C. (2015). Telemedicine in parkinson's disease: A patient perspective at a tertiary care centre. *Parkinsonism & Related Disorders*.

[Salvatore Loreto, 2014] Salvatore Loreto, S. P. R. (2014). *Real-Time Communication with WebRTC: Peer-to-Peer in the Browser*. O'Reilly Media, Inc.

[Srisuresh and Egevang, 2001] Srisuresh, P. and Egevang, K. (2001). Traditional ip network address translator (traditional nat). RFC 3022, RFC Editor.

[Thomas and Capistrant, 2014] Thomas, L. and Capistrant, G. (2014). 50 state telemedicine gaps analysis coverage & reimbursement. pages 1–100.

[William et al., 2014] William, M., Chen, E., Krukas, M., Tayama, D., Ernst, F., and Wagner, J. (2014). Comparison of thrombolytic treatment for acute ischemic stroke pre-and post-telemedicine implementation in the spoke hospital setting (pl1. 002). *Neurology*, 82(10 Supplement):PL1–002.

[Zundel, 1996] Zundel, K. M. (1996). Telemedicine: history, applications, and impact on librarianship. *Bulletin of the Medical Library Association*, 84:71–79.