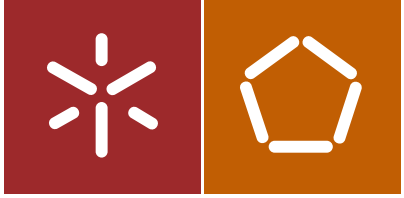
Universidade do Minho
Escola de Engenharia

João Rafael Pinheiro Vilela

QoS no Acesso à Internet em Redes Celulares
QoS on Internet Access Through Cellular Networks

Outubro de 2014

QoS no Acesso à Internet em Redes Celulares
QoS on Internet Access Through Cellular Networks
João Rafael Pinheiro Vilela
UMinho | 2014



Universidade do Minho
Escola de Engenharia

João Rafael Pinheiro Vilela

QoS no Acesso à Internet em Redes Celulares
QoS on Internet Access Through Cellular Networks

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia de Comunicações

Trabalho efetuado sob a orientação do
Professor Doutor Adriano Moreira
Professor Doutor Nuno Vasco Lopes

Outubro de 2014

Anexo 3

DECLARAÇÃO

Nome

João Rafael Pinheiro Vilela

Endereço electrónico: a58665@alunos.uminho.pt Telefone: 914390051

Número do Bilhete de Identidade: 13917135

Título dissertação □/tese □

QoS no Acesso à Internet em Redes Celulares

QoS on Internet Access Through Cellular Networks

Orientadores:

Professor Doutor Adriano Moreira

Professor Doutor Nuno Vaco Lopes

Ano de conclusão: 2014

Ciclo de Estudos Integrados Conducentes ao Grau de Mestre em Engenharia de Comunicações

Nos exemplares das teses de doutoramento ou de mestrado ou de outros trabalhos entregues para prestação de provas públicas nas universidades ou outros estabelecimentos de ensino, e dos quais é obrigatoriamente enviado um exemplar para depósito legal na Biblioteca Nacional e, pelo menos outro para a biblioteca da universidade respectiva, deve constar uma das seguintes declarações:

1. É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

2. É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA TESE/TRABALHO (indicar, caso tal seja necessário, nº máximo de páginas, ilustrações, gráficos, etc.), APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

3. DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA TESE/TRABALHO

Universidade do Minho, ___/___/_____

Assinatura: _____

“Dreams are the touchstones of our character”

Henry David Thoreau

Agradecimentos

Finda esta etapa da minha vida que dá por concluído o meu percurso académico, quero agradecer e reconhecer o apoio de todos aqueles que estiveram ao meu lado, que de alguma forma, contribuíram para a realização deste projeto. Esta foi uma experiência memorável com um valor inestimável para a minha vida, tanto a nível pessoal como a nível profissional.

Gostaria de agradecer ao Professor Doutor Adriano Moreira e ao Professor Doutor Nuno Vasco Lopes pelo valioso apoio durante este projeto, bem como a paciência, o profissionalismo, a dedicação e a disponibilidade. Foi um privilégio e estou bastante grato por ter tido a oportunidade de trabalhar sobre a orientação de ambos.

A um nível mais pessoal, agradeço ao meu pai Luís, à minha mãe Manuela, ao meu irmão André e à minha Avó Fernanda. Estas são as pessoas em quem a minha vida assenta e que sempre estiveram e estarão ao meu lado nas diversas fases do meu percurso e cujo o apoio constante me impele a ir cada vez mais longe. Ao meu falecido Avô António deixo um agradecimento especial pela educação e pelos valores que me incutiu.

Deixo um obrigado a todos os meus amigos e colegas, destacando três pessoas com que partilhei muitos momentos e que foram excelentes companheiros ao longo do meu percurso académico, António Maio, Joel Rodrigues e Paulo Alves. Obrigado pelo vosso apoio e sobretudo pela amizade.

Resumo

O principal objetivo deste projeto é a avaliação da Qualidade de Serviço de conexões à Internet usando redes móveis celulares. Para tal recorre-se a um conjunto de ferramentas.

Atualmente as redes móveis estão em larga expansão e o acesso a conteúdo através de dispositivos móveis é cada vez mais uma parte essencial na realização de tarefas diárias. Avaliar a Qualidade com que um determinado serviço é prestado torna-se uma necessidade.

Esta dissertação centra-se no desenvolvimento de uma aplicação para dispositivos móveis, que permite a realização de testes de QoS através de quatro métricas: latência, perda de pacotes, débito em download e débito em upload. Os dados são enviados para um servidor, onde são processados e armazenados. Além da aplicação móvel é desenvolvido um *Web* Site que permite visualizar os dados recolhidos e efetuar um conjunto de análises estatísticas.

Este é um serviço de largo espectro focado nas redes móveis, mas com ferramentas que dão suporte e estendem as capacidades da aplicação móvel. O sistema no seu todo é o que o diferencia das ferramentas que existem no mercado. É um sistema cujo valor real se encontra sobre os dados recolhidos. Para tal é necessário atrair o máximo de utilizadores, pelo que um ponto fundamental é criar uma aplicação visualmente atrativa e simples, com funcionalidades bem definidas.

Adicionalmente, é imperativo que todo o sistema, e a aplicação móvel em particular cumpra um conjunto de requisitos restritos, ao nível do consumo de bateria, memória e processamento e sobretudo de tráfego no instante de realização dos testes. Tudo isto é conseguido através de mecanismos e algoritmos de recolha, processamento e tratamento de dados eficientes.

Em suma trata-se de um sistema colaborativo de aferição de QoS, composto pela aplicação móvel que efetua os testes e todo um serviço de armazenamento e consulta dos dados pelo utilizadores.

Abstract

The main goal of this project is to evaluate the Quality of Service of Internet connections through mobile networks. For that a set of tools has been used.

Nowadays mobile networks are in expansion and the access to content through mobile devices is starting to get essential to our day-to-day tasks. Therefore, evaluating the quality of a service in cellular networks is crucial now.

This project focused on the development of an application for mobile devices, which allows users to perform QoS tests using four metrics, such as, latency, packet loss, *download* throughput and *download* throughput. The application sends the data to one server for being processed and stored. Besides the mobile application, the system also has integrated a *Web Site*, which allows users to consult and analyze their personal data by using a set of statistical analysis views.

This is a service for measuring QoS focused on cellular networks, with a set of tools, which give support and extends the application capabilities. The overall system with all its components is what differentiates it from other tools on the market. The main value of the project relies on the importance that the collected data has to measure the quality of the service provided by different ISP's on the market. Therefore, it is very important to have an attractive, simple and well-defined application.

Furthermore, it is also imperative that the overall system, specifically the mobile application, meets a strict set of requirements, such as, battery, memory processing and light-weighted QoS tests. This is possible using efficient mechanisms and algorithms to collect and process data.

Índice de conteúdos

Lista de figuras

Lista de tabelas

Lista de abreviaturas

3GPP	3rd Generation Partnership Project
AMC	Adaptive Modulation and Coding
ANACOM	Autoridade Nacional das Comunicações
API	Application Programming Interface
AuC	Authentication Center
BER	Bit Error Rate
BSC	Base Station Controller
BTS	Base Transceiver Station
CDMA	Code Division Multiple Access
CN	Core Network
CSD	Circuit Switched Domain
CSMA	Carrier Sense Multiple Access
DECT	Digital Enhanced Cordless Telecommunication
EiR	Equipment Identify Register
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
FCCN	Fundação para a Computação Científica Nacional
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FTP	File Transfer Protocol
GERAN	GSM EDGE Radio Access Network
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HARQ	Hybrid ARQ
HLR	Home Location Register
HSPA	High Speed Packet Access
HTTP	Hypertext Transfer Protocol

ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IEPM	Internet End-to-End Performance Measurement
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU	International Telecommunications Union
LAN	Local Area Network
MT	Mobile Terminal
NRA	National Regulatory Agency
Ofcom	Office of Communications
OMC	Operation and Maintenance Center
PDN	Public Data Network
PER	Packet Error Rate
PingER	Ping End-to-End Reporting
PLR	Packet Loss Rate
PSD	Packet Switched Domain
PSTN	Public Switched Telephone Network
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RDIS	Rede Digital com Integração de Serviços
RNC	Radio Network Controller
RTT	Round Trip Time
SFTP	Secure File Transfer Protocol
SLAC	Stanford Linear Accelerator Center
TDD	Time Division Duplex
TE	Terminal Equipment
UMTS	Universal Mobile Telecommunication System
UTRAN	UMTS Terrestrial Radio Access Network
VLR	Visitor Location Register

W-CDMA Wideband Code Division Multiple Access

1. Introdução

1.1 Enquadramento e Motivação

Atualmente, no domínio das redes móveis assiste-se a uma convergência da tendência em explorar cada vez mais o conceito de mobilidade, tanto do utilizador como do dispositivo. Dado o papel relevante que a Internet e a suas aplicações têm vindo a representar na sociedade moderna, conjugar as necessidades dos utilizadores ao mesmo tempo que se explora o conceito de mobilidade requer um maior controlo sobre a qualidade dos serviços prestados, uma vez que este paradigma de comunicação e ubiquidade apresenta novos desafios e exigências do ponto de vista tecnológico e de utilização.

Quando se aborda o conceito de QoS (*Quality of Service)* é necessário ter em consideração as suas duas variantes. Por um lado é necessário compreender a perceção que um utilizador tem do serviço, referido como QoE (*Quality of Experience*), por outro lado existem métricas e parâmetros objetivos que permitem avaliar e quantificar a qualidade de um serviço de comunicações [1]. Neste contexto, surgem serviços com características específicas, apresentando requisitos distintos. A tendência, nas redes móveis, é integrar vários serviços de alta qualidade na mesma rede tais como voz, dados e vídeo.

A implementação de mecanismos que permitem verificar os parâmetros de avaliação de QoS visam garantir que os serviços são prestados de forma consistente e previsível ao longo do tempo. Com este objetivo em mente assiste-se a uma crescente necessidade de regulamentação sobretudo por entidades governamentais, denominadas de NRA's (*National Regulatory Agencies*) [2]. Estas, levam a cabo estudos e testes periódicos para avaliar a QoS dos ISP's (*Internet Service Providers*) com vista a uma melhor compreensão dos serviços disponibilizados evidenciando se os requisitos em termos de qualidade estão a ser ou não devidamente cumpridos. Estas agências, nas quais se insere, no território Português, a ANACOM (Autoridade Nacional das Comunicações), têm um papel fundamental na regulamentação dos sistemas de comunicações em todas as suas vertentes, garantindo que os serviços

contratados correspondem na prática ao que foi contratualizado e é anunciado pelos operadores, tornando assim o mercado mais competitivo.

Na ótica do utilizador, devido ao fácil acesso a informação tecnológica, o custo dos serviços deixa de ser, por si só, um fator decisivo [2], assistindo-se à introdução de novos fatores, dos quais se destaca a QoS, fazendo com que os utilizadores assumam uma posição mais crítica sobre os serviços que compram. Neste ponto introduz-se o conceito de QoE, uma vez que os serviços são desenhados e desenvolvidos para o público, em última análise a sua opinião sobre a experiência de utilização dos mesmos é determinante para atestar a qualidade com que um determinado serviço de comunicações é prestado. Esta nova realidade vem aumentar a pressão sobre os operadores para garantirem qualidade nos serviços que prestam, forçando-os a adoptar uma postura mais competitiva no desenvolvimento de novas ofertas comerciais, novos produtos, com mais qualidade, com custos mais reduzidos, por forma a destacarem-se no mercado face aos seus concorrentes. Apesar desta tendência, ainda há alguma resistência por parte dos operadores em efetuar e divulgar ao público testes sobre a QoS dos serviços que prestam, fazendo com que o papel das NRA's seja essencial.

Dado o crescente relevo que a QoS tem vindo a representar nos serviços de comunicações tornou-se necessário definir de forma concreta os requisitos e a forma como os fluxos de tráfego são tratados no domínio de um sistema de comunicações [3]. As diferentes tecnologias de redes móveis definem mecanismos concretos para providenciar QoS, tanto na rede de acesso rádio como no núcleo. Por exemplo, o UMTS define classes de QoS distintas para os serviços de voz, dados, interativos e de *background*, uma vez que estes apresentam características distintas e prioridades diferentes dentro do sistema de comunicações [2]. Com o crescente desenvolvimento das aplicações de dados sobre a Internet, cresce a necessidade de implementação de modelos e arquiteturas que permitam melhorar o desempenho dos sistemas implementados providenciando melhor QoS. Os dispositivos e as redes móveis estão cada vez a ganhar mais popularidade [4].

De forma geral, é possível reconhecer a crescente importância da QoS no domínio dos sistemas de comunicações, quer a nível tecnológico, através do desenvolvimento e implementação de novos modelos e esquemas, quer seja do ponto de vista

regulamentar, havendo cada vez mais a preocupação com a elaboração de estudos para aferir a QoS dos ISP's. A experiência de utilização (QoE) ganha um papel cada vez mais relevante na avaliação de um serviço de telecomunicações, uma vez que os utilizadores são os principais consumidores das ofertas comerciais feitas pelos operadores. O papel das conexões em banda larga assume um lugar cada vez mais relevante e determinante também no domínio social, apresentando-se já como um indicador do desenvolvimento socioeconómico de uma determinada região [5]. A QoS é uma das vertentes vitais no ramo da engenharia de comunicações. O seu estudo, em toda a sua plenitude, apresenta-se como um desafio fundamental e necessário para um progresso sustentado e significativo na evolução das redes de telecomunicações do futuro.

1.2 Objetivos

Esta dissertação tem como objetivo contribuir para uma melhor compreensão e perceção do conceito de QoS no domínio das redes celulares terrestres, fundamentalmente ao nível do acesso à Internet. Explora o conceito em si e detalha as vertentes que o constituem focando-se sobretudo na avaliação dos parâmetros que permitem quantificar e classificar a qualidade de um determinado serviço de comunicações. Neste processo é também considerado o conceito de QoE e qual o seu relevo na avaliação da qualidade de um serviço de comunicações. Como ponto fundamental para percepcionar a QoS será desenvolvido um *software* para *smartphones* que terá como objetivo a recolha de dados da rede, por forma a que seja possível armazená-los e agregá-los num serviço *web* para uma posterior avaliação e estudo dos mesmos. Os resultados dos dados recolhidos estarão disponíveis num *web site* para consulta quer por parte dos utilizadores do serviço, onde podem analisar exclusivamente os seus dados, quer por parte do público em geral, que têm acesso a um conjunto de dados a avaliações de domínio público. Esta componente tecnológica implica a participação de utilizadores de serviços de Internet móvel, apresentando-se, assim, como uma nova forma de avaliar a qualidade de um serviço que se distancia dos métodos de avaliação que as NRA's e operadoras atualmente apresentam, desenvolvendo-se, assim, um novo paradigma para aferição de QoS.

Os principais objetivos deste trabalho estão estruturados da seguinte forma:

- Desenhar uma solução que permite aferir a QoS associada aos serviços de acesso à Internet disponibilizados através de redes móveis celulares;
- Envolver os utilizadores desses serviços no próprio processo de aferição;
- Tornar possível que a aferição de QoS seja efectuada de forma contínua e distribuída no espaço.

1.3 Abordagem

Para que se possa alcançar os objetivos propostos e obter os resultados esperados, a especificação das tarefas a realizar assume particular destaque. As tarefas necessárias à realização do trabalho são:

1. Recolha de Bibliografia e estudo do Estado-da-Arte associado com o objetivo de identificar as principais características do conceito de QoS. Com esta tarefa pretende-se desenvolver o conhecimento sobre o tema, explorando a tecnologia UMTS, aprofundar a compreensão sobre como são feitas atualmente avaliações e estudos de QoS, ao mesmo tempo que são analisadas as métricas e as metodologias usadas nos testes. É procurado, também, compreender o impacto da avaliação da qualidade dos serviços de comunicações, tanto ao nível dos utilizadores, como ao nível dos ISP's ou NRA's, bem como perceber o papel de cada um destes intervenientes na avaliação de QoS;
2. Definir os requisitos e modelo do sistema a desenvolver, tendo em conta todas as variáveis que se poderão apresentar como obstáculos, procurando esclarecer as possíveis abordagens de implementação e como é que estas permitirão alcançar os objetivos pretendidos, tentando optar por aquela que oferece mais garantias de sucesso, ao mesmo tempo que permite construir um sistema eficiente que vá de encontro àquilo se propõe como tema;
3. Desenhar o sistema a implementar. Após definidos os principais requisitos e a abordagem a seguir, é necessário construir um plano de como será desenvolvido o sistema e as suas componentes em concreto, explorando o

tipo de tecnologias e serviços que serão necessários. O desenho será o guia para a forma como o sistema irá ser produzido e apresentando no final;

4. Implementação do sistema. Será uma das tarefas mais importantes e uma das que mais se destaca no âmbito desta dissertação. Após ser definido como serão desenvolvidas as principais componentes do sistema é essencial implementar o mesmo seguindo o desenho inicial, recorrendo a diversas tecnologias e mecanismos, de forma organizada e estruturada, por forma a concluir o sistema dentro dos prazos previstos. Esta tarefa inclui, inerentemente a este tipo de sistemas, testes de desenvolvimento que permitem verificar as capacidades do mesmo, por forma a garantir que são obtidos os resultados esperados;
5. Experimentação no terreno. Após ser concluído o sistema e testado em ambiente de desenvolvimento é imprescindível efetuar testes experimentais no terreno, recorrendo a sujeitos de teste, que terão uma participação extremamente importante no campo de ação do projeto. Após a realização destes testes, poderão ser efetuados correções menores ao sistema com vista a aperfeiçoar todas as suas vertentes;
6. Estudo e avaliação dos resultados obtidos. Na parte final deste projeto é necessário efetuar uma introspeção com o intuito de se poder avaliar o que foi alcançado durante o desenvolvimento de todo o projeto e de como é que este se aproxima dos resultados previstos na fase inicial;
7. Redação da dissertação. Esta é a componente formal, onde se expõe de forma concreta e concisa todas as fases de trabalho e de como foi realizado o projeto. Permite descrever detalhadamente todos os processos desenvolvidos, desde a fase conceptual até à fase experimental, exibindo os resultados obtidos. Em suma, é um documento que agrega todo o trabalho realizado no domínio da dissertação.

1.4 Estrutura do Documento de Dissertação

No capítulo 1 é feita uma breve introdução ao tema de Dissertação. É explicado o enquadramento ao nível dos conceitos que se visam explorar e estudar, descrevendo as motivações para a realização deste trabalho. Neste capítulo são apresentados os principais objetivos a alcançar e a abordagem ao projeto. O capítulo 1 termina com a descrição da estrutura adoptada para o documento desta Dissertação.

O capítulo 2 visa apresentar o estado da arte relativamente aos conceitos fundamentais sobre os quais o tema assenta, dos quais se destaca: QoS e QoE. É também destacado algum trabalho relacionado nesta área, mais concretamente projetos desenvolvidos cujos objetivos se enquadram nos princípios fundamentais e no âmbito deste trabalho. Outro dos componentes que se insere neste capítulo é a análise à tecnologia 3G UMTS e como é que se providencia QoS no seio da mesma. Neste capítulo apresenta-se ainda uma visão sobre o papel das NRA's, como a ANACOM, ou de empresas privadas como a SamKnows, que efetuam testes e avaliam os serviços de comunicações, tentando perceber como são importantes esses estudos e que impacto têm nos utilizadores, nos ISP's e no próprio mercado.

No terceiro capítulo deste documento são apresentados os principais pressupostos do sistema. O processo de os identificar inicia-se com a definição das métricas que serão utilizadas para avaliar a QoS. Posteriormente é proposta uma arquitetura de alto nível que agrega todos os componentes que se identificam como necessários à elaboração do projeto.

O quarto capítulo descreve os requisitos ao nível do sistema. São definidos os requisitos funcionais e não funcionais com maior destaque para a componente da aplicação móvel a desenvolver, apresentando um pequeno estudo sobre o sistema operativo no sector móvel mais adequado.

O capítulo cinco contém todo o desenho do sistema e como foi implementado. É exposta a arquitetura geral e posteriormente são analisados em detalhe cada um dos elementos e como se relacionam entre si, abordando as tecnologias, a estrutura e a organização interna de cada um dos mesmos.

O capítulo seis tem como objetivo apresentar os resultados obtidos através da aplicação e como foi recebida pelos utilizadores, bem como as estratégias de divulgação usadas.

O sétimo capítulo apresenta as conclusões bem como o trabalho futuro relativo a este projeto.

2. Aferição da Qualidade de Serviço

2.1 Introdução

Atualmente existe um grande esforço para a integração de diferentes serviços na mesma rede, tais como voz, dados e vídeo. Sendo serviços diferentes, com diferentes características e objetivos, apresentam requisitos distintos em relação à qualidade. A implementação de mecanismos que permitem verificar os parâmetros de avaliação da qualidade de serviço visam garantir que os mesmos possam ser prestados de forma consistente e previsível, assegurando que utilizador obtém uma percepção positiva do serviço.

O *International Telecommunication Union* (ITU) define Qualidade de Serviço como: "*O total de características de um serviço de telecomunicações que afetam a sua capacidade de satisfazer as necessidades implícitas e explícitas do utilizador do serviço*" [1]. É necessário compreender a percepção que um utilizador tem do serviço, sendo esta uma visão subjetiva deste conceito. Contudo, existem parâmetros objetivos que permitem avaliar e quantificar a qualidade de um serviço de telecomunicações.

Na atual era da indústria das telecomunicações o preço deixa de ser o fator decisivo na compra de um serviço. A QoS assume cada vez mais um papel importante. Contudo, avaliações sobre a qualidade de serviço são ainda um pouco escassas [2]. Existem cada vez mais NRA's a criar estudos sobre a QoS com vista a uma maior compreensão dos serviços disponibilizados pelos operadores e as suas características, evidenciando se os requisitos estão a ser ou não cumpridos. Estas também estabelecem parâmetros que os operadores e as respetivas ofertas comerciais devem corresponder. Agências como a ANACOM assumem um papel importante na obtenção de dados para avaliação da métricas de QoS. Estes dados são publicados, mas tendem a passar despercebidos ao utilizador comum, que geralmente conhece a "realidade" facultada pelos operadores que apenas apresentam dados que não desacreditam os ofertas comerciais que disponibilizam. Existe falta de informação e

divulgação, não permitindo uma grande penetração do trabalho desenvolvido pela ANACOM nos utilizadores comuns. Contudo existem testes e avaliações de NRA's que possibilitam aos utilizadores contratar um serviço sabendo todas as suas características, tanto teóricas como práticas, ao nível do desempenho e em toda a sua plenitude. Com as alterações e evoluções do mercado o papel da QoS assume particular destaque, sendo que os utilizadores estão cada vez mais conscientes e mais informados sobre os serviços que têm à sua disposição e deixam de ser "obrigados" a contratar um serviço com base apenas na informação que o prestador dos mesmos disponibiliza. Todavia, os operadores parecem deixar escapar o elevado potencial das avaliações de QoS e as devidas comparações com os seus concorrentes de mercado [2].

2.2 Conceitos Básicos

2.2.1 Parâmetros de Qualidade de Serviço

A qualidade de serviço é afetada por vários fatores que definem as características de um sistema de telecomunicações. No domínio das redes móveis existem parâmetros específicos que são usados para determinar a QoS. Contudo, dada a integração dos serviços de dados móveis com a Internet, em toda a arquitetura do sistema móvel *end-to-end*, é necessário ter em conta todos os parâmetros de QoS. Existem três grandes domínios onde se inserem estes parâmetros: Atraso, Largura de Banda e Fiabilidade.

2.2.1.1 Atraso

Latência ou atraso, designa o tempo gasto por um pacote de informação para percorrer o caminho desde a sua origem até ao seu destino. Quanto maior for a latência, maior será a quantidade de dados em circulação na rede num determinado instante de tempo. Os parâmetros que se inserem neste domínio reflectem alguns fatores inerentes ao sistema de comunicações, seja ao nível do tipo de equipamentos de rede e capacidade de processamento dos mesmos, quer ao nível da implementação

da própria rede, mais concretamente o tipo de topologias usadas ou protocolos de comunicação implementados, ou ainda do tipo de tecnologias físicas para o transporte dos dados. [5]:

- Atraso de Transmissão: Designa o tempo que um pacote de dados necessita para ser injetado na rede, em função do seu tamanho, em bits, e do débito binário da ligação.
- Atraso de Propagação: Tempo que um pacote de dados necessita para percorrer uma determinada distância no meio físico de transmissão.
- Atraso de Processamento: Designa o tempo que um pacote de dados necessita para ser processado.
- Atraso nas Filas de Espera: Tempo de atraso que se deve à existência de filas de espera, necessárias para efetuar políticas de escalonamento.
- Atraso no acesso ao meio: Designa o tempo em que um pacote de dados fica em espera para que possa ser injetado no meio do sistema de comunicação.

O *jitter* designa a variação do atraso na rede de comunicação sofrido pelos pacotes de dados. Quando se verificam valores elevados de *jitter* o número de *timeouts* nos protocolos de transporte podem aumentar, levando a ineficiência no transporte de dados, originando uma recepção não regular dos pacotes. Uma das formas de atenuar a variação do atraso é recorrer à implementação de *buffers* no receptor e nós intermédios da rede, que armazenam os dados e os encaminham à mesma cadência. Para aplicações que operam em tempo real este parâmetro apresenta um peso elevado. Os pacotes que ultrapassam o tempo máximo de atraso são descartados [5].

2.2.1.2 Largura de Banda

No domínio da largura de banda existem dois conceitos a considerar: *goodput* e *throughput* [5].

O *throughput* designa o débito bruto da rede, ou seja, considera o tamanho dos pacotes incluindo cabeçalhos associados aos vários protocolos (*overhead*).

O *goodput* designa o débito útil que é efetivamente fornecido às aplicações, ou seja, só os dados transferidos.

Ao nível das aplicações, quanto à largura de banda, podem ser distinguidos dois tipos: aplicações elásticas e aplicações inelásticas. As aplicações elásticas possuem a capacidade de adaptar o seu débito de forma dinâmica em função da capacidade do canal de transmissão. As aplicações inelásticas, não possuem a capacidade de adaptar o seu débito, podendo sofrer, ao longo do tempo, uma severa degradação da qualidade do serviço, se se verificar que o débito produzido é superior à capacidade do canal de transmissão [5].

2.2.1.3 Fiabilidade

No domínio da fiabilidade os parâmetros a considerar são: as taxas de erros nos pacotes de dados e o *delivery ratio* [5].

A taxa de erros nos pacotes pode dever-se a vários fatores, como: interferências no canal de transmissão, presença de ruído que causa erros nos bits transmitidos, e colisões de pacotes quando os protocolos não as conseguem evitar, entre outros. Ao nível das taxas de erros destacam-se: o BER (*Bit Error Rate*), o PER (*Packet Error Rate*) e o PLR (*Packet Loss Rate*).

O parâmetro do *delivery ratio* designa a proporção de mensagens transmitidas que chegam ao receptor sem erros.

Existem aplicações que são intolerantes a erros, logo, requerem a retransmissão de pacotes com erros, aumentado a latência na rede. Contrariamente, as aplicações que são tolerantes a erros, no caso de uma taxa de erros elevada, degradarão a qualidade do serviço prestado.

2.2.2 Parâmetros de Medição de QoS

Dos domínios anteriormente mencionados existem parâmetros que se distinguem e assumem um papel fundamental para análise da qualidade de um serviço de comunicações. Para que se possa avaliar estes parâmetros é necessário recorrer a ferramentas que permitam interrogar a rede de forma a obter essa informação.

Os parâmetros de QoS relevantes que são comummente usados para avaliar um serviço de comunicações são [6]:

- Índice de Velocidade Relativa (IVR) de *download* e *upload* de um ficheiro por FTP. Este representa um dos testes fundamentais quando se avalia QoS uma vez que na ótica do utilizador é um dos que mais se destaca dada a simplicidade do resultado e das ilações que se podem retirar desse valor. É um resultado que a generalidade dos utilizadores é capaz de entender permitindo-lhe fazer uma rápida avaliação do serviço [6].
- Índice de Velocidade Relativa (IVR) em P2P: Atualmente as aplicações para partilha de ficheiros através de BitTorrent têm vindo a conhecer um elevado crescimento, tornando-se, assim, um dos elementos mais importantes na medição e avaliação de QoS [6].
- Tempos de Carregamento de Página *Web*: Este teste assume um papel muito importante dado que a "navegação" na Internet (*web browsing*) representa a maior porção do tráfego total transacionado a nível global, cerca de 33,7%", segundo a *Sandvine*, em *Intelligent Broadband Networks* [7].
- Latência da Rede: Como já foi referido acima a latência caracteriza-se como sendo um parâmetro no domínio do tempo que é afetado por alguns fatores como as características do meio, a capacidade da rede, tráfego, etc.
- Taxas de Erros: Através da avaliação das taxas de erros, e em concreto sabendo o número de pacotes perdidos é possível efetuar uma apreciação da qualidade de uma conexão, sendo que este valor permite retirar algumas conclusões de imediato e constitui uma excelente base para uma comparação entre diferentes conexões.

Estes testes são amplamente usados para avaliar a qualidade de serviço de um sistema de comunicações. Apresentam-se como um conjunto de testes simples, mas eficazes, no ponto de vista do objetivo a alcançar. São medições que são passíveis de ser facilmente interpretadas pela generalidade dos utilizadores ao mesmo tempo que oferecem garantias de avaliação de QoS.

2.2.3 Quality of Experience

Quality of Experience (QoE) é um conceito que remete para a avaliação que um utilizador faz do seu serviço em função da sua experiência de utilização. Este é um termo que carece de alguma definição concreta e homogénea, mas o ITU define-o (G.1011 e G.1000) como "*a aceitação geral de um serviço ou aplicação experienciada subjetivamente pelo utilizador*".

Avaliar e medir a qualidade de um serviço sempre foi uma preocupação das entidades que regulam os sistemas de comunicações. Existem ferramentas e modelos que permitem efetuar testes e previsões sobre como a qualidade de um serviço evolui ao longo do tempo. Contudo, uma vez que os serviços são desenhados e desenvolvidos para o público, em última análise a sua opinião sobre a experiência de utilização dos mesmos é determinante para atestar a qualidade com que um determinado serviço de comunicações é prestado.

Para se poder obter as opiniões dos utilizadores é necessário recorrer a modelos de avaliação. O modelo que o ITU recomenda (P.800) é conhecido como *Mean Opinion Score* (MOS). Este designa um teste que tem como base uma medição subjetiva, cujo princípio é obter a percepção do utilizador face à qualidade de um serviço. O MOS define-se (ITU P.800.1) como " *um conjunto de valores numa escala pré-definida onde os utilizadores oferecem a sua opinião de performance de um sistema de transmissão*". Inicialmente foi desenvolvido para testes de áudio e vídeo, mas pode ser transposto para testes de serviços de natureza diferente. O MOS é expresso como um único valor, numa gama de 1 a 5, onde 1 é o valor de qualidade experienciada mais baixo e 5 o maior. Para cada valor é estabelecido um enquadramento que guia os utilizadores em relação à experiência que obtêm do serviço.

O modelo MOS para além da sua competência para avaliar um serviço de forma subjetiva possui mecanismos de avaliação objectiva e mecanismos de avaliação baseados em previsões (ITU P.800.1).

Tabela 1 – Avaliação de QoE segundo o Modelo MOS (ITU P.800).

MOS	Qualidade	Enquadramento
5	**Excelente**	**Imperceptível**
4	**Bom**	**Pouco perceptível**
3	**Aceitável**	**Pouco irritante**
2	**Pobre**	**Irritante**
1	**Mau**	**Muito irritante**

2.3 Universal Mobile Telecommunication System

2.3.1 Introdução

A tecnologia UMTS (*Universal Mobile Telecommunication System*) designada como a terceira geração das comunicações móveis. Foi desenvolvida e mantida pelo 3GPP (*Third Generation Partenership Project*). Esta tecnologia surgiu com a necessidade de uma nova geração de redes móveis que proporcionasse serviços mais sofisticados com uma forte componente multimédia (vídeo e áudio). Tem como objetivo fazer convergir vários tipos de tráfego no mesmo meio, enquanto tem em consideração os diferentes requisitos de QoS para cada serviço. Cada aplicação possui os seus próprios requisitos de QoS. O conceito de Qualidade de Serviço é um conceito *end-to-end* que necessita de ser satisfeito através da cooperação entre todas as infraestruturas por onde os dados fluem [8].

Nas redes UMTS a QoS tem que ser providenciada tanto para voz como para dados. O serviço de voz, ainda assim, tem prioridade, uma vez que representa o serviço primário e é bastante sensível ao *delay*, sendo este um serviço em tempo real. O serviço de dados é menos exigente em termos de QoS [9].

O UMTS apresenta como fatores vantajosos o suporte de débitos mais elevados, maior eficiência espectral e maior controlo sobre a Qualidade de Serviço.

O UMTS-HSPA é considerado como 3.5G, sendo um melhoramento à tecnologia UMTS.

No Anexo A é feita uma breve análise à arquitetura do UMTS, que permite compreender melhor os mecanismos que providenciam QoS, apresentado a seguir.

2.3.2 QoS em redes UMTS

Utilizando mecanismos de controlo de QoS, o fornecedor de serviços de rede ou os administradores podem usar os recursos de forma eficiente, providenciando serviços com uma boa qualidade, pois a comunicação na sociedade moderna desempenha um papel de alto relevo e a QoS surge como um elemento fundamental dos sistemas de comunicação.

Quando se refere a QoS é necessário considerar aplicações específicas, uma vez que há algumas que são mais preponderantes, logo, são mais críticas. Conclui-se assim, que há tráfego prioritário [9]. Com os mecanismos de QoS, o objetivo passa por providenciar, preferencialmente, "entrega" de serviços prioritários, utilizando taxas de transmissão de dados suficientes, controlo de latência, ao mesmo tempo que se garante perdas de dados mínimas [9].

Nas redes UMTS existem três classes de serviços distintas:

- *Teleservices*: Inclui os serviços de voz, chamada de emergência e SMS.
- *Supplementary Services*: Inclui serviços suplementares como identificação da linha, encaminhamento de chamada, entre outros.
- *Bearer Services*: Nesta classe estão incluídos os serviços de dados.

Para serviços baseados em redes celulares a QoS depende de diversos fatores como: Atraso, Largura de Banda e Fiabilidade, tal como foi já mencionado.

Quando se considera QoS no domínio das redes celulares, os maiores desafios prendem-se com: variações nas características de transmissão do canal, alocação de largura de banda e suporte de *handoff/handover* entre redes heterogéneas. Cada uma das camadas da pilha protocolar possui mecanismos específicos para fornecer um serviço com qualidade. Desta forma providencia-se um serviço de rede flexível e mais tolerante a falhas. Estes mecanismos assumem um papel cada mais relevante dada a crescente necessidade de suporte de voz e dados de alta qualidade.

O *Bearer Service*, é responsável por providenciar a QoS nos serviços de dados. Para que o *Bearer Service* seja capaz de providenciar qualidade de serviço, deve ser planeado e implementado com funcionalidades definidas de forma clara pelo prestador de serviços, para se poder estabelecer um sistema de comunicações *end-to-end* fiável. Um serviço *end-to-end* designa a comunicação efetuada entre TE's (*Terminal Equipments*) [9].

A arquitetura do *Bearer Service* é composta por várias camadas (ver Fig. 1).

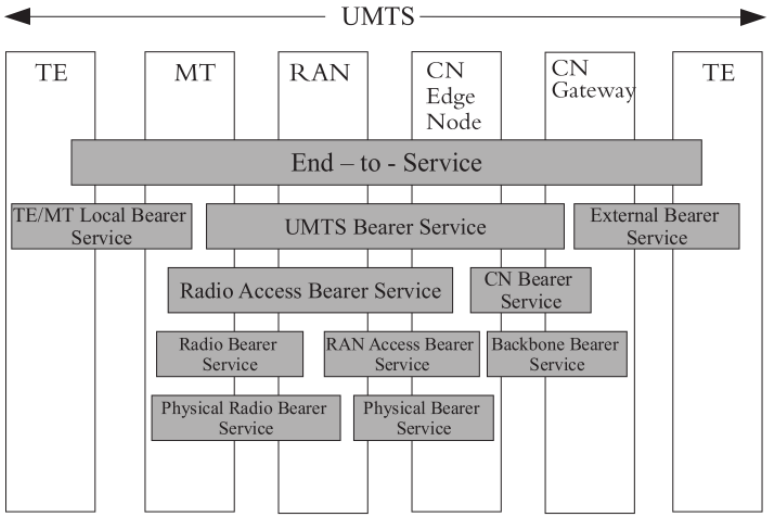


Figura 1 – QoS End-to-End (UMTS) [9].

A camada *End-to-End* é a camada superior da arquitetura de QoS. Esta torna possível a comunicação entre os TE's. Por sua vez, os TE's estão ligados à rede UMTS através de MT's (*Mobile Terminals*). Os serviços afetos à camada *End-to-End* vão ser realizados pelas camadas abaixo. O operador UMTS oferece serviços providenciados pelo UMTS *Bearer Service*, sendo que, este providencia, por sua vez, QoS UMTS [12].

O UMTS *Bearer Service*, é constituído por duas partes distintas: o RAB *Service* (*Radio Access Bearer*) e o CNB *Service* (*Core Network Bearer*) [9].

O RAB *Service* é responsável por providenciar transporte seguro de sinalização e dados do utilizador entre o MT (*Mobile Terminal*) e o *Edge Node* do CN (*Core Network*), com QoS adequado. O RAB *Service* depende das características da interface de rádio e também providencia proteção contra erros, se for implementada essa componente.

O CNB *Service* conecta o *Edge Node* do CN com o *gateway* do CN para a rede externa. Este serviço controla e utiliza eficientemente a rede de *backbone* de modo a providenciar o serviço de 3G UMTS contratado [12][9].

O RAB *Service* é providenciado pelo RB *(Radio Bearer) Service* e pelo RAN (*Radio Acces Network*) *Service* . O RB *Service* cobre todos os aspectos da interface de transporte rádio. Para providenciar a proteção contra erros, o RAN e o MT possuem a

capacidade para segmentar/reconstruir os fluxos dos utilizadores em sub-fluxos requisitados pelo RAB *Service*.

O CNB *Service* usa um *Backbone Network Service* genérico que cobre todas as funcionalidades da camada 1 e camada 2 e é selecionado de acordo com a escolha do operador para cumprir os requisitos do CNB *Service* [9].

Por forma a distinguir serviços de natureza diferentes e com requisitos de QoS específicos foram desenvolvidas classes que os diferenciam. Existem quatro classes de QoS no UMTS:

- Conversação;
- *Streaming*;
- Interativa;
- *Background*.

Uma das maiores diferenças entre as classes é a sensibilidade ao *delay*. A classe de Conversação e a de *Streaming* são adequadas para tráfego muito sensível. Estas são usadas para tráfego em tempo real.

As classes Interativa e de *Background* são menos sensíveis ao *delay* e são usadas para aplicações como: *email*, notícias, FTP, etc. Devido a esta característica, ambas apresentam taxas de erros muito menores que a restantes classes, devido a técnicas de codificação e retransmissão. Isto significa que ocorre uma retransmissão de um pacote sempre que ocorre um erro ou perda de dados. Apesar de serem insensíveis ao *delay* requerem um elevado *throughput* e baixas taxas de erros.

Na classe conversacional inserem-se serviços como telefonia por voz, VoIP e vídeo conferência. A conversação em tempo real é sempre realizada entre grupos de humanos, tornando assim este domínio único, dado que as características requeridas são estritamente obtidas pela percepção humana. O *delay* máximo de transferência é ditado pela quantidade que os humanos toleram para áudio e vídeo. Desta forma, as exigências para este parâmetro são bastante restritas. Se o *delay* de transferência não for baixo o suficiente a qualidade da transmissão irá ser afetada, devendo este ser mais baixo que o *delay* de *round trip* [12].

Uma das aplicações mais comuns que se insere na classe de *streaming* é a transmissão de um vídeo em tempo real. Esta classe caracteriza-se pelo facto das relações de tempo (variações) entre entidades de informação necessitarem de ser

preservadas. A variação do *delay* do fluxo *end-to-end* deve ser limitado para preservar a variação de tempo entre entidades de informação do *stream*.

A aplicação que mais se destaca na classe interativa é o *web browsing*. O esquema de comunicação interativo é caracterizado pelo padrão de pedido/resposta do utilizador. O *delay* de *round trip* é o atributo mais importante desta classe. Outro atributo também relevante neste contexto é a taxa de erros, que deve ser bastante baixa aquando da transferência de dados.

A classe *background* é uma classe de serviço onde as aplicações correm em *background*. Um dos exemplos mais comuns é o serviço de *email*. As aplicações que se inserem nesta classe dormem a maior parte do tempo e são acordadas sempre que ocorrem eventos ou em períodos de tempo pré-definidos. O tráfego de *Background* é caracterizado pelo facto do destinatário não estar à espera da recepção de dados. Esta classe é a menos sensível ao *delay* e os conteúdos devem ser entregues com baixas taxas de erro.

Os algoritmos de escalonamento de recursos dão mais prioridade à classe Interativa do que à de *Background*. Devido a esta razão, a última usa recursos quando as aplicações interativas não necessitam dos mesmos.

2.4 ANACOM e o Consumidor

O trabalho desenvolvido pela ANACOM não se cinge apenas à regulamentação e à execução de testes de QoS sobre os serviços prestados pelos operadores. Outra componente do trabalho realizado pela ANACOM é a prestação de informação aos utilizadores. O "Portal do Consumidor" apresenta-se como um meio de informação muito relevante na tarefa de consciencialização dos utilizadores para o tipo de serviço que pretende contratar, informando o utilizador sobre o que procurar num oferta comercial de um operador, tendo em conta aquilo que se adequa às necessidades do mesmo. Outro ponto relevante é apresentar um conjunto de parâmetros de QoS que caracterizam o serviço que o utilizador contrata, seja telefone móvel, telefone fixo, Internet e televisão. Depois de apresentada alguma informação sobre estes parâmetros, para mais informação são colocados *links* para os relatórios de avaliação de QoS efetuados permitindo ao utilizador fazer um escolha de serviço fundamentada e adequada às suas necessidades. É possível através do portal consultar e pesquisar

tarifários para os diversos serviços de todos os operadores nacionais através da área "com.Escolha", bem como efetuar simulações.

No portal da ANACOM existe também uma secção de dúvidas, onde são apresentadas algumas FAQ (*Frequently Asked Questions*), mas, caso a questão de um utilizador não tenha correspondência com algumas destas, é possível colocar uma questão, na área "A ANACOM Responde".

Por fim, outra das possibilidade disponibilizada é a possibilidade de apresentar uma reclamação contra um determinado operador.

O papel da ANACOM, também no domínio da informação e prestação de esclarecimentos, bem como consulta de tarifários ou até a possibilidade de efetuar reclamações sobre um serviço de um operador constitui um instrumento muito relevante na óptica do utilizador, pois tem um elevado grau de apoio numa área tecnológica onde por vezes pode ser difícil tomar as decisões acertadas. Neste aspecto, o papel das NRA's é também fundamental, podendo prestar informações de forma imparcial que tem como único objetivo consciencializar e apoiar os utilizadores no domínio das tecnologias de comunicação.

2.5 Trabalho Relacionado

Esta secção apresenta uma análise a trabalhos desenvolvidos na área da avaliação de Qualidade de Serviço. Explora algumas metodologias de teste usadas na aferição, bem como ferramentas disponíveis no mercado.

É feita uma análise a relatórios e a documentos relativos à aferição de QoS em diferentes partes do Mundo, com maior incidência na Europa e em Portugal.

2.5.1 QoS na União Europeia

Na Comunidade Europeia tem havido uma crescente necessidade de criar planos de desenvolvimento sustentado para os Estados Membros na tentativa de criar uma comunidade mais homogénea do ponto de vista social, político e económico. Desta forma, a Comissão Europeia adoptou, em Março de 2010, a estratégia "Europa 2020" com vista ao desenvolvimento económico e social para promover crescimento e sustentabilidade inteligente com o intuito de estimular o emprego e proporcionar

coesão social nos Estados Membros. Uma das principais iniciativas é alcançar acesso de banda larga de alto débito, tipicamente com valores acima dos 30 Mbits/s, havendo intenção, de num futuro próximo, se alcançar os 100 Mbits/s na generalidade dos agregados familiares dos países europeus. Segundo estudos apresentados pela UE, 95,7% dos Europeus tem acesso banda larga de alto débito em áreas urbanas, 78% em zonas rurais, sendo que, em muitos países a cobertura de acesso em banda larga cobre menos de 60% da população em zonas rurais [13].

2.5.1.1 Estudos de Avaliação de QoS

Com vista a alcançar os objetivos pretendidos e por forma a avaliar o serviço em geral disponibilizado pelos ISP's europeus são levados a cabo estudos nesse sentido, fazendo parte das iniciativas da Comissão Europeia a monitorização do acesso em banda larga, que, desde Julho de 2002, são da responsabilidade do Comité das Comunicações [13]. Um dos últimos estudos foi comissionado à empresa SamKnows, por forma a obter dados estatísticos concretos sobre a performance do acesso em banda larga nos Estados Membros da União Europeia. As metodologias da SamKnows são globalmente aceites pelos governos e entidades reguladoras na Europa, Estados Unidos da América, América do Sul e Ásia para efetuar medições sobre a performance no acesso fixo e móvel em banda larga [13].

Como consequência destes estudos, governos e NRA's são capazes de cooperar de forma positiva com a indústria das comunicações, ISP's, Universidades e grupos de consumidores, não só para educar as várias partes envolvidas sobre as limitações das várias tecnologias disponíveis, mas também para promover mais investimento em acessos de banda larga mais consistentes e com maiores débitos. O estudo apresentado pela SamKnows em Março de 2012 veio providenciar um importante ponto de referência para as contínuas investigações sobre a qualidade do acesso à Internet na Europa, resultando numa maior transparência e abertura do sector das comunicações. Este estudo contou com a participação de 250 ISP's distribuídos por todo o continente Europeu, mas, espera-se que muitos mais se possam aliar a este projeto no futuro.

2.5.1.2 Metodologias de Teste

Uma das escolhas fundamentais quando se desenvolve uma solução para a avaliação de parâmetros de QoS de um acesso de banda larga é a abordagem: por *hardware* ou por *software* [13].

As abordagens por *software* são as mais comuns e permitem que se atinja uma amostra bastante ampla com relativa simplicidade, sendo que as aplicações baseadas em *Web* são as que mais se destacam nesta abordagem. Estas, tipicamente usam Flash ou Java Applets [13]. Geralmente os clientes fazem um *download* de um ficheiro de um servidor remoto que mede o *throughput* da transferência. Os testes mais completos também efetuam medições de *upload* e latência da rede. As soluções por *software* apresentam desvantagens em relação às soluções por *hardware*, das quais se destacam:

- O *software* geralmente não tem em conta múltiplas máquinas na rede;
- O *software* pode ser afetado pela qualidade e construção da máquina;
- Configurações de rede ou equipamento de rede obsoleto produzem dados pouco fiáveis;
- O consumidor tende a movimentar a máquina alterando as condições iniciais de teste, afetando, assim, a performance;
- Os testes são efetuados apenas quando a máquina está ligada, impedindo criar um perfil de avaliação de 24 horas. No caso de aplicações manualmente executadas, o perfil de teste é ainda mais duvidoso, pois, será criado em situações em que o utilizador encontra problemas no seu serviço.

Os testes realizados com base em *hardware* apresentam maior precisão, recorrendo a equipamentos específicos que são colocados na ligação do consumidor à Internet. Estes, não são afetados pelo fato do equipamento de trabalho do consumidor estar ou não ligado, permitindo uma recolha de dados constante, elaborando perfis de 24h. Umas das principais desvantagens é o custo deste tipo de equipamento de medição ao qual acresce a necessidade de uma instalação especializada.

As métricas chave que são usadas nas aplicações da SamKnows são:

- Débito em *Download*;
- Débito em *Upload*;

- Latência da Rede;
- Perda de Pacotes na Rede.

2.5.2 Internet Móvel em Portugal

A nível nacional existem, atualmente, três grandes operadores de telecomunicações: Vodafone, Optimus (agora denominado de NOS) e TMN (agora denominado de MEO).

Portugal, é um país que investe bastante na área das telecomunicações, acompanhando constantemente as tendências mais recentes das tecnologias nesta área. Exemplo disso é a implementação da tecnologia LTE por parte dos três grandes operadores no primeiro trimestre de 2012 [14]. O sector das comunicações, em Portugal, representa uma elevada fatia do PIB (Produto Interno Bruto) nacional, geralmente entre o 4,5% e os 5%, mas tendo vindo a sofrer um decréscimo desde 2009 [15] muito devido ao panorama de crise económica que o País atravessa.

A ANACOM é a entidade responsável pelo sector das comunicações a nível nacional. À mesma compete realizar periodicamente testes às redes e serviços dos operadores. O último relatório efetuado data de Julho de 2010 [6]. Nele é revelada a QoS dos operadores ao nível de acesso à Internet em redes celulares. Atualmente, este tipo de estudo assume um papel bastante relevante para a compreensão de como o a qualidade do acesso à Internet em redes celulares é afetado por diferentes factores, seja de natureza tecnológica ou física.

2.5.2.1 Análise ao Relatório da ANACOM sobre QoS

O relatório providenciado pela ANACOM fornece um estudo sobre os acessos móveis 3G. A tecnologia utilizada pelos três operadores é UMTS-HSDPA.

Actualmente, no panorama nacional "*verifica-se que a maior parte dos clientes estão associados a ofertas comerciais de velocidades bastante inferiores aos 21.6Mbits/s. Tendo por referência esta informação, o estudo baseou-se, para os três operadores, numa oferta comercial pré-paga de 2Mbits/s / 384Kbits/s"* [6].

O relatório disponibilizado pela ANACOM tem como objetivo aferir os parâmetros de QoS dos serviços dos três maiores operadores a nível nacional. Para tal

foram efetuados testes em locais *indoor* e *outdoor* nos concelhos de: Lisboa, Porto e Faro.

Nos testes efetuados foram utilizados três servidores, um nacional (Servidor da FCCN) e outros dois internacionais, um nos E.U.A (Houston) e o outro na União Europeia (Londres).

2.5.2.1.1 Velocidade Média de Transferência de Ficheiros

Segundo os dados recolhidos os três operadores apresentam valores muito próximos aos valores contratados (2Mbits/s), sendo que o acesso aos servidores estrangeiros apresentam menores taxas de transmissão que o acesso ao servidor nacional. Os operadores Vodafone e Optimus apresentam melhores resultados, sendo que o primeiro, no Concelho do Porto, em média, garante velocidades superiores ao valor contratado.

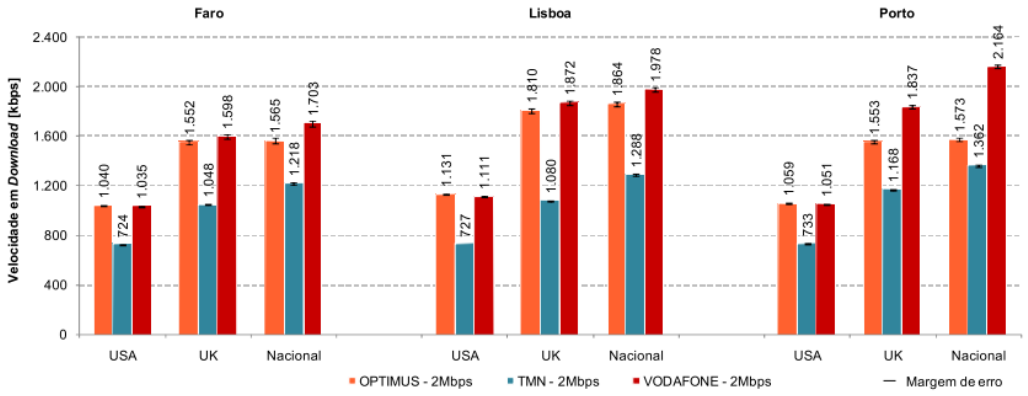


Figura 2 – Velocidade Média de transferência de ficheiros em *Download* [6].

De forma geral, os operadores superam os valores máximos contratados para *Upload* de 348kbits/s.

A destacar os desempenhos da Vodafone no concelho do Porto e da Optimus nos concelhos de Lisboa e Faro.

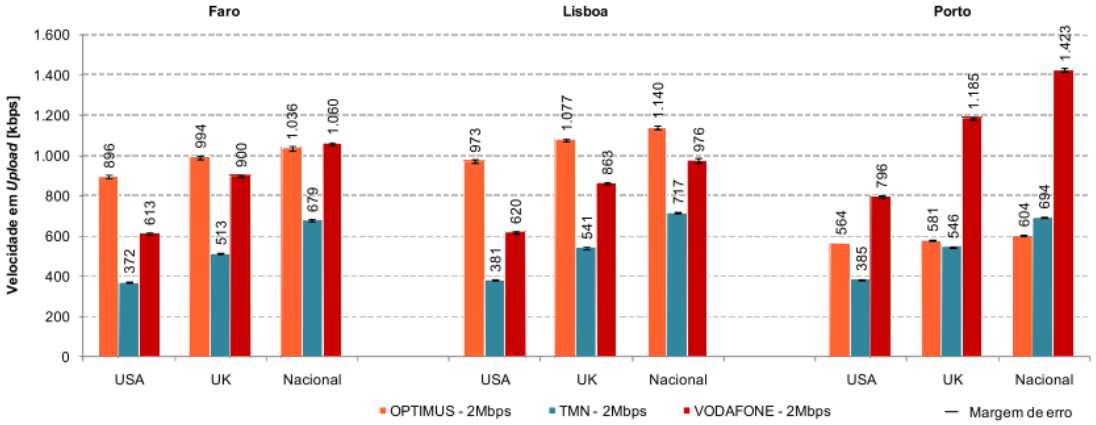


Figura 3 – Velocidade Média de transferência de ficheiros em *Upload* [6].

2.5.2.1.2 Latência da Rede

Para o servidor nacional e o europeu (Londres) os valores de latência não ultrapassam os 80 ms. Como seria espectável, para o servidor mais afastado (E.U.A.) a latência é muito mais elevada, ainda assim dentro de valores aceitáveis [6].

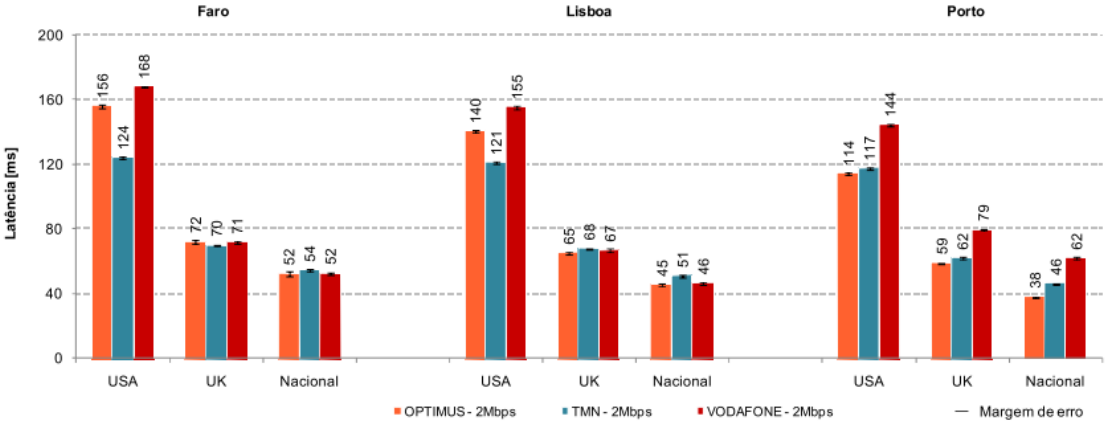


Figura 4 – Latência Média [6].

2.5.2.1.3 Disponibilidade do Serviço

Para que seja possível a conexão à Internet em acessos móveis é necessário o estabelecimento de ligação para atribuição de recursos e autenticação.

A taxa de sucesso no estabelecimento permite avaliar a disponibilidade do serviço. Segundo relatório da ANACOM para este parâmetro, o sucesso é superior a 99%, garantindo uma elevada percentagem de disponibilidade. As diferenças apuradas para os três operadores não são significativas, indicando um serviço semelhante entre os mesmos. O gráfico da Fig. 5 permite avaliar a taxa de sucesso dos operadores.

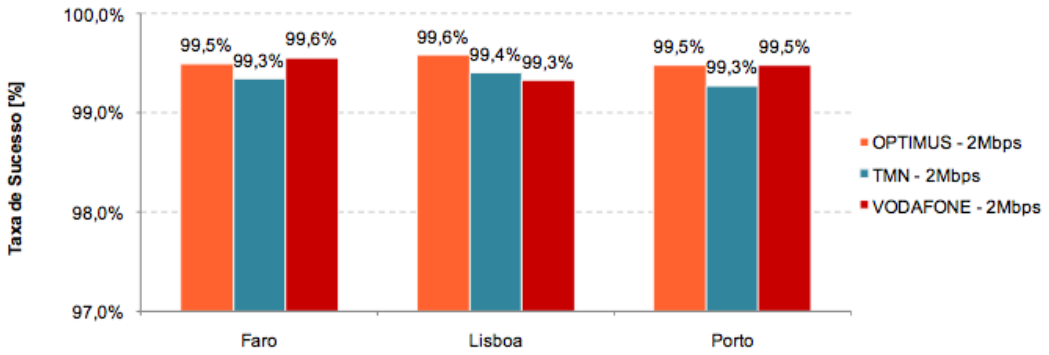


Figura 5 – Taxa de Sucesso [6].

2.5.2.1.4 Tempo de estabelecimento/ativação da ligação

O tempo de estabelecimento/ativação da ligação à Internet é considerado um tempo de atraso percepcionado pelo utilizador, sendo um parâmetro bastante relevante para qualquer serviço.

Na Fig. 6 é possível apurar que a TMN nos concelhos de Faro e Lisboa apresenta tempos mais baixos para estabelecer a ligação, na ordem de 1 segundo a menos, relativamente aos concorrentes. No entanto, no concelho do Porto o valor é muito mais elevado, superior à Vodafone e Optimus.

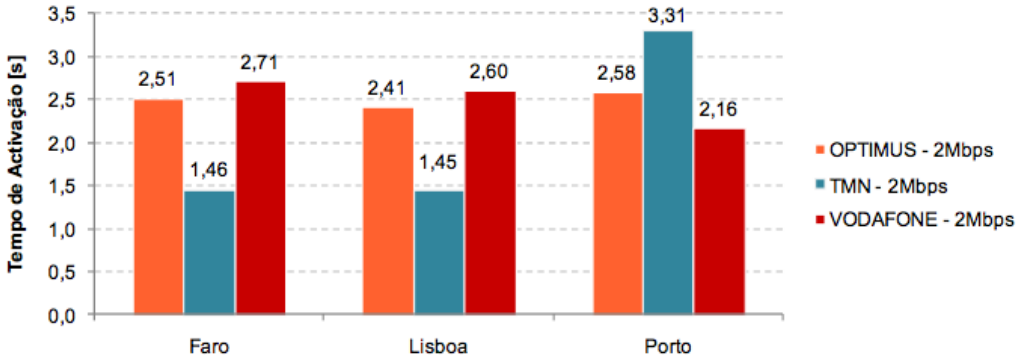


Figura 6 – Tempo de ativação/estabelecimento da ligação [6]

2.5.3 Internet no Mundo

2.5.3.1 Internet em África

O continente africano apresenta o menor índice de desenvolvimento do Mundo. Os seus países apresentam taxas de subdesenvolvimento em vários sectores, incluindo a saúde, política, economia, educação, bens de consumo, entre outros [16]. O sector das telecomunicações é, também, muito precário, muitas vezes monopolizado, sem concorrência de mercado, o que obriga os cidadãos a subscrever serviços com baixa qualidade, com custos demasiado elevados, e superiores aos de países desenvolvidos (quando comparados sobretudo com a Europa e Estados Unidos da América, com serviços muito superiores).

O Dr. R. Les Cottrell no seu artigo "*How Bad is Africa's Internet*" [17] conta a sua experiência em Kinshasa, na República Democrática do Congo. No período em que permaneceu neste país, constatou que na Universidade de Kinshasa, para um "*staff*" (alunos, funcionários e professores) de cerca de 3000 pessoas, só existiam 800 computadores. O uso da Internet cingia-se, apenas, a uso interno, sobretudo para a troca de *email*s e informação académica. O acesso externo era restrito a cerca de 200 pessoas com privilégios mas, devido à qualidade do acesso era praticamente impossível o seu uso - "*the connection was barely usable*" [17]. Neste artigo, o autor

refere que a situação da Universidade que visitou era uma "realidade em muitas mais instituições, organizações, e comunidades em várias partes subdesenvolvidas do Globo". Afirmando que : "*Without these connections, many developing countries miss out on innovations that offer higher standards of living, such as telemedicine, remote learning, and online commerce*" [17].

O Dr. R. Les Cottrell, aponta ainda alguns dados relevantes para a percepção da qualidade do acesso à Internet em África. Refere que menos de 16% dos Africanos possuem acesso à mesma, ao contrário dos 63% na Europa, 79% nos E.U.A. e mesmo dos 32% verificados na Ásia. A *performance* no acesso é um dos principais motivadores para estas estatísticas, pois, afirma que em África é apenas de 1/70 da verificada nos países Europeus. Outra questão relevante é o custo, muito mais elevado. É mesmo dado um exemplo concreto: numa universidade na Alemanha, por uma largura de banda de 1Gbits/s, num mês, o custo é de 40.000$ (dólares americanos), quando no Quénia, para a mesma largura de banda, o custo é de 200.000$ (dólares americanos).

É necessário tentar medir de forma eficiente, com baixos custos a *performance* do acesso à Internet em África. O projeto PingER (*Ping End-to-End Reporting*) é isso que tenta concretizar. Os dados deste projeto são recolhidos e comparados com os de outras regiões permitindo avaliar mudanças ao longo do tempo. O PingER baseia-se no teste Ping comum (ICMP), possibilitando, avaliar regularmente a qualidade com que os dados fluem na rede, tal como é explicado pelo Dr. R. Les Cottrell. "*A ubiquidade e facilidade do teste "ping" torna-o especialmente adequado para uma monitorização alargada da Internet, particularmente em regiões como África, onde aplicações mais avançadas seriam impraticáveis*" [17]. Com vários testes Ping sucessivos é possível determinar: o "*round-trip time*" (latência) de cada pacote de informação, o *jitter* (medida de variação do atraso) e a percentagem de pacotes perdidos na rede. Um *jitter* elevado e perdas constantes de pacotes, indicam congestionamento na rede ou largura de banda insuficiente. Este conjunto de parâmetros é essencial para determinar a qualidade do serviço.

Como conclusão do seu artigo, o Dr. R. Les Cottrell afirma que, no ano de 2009, ano em que escreveu o mesmo, o *througput* em África era semelhante ao da Europa em 1994. Em nota de projeção prevê que a qualidade do acesso à Internet em África irá alcançar a da Europa apenas em 2030.

2.5.3.2 Projeto PingER

Este projeto tem como objetivo avaliar as ligações à Internet de um vasto conjunto de *hosts* e é desenvolvido pelo grupo IEPM (*Internet End-to-End Performance Measurement*) do *Stanford Linear Accelerator Center* (SLAC). Com este projeto é possível monitorizar mais de 300 *hosts* globalmente. Este projeto tem vindo a crescer de forma exponencial nos últimos anos e tem alcançado uma vasta dimensão, abrangendo bastantes instituições e pessoas um pouco por todo o Mundo.

Na página oficial são apresentados diversos relatórios e artigos sobre o projeto, bem como resultados de avaliações efetuadas com pormenores detalhados dos *hosts* e servidores [18]. É possível consultar e utilizar ferramentas que permitem visualizar esses dados de forma fácil e simples recorrendo a gráficos, tabelas, entre outros. Tudo isto demonstra um cuidado elevado e o empenho dos envolvidos.

Tal como foi referido anteriormente este projeto baseia-se no uso do ICMP (*Internet Control Message Protocol*) Ping.

No contexto do projeto PingER, o teste Ping é usado para medir o tempo de resposta, a percentagem de pacotes perdidos, a variação do tempo de resposta e a disponibilidade da rede.

Devido à dimensão que o projeto PingER atingiu ao longo dos anos, com os dados que foram sendo recolhidos foram desenvolvidas ferramentas e mecanismos de observação desses dados, seja através de gráficos, esquemas e até ferramentas que permitem interatividade com o utilizador. É possível observar de forma simples os dados e constatar como evoluem alguns dos parâmetros de medição de QoS no tempo, num determinado País ou Continente. Este tipo de ferramentas vocacionado para os utilizadores constitui uma forma bastante apelativa para o relacionamento e comparação de dados. A abertura e exposição dos dados recolhidos para consulta do público em geral representa uma das mais valias deste projeto.

Uma das ferramentas de observação que mais se destaca é a *PingER Metrcis Motion Chart*, criada pela Google que permite a visualização de dados recolhidos pelo projeto PingER nos últimos 15 anos. Apresenta dois eixos de visualização que podem ser alterados para apresentar diferentes tipos de dados técnicos como: *througput* ou *jitter* ou indicadores sociais como: HDI (*Human Developmnet Index*) ou CPI (*Corruption Perception Index*). A escala em cada eixo pode ser linear ou logarítmica.

Desta forma é possível conjugar não só dados tecnologicamente relevantes como é possível relacioná-los com índices sociais de desenvolvimento, que remetem para as origens deste projeto, medir e avaliar a qualidade de acesso à Internet à escala global com forte incidência nos países em desenvolvimento onde a recolha e avaliação de parâmetros de QoS é escassa, fornecendo, assim, um panorama geral sobre estes países, quando nem instituições governamentais nem empresas de domínio privado o conseguem fazer. A taxa de penetração do projeto PingER é muito significativa, tendo a tendência para evoluir cada vez mais à medida que novos *hosts* vão sendo adicionados e monitorizados. Pela Fig. 7 é possível observar um gráfico que compara a evolução de dois parâmetros de QoS entre os anos de 2000 e 2012 em Portugal.

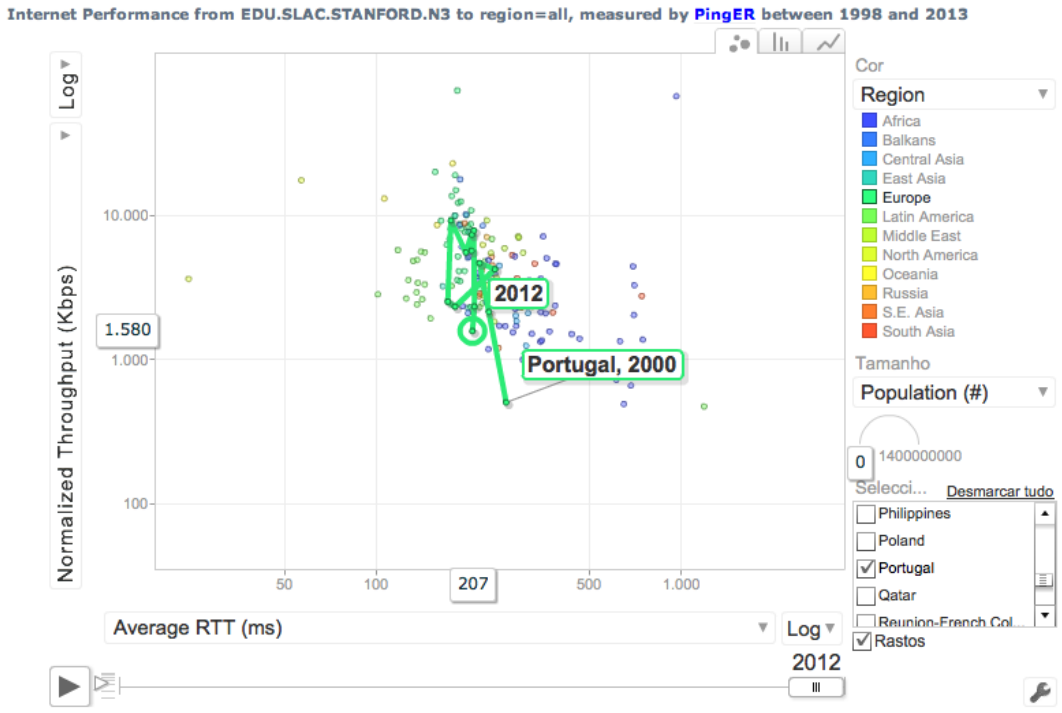


Figura 7 – PingER Metrics Motion Chart [18].

Os parâmetros observáveis são: o *Throughput* (Kbits/s) representado no eixo das ordenadas (*yy*) e o RTT médio (ms) representado no eixo das abcissas (*xx*). Os eixos estão representados na escala logarítmica. Ao nível da apresentação gráfica é possível observar os dados em "bolha" como na Fig. 8, histograma ou em gráficos de barras.

Com o uso do Google Maps é possível analisar parâmetros de avaliação de QoS através da visualização de um mapa mundial. As métricas observáveis são as mesmas que na ferramenta *Metrics Motion Chart*. Esta apresenta-se como uma forma simplificada de estudar os dados obtendo uma cenário generalizado distribuído pelo mapa.

Na Fig. 8 é possível analisar dois parâmetros de QoS: *Throughtput* e RTT médio. É possível observar a distribuição global, sendo os dados relacionados em forma de "bolha" através da dimensão e do grau da cor, quanto mais escura maior é o valor de RTT. Quanto maior for o diâmetro da bolha maior é o valor de *Throughput*.

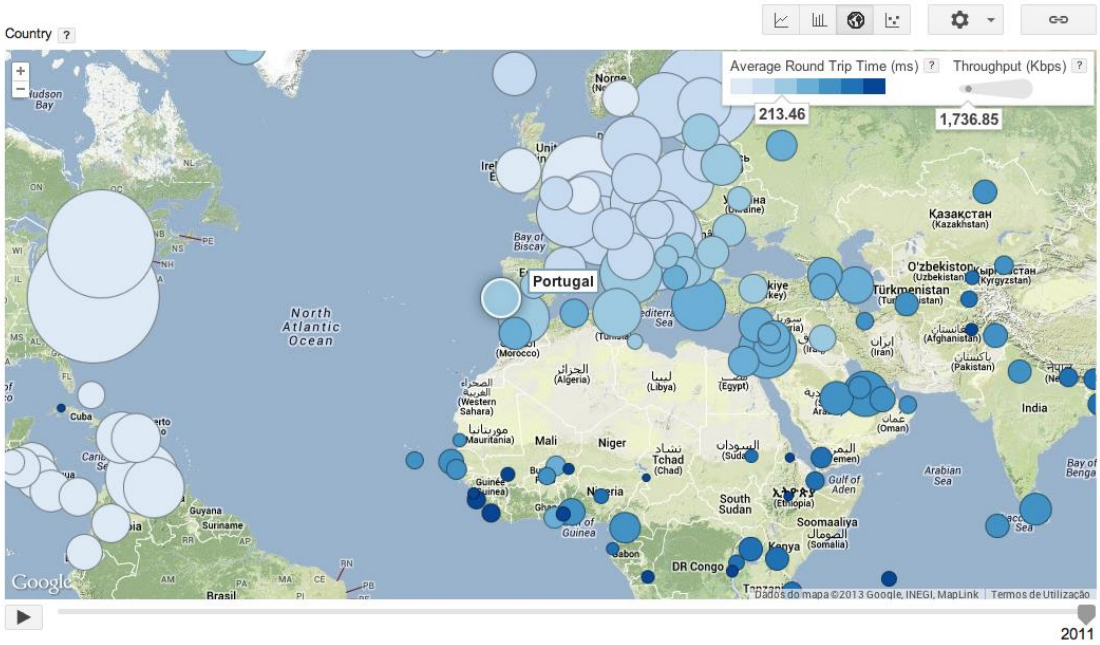


Figura 8 – PingER Metrics Motion Chart [18].

Outro dos mecanismos que permitem a visualização de dados é através de tabelas de Ping, que contêm os dados recolhidos. É possível manipular os parâmetros observáveis. Contudo, é uma ferramenta mais complexa e de visualização mais dificultada, sendo vocacionada para um público alvo mais restrito com conhecimento mais abrangente sobre a matéria em questão. Através de mecanismos disponibilizados consegue-se identificar os servidores de monitorização bem como os *hosts*, sabendo a sua localização, endereço IP entre outras informações.

2.5.4 Ferramentas Comerciais

Ao longo do presente capítulo foram sendo abordados estudos e parâmetros de avaliação de QoS. Foi abordada a temática que refere a necessidade de efetuar testes e de se avaliarem serviços de comunicações, a importância das NRA's e também de empresas que levam a cabo esses testes sobre ISP's por ordem de entidades governamentais. Como forma de tentar esclarecer como são feitos esses testes e que tipo de mecanismos ou aplicações são usadas para o efeito, vão ser analisadas algumas ferramentas. Como já foi referido, existem dois tipos de mecanismos: por *software* e por *hardware*.

Estudos levados a cabo por NRA's como a ANACOM, em Portugal, a FCC (*Federal Communications Commission*) nos Estados Unidos da América ou Ofcom (*Office of Communications*) no Reino Unido recorrem a mecanismos baseados em *hardware*, dada a maior precisão nos dados recolhidos. Por outro lado, ao nível dos mecanismos de *software*, a gama de produtos é muito maior e foca-se, geralmente, na recolha de dados para testes instantâneos para visualização do utilizador, como por exemplo, o SpeedTest (da Ookla).

2.5.4.1 Mecanismos baseados em Hardware

Os testes que recorrem a mecanismos baseados em *hardware* levados a cabo pelas NRA's usam equipamentos especificamente desenhados para o efeito. Geralmente, as NRA's associam-se às empresas que os desenvolvem e efetuam estudos próprios com base nestes equipamentos. Por outro lado, os estudos podem ser comissionados a empresas especializadas. Uma das empresas mais proeminentes neste domínio é a SamKnows, "*líderes no mercado de medições de banda larga*" [19]. É uma empresa que vai de encontro às exigências governamentais, sendo responsável por estudos nos Estado Unidos da América, Brasil, Europa, Reino Unido e Singapura. Todo o projeto assenta em testes realizados pelos seus equipamentos colocados nas instalações dos consumidores. Apresentam como objetivo expor uma visão o mais precisa possível sobre a performance do acesso em banda larga a nível mundial. Uma das principais preocupações da SamKnows é apresentar os dados recolhidos de forma transparente. Trabalha com NRA's, governos, indústria e universidades para se desenhar e construir

metodologias de teste *standard* e abertas [20]. Todos os estudos feitos com base nos dados que recolhem são publicados indo de encontro às suas políticas de abertura e transparência.

Para a recolha de dados a SamKnows apresenta uma *Whitebox*, que é um equipamento de teste que corre um *software* aberto. Qualquer utilizador que esteja interessado em participar nos estudos, se preencher os requisitos impostos, pode pedir uma *Whitebox* e instalá-la na sua casa (ver Fig. 9). Os dados recolhidos serão submetidos para os servidores da SamKnows. Os dados dos testes efetuados, para além do uso em estudos alargados efetuados pela SamKnows, são enviados para o utilizador, via *email*, acesso *web*, ou através de uma aplicação móvel. Outra das possibilidades assenta no facto de o utilizador, detetando irregularidades no seu serviço, com os dados recolhidos, pode submetê-los diretamente para o seu ISP, por forma a que sejam resolvidos.

A *Whitebox* conduz uma série de testes à rede de forma periódica, por forma a recolher a maior quantidade de dados possível, não interrompendo o uso normal da Internet, não necessitando de gerar muito tráfego.

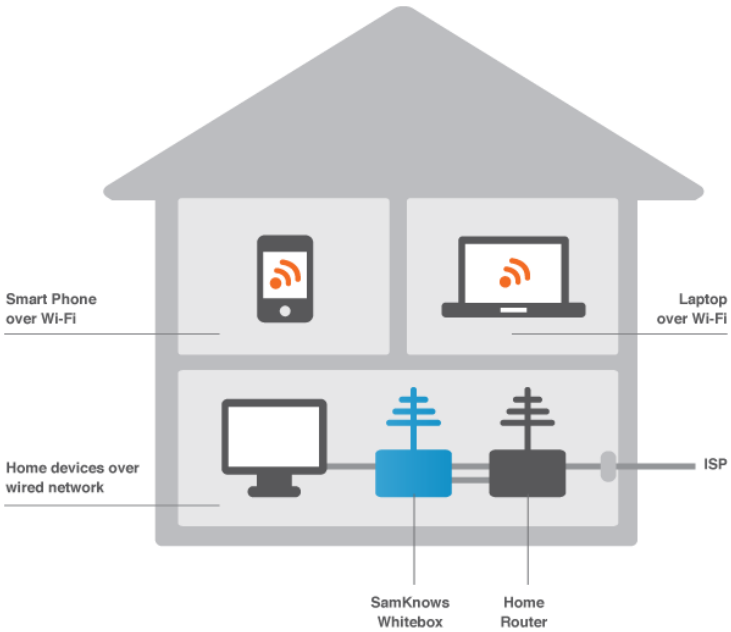


Figura 9 – Ilustração do funcionamento da Whitebox.

O estudo feito pela ANACOM, analisado anteriormente, assenta na tecnologia da Ascom, o TEMS Automatic [21]. Este é um sistema autónomo que permite a recolha

de dados para aferição da qualidade de serviço percepcionada pelo utilizador. Apresenta um largo volume de dados que serve para a elaboração de comparações estatísticas entre os vários operadores nacionais.

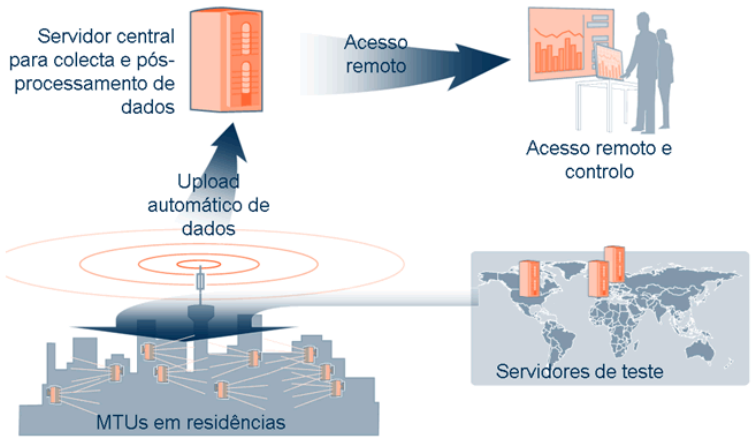


Figura 10 – Arquitetura de Teste (ANACOM) [30].

O TEMS Automatic é uma solução que efetua testes à rede 24 horas por dia, 7 dias por semana, gerando uma quantidade de dados relevante para criar estudos estatísticos sobre os parâmetros de QoS avaliados. É uma solução completa que oferece monitorização de QoS eficiente, permitindo avaliar os diferentes ISP's a operar em Portugal. Todo processo é automatizado não havendo qualquer intervenção humana.

No domínio dos testes efetuados pela ANACOM, foram usados dispositivos em modo estático. Os dados recolhidos são carregados para os servidores para serem processados e servirem de base aos estudos publicados.

2.5.4.2 Mecanismos baseados em Software

Este tipo de mecanismos é muito comum. É um mecanismo amplamente usado e apresenta dados relevantes, mas muito aquém daqueles que são apresentados pelos mecanismos por *hardware*, dadas as suas limitações. Um dos principais pontos onde exibe maior imprecisão é no instante em que ocorre a recolha de dados em si. Uma vez que geralmente são usados pelos utilizadores em situações que experienciam uma fraca qualidade de acesso à Internet, logo, os resultados são bastantes desviados da

realidade do serviço no seu todo. Contudo, não deixa de ser uma ferramenta com grandes vantagens, sobretudo, a capacidade de medições instantâneas que permitem ao utilizador em pouco tempo percepcionar a qualidade do seu serviço. Geralmente, este tipo de aplicações baseia-se em testes Ping e recorrem a servidores, segundo o protocolo FTP para a troca de ficheiros, por forma a avaliar a velocidade de *download* e de *upload*.

Neste domínio pode-se destacar a aplicação SpeedTest by Ookla, sendo uma das mais conhecidas e amplamente usadas, recorre a diversos servidores para efetuar os testes à rede e obter os resultados.

No final do ano de 2013 a ANACOM, em parceria com a FCCN lança uma aplicação, chamada de Netmede, também para efetuar testes de QoS, tanto à rede fixa como rede móvel permitindo comparar a velocidade máxima medida com a que é contratada ao ISP. Uma das características que merece destaque na aplicação desenvolvida pela ANACOM é a possibilidade de fazer testes de *traffic shapping*[1].

Este tipo de aplicações tem vindo a ser desenvolvido para sistemas operativos móveis e são cada mais usados. Com a crescente exigência dos utilizadores na qualidade dos serviços que contratam, que muito se deve à maior acessibilidade de informação de conteúdos, este tipo de aplicações apresenta um papel de relevo, assumindo-se como um dos principais mecanismos de avaliação de parâmetros de QoS no acesso à Internet.

[1] Técnica que classifica e prioriza certos fluxos de tráfego em detrimento de outros, adaptando a largura de banda com o objetivo de a otimizar.

3. Pressupostos

3.1 Introdução

O presente capítulo visa apresentar a abordagem inicial ao projeto, idealizando uma arquitetura de alto nível que permita, posteriormente, identificar e definir os requisitos dos diversos constituintes do sistema. O projeto centra-se no desenvolvimento de um sistema de aferição colaborativa, que recorre a uma aplicação móvel para a recolha de dados e a um serviço de gestão e divulgação de dados.

A análise e estudo dos objetivos do sistema colaborativo permitem isolar e identificar os diferentes componentes do mesmo, tornando possível desenvolver a proposta de alto nível para a sua arquitetura apresentada neste capítulo.

Neste contexto acresce a necessidade de apresentar as métricas que serão usadas e com devem ser interpretadas no âmbito do que se pretende com este projeto.

3.2 Arquitetura de Alto Nível

A arquitetura de alto nível apresenta-se como uma proposta do sistema a desenvolver e tenta mostrar uma visão geral de como este será construído e como se deverão interligar os diferentes constituintes.

O que se pretende neste modelo preliminar da arquitetura é isolar os componentes chave e defini-los de acordo com o que se espera dos mesmos em termos das funcionalidades que asseguram ao sistema.

Nesta arquitetura considera-se os seguintes constituintes:

- Aplicação Móvel;
- Servidor: Este coaduna um subconjunto de constituintes:
    - Servidor de Testes: Este deve implementar um conjunto de ferramentas que permitam à aplicação móvel fazer testes de avaliação da conexão à Internet, baseando-se num conjunto de métricas;

- Servidor de Dados: tem como função efetuar o tratamento de dados, desde os receber da aplicação móvel e comunicar com restantes componentes, sempre que necessário;
- Base de Dados: essencial para salvaguardar dados relativos aos testes dos utilizadores e informação diversa afeta ao sistema;

- *Web Site*: aplicação *web*, mais concretamente um *Web Site* para publicação e divulgação de dados recolhidos pela aplicação móvel.

Estabelecida esta arquitetura de alto nível e identificados os constituintes do sistema é possíjectovel desenvolver um diagrama de blocos funcionais. Na Fig. 11 é possível identificar os blocos do sistema e ver como estes se relacionam entre si, mais concretamente, os tipos de operações básicas que terão que ser desenvolvidas.

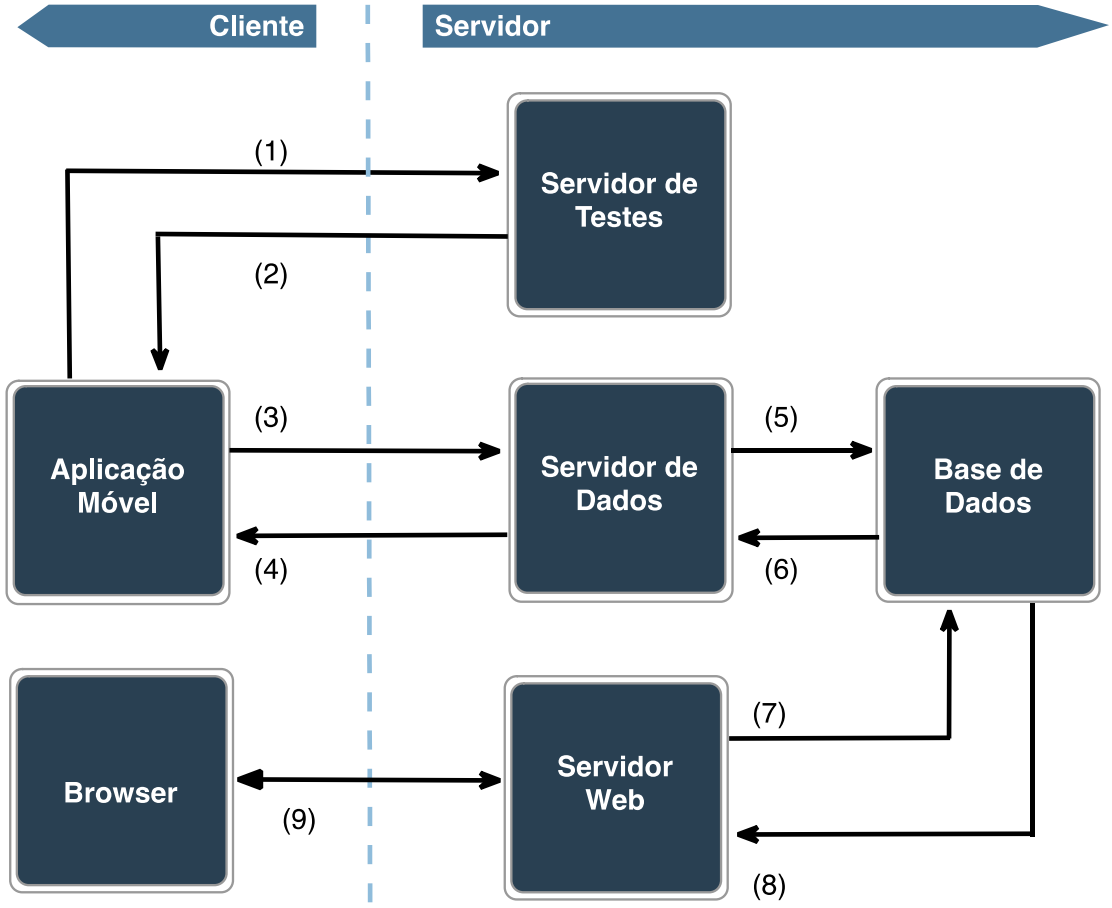


Figura 11 – Diagrama de Blocos Funcionais.

A arquitetura do sistema segue o modelo Cliente-Servidor, onde do lado Cliente se encontra a Aplicação Móvel e o Browser (Cliente *Web*), já no lado do servidor se encontra os Servidor de Testes, o Servidor de Dados, a Base de Dados e Servidor *Web*. A aplicação móvel é o elemento central e de maior destaque em todo o sistema, visto que é a ferramenta de recolha de dados e o principal elemento de interação com o utilizador.

A comunicação entre os componentes é essencial para o seu devido funcionamento. São identificadas oito interações básicas:

- (**1**) – **Execução do Teste de QoS**: Esta operação designa um teste de QoS executado a um servidor de teste. Os tipos de testes e a anatomia de cada um dos mesmos será descrita adiante no presente documento;
- (**2**) - **Resposta do Teste de QoS**: Em (1) o pedido é feito no sentido do servidor. A resposta a esse pedido específico é o que permite efetivamente avaliar uma determinada métrica. Os dados recebidos e como são recebidos são analisados pela Aplicação Móvel e sobre os mesmos são feitos cálculos para determinar a qualidade do teste efectuado;
- (**3**) – ***Upload* de Dados/Pedido de Dados**: Esta operação subdivide-se em duas outras distintas. O *upload* de dados remete para a componente de envio de dados para o servidor, sejam resultados dos testes efectuados, sejam dados do utilizador nas situações de registo ou de *login*. Estes dados são posteriormente armazenados na Base de Dados. O pedido de dados refere-se a pedidos feitos pela aplicação ao servidor de dados, mais concretamente, quando é necessário apresentar dados estatísticos ou outros determinados a partir de informação contida na Base de Dados;
- (**4**) – **Resposta do Servidor de Dados**: O servidor de dados, na interação com a aplicação móvel, tem que responder aos pedidos efectuados. As respostas, de acordo com o pedido pode conter informação útil (dados para consumo pela aplicação móvel) e informação de sinalização ou apenas informação de sinalização, no caso da resposta não necessitar de informação útil;

- (**5**) – **Inserção na Base de Dados**: Esta operação designa a inserção de dados na Base de Dados. A informação recolhida pela aplicação aóvel, caso seja do interesse do utilizador é enviada para o servidor de dados para ser posteriormente armazenada. A aquisição deste tipo de dados é o maior objetivo do projeto em si. Sobre estes é possível traçar um mapa global sobre a Qualidade de Serviço sobre as quatro métricas avaliadas nos testes. A Base de Dados também armazena informação pessoal do utilizador, que está associada aos testes efectuados;
- **(6)** – **Leitura da Base de Dados**: A operação de leitura é executada sempre que há a necessidade de obter dados da Base de Dados. Esta operação pode ser executado como parte do processo de um pedido feito pela Aplicação Móvel ou pelo Servidor *Web*;
- (**7**) – **Pedido de dados**: A operação de pedido de dados efectuado pelo Servidor *Web* designa os conjunto de operações que permitem ao mesmo obter dados que estão na Base de Dados;
- **(8)** – **Resposta ao Pedido de Dados:** A operação de resposta aos pedidos de dados permitem obter os dados que foram requisitados na operação de pedido. Estes dados permitem construir toda a interface de interação do *Web Site*. Sobre estes são calculados dados estatísticos, apresentados gráficos para análise entre outros.
- **(9) – Pedido/Resposta do Browser:** Através de um Browser genérico é possível aceder ao *Web Site*. Para tal existe um conjunto de pedidos e respostas entre o cliente e o Servidor *Web*.

3.3 Métricas e Modelo de Avaliação

Após uma análise aos objetivos do sistema e à arquitetura de alto nível, a definição de quais as métricas a avaliar durante os testes tornou-se, incontornavelmente, um dos primeiros passos para a concepção e estudo dos requisitos. Após estabelecidas as métricas é necessário abordar como os dados

gerados são tratados e interpretados. Isto consegue-se estabelecendo modelos de avaliação que permitem avaliar os dados recolhidos.

Quando se remete para o conceito de métricas, o objetivo passa por escolher aquelas que melhor se adequam, tanto ao nível de interação com o utilizador, ou seja, providenciar dados simples de serem interpretados, mas que ao mesmo tempo possam ter relevância num domínio mais técnico.

As métricas que serão utilizadas para avaliar a qualidade de serviço são:

- Latência da rede (ms);
- Débito binário médio em *download* (bits/s);
- Débito binário médio em *upload* (bits/s);
- Taxa de pacotes perdidos (%).

Este conjunto de métricas permite avaliar de forma concisa uma ligação à Internet, oferecendo um excelente agregado de informação. Estas, permitem estimar de forma concreta e objetiva a QoS da conexão estabelecida.

As métricas relativas aos débitos, permitem determinar a quantidade de dados por unidade de tempo a que o utilizador tem acesso, em ambos os sentidos. Os valores recolhidos para estas métricas podem ser comparados diretamente com aqueles que são anunciados pelos operadores. É de salientar que os valores relativos aos débitos deixaram de ser um factor diferenciador relativamente às ofertas comerciais oferecidas, passando o tráfego mensal/semanal a assumir esse papel.

Na generalidade dos operadores assiste-se à tendência para uma elevada discrepância entre os débitos de *download* e *upload* disponibilizadas. No caso do acesso móvel o débito de *upload* tem um significado menos preponderante, uma vez que o tipo de aplicações geralmente usadas no domínio das redes celulares geram pouco tráfego no sentido ascendente. Devido a este facto, quando se considera um modelo de avaliação de QoS qualitativa, no contexto deste projeto, o peso que é atribuído a este parâmetro deve ser menor que aos restantes.

Para se desenvolver um modelo de referência deve ser tido em conta a realidade atual dos serviços disponibilizados em território nacional. Os operadores apresentam diversas ofertas comerciais com a acesso a redes 3G e 4G, o que leva a diferenças acentuadas nos débitos, dependendo da rede que é usada no instante do teste. Desta

forma é necessário criar um modelo que satisfaça os utilizadores de ambas as tecnologias. Para tal deve ser considerado o tipo de serviços mais usados no domínio das redes celulares, que são sobretudo o acesso a páginas *web* (*web browsing*).

Analisando a generalidade das ofertas comerciais para serviços de dados fornecidos pelos operadores nacionais, verifica-se que existe uma elevada falta de explicitação sobre os débitos associados a essas ofertas comerciais. Devido a este facto a avaliação dos débitos em ambos os sentidos assume uma relevância acrescida.

A avaliação da taxa de pacotes perdidos é um parâmetro que oferece uma excelente perspetiva da qualidade da ligação. A perda é tipicamente causada por congestionamentos que por sua vez fazem que com que filas de espera nos equipamentos, tipicamente nos *routers*, alcancem os valores máximos, levando pacotes a ser descartados. A perda de pacotes pode também ser causada pela entrega de uma cópia imperfeita de um pacote, geralmente causada por erros nas ligações, erros nos equipamentos de rede ou falhas na transmissão de sinais. Quando se obtém um valor para a taxa de perda de pacotes de 0% a rede é considerada "não ocupada". Medindo a frequência com que uma rede está "não ocupada" pode-se obter uma indicador sobre o comportamento e estado da mesma num determinado intervalo de tempo. Em termos qualitativos o maior foco recai sobre os valores instantâneos de perda de pacotes, aquando de um teste. Um elevado número de utilizadores usa aplicações em tempo real, que necessitam de valores muito baixos ao nível da percentagem da perda de pacotes. Para este tipo de aplicações são, geralmente, considerados diferentes níveis qualitativos

Para determinar o valor da latência da rede, é necessário recorrer ao RTT (*Round Trip Time*). Este parâmetro está relacionado com a distância entre locais e o *delay* em cada salto ao longo do percurso *end-to-end*. A latência designa o tempo gasto por um pacote de informação para percorrer o caminho desde a sua origem até ao seu destino. Quanto maior for a latência, maior será a quantidade de dados em circulação na rede num determinado instante. O *delay* em cada salto depende essencialmente da capacidade de processamento do *router* e o escalonamento nas filas de espera. Desta forma é possível avaliar a latência na rede de comunicação, que se apresenta como um dado relevante e bastante importante quando se pretende avaliar a QoS de uma ligação

à Internet. Este é um dos parâmetros mais importantes, pois afeta diretamente todas as aplicações. Geralmente, cada uma possui os seus próprios requisitos de atraso sofrido na rede. O ITU-T, na recomendação G.114 refere que se o *delay* for mantido abaixo dos 150 milissegundos, a generalidade das aplicações não são significativamente afetadas, e ainda que qualquer rede não deva apresentar um valor de *delay* superior a 400 milissegundos. Quando se fazem testes a uma rede para se avaliar a latência é necessário ter em conta alguns fatores, como por exemplo, a distância. Este é um fator relevante, dado que, quanto mais extensa for a distância do percurso *end-to-end*, maior será a latência. Para que se possa aferir a latência é necessário estabelecer valores de referência para este parâmetro, nomeadamente através das recomendações do ITU e de testes Ping a diversos servidores nacionais e internacionais, bem como uma análise aos dados publicados pela ANACOM [5]. O modelo de avaliação não pode ser extrapolado para situações de comunicações entre dois pontos que distam uma distância superior à do território nacional, uma vez que, como já foi referido, a distância é um fator determinante.

4. Requisitos do Sistema

4.1 Introdução

Este capítulo visa apresentar os requisitos do sistema. Estes correspondem a um conjunto de condições definidas na fase da abordagem ao projeto, por forma a que o sistema alcance os objetivos para os quais é desenvolvido. Os requisitos assumem, desta forma, um papel de delineação das diversas funcionalidades e sobretudo das estratégias e processos a adoptar durante a implementação, apresentando estes uma função determinante. O seu estudo permite antecipar que tipo de restrições, potenciais desafios e obstáculos podem surgir durante a implementação.

Dentro do domínio dos requisitos é necessário distinguir entre os funcionais e os não funcionais. Os primeiros são aqueles que se referem aos processos que permitem a interação direta com o utilizador, em suma, ao nível aplicacional. Os segundos referem-se a processos de domínio técnico que estão ocultos ao utilizador, como protocolos implementados ou mecanismos de segurança. Estes são na sua maioria, processos de níveis inferiores ao aplicacional.

4.2 Aplicação Móvel

A aplicação móvel será desenvolvida para sistemas operativos de *smartphones*. A preferência do Sistema Operativo (SO) recai sobre Android, dada a elevada percentagem de dispositivos no mercado com este SO, cerca de 81% (ver Fig. 12). Pode-se verificar que a partir do fim do ano de 2012 até ao terceiro trimestre de 2013 o Android tem verificado um elevado aumento, na ordem dos 10% na fatia de mercado a nível mundial, que contrasta com um decréscimo proporcional do iOS. No início do ano de 2014 o Android volta a aumentar a sua quota de mercado, havendo previsões de um contínuo aumento até ao fim do ano de 2014 [24][25].

O número de *smartphones* vendidos continua a crescer. Segundo os dados do IDC (*International Data Corporation*), no 2Q14 foram vendidos 301.3 milhões de equipamentos em todo o mundo.

Verificou-se um decréscimo no preço médio dos *smartphones*, rondando agora os $316 (dólares americanos).

A nível global continua-se a assistir à tendência do crescimento do sector de *smartphones* alcançando agora os 63.1% do mercado de telemóveis, um crescimento considerável em comparação com igual período do ano anterior, que se situava nos 50.7% [25].

Ao nível dos fabricantes, a Samsung continua a liderar o sector de vendas de equipamentos com Android, no entanto, os fabricantes chineses assumem-se como grandes concorrentes, uma vez que têm apresentado resultados bastante favoráveis, dado que, Huawei, Lenovo, Coolpad, Xiaomi, ZTE e OPPO encontram-se no top 10 de vendas no primeiro trimestre de 2014 [25].

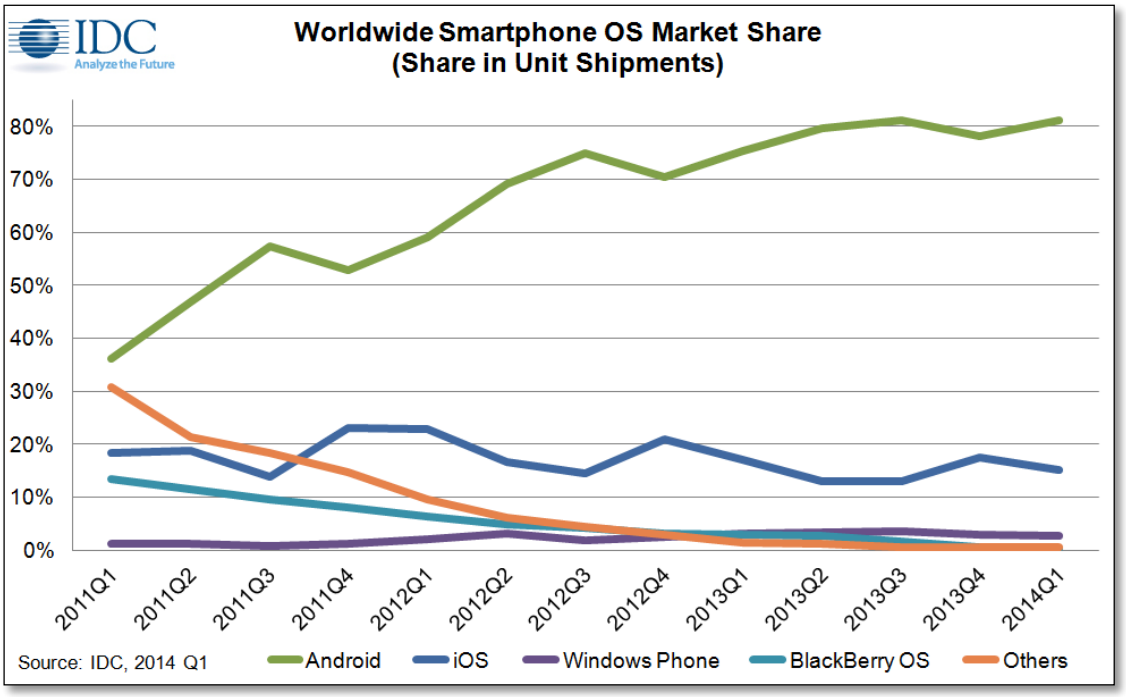


Figura 12 – Cota de Mercado de Sistema Operativos (2011-2014) [25].

A aplicação móvel tem como objetivo efetuar testes à rede, avaliando o conjunto de métricas descritas anteriormente. Esta apresentará duas opções de utilização

distintas, sendo uma vocacionada para efetuar teste instantâneos à rede, a outra tem como objetivo a recolha periódica de dados, para posterior encaminhamento para um serviço que permitira agregar todos os dados recolhidos dos utilizadores para serem analisados e apresentados para consulta através de um serviço *web*.

Para ambas as componentes aplicacionais, as métricas a avaliar durante os testes à rede serão aquelas que foram descritas anteriormente.

4.3 Componente Aplicacional de Testes Instantâneos

A componente aplicacional de testes instantâneos da aplicação móvel tem como objetivo a realização de testes num curto período de tempo, por forma a permitir uma aferição de QoS instantânea. Esta componente visa essencialmente permitir ao utilizador obter uma percepção imediata da qualidade da sua conexão à Internet.

Para esta componente aplicacional definem-se um conjunto restrito de requisitos específicos:

- Interface simples, de fácil utilização;
- Os dados obtidos devem ser apresentados de forma concisa e simplista, possibilitando uma fácil interpretação dos mesmos;
- O processamento da aplicação deve ser expedito, mais concretamente, o período de processamento, desde o instante em que o utilizador inicia o teste até serem apresentados os resultados, deve ser bastante baixo;
- Os resultados dos testes devem ser passíveis de serem guardados, para que o utilizador possa analisá-los mais tarde.

4.4 Componente Aplicacional de Testes Periódicos

A componente da aplicação móvel que efetua testes periódicos tem como objetivo a recolha periódica de dados ao longo de um intervalo de 24h. A aplicação para efetuar os testes gerará tráfego, no entanto deverá ser uma quantidade vestigial em relação ao *plafond* contratado pelo utilizador. No fim desse período os resultados dos testes deverão ser submetidos para o servidor do sistema.

Esta componente aplicacional apresenta um conjunto de requisitos específicos:

- Os dados recolhidos devem ser armazenados no dispositivo até que possam ser submetidos para o servidor;
- Garantir que o tráfego gerado pela aplicação seja reduzido;
- Após a primeira ativação pelo utilizador, a aplicação deve correr em *background*, não interferindo de forma alguma com as atividades das restantes aplicações do dispositivo;
- O período que a aplicação passa em *idle*, entre testes, deve garantir que o consumo de recursos (processamento, memória, entre outros) seja o mais baixo possível, para efeitos de poupança energética;
- A cada teste deve ser associado, caso seja possível e caso o utilizador autorize, informação espacial (coordenadas geográficas) e informação temporal. Os testes geo-referenciados são um dos pontos essenciais deste projeto.

4.5 Requisitos Funcionais

Os requisitos funcionais referem-se àqueles que estão diretamente associados com a interação do utilizador com a aplicação, ou seja, aqueles que se referem sobretudo às capacidades das aplicações que o utilizador usa. Deste modo, são identificados os seguintes requisitos:

- A aplicação móvel apresenta dois tipos de componentes aplicacionais, dos quais o utilizador poderá optar. Uma permite a realização de teste instantâneos, a outra permite a realização de testes periódicos;
- A aplicação de testes periódicos requer que o utilizador se registe no serviço, por forma a que os dados recolhidos pela sua aplicação possam ser consultados através do Cliente *Web*;
- A aplicação de testes periódicos deve possibilitar a associação dos resultados dos testes a referências temporais e espaciais, uma vez que a localização e instante em que ocorrem os testes têm elevada relevância na avaliação da evolução da qualidade de um determinado serviço de comunicações;
- O utilizador, registando-se no serviço, através do cliente *Web* tem a possibilidade de consultar os seus dados, relativamente ao testes que a sua aplicação

de testes periódicos e comparar esses dados com os que são divulgados pelo seu operador de comunicações móveis;

- O cliente *Web* deve apresentar uma interface simples, mas, ao mesmo tempo, através de análises estatísticas efetuadas sobre os dados recolhidos nos testes, devem possibilitar uma análise concisa e detalhada;

O cliente *Web* além de permitir aos utilizadores a consulta dos seus dados pessoais, deve permitir a todos os visitantes do *Web Site* a visualização desses dados, de forma simplificada, não permitindo identificar os utilizadores/autores dos dados. Este serviço de consulta pelo público em geral é uma das componentes essenciais do projeto.

4.6 Requisitos Não Funcionais

Os requisitos não funcionais são aqueles que remetem para um domínio mais técnico, que são essenciais ao projeto mas que não estão diretamente associados à uso dos utilizadores.

Neste contexto são identificados os seguintes requisitos não funcionais:

- Como o sistema terá que lidar com dados pessoais dos utilizadores, uma vez que uma das componentes aplicacionais requere que os mesmos se registem, as questões de segurança, como privacidade e integridade dos dados, são pontos fundamentais e de extrema relevância no contexto do sistema. Este deve implementar mecanismos que garantam que os dados pessoais são mantidos de forma segura, preservando a identidade dos utilizadores, a privacidade e integridade dos seus dados pessoais;
- Ambas as componentes aplicacionais têm como base as métricas definidas anteriormente e serão as únicas que serão usadas para avaliar a qualidade de serviço da conexão à Internet, garantindo que os dados apresentados ao utilizador são coerentes no que se refere a todas a componentes do sistema;
- A aplicação de testes instantâneos deve ter uma capacidade de processar os dados e de efetuar os testes no menor intervalo de tempo possível, para evitar que o utilizador espere demasiado pelos resultados;

- A componente aplicacional de testes periódicos deve ser capaz de identificar se o dispositivo se encontra ligado a um rede Wi-Fi ou a uma rede de dados móvel, pois, os testes só podem ocorrer quando se verificar a segunda situação;
- Esta componente aplicacional deve, ainda, garantir que o seu estado de *idle* entre testes consome o mínimo de energia possível;
- Uma vez que os *plafonds* de dados móveis são geralmente limitados em termos de tráfego, a aplicação de testes periódicos deve usar a menor quantidade de dados possível para efetuar os testes à rede;
- Os testes efectuados serão baseados em dois protocolos: ICMP e HTTP, dado que se apresentam como aqueles que melhor permitem implementar as metodologias de teste que serão usadas para avaliar as métricas descritas.

5. Desenho do Sistema

5.1 Introdução

Este capítulo tem como objetivo apresentar a descrição detalhada de todo o sistema. Pretende explicar individualmente cada um dos componentes e como é que estes se interligam entre si e comunicam.

Numa fase inicial é apresentada a arquitetura geral do sistema desenvolvido. Mostra as principais características e permite de forma geral ter uma noção do sistema no seu todo observando os diferentes componentes. É feita uma análise tecnológica, na medida em que são abordadas algumas questões relativamente às escolhas feitas em termos de ferramentas usadas para desenvolver alguns dos componentes. Isto permite descrever como é que o sistema foi desenvolvido.

Numa fase posterior, depois de ter sido abordada a arquitetura, os componentes e como é que estes comunicam, é apresentada detalhadamente a arquitetura interna de cada um dos componentes e como é que estes operam, explicando todo o processo de desenvolvimento.

5.2 Arquitetura

A arquitetura do sistema resume o modelo criado para o sistema colaborativo de aferição de Qualidade de Serviço em *smartphones* com sistema operativo Android.

A arquitetura desenvolvida contém dois blocos essenciais: a aplicação móvel, denominada de QoS.Test e os servidores (ver Fig. 13).

Existem dois servidores distintos:

- Servidor de testes da Google: Este é o servidor usado para poder efetuar os testes de *download*, latência e perda de pacotes. É um servidor de domínio nacional;

- Servidor QoS.Test: Este é o servidor desenvolvido como parte do projeto. Contém os seus próprios módulos: QoS.Test *Web Service*, QoS.Test *Web Site* e o Servidor de Testes QoS.Test. Este último é usado para se poder efetuar os testes de *upload*.

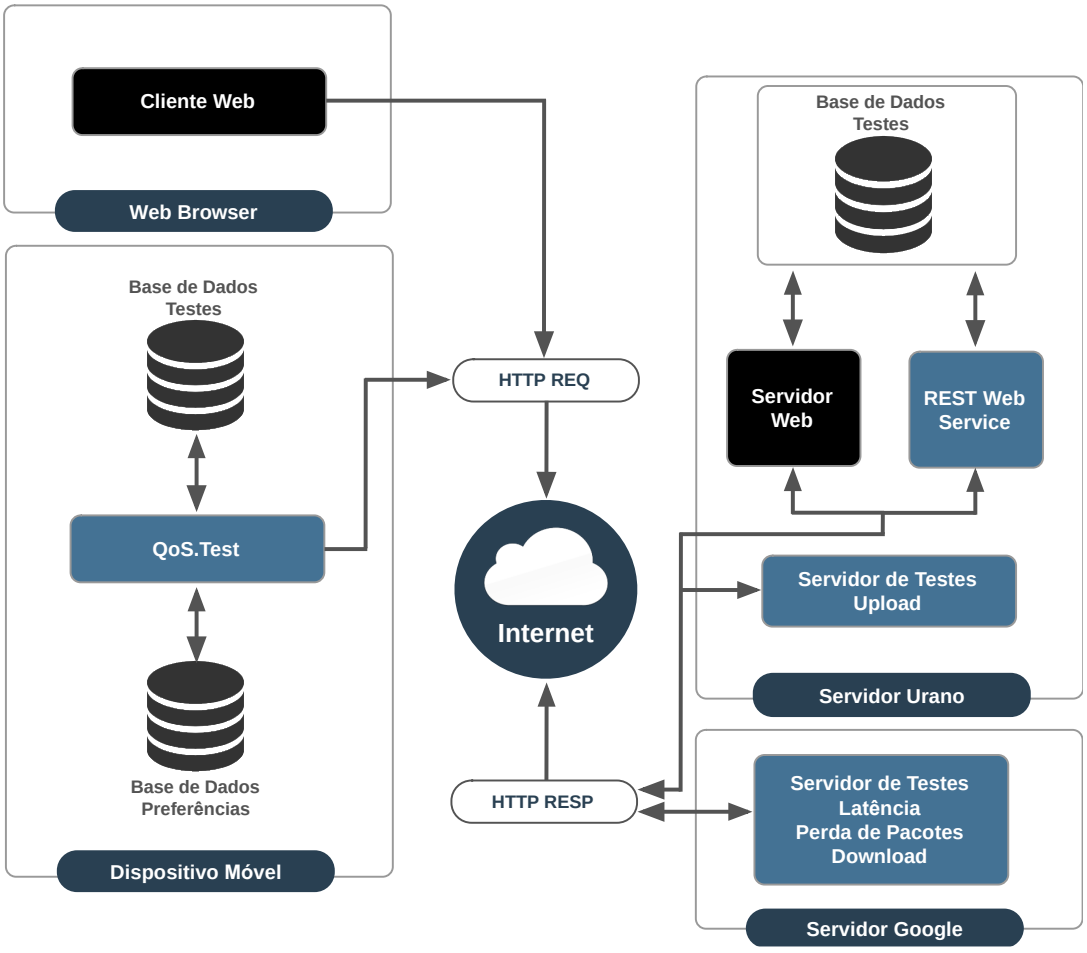


Figura 13 – Arquitetura do Sistema Final.

5.2.1 Componentes do Sistema

5.2.1.1 Aplicação Móvel

A aplicação móvel, tal como havia sido referido assume a função de maior destaque em todo o sistema, tornando-se assim o elemento fundamental.

Atualmente vive-se numa era em que as comunicações móveis estão em franco crescimento, sendo talvez umas das áreas com maior investimento e desenvolvimento a nível tecnológico. Os dispositivos disponíveis no mercado possuem recursos de *hardware* bastante avançados, com diversos periféricos e com capacidade de processamento elevada. Isto, associado ao desenvolvimento de novas redes de comunicação permite aos utilizadores uma experiência completamente distinta do que era habitual à apenas alguns anos.

Após uma análise ao mercado dos dispositivos móveis e dos respectivos sistemas operativos foi escolhido o Android como plataforma alvo de desenvolvimento. O Anexo B introduz a plataforma Android, bem como algumas das suas principais características e elementos fundamentais.

5.2.1.2 Servidores

O conjunto de servidores que servem a aplicação é determinante para a implementação do projeto. Como foi referido existe uma separação em termos de servidores: Servidor QoS.Test e o servidor de testes da Google. O último é utilizado como ferramenta para se efetuarem os testes de *download*, latência e perda de pacotes. O servidor QoS.Test para além de permitir a execução dos testes de *upload*, tem mais dois componentes determinantes: QoS.Test *Web Service* e o Servidor *Web* QoS.Test (*Web Site*).

Sabendo que muito do processamento do sistema passa pelo QoS.Test *Web Service* foi necessário escolher as melhores ferramentas para a sua implementação.

Após uma análise às diferentes ferramentas que poderiam ser utilizadas para desenvolver este componente do servidor QoS.Test, tal como á apresentado no Anexo C, foi adoptada uma solução baseada em REST e JSON.

As capacidades do REST e a simplicidade no acesso aos recursos, o uso dos métodos tradicionais do HTTP, bem como a crescente adoção por diversas entidades desta arquitetura, levaram a que o *Web Service* desenvolvido neste projeto se baseasse neste modelo, enquadrando-se com a tendência no desenvolvimento para plataforma *Web*.

Quanto à linguagem de marcação a escolha recaiu sobre o JSON, dado que os dados trocados entre cliente e servidor são bastante simples, com tipos de dados muito específicos e enquadrados com os tipos de dados do JSON. Ao mesmo tempo a sua capacidade para lidar com *arrays* bem como a simplicidade na manipulação de dados e o abrangente suporte para esta linguagem de marcação levaram a que fosse a opção escolhida para a troca de dados com a aplicação móvel. No Anexo C é feita uma análise ao JSON e ao XML, bem como uma comparação em termos de potencialidades destas duas tecnologias.

Ao nível do desenvolvimento interno do servidor, o PHP foi escolhido como plataforma de desenvolvimento, uma vez que satisfaz todas as necessidades no domínio do projeto. A sua simplicidade e suporte multiplataforma tornam esta linguagem bastante adequada, tal como se pode analisar no Anexo C.

5.2.2 Interfaces

A aplicação faz uso de um diverso conjunto de interfaces. Estas permitem obter dados e efetuar a comunicação com os restantes componentes do sistema.

As interfaces que a aplicação usa são as seguintes:

- Interface de Rede Móvel: Usada para a execução dos testes de QoS segundo as métricas definidas, serviços de localização do utilizador e troca de dados com o servidor.
- Interface de WiFi: Usada para obter localização do utilizador e troca de dados com o servidor.
- Interface GPS: Usada para determinar a localização do utilizador.

A comunicação efectuada entre o Dispositivo Móvel e o Servidor é feita sobre a Internet, com recurso à interface de Rede Móvel ou de WiFi, dependendo da disponibilidade de ambas no momento da interação.

A localização recorre a três interfaces, tendo como prioridade o uso do WiFi, depois a Rede Móvel e por último, apenas em caso de necessidade e de indisponibilidade das anteriores, o GPS.

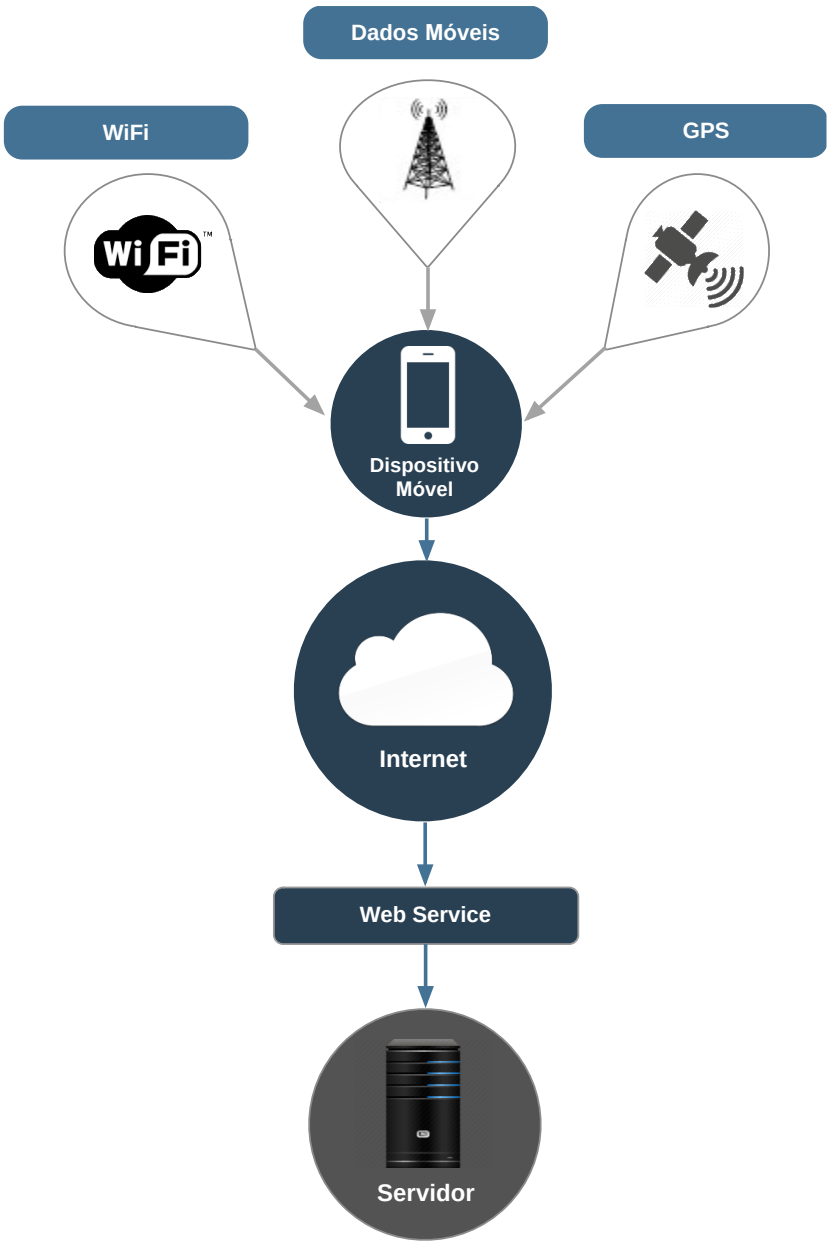


Figura 14 – Interfaces do dispositivo móvel usadas.

5.3 Aplicação Móvel

5.3.1 Introdução

A aplicação móvel desenvolvida assume o nome de QoS.Test. A seguir, neste documento, será explicada toda a arquitetura da aplicação, sendo identificados e analisados detalhadamente os diferentes módulos e como é que estes interagem.

As principais características da aplicação móvel são:

- Plataforma: Android;
- Android SDK: Versão 4.0 (API level 19);
- Linguagem de desenvolvimento: Java;
- IDE: Eclispe (Mac OS X 10.7.5);
- Idioma: Inglês.

Durante o processo de desenvolvimento foi tido em atenção a grande gama de dispositivos disponíveis no mercado com este sistema operativo, nomeadamente, em termos de resolução do ecrã, entre outras características, por forma a que a aplicação pudesse ser usada pelo maior número possível de utilizadores. Contudo, o equipamento usado para testes e *debug* foi um Samsung Galaxy Note 3 SM-N9005, logo a aplicação está aprimorada para este dispositivo quer em termos de capacidade de processamento quer em termos de resolução e apresentação da interface gráfica.

A escolha sobre a versão 4.0 e superiores deve-se sobretudo às estatísticas apresentadas anteriormente, uma vez que poderia ser alcançada uma quota de cerca de 73% dos dispositivos no mercado.

Foi desenvolvida em inglês, com a intenção de no futuro, caso seja possível, tornar a aplicação disponível e totalmente operacional em qualquer país. A aplicação QoS.Test foi desenvolvida tendo em conta as características da rede móvel em Portugal.

5.3.2 Modelo de funcionamento

A aplicação QoS.Test tem como principal objetivo a execução de testes de Qualidade de Serviço à rede móvel dos utilizadores. As métricas avaliadas são: latência, perda de pacotes , *download* e *upload*.

Utilizando estas métricas estabeleceram-se os testes a ser efectuados, tanto na componente de testes instantâneos como de testes periódicos. Estas duas componentes são explicadas a seguir.

5.3.2.1 Testes Instantâneos

Com as métricas estabelecidas, escolheram-se os tipos de testes que o utilizador pode efetuar:

- Teste de latência;
- Teste de perda de pacotes;
- Teste de *download*;
- Teste de *upload*;
- *Full QoS Test*: Conjugação de todos os anteriores.

Estes são os tipos de testes que o utilizador pode efetuar a partir do menu principal da aplicação QoS.Test (ver Fig. 15). Esta é a componente de testes instantâneos definida nos requisitos.

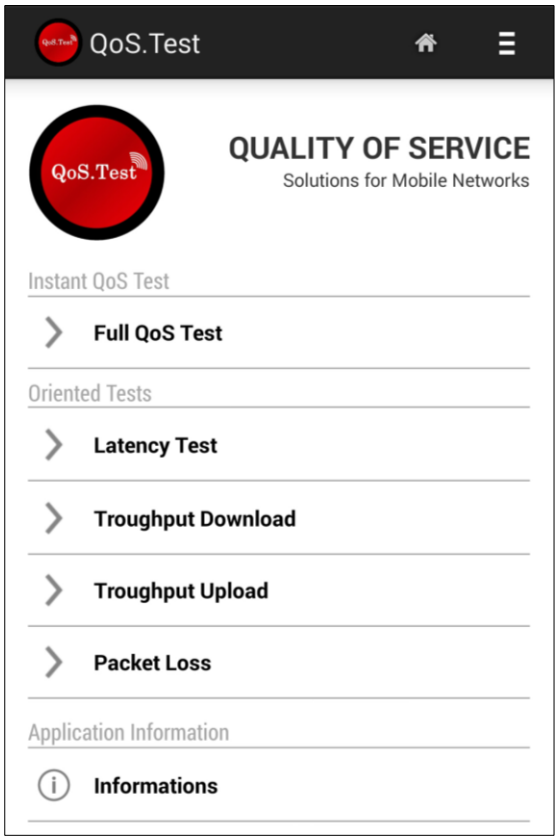


Figura 15 - Menu principal da aplicação QoS.Test

Os resultados dos testes são posteriormente apresentados num ecrã com duas secções navegadas através do movimento horizontal, denominado de *swipe* (ver Fig. 16):

- Informações e resultados quantitativos e qualitativos: Nesta secção são apresentados dados específicos sobre o teste realizado. Contém valores numéricos, ao mesmo tempo apresenta ao utilizador um valor qualitativo sobre a qualidade do teste efectuado. Os resultados qualitativos possuem cinco níveis distintos: *Excellent* (Excelente), *Very Good* (Muito Bom), *Good* (Bom), *Acceptable* (Aceitável) e *Bad* (Mau). A tradução dos valores quantitativos em valores qualitativos é feita segundo um conjunto de modelos desenvolvidos para este fim, apresentados adiante no presente documento. Independentemente do teste é sempre apresentado o tipo de rede e o nome do ISP;
- Localização do dispositivo móvel no instante do teste: Nesta secção, recorrendo aos mapas da Google é marcado o local onde ocorreu o teste. São adicionalmente apresentadas as coordenadas geográficas (latitude e longitude).

Para cada um dos testes que estão ao dispor do utilizador a interface de observação dos resultados é igual à apresentada na Fig. 16.

Para cada teste existe um conjunto específico de resultados quantitativos:

- Latência: *Round Trip Time* médio (em milissegundos), o *Round Trip Time* máximo (em milissegundos), *Round Trip Time* mínimo (em milissegundos) e o desvio padrão (em milissegundos);
- Perda de Pacotes: Pacotes enviados, pacotes recebidos, percentagem de perda de pacotes e o tempo de duração do teste (em milissegundos);
- *Download*: Débito (em bits/s), duração do teste (em milissegundos) e a quantidade total de dados recebidos (em bits);
- *Upload*: Débito (em bits/s), duração do teste (em milissegundos) e a quantidade total de dados enviados (em bits).

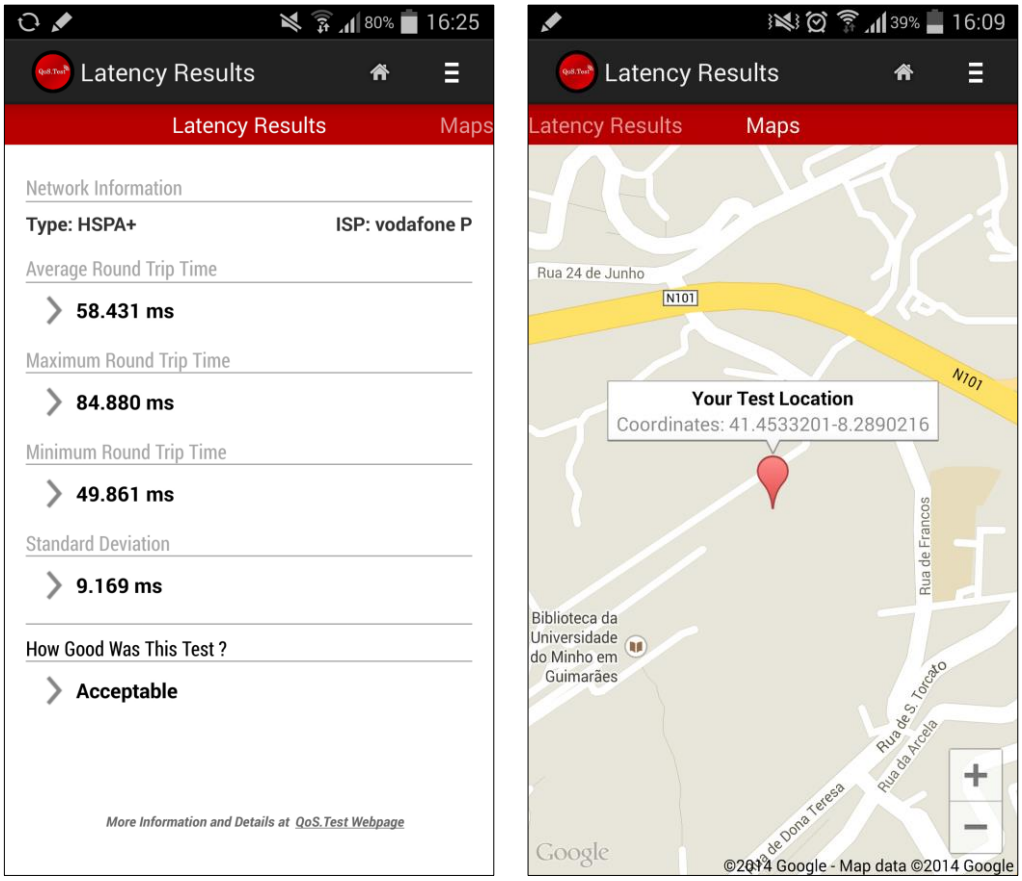


Figura 16 - Ecrãs com resultados dos testes (teste de latência).

Os dados gerados através destes testes podem ser enviados para o servidor e consequentemente para a base de dados. Para tal, é necessário que o utilizador se registe no serviço e efetue o *login* (ver Fig. 17). Depois da operação de *login*, tem um menu com diferentes possibilidades de escolha e preferências da aplicação (Fig. 18).

A operação de registo requer quatro parâmetros:

- Primeiro nome;
- Último nome;
- *Email*;
- *Password*.

A operação de *login* é efectuada usando apenas o *email* e a *password*.

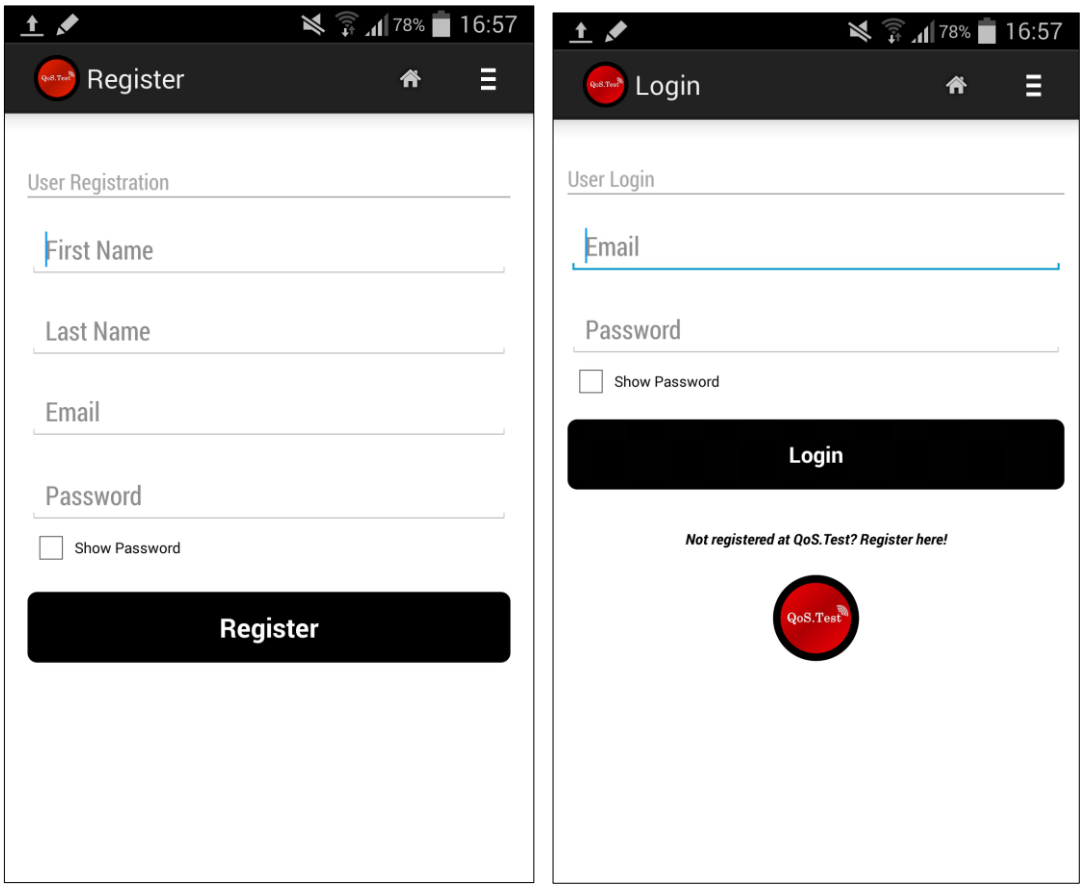


Figura 17 - Ecrãs de registo (esquerda) e *login* (direita).

5.3.2.2 Testes Periódicos

Uma das componentes da aplicação é a possibilidade de serem efectuados testes periódicos. Estes variam entre um período de 18 a 30 horas, por forma a que seja realizado aproximadamente um teste por dia, com um factor de aleatoriedade para elevar a relevância dos dados recolhidos.

Esta componente pode ser ativada no menu do utilizador, que só está acessível se este fizer *login* na aplicação (ver Fig. 18). É uma componente exclusiva para utilizadores registados no sistema. Os resultados destes testes são forçosamente enviados para o servidor, uma vez que a base de dados interna é limitada, tal como é explicado a seguir.

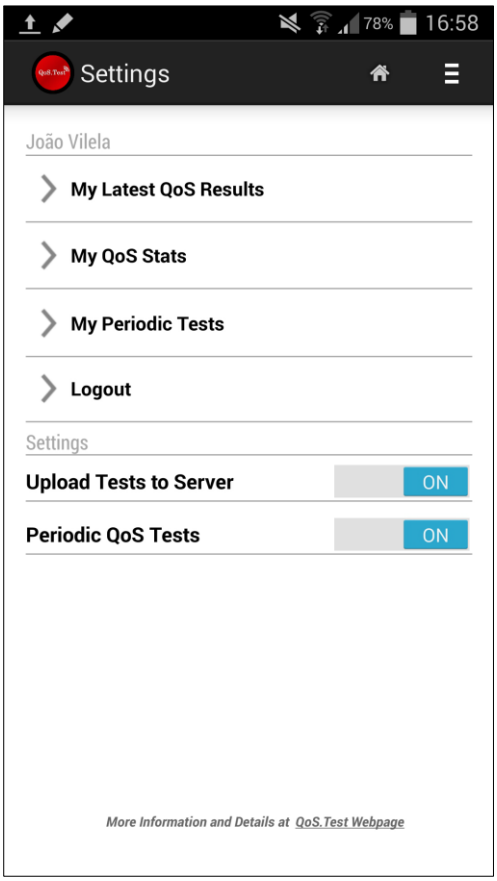


Figura 18 – Menu pessoal do utilizador após o *login*.

5.3.2.3 Armazenamento dos resultados

Se um utilizador se registar e assim o pretender, pode autorizar a que os dados sejam enviados para o servidor e consequentemente para a base de dados. Contudo, independentemente desta opção, utilizadores com *login* efectuado ou não, têm a possibilidade de armazenar os resultados dos testes localmente, até a um máximo de cinquenta. Após este valor os resultados mais antigos vão sendo sobrepostos pelos mais recentes. O armazenamento dos resultados dos testes realizados é efectuado numa base de dados local. Esta armazena os dados quer estes sejam provenientes dos testes instantâneos ou dos testes periódicos. O acesso é feito através do menu pessoal do utilizador. No caso dos utilizadores sem o *login* efectuado, não é possível ver dados relativos a testes periódicos, apenas dos testes instantâneos.

Independentemente do tipo de resultados que o utilizador pretende visualizar, estes são sempre apresentados num ecrã com uma lista com os mesmos. No caso do testes instantâneos, visto que existem diferentes testes que podem ser realizados, o ecrã possui uma secção para cada tipo de teste. A navegação é feita com movimentos horizontais entre elas (*swipe*) (ver Fig. 19).

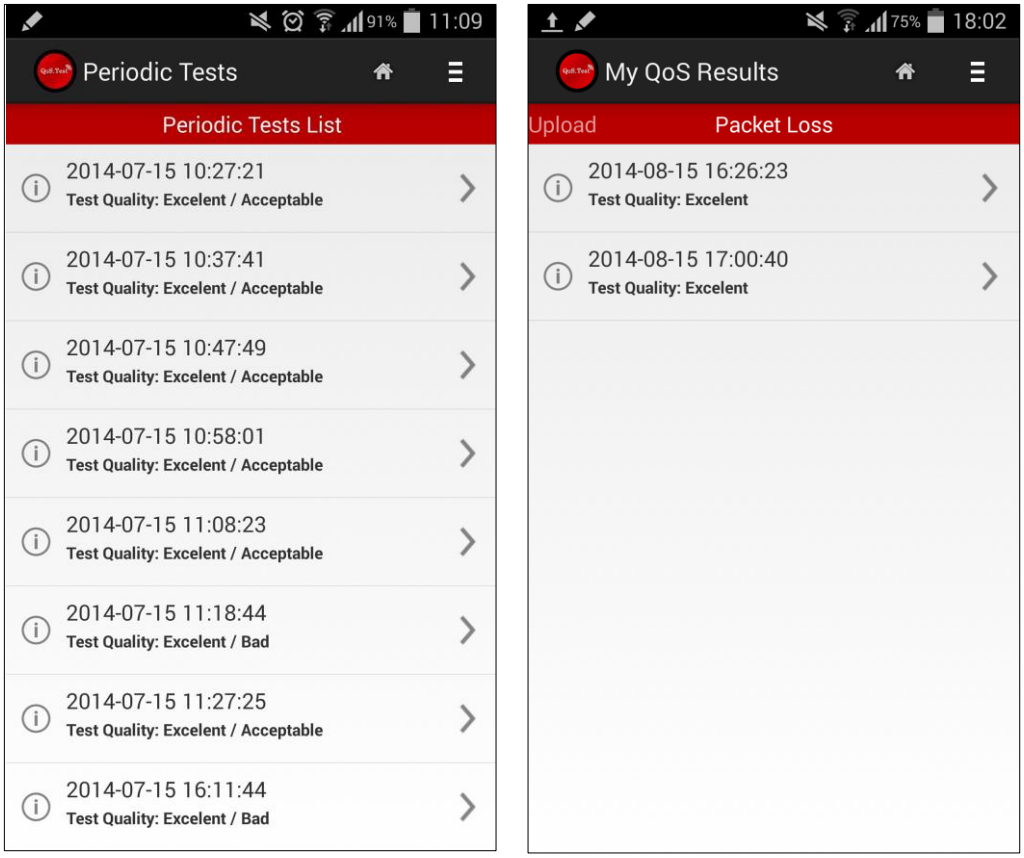


Figura 19 - Ecrãs de resultados de testes periódicos (esquerda) e testes instantâneos (direita).

Se o utilizador pretender ver com mais detalhe os resultados dos testes apresentados, pode clicar em cada elemento da lista e será redirecionado para o ecrã com os resultados quantitativos e qualitativos, bem como com a localização, iguais aos apresentados após um teste demonstrados na Fig. 16.

5.3.2.4 Estatísticas de QoS

O utilizador, com os dados recolhidos pela sua aplicação, pode obter uma análise estatística para cada uma das métricas avaliadas. Os resultados são acompanhados de dois gráficos, um de linhas e um circular. O primeiro agrega os últimos dez resultados e o último oferece uma estatística qualitativa sobre todos os dados.

As análises estatísticas estão apenas disponíveis para utilizadores cujos dados foram enviados para o servidor. Não é possível apresentar dados estatísticos com os valores armazenados na base de dados local da aplicação.

Os resultados estatísticos são apresentados num ecrã com duas secções. A primeira contém o gráficos e a segunda contém os valores numéricos e também qualitativos (ver Fig. 20).

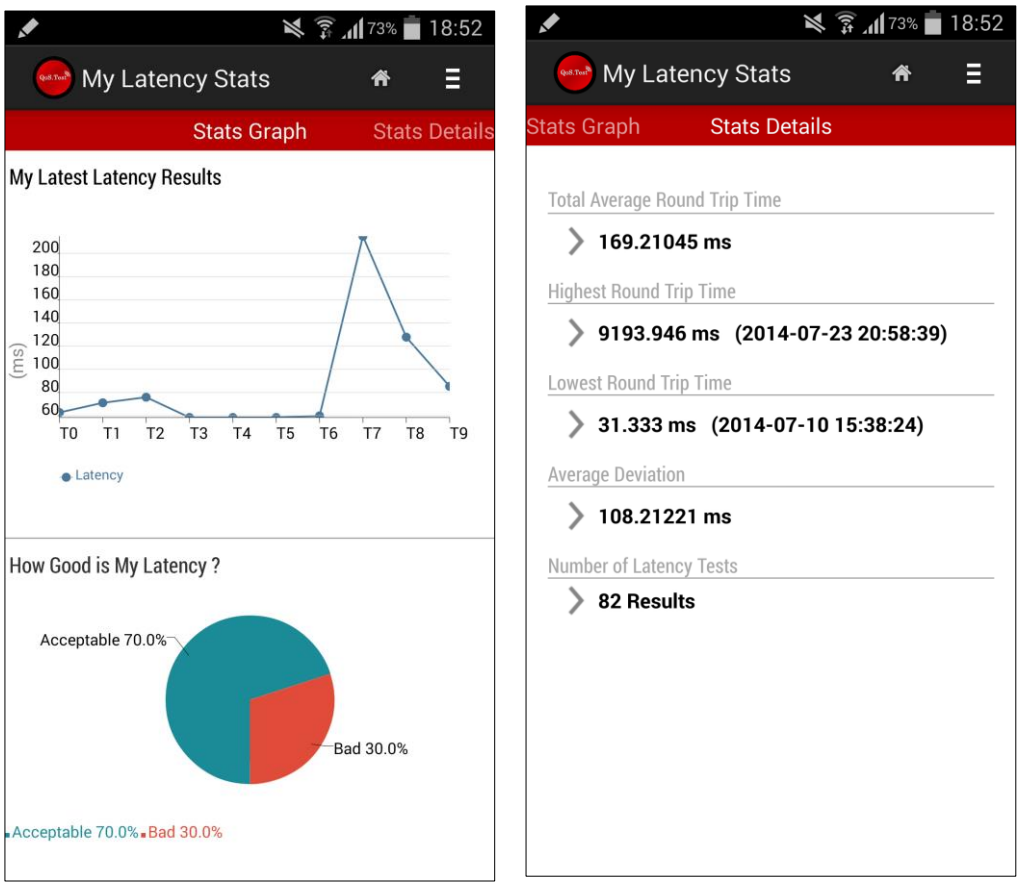


Figura 20 - Resultados estatísticos, gráficos (esquerda) e valores (direita).

Uma particularidade da secção com os gráficos é a possibilidade de o utilizador rodar o equipamento e o *layout* passa a horizontal, podendo ver exclusivamente o gráfico de linhas com mais detalhe (ver Fig. 21). O gráfico circular fica oculto no *layout* na horizontal.

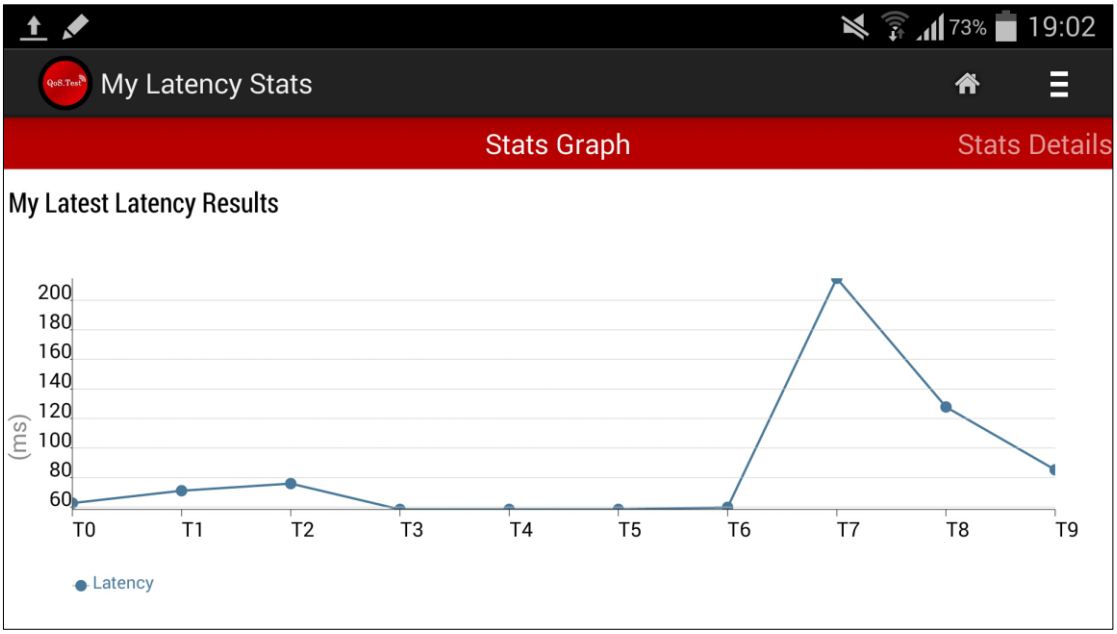


Figura 21 - Gráfico de linhas em plano horizontal.

Para cada um dos tipos de testes existe um conjunto distinto de análises estatísticas apresentadas:

- Latência: Média total de *Round Trip Time* (em milissegundos), o *Round Trip Time* máximo registado (em milissegundos), *Round Trip Time* mínimo registado (em milissegundos), média total dos valores do desvio padrão (em milissegundos) e o número total de resultados de latência;
- Perda de Pacotes: média total de percentagem de perda de pacotes, duração média de teste (em milissegundos), percentagem máxima de perda de pacotes registada, percentagem mínima de perda de pacotes registada e o número total de resultados de perda de pacotes;
- *Download*: Débito médio (em bits/s), débito máximo registado (em bits/s), débito mínimo registado (em bits/s), a quantidade total de dados recebidos (em bits) e o número total de resultados de *download*;

- *Upload*: Débito médio (em bits/s), débito máximo registado (em bits/s), débito mínimo registado (em bits/s), a quantidade total de dados enviados (em bits) e o número total de resultados de *upload*;

5.3.3 Arquitetura Interna

A arquitetura da aplicação foi desenvolvida segundo uma estrutura simples, baseada em níveis, tendo estes uma organização assente em módulos distintos, com características e objetivos bem definidos. Os módulos são independentes e podem ser alterados sem modificar a estrutura e os processos de interação entre os restantes.

A aplicação QoS.Test apresenta uma arquitetura com dois níveis:

- *Graphical User Interface and High Level Functions*: Este nível inclui toda a interface do utilizador bem como algumas funções e operações de alto nível.
- *Management Services*: Este nível é responsável pela execução da maioria das operações da aplicação. Está estruturado em três módulos distintos:
    - *Application Services*: Módulo responsável por serviços de apoio à aplicação, nomeadamente de cálculos de Qualidade de Serviço, bem como gestão de processos em *background*;
    - *Data and Communication Services*: Este módulo lida com os dados da aplicação, mais concretamente, gere as duas bases de dados internas e é responsável por efetuar a comunicação com o *Web Service*;
    - *Location and State Services*: Módulo responsável pelos serviços de localização e de conectividade das interfaces.

Cada um dos três módulos descritos possui sub-módulos, mais concretamente classes Java, que contêm um conjunto de métodos essenciais à execução da aplicação. Dessas classes, algumas recorrem a API's Android específicas para determinadas tarefas.

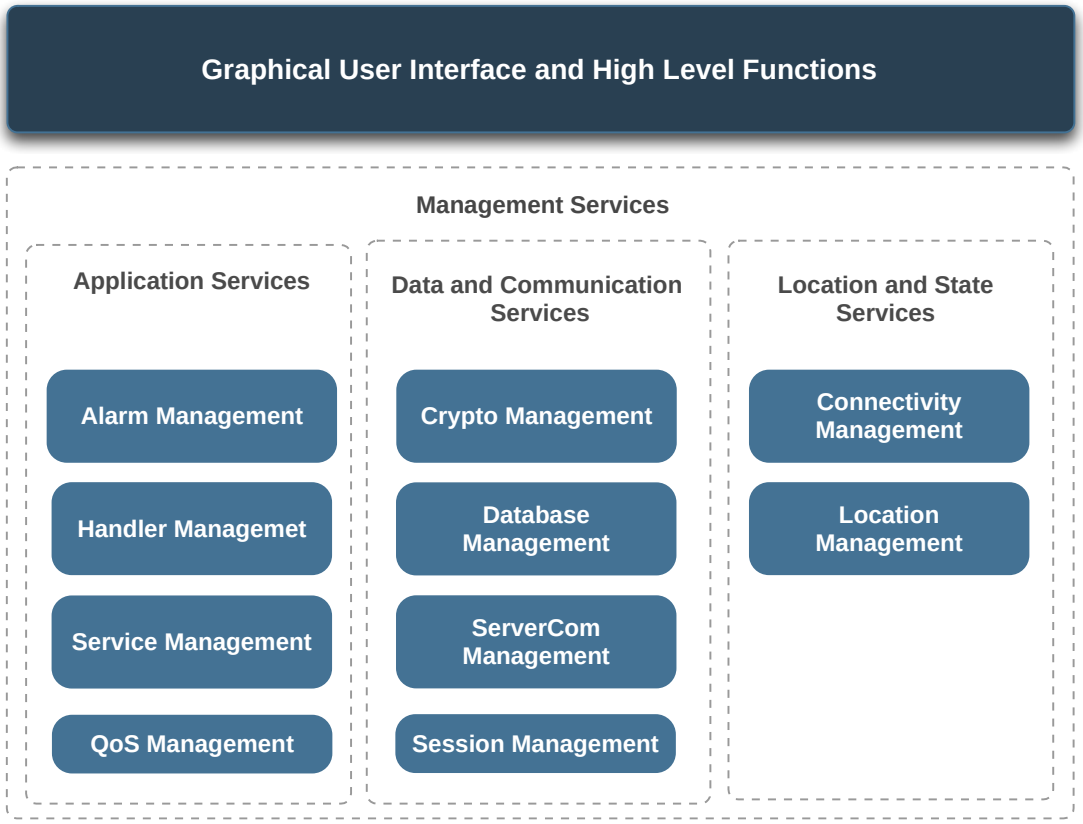


Figura 22 - Arquitetura da aplicação móvel QoS.Test.

O sub-módulo *Connectivity Management* permite a gestão de interfaces de rede, como o WiFi e dados móveis. Permite aceder e alterar o estado dessas interfaces bem como obter informação detalhada. Na Tab. 2 são descritas as classes das API's Android utilizadas neste sub-módulo.

Tabela 2 - API's Android usadas para gestão de interfaces de rede (Connectivity Management).

API	Classe	Descrição
android.net	**ConnectivityManager**	Classe que responde a pedidos sobre estado de conexão da rede. Uma das principais responsabilidades desta classe é monitorizar conexões de rede (WiFi, GPRS, UMTS, etc.)
android.net	**NetworkInfo**	Permite obter informação sobre uma conexão de rede específica, como o nome da rede, tipo de rede (Wi-Fi ou móvel), se está em roaming, etc.
android.net.wifi	**WiFIManager**	Esta classe providencia mecanismos primários para a gestão de todos os aspetos da conexão WiFi.
android.net	**TelephonyManager**	Providencia acesso a informação sobre os serviços telefónicos do dispositivo. Permite gerir e obter informação sobre a conexão de rede móvel.

O sub-módulo *Location Management*, responsável pelos serviços de localização, recorre à API android.location para obter as coordenadas geográficas. As classes desta API permitem conseguir dados de localização através das interfaces de rede (WiFi e móvel) e GPS (ver Tab. 3).

Tabela 3 - API's Android usadas para gestão da localização (Location Management).

API	Classe	Descrição
android.location	**Location**	Classe de dados que representa uma localização geográfica, que consiste em coordenadas geográficas, altitude, velocidade, etc.
android.location	**LocationManager**	Classe que providencia o acesso aos serviços do sistema de localização. Permite à aplicação obter *updates* periódicos da localização do dispositivo. A informação pode ser obtida através das interfaces de rede ou GPS.

Como já foi referido anteriormente, a aplicação possui uma base de dados local, criada em SQLite. Esta permite armazenar um conjunto de resultados de testes efectuados. Para a implementação da mesma recorreu-se à API android.database.sqlite, descrita na Tab. 4.

Tabela 4 - API usada para a gestão da base de dados local (Database Management).

API	Classe	Descrição
android.database.sqlite	**SQLiteDatabase**	Permite que a aplicação faça a gestão de uma base de dados privada.

O sub-módulo *Session Management* possui um conjunto de métodos que permitem criar uma base de dados. Permite que a aplicação guarde um conjunto de informação sobre o utilizador, como o *email*, nome, e preferências de utilização (ativar ou não testes periódicos ou enviar dados para o servidor). Estes dados estão sempre ao dispor da aplicação. É uma base de dados de preferências e informação pessoal.

Tabela 5 - API usada para gestão de preferências do utilizador (Session Management).

API	Classe	Descrição
android.content	**SharedPreferences.Editor**	Classe que permite aceder e modificar conjunto de dados de preferências.

O sub-módulo *Alarm Management* tem como principal responsabilidade a gestão de um alarme, que permite que a aplicação execute uma tarefa, mesmo quando o dispositivo se encontra num estado de repouso. É definido um instante em que o dispositivo acorda e executa essa tarefa. A API e as classes que permitem a implementação deste serviço estão descritas na Tab. 6.

Tabela 6 - API's usadas para gestão de alarmes e serviços (Alarm Management).

API	Classe	Descrição
android.app	**AlarmManager**	Providencia acesso aos alarmes do sistema. Permite que se possa configurar a aplicação para executar automaticamente no futuro.
android.app	**PendingIntent**	Descrição de um *Intent* que contém uma ação que será executada no futuro.
android.app	**Service**	É um componente da aplicação que representa uma operação de execução de longa duração sem interação com o utilizador, ou uma função suplementar para outras aplicações usarem.

Quanto à componente de construção de gráficos que acompanham as análises estatísticas, foram construídos usando a libraria AChartEngine. Esta permite construir um vasto conjunto de gráficos focados para aplicações Android. É bastante versátil e bem documentada permitindo uma implementação simples, ao mesmo tempo possibilitando uma grande capacidade de customização.

5.3.3.1 Testes de QoS

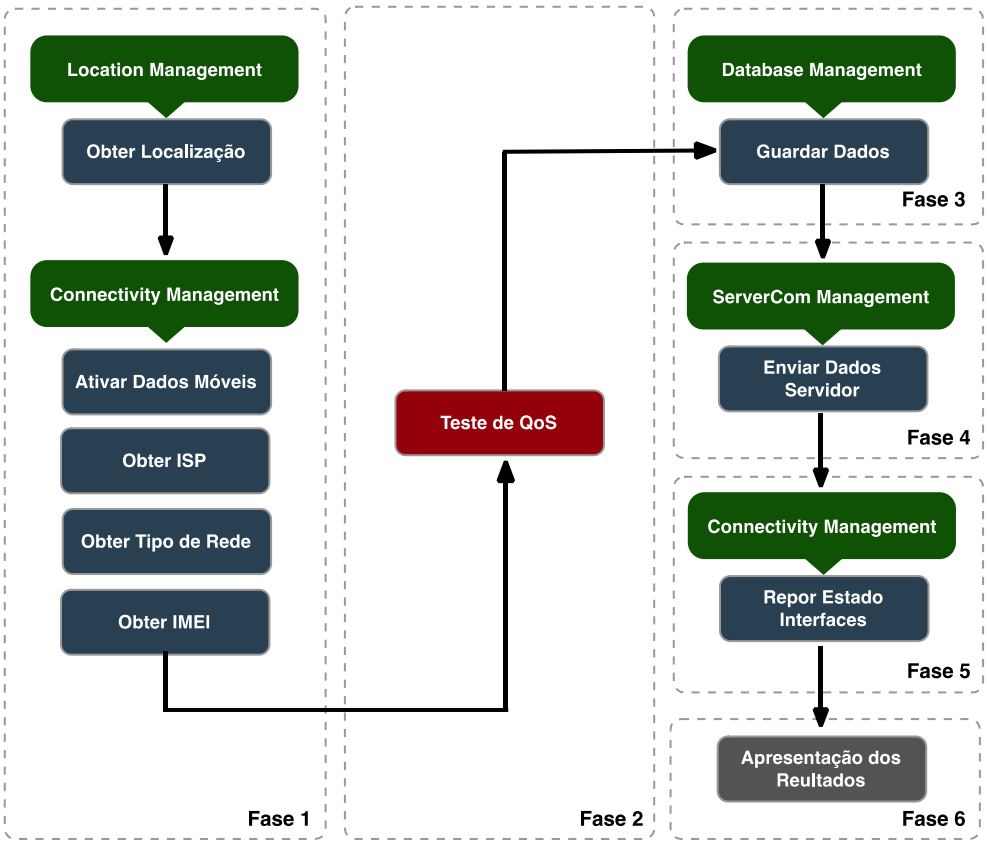


Figura 23 - Diagrama ilustrando o procedimento de um teste de QoS.

A base da aplicação é a execução de testes de Qualidade de Serviço à rede móvel. As métricas usadas para este efeito foram já descritas. Para cada um dos testes a executar o processamento por parte da aplicação é sempre igual.

A execução de um teste, independentemente de qual, ou quais as métricas a ser avaliadas, passa por seis fases distintas (ver Fig. 23):

1) Na primeira fase é necessário verificar o estado atual das interfaces, alterar se for caso disso, e ainda obter a localização do utilizador. Este processo inicia-se verificando se a interface de WiFi está ativa. Se assim for, tenta-se obter a localização do utilizador e logo após, desliga-se esta interface. Caso não esteja ativa, passa-se diretamente para a verificação do estado da interface de dados móveis. Esta tem que estar obrigatoriamente ativa durante o teste, visto que o objetivo é avaliar a rede móvel. Se estiver desligada é ligada automaticamente. Caso anteriormente não tenha sido possível obter a localização do utilizador, tenta-se novamente através da

rede móvel. Se ainda não for obtida uma localização e o GPS estiver desativado é solicitado ao utilizador que o ligue, uma vez que o Android impede que esta interface possa ser ativada automaticamente através do código. Após ativar o GPS, o utilizador tem que voltar a tentar executar o teste. Tendo o GPS ativo, se ainda não for possível obter a localização, é impossível ao utilizador executar qualquer teste. Caso contrário, após ser obtida a localização e de as interfaces estarem no estado correto para que se efetue o teste é necessário obter o nome do ISP, tipo de rede (HSDPA, HSPA+, UMTS, etc) e o IMEI do dispositivo. Para obter os dados da localização (coordenadas geográficas) a aplicação recorre aos métodos do sub-módulo *Location Management*. Para obter os restantes dados, o sub-módulo utilizado é o *Connectivity Management*. Sintetizando, no fim desta fase é necessário obter a localização do utilizador, nome do ISP, tipo de rede e o IMEI. É necessário que a interface de dados móveis esteja ativa e a de WiFi desligada. A Fig. 24 ilustra o processo de execução desta fase.

2) Na segunda fase é iniciado o teste de QoS, independentemente da métrica a avaliar. O teste dará origem aos resultados apresentados ao utilizador. Posteriormente será descrito o processo de cada teste individualmente.

3) Armazenamento de dados localmente. Os dados são armazenados obrigatoriamente na base de dados local da aplicação, para poderem posteriormente ser acedidos pelo utilizador. Este processo é gerido pelo sub-módulo *Database Management*.

4) Armazenamento dos dados no servidor. Como já havia sido referido, o utilizador no menu das suas preferências pode permitir que os dados sejam enviados para o servidor para serem posteriormente armazenados na base de dados. Sendo este o caso, a aplicação tenta estabelecer uma conexão por WiFi (*timeout* de 6 segundos) para tentar enviar os resultados sem consumir tráfego dos dados móveis. Caso não seja possível estabelecer uma conexão, estes são enviados pela rede móvel. Mais uma vez, a manipulação do estado das interfaces é feito com recurso ao sub-módulo *Connectivity Management*. O sub-módulo responsável pelo envio dos dados para o servidor é o *ServerCom Management*.

5) Na quinta fase a aplicação, recorrendo ao sub-módulo *Connectivity Management*, altera o estado das interfaces para o estado inicial, mais concretamente, imediatamente antes do teste ter sido iniciado. A alteração do estado das interfaces é oculta. O utilizador, no início de cada teste, é avisado que o estado das interfaces vai ser alterado. Se aceitar, dá-se início ao teste.

6) Na última fase, a aplicação apresenta os resultados do teste efetuado ao utilizador.

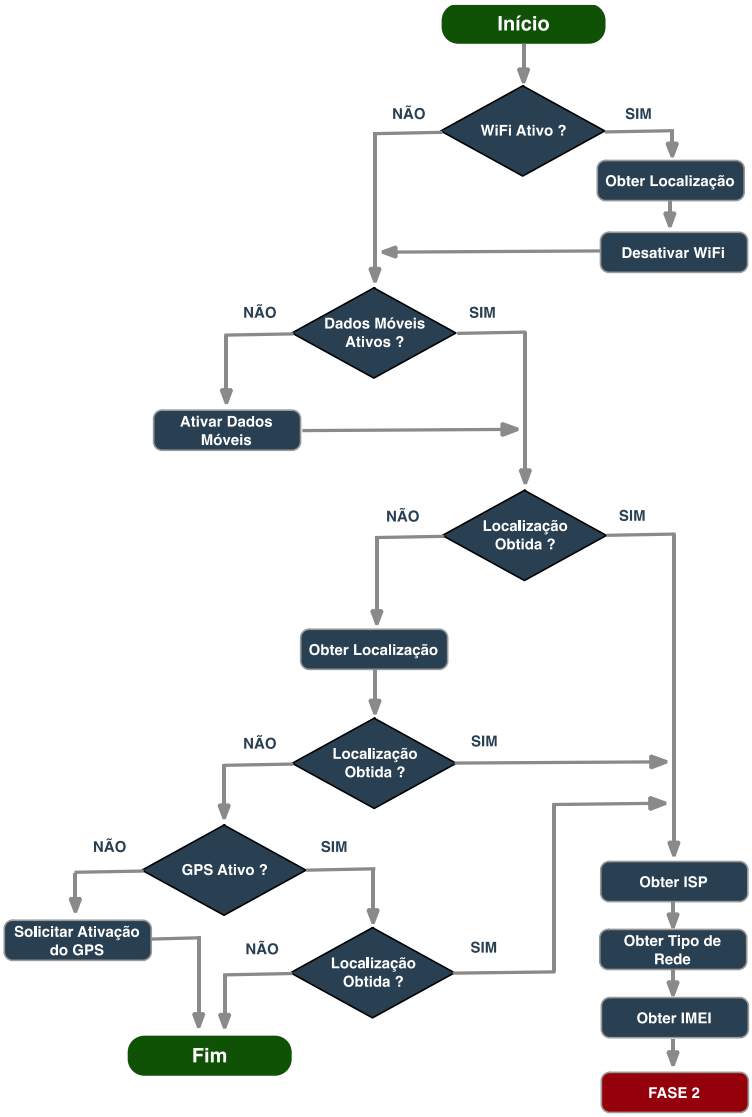


Figura 24 - Fluxograma ilustrando a Fase 1.

O procedimento descrito nestas seis fases distintas é o mesmo para qualquer um dos testes de QoS que a aplicação executa. A execução dos testes e a apresentação de resultados ao utilizador é feita no nível mais alto da aplicação, *Graphical User Interface and High Level Functions*.

A segunda fase do procedimento correspondente à execução do teste de QoS é analisada com mais detalhe na próxima secção. Individualmente, serão detalhados os processos envolvidos em cada um dos diferentes testes.

5.3.3.2 Teste de QoS – Latência

O objetivo deste teste é obter valores relacionados com a latência. Esta designa o tempo gasto por um pacote de informação para percorrer o caminho desde a sua origem até ao seu destino.

Para a execução deste teste foi escolhido o Ping. Este implementa o protocolo ICMP e é bastante simples, tanto ao nível de implementação como de interpretação de resultados. O Ping recorrendo ao protocolo ICMP envia pacotes (*echo request*) para um servidor e espera por uma resposta. O tempo decorrido entre estes dois eventos permite obter o valor de *Round Trip Time*.

A utilização do Ping para o envio de uma sequência de pacotes permite obter informação importante sobre a latência. Com a utilização do teste Ping obtém-se os seguintes valores:

- *Average Round Trip Time*: Tempo médio obtido no conjunto de pacotes trocados com o servidor;
- *Maximum Round Trip Time*: Tempo máximo medido.
- *Minimum Round Trip Time*: Tempo mínimo medido.
- *Standard Deviation*: Valor do desvio padrão.

No domínio da aplicação, em cada teste de latência são trocados dez pacotes com o servidor de testes.

O servidor de testes usado é um servidor da Google com o endereço 173.194.41.215. Este servidor está em domínio nacional. A escolha de um servidor gerido por terceiros foi sobretudo na tentativa de tornar a aplicação o mais universal possível. A aplicação foi desenvolvida para funcionar em pleno em Portugal, mas o

objetivo era poder torná-la funcional globalmente. Para tal seria necessário recorrer a um conjunto de máquinas dispersas pelos diferentes países, para que os testes fossem feitos a servidores o mais próximo possível para minimizar efeitos de congestionamento de tráfego, distância, etc. O uso de servidores da Google tornou-se a solução, na medida em que esta empresa possui equipamento em todos os continentes e em grande parte dos países a nível mundial. Recorrer a estes equipamentos foi a abordagem adoptada, que para além de ser uma solução viável e homogénea em praticamente todos os países, não possui qualquer custo. Em suma, no caso de uma expansão da aplicação, a anatomia do testes mantem-se, podendo usar os servidores da Google mais próximos do utilizador no instante do teste, sabendo que o tempo de resposta e a capacidade de processamento em cada servidor será semelhante independentemente de onde estão localizados.

No fim da execução de um teste de latência, para além dos valores obtidos através do teste Ping, a aplicação recolhe um conjunto adicional de dados, dos quais quatro foram obtidos na fase que antecede a execução do teste:

- ***Timestamp***: É um parâmetro que contém a data e hora em que o teste se realiza. O formato é: Ano-Mês-Dia Horas:Minutos:Segundos. Este, para além dos valores de latência, é o único parâmetro que é obtido nesta fase.
- **Localização Geográfica**: Coordenadas geográficas obtidas na fase que antecede execução do teste de latência.
- **Tipo de rede**: Tipo de rede usada no momento do teste, como por exemplo: HSDPA, HSPA+, UMTS, etc.
- **Nome do ISP**: Nome do operador ao qual o utilizador está associado.
- **IMEI**: Este parâmetro é usado como identificador do dispositivo na base de dados.

Este conjunto de dados é associado ao teste e são estes que são armazenados na base de dados local e/ou na base de dados do servidor.

Os dados representados na Fig. 25 resumem o conjunto total recolhido pela aplicação até ao instante imediatamente após a execução de um teste de latência. São representados os dados obtidos na primeira fase e os dados obtidos na segunda fase. Este conjunto total de dados é o que será usado nas fases 3 e 4 do teste.

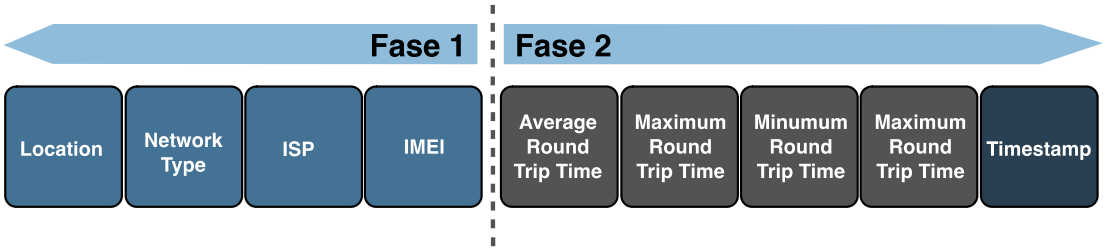


Figura 25 - Conjunto de dados recolhidos pela aplicação após um teste de latência.

O diagrama da Fig. 26 apresenta o modelo usado para executar o teste de latência. São lidos os dados obtidos na fase 1, a esses dados é adicionado o *timestamp* e dá-se inicio ao teste Ping, que permite obter os valores de RTT médio, RTT máximo, RTT mínimo e Desvio Padrão. Após estes dados serem obtidos são guardados na base de dados local da aplicação. O sub-módulo *Database Management* é responsável por esta operação. Como foi referido, os dados podem ou não ser enviados para o servidor. Caso os dados sejam enviados para o servidor o processo é executado pelo sub-módulo *ServerCom Management*.

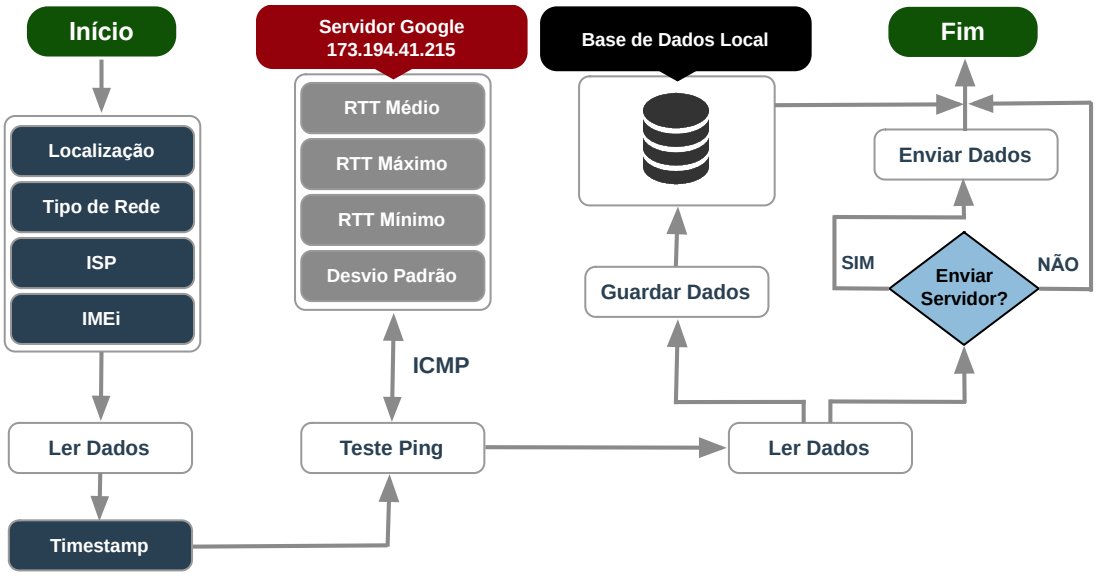


Figura 26 - Modelo de execução de um teste de latência.

Tabela 7 - Parâmetros do teste de latência.

Parâmetro	Descrição
Protocolo	ICMP
Pacotes Enviados	10
Duração do teste	Aprox. 16 segundos
Valores Obtidos	RTT Médio RTT Máximo RTT Mínimo Desvio Padrão
Quantidade de dados trocados	20 Kbits.
Servidor de Testes	Servidor Google (173.194.41.215)

5.3.3.3 Teste de QoS – Perda de Pacotes

A taxa de perda de pacotes é uma das métricas avaliadas. Tal como o teste de latência, o teste de perda de pacotes usa o Ping. A ubiquidade desta ferramenta é elevada, e tal como foi abordado anteriormente faz que com que seja usada em diversas aplicações que se inserem na avaliação de QoS. Os dados obtidos neste teste são:

- Número de pacotes enviados;
- Número de pacotes recebidos;
- Percentagem de perda de pacotes;
- Duração do teste.

Estes são os dados recolhidos pela aplicação. Para a execução do teste é, tal como acontece no teste de latência, usado o servidor da Google com o endereço 173.194.41.215, sendo trocados dez pacotes. Um dos pontos importantes era que os testes fossem executados nas mesmas condições, para que qualquer variação nos dados resultantes não dependesse dos servidores usados.

O conjunto total de dados recolhidos pela aplicação imediatamente após um teste de perda de pacotes difere de um outro teste apenas nos dados recolhidos na fase 2. Na fase 1 os valores são os mesmos (ver Fig. 27).

Após serem obtidos os valores sobre a perda de pacotes através do teste Ping, é calculado o tempo de duração do teste. Posteriormente, os dados são guardados na

base de dados da aplicação. Caso a opção esteja ativa os dados que resultam do teste são enviados para o servidor (ver Fig. 28).

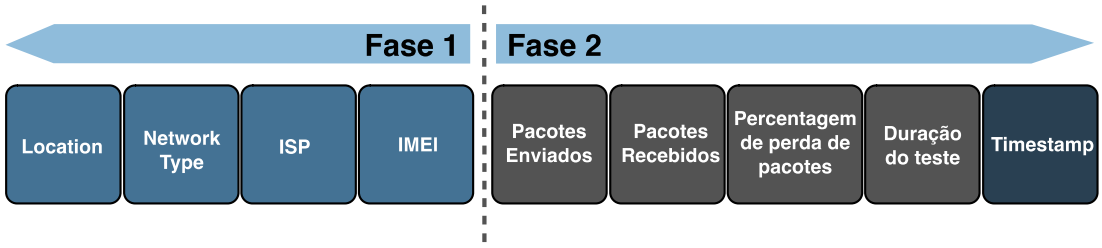


Figura 27 - Conjunto de dados recolhidos pela aplicação após um teste de perda de pacotes.

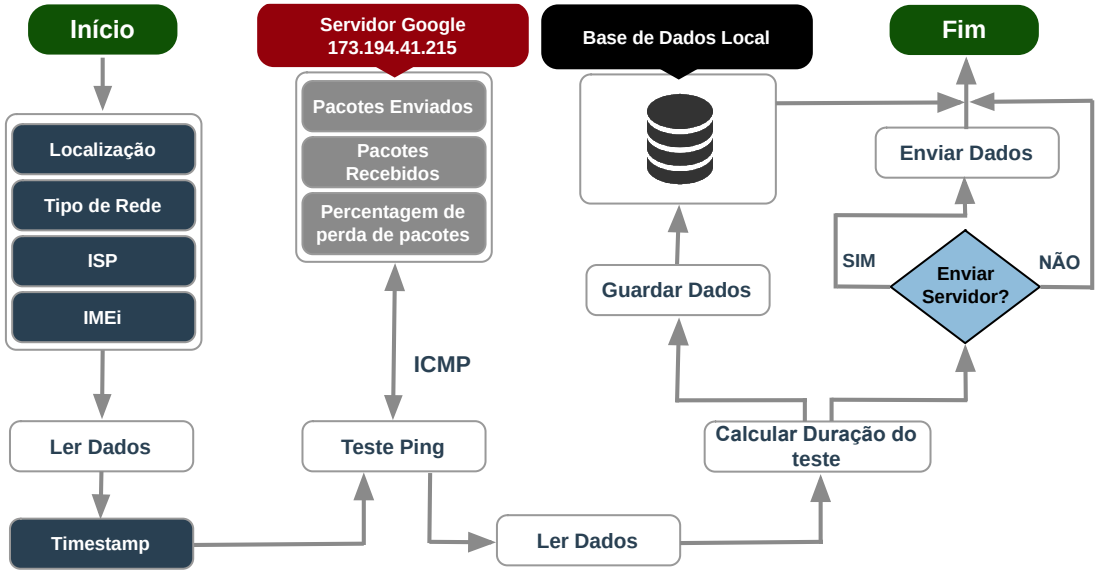


Figura 28 - Modelo de execução de um teste de perda de pacotes.

Tabela 8 - Parâmetros do teste de perda de pacotes.

Parâmetro	Descrição
Protocolo	ICMP
Pacotes Enviados	10
Duração do teste	Aprox. 10 segundos
Valores Obtidos	Pacotes Enviados Pacotes Recebidos Percentagem de Perda de Pacotes Duração do teste
Quantidade de dados trocados	20 Kbits.
Servidor de Testes	Servidor Google (173.194.41.215)

5.3.3.4 Teste de QoS – Download

De entre os testes de QoS que a aplicação executa, o teste de *download* é o que mais se destaca, apresentando uma das métricas que, na óptica dos utilizadores, é a mais importante. O valor apresentado é a quantidade de dados lidos do servidor sobre a unidade de tempo (*bits* por segundo).

O teste é efectuado com a intensão de "inundar" a conexão com pedidos HTTP GET ao servidor da Google (173.194.41.215). Os pedidos são feitos em "simultâneo"[2]. Os dados contabilizados referem-se ao *throughput*, o tempo contabilizado contém *overhead* relacionado com o estabelecimento da ligação TCP entre o cliente e o servidor.

Os dados recolhidos neste teste são:

- Débito binário medido em bits/s;
- Duração do teste completo em milissegundos;
- Quantidade de dados recebida do servidor em bits.

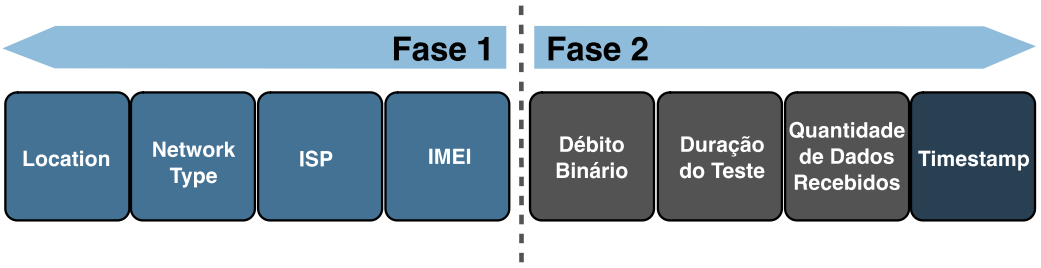


Figura 29 - Conjunto de dados recolhidos pela aplicação após um teste de *download*.

O modelo de execução de um teste de *download* é apresentado na Fig. 30 e é semelhante ao modelo dos testes anteriores. O pedido é feito através do protocolo HTTP, recorrendo ao método GET, obtendo-se um conjunto de dados relativo a uma página HTML. Esses dados são recebidos pela aplicação e sobre eles é calculado o débito. No domínio deste teste são recolhidos dois tempos. Um representa o tempo necessário para a execução dos pedidos HTTP, outro representa o tempo de execução de todo o teste, ou seja, desde que o utilizador inicia até que lhe sejam apresentados os dados., incluindo todo o processamento necessário pela aplicação. Na Fig. 30 está representado o primeiro.

[2] Considera-se em simultâneo, mas são iniciados com diferenças de poucos milissegundos.

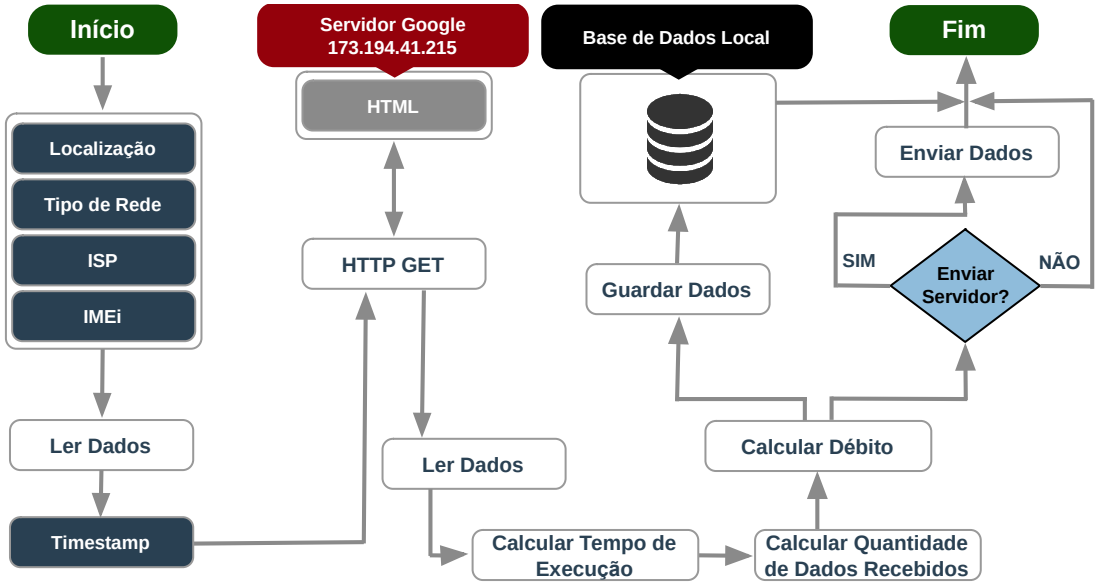


Figura 30 - Modelo de execução de um teste de *download*.

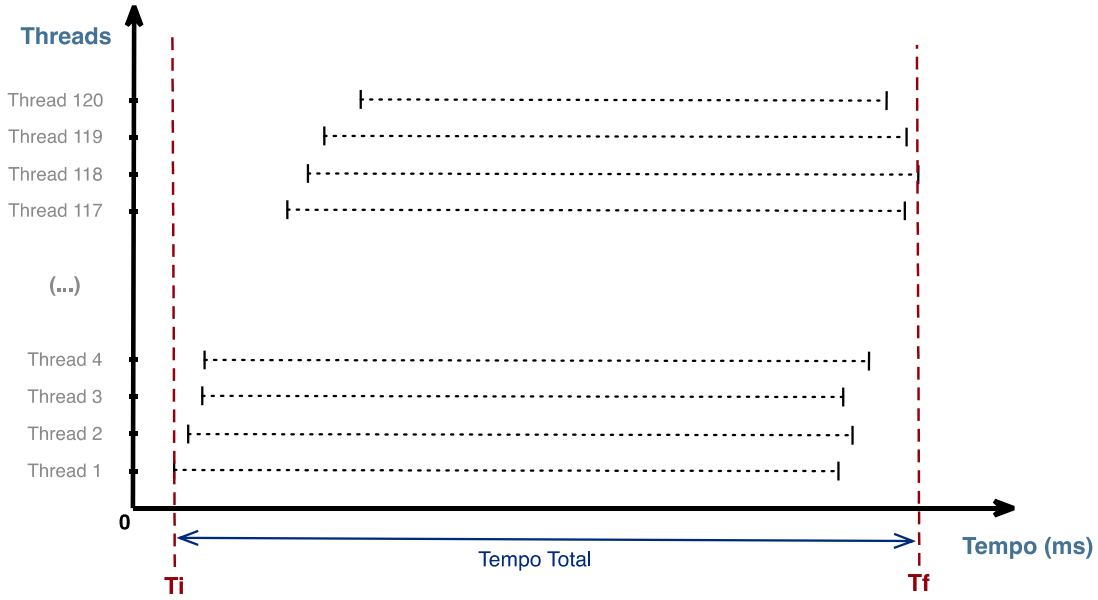


Figura 31 - Gráfico ilustrativo do teste de *download*.

O gráfico da Fig. 31 representa o processo pelo qual é obtido o tempo total de *download* (tempo de execução dos pedidos HTTP GET) que é usado para calcular a quantidade de dados sobre a unidade de tempo.

O processo pelo qual é realizado o teste é o seguinte:

- A aplicação lança 120 *threads*;
- Cada *thread* uma executa um pedido HTTP GET ao servidor da Google e é registado o tempo inicial do pedido;
- O servidor responde com os dados, correspondentes a uma página HTML;
- Em algumas situações, por limitações do dispositivo, nem sempre é possível executar todas as 120 *threads*. Neste caso o teste é feito apenas com as *threads* que o sistema operativo permite nesse instante.
- Após as *threads* receberem a resposta, é calculado o tempo final de cada uma. O tempo inicial do teste de *download* é dado pelo tempo inicial da primeira *thread* a iniciar o pedido HTTP GET. O tempo final do teste é dado pelo tempo final da última *thread* a executar e receber a resposta do servidor. Com estes valores obtém-se o tempo total do teste de *download*. O tempo total que cada *thread* necessita para executar o pedido é diferente, logo, a última *thread* a ser lançada não é necessariamente a última a executar.
- Depois de obtido o tempo total de teste, contabiliza-se a quantidade de dados trocados e calcula-se o débito em *download*.

$$Download\ (bits/s) = \frac{Tamanho\ Total\ dos\ Dados\ (bits)}{Tempo\ Total\ (s)} \qquad \textbf{(1)}$$

Na Eq. 1, o Tamanho Total de Dados é a soma total da quantidade de dados que a aplicação recebe como resposta ao conjunto de pedidos HTTP GET efectuado pelas *threads*. O Tempo Total é o valor obtido desde que a primeira *thread* executa o pedido até à última *thread* a fazê-lo, como é apresentado na Fig. 31.

Tabela 9 - Parâmetros do teste de *download*.

Parâmetro	Descrição
Protocolo	HTTP
Threads	120
Duração do teste	Aprox. 15 segundos.
Total de Dados Recebidos	Aprox. 11 Mbits
Valores Obtidos	Débito Binário (bits/s) Duração do Teste (Pedidos HTTP GET) (ms) Quantidade de Dados Recebidos (bits)
Servidor de Testes	Servidor Google (173.194.41.215)

Como já foi referido, para executar o teste de *download* são lançadas 120 *threads*. Para se chegar a este valor, foi feito um estudo com vários testes. Foram realizados 15 testes, com 20, 50, 100, 120 e 150 *threads*. Posteriormente foi calculado o débito médio obtido nos 15 testes realizados. Na Fig. 32 é apresentado um gráfico que traduz os resultados dos testes efectuados. Verificou-se que com 120 *threads* em simultâneo se obtinha o valor máximo de *download*, aproximadamente 8.8 Mbits/s. Com 150 *threads*, o valor desceu para os 8.4 Mbits/s. Determinou-se que a execução dos testes com 120 *threads* permitia obter um resultado coerente. O objetivo era saturar a conexão e o resultado que melhores valores apresentou foi recorrendo a 120 *threads*.

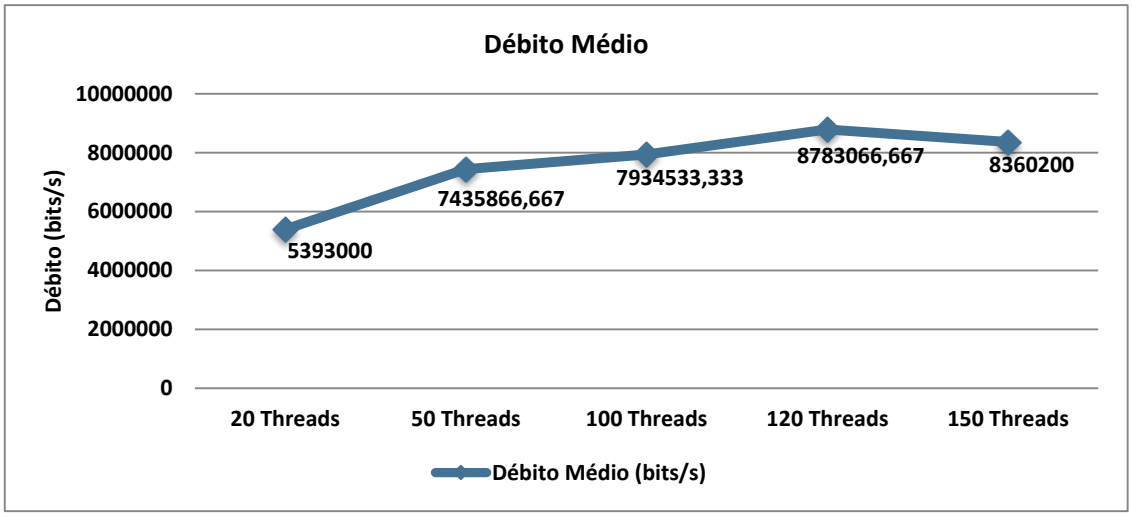


Figura 32 - Débito médio em *download* (bits/s) para testes com diferentes *threads*.

Para cada um dos testes foi analisado o tempo de execução. Pelo gráfico da Fig. 33 verifica-se que com 120 *threads* o tempo médio de execução é de 1.3 segundos, aproximadamente. No entanto, com 150 *threads* o valor aumenta consideravelmente, para cerca de 1.9 segundos, explicando o débito apresentado na Fig. 32. O tempo médio de execução de um pedido HTTP GET é de 371 ms, aproximadamente.

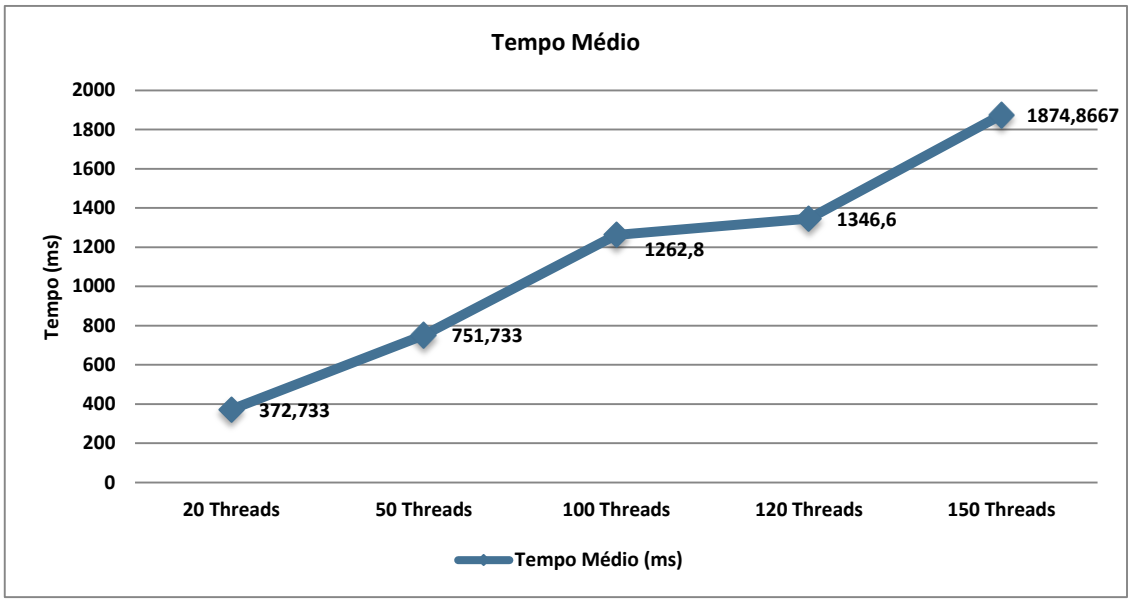


Figura 33 - Tempo médio de *download* (ms) para testes com diferentes *threads*.

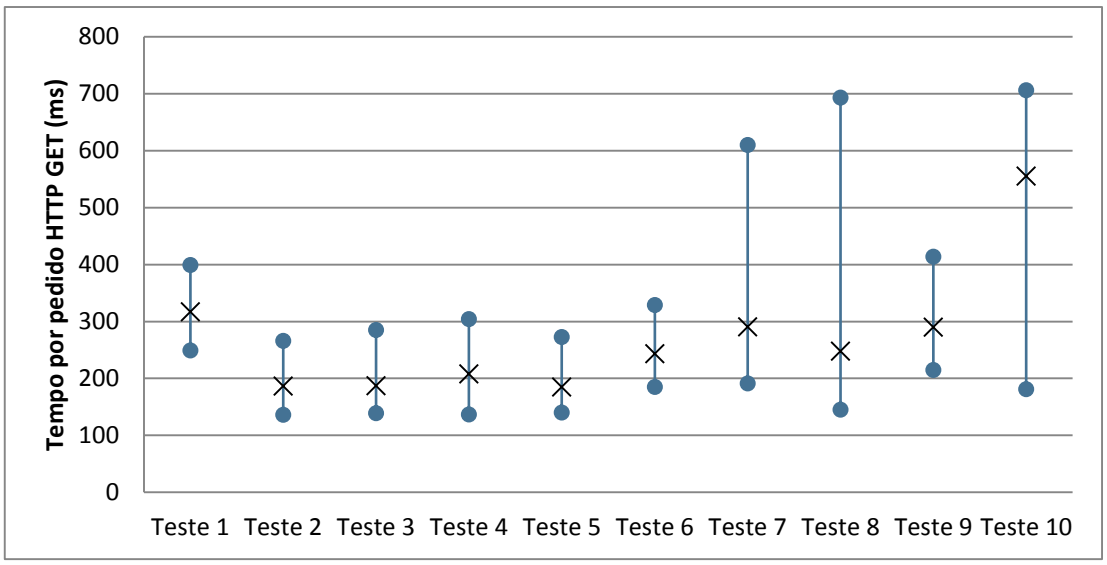


Figura 34 - Gráfico com valores máximo, mínimo e médio por pedido HTTP GET.

O gráfico da Fig. 34 apresenta um conjunto de 10 testes executados com 120 *threads*, apresentado o tempo máximo, mínimo e médio em cada um. Verifica-se que apesar de haver, por vezes, valores máximos acentuados, como acontece no Teste 7, 8 e 10 (tempo máximo), o valor médio é constante, só no Teste 10 é que apresenta um valor díspar.

Estes testes permitiram escolher qual a melhor abordagem a seguir. Ao mesmo tempo, estes resultados foram comparados com outras aplicações no mercado, como o Speedtest (Ookla) ou o Speed Checker (Speedchecker Ltd.) e os resultados foram sempre muitos semelhantes.

Outra abordagem foi ponderada na execução dos testes de *download*. Ao contrário da abordagem anterior, o cálculo seria feito em tempo real, ou seja, sempre que uma *thread* terminava a execução de um pedido HTTP GET, era instantaneamente calculado o débito. Com esta abordagem seria definido um número de *threads* a utilizar de forma dinâmica. O teste terminaria quando se verificasse um decréscimo constante no débito calculado. Com esta abordagem, era possível ter um teste dinâmico e não estático. O teste era adaptativo e permitia obter um resultado de acordo com os valores que iam sendo calculados durante a sua execução. Contudo, o algoritmo e a condição de paragem seriam bastante complexos, exigiriam uma enorme capacidade de processamento por parte do dispositivo e poderia levar a que os resultados demorassem bastante tempo a ser apresentados ao utilizador e acabaria por ir contra um dos objetivos da aplicação, ser expedita na execução e apresentação dos testes.

As Fig. 35 e Fig. 36 apresentam um estudo realizado sobre esta abordagem. Pelo segundo gráfico é possível observar que o débito alcançou o pico máximo ao fim da execução de 100 *threads*, onde teve uma descida abrupta, voltando a aumentar linearmente até alcançar as 120 e finalizar. Pelo contrário, no primeiro gráfico é possível observar um comportamento diferente, onde não se regista nenhuma queda abrupta, e o débito aumenta linearmente até alcançar as 120 *threads* e finalizar. Estas alterações e variações levariam a uma elevada complexidade na estipulação das condições de paragem e implementação de um algoritmo.

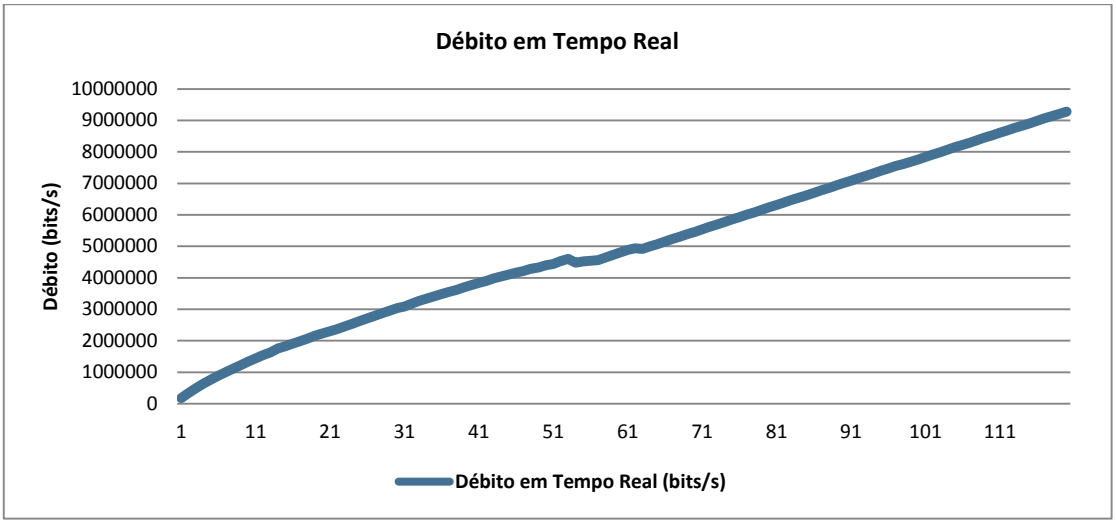


Figura 35 - Gráfico do cálculo do débito em *download* em tempo real, teste 1.

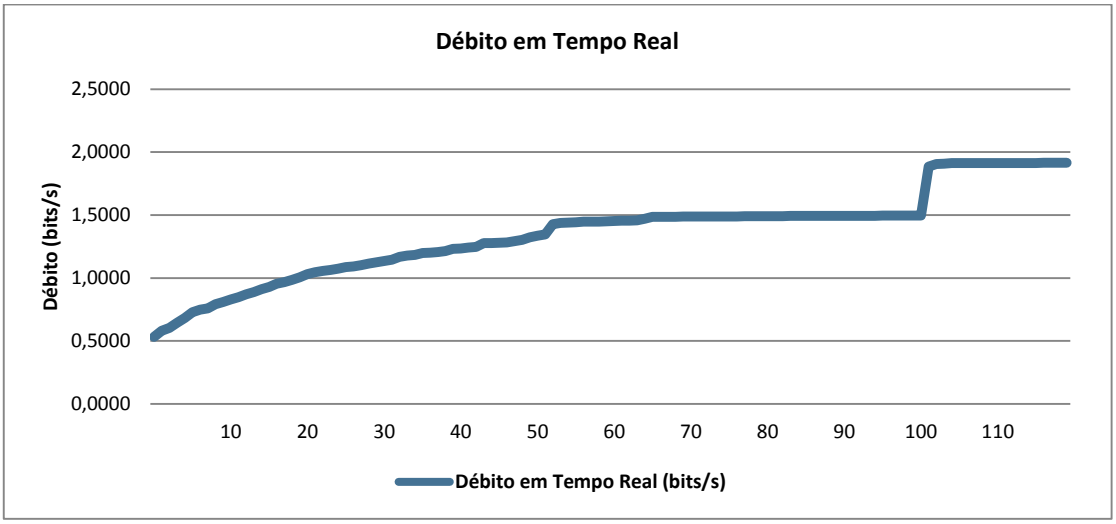


Figura 36 – Gráfico do cálculo do débito em *download* em tempo real, teste 2.

O gráfico da Fig. 37 mostra o momento em que são lançadas as *threads* pelo dispositivo.

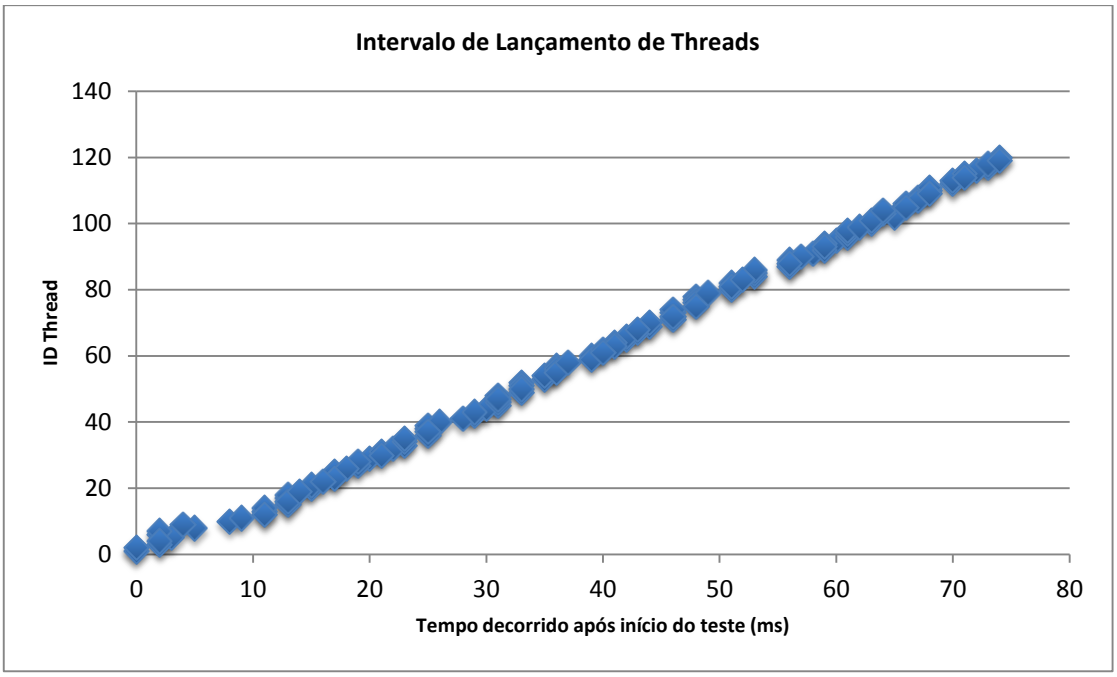


Figura 37 - Gráfico do Intervalo de tempo de lançamento de *threads*.

Como parte do estudo realizado foram criados os gráficos da Fig. 38 e da Fig. 39 para se poder observar o comportamento das *threads* no instante em que estas terminam a sua execução. O comportamento observado foi de difícil interpretação. No gráfico da Fig. 39 verifica-se que existem dois instantes em que várias *threads* terminam, 1500 ms e 1900 ms após o início do teste. Quanto ao gráfico da Fig. 38 verifica-se um comportamento bastante diferente, onde várias *threads* terminam o seu processamento de forma progressiva, até que surge um instante em que várias terminam em simultâneo, cerca de 1800 ms após o início do teste.

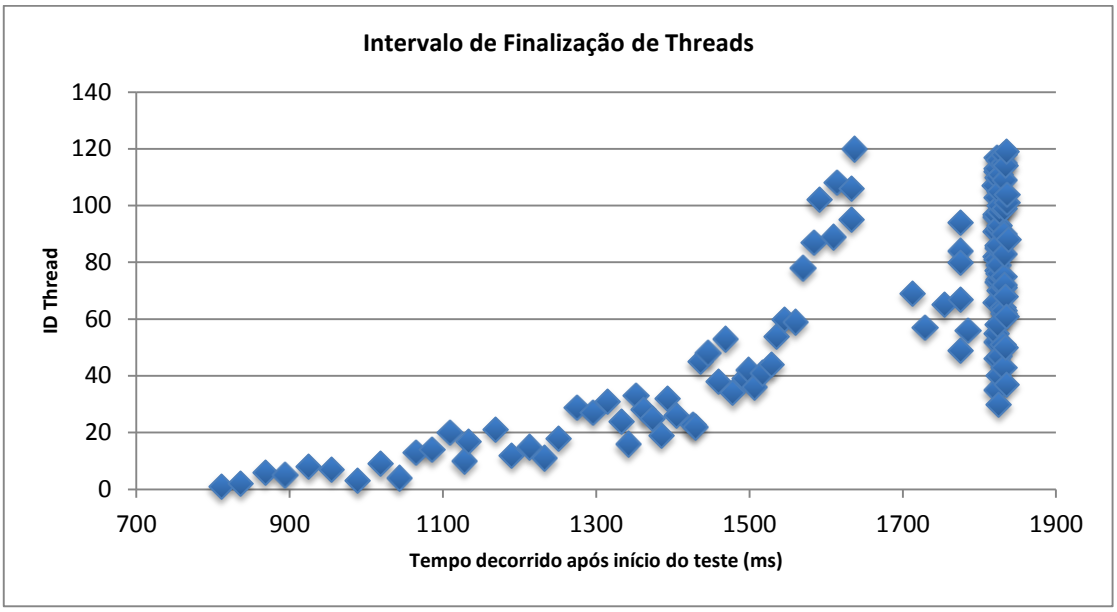


Figura 38 - Gráfico do intervalo de tempo de finalização de *threads*, teste 1.

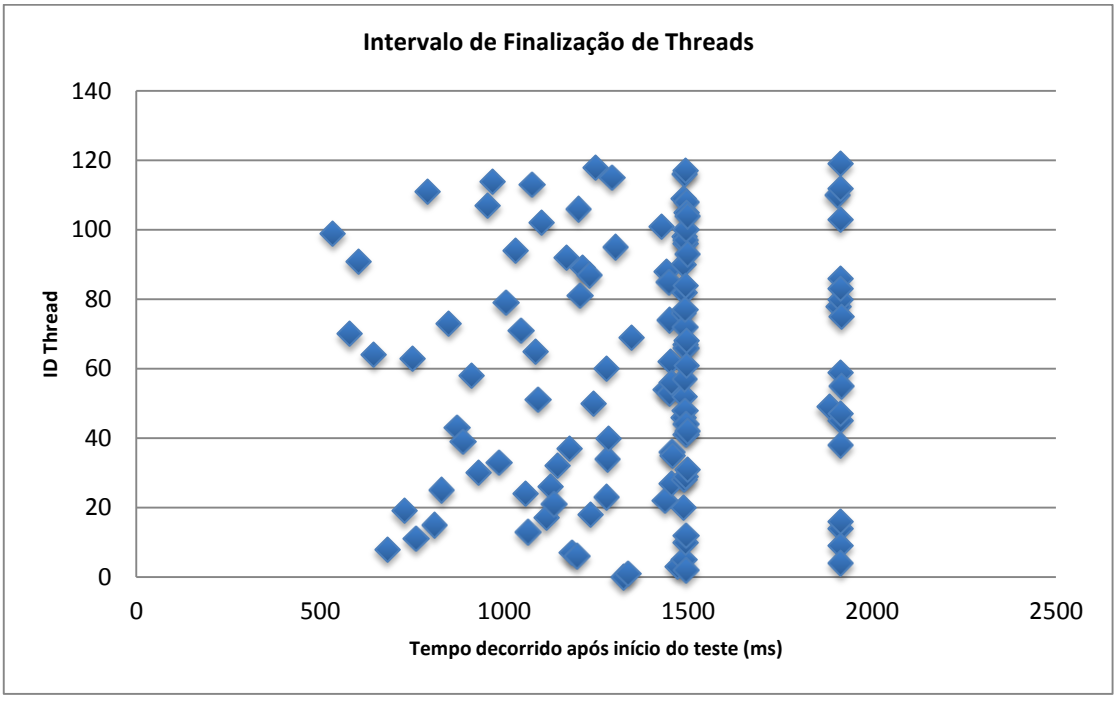


Figura 39 - Gráfico do intervalo de tempo de finalização de *threads*, teste 2

Para tentar determinar se o comportamento era dependente do servidor ao qual os testes estavam a ser realizados, neste caso concreto ao servidor da Google, foram realizados os mesmos testes a outros dois servidores, do Bing e do Facebook,

contudo, os resultados não foram conclusivos. Verifica-se que os tempos de resposta aos pedidos HTTP GET dependem bastante do servidor, mas ainda assim não se consegue explicar o comportamento das *threads*, nem tão pouco perceber porque é que *threads* lançadas primeiro terminam o seu processamento bastante tempo depois de outras lançadas muito depois. Seria de esperar, e dado o gráfico que mostra o intervalo de lançamento das *threads* (ver Fig. 37), que o comportamento das *threads* no fim da sua execução fosse semelhante, mas tal não se verifica.

Após a realização deste estudo verificou-se que se tornaria impossível a implementação desta metodologia para o cálculo do débito, visto os resultados não serem conclusivos. Para se poder transitar para esta abordagem seria necessário efetuar mais estudos para se tentar compreender o comportamento das *threads* e como os servidores atendem e processam os pedidos.

5.3.3.5 Teste de QoS – Upload

O teste de *upload* tem como objetivo obter o débito binário no sentido ascendente. É uma métrica que no domínio dos dispositivos móveis não é muito crítica, uma vez que a generalidade das aplicações usadas não têm grandes requisitos.

À semelhança do teste de *download*, é usada uma estratégia que assenta na execução de um conjunto de pedidos em "simultâneo"[3] a um servidor, por forma a "inundar" a conexão, para que esta sature. O valor apresentado é a quantidade de dados enviados para o servidor sobre a unidade de tempo (*bits* por segundo). Ao contrário dos restantes testes, tal como já tinha sido abordado, este não recorre ao servidor da Google, mas sim a um servidor da Universidade do Minho (urano.dsi.uminho.pt).

O teste baseia-se num conjunto de pedidos HTTP POST. Os dados enviados são referentes a um ficheiro de texto (file_*download*.txt) que se encontra numa pasta da aplicação, e que tem um tamnaho de 44 614 bytes. A referida pasta chama-se de *assets* e é destinada a armazenar ficheiros ou dados que são necessários à aplicação.

Os dados recolhidos neste teste são:

[3] Considera-se em simultâneo, mas são iniciados com diferenças de poucos milissegundos.

- Débito binário obtido em bits/s;
- Duração do teste completo em milissegundos;
- Quantidade de dados enviada para o servidor em bits.

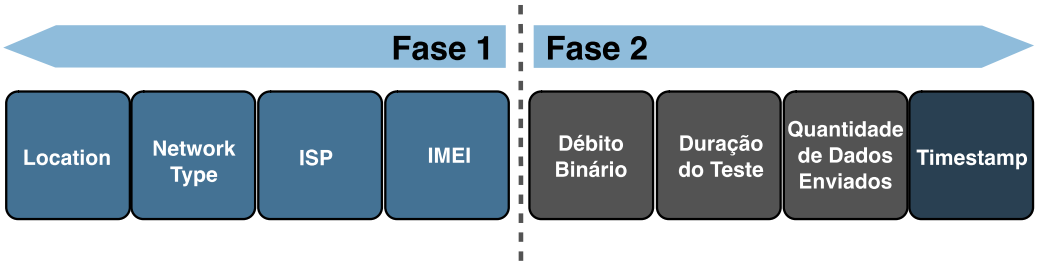


Figura 40 - Conjunto de dados recolhidos pela aplicação após um teste de *upload*.

O modelo de execução do teste de *upload* é semelhante ao do teste de *download*, apenas alterando na interação com o servidor (ver Fig. 41).

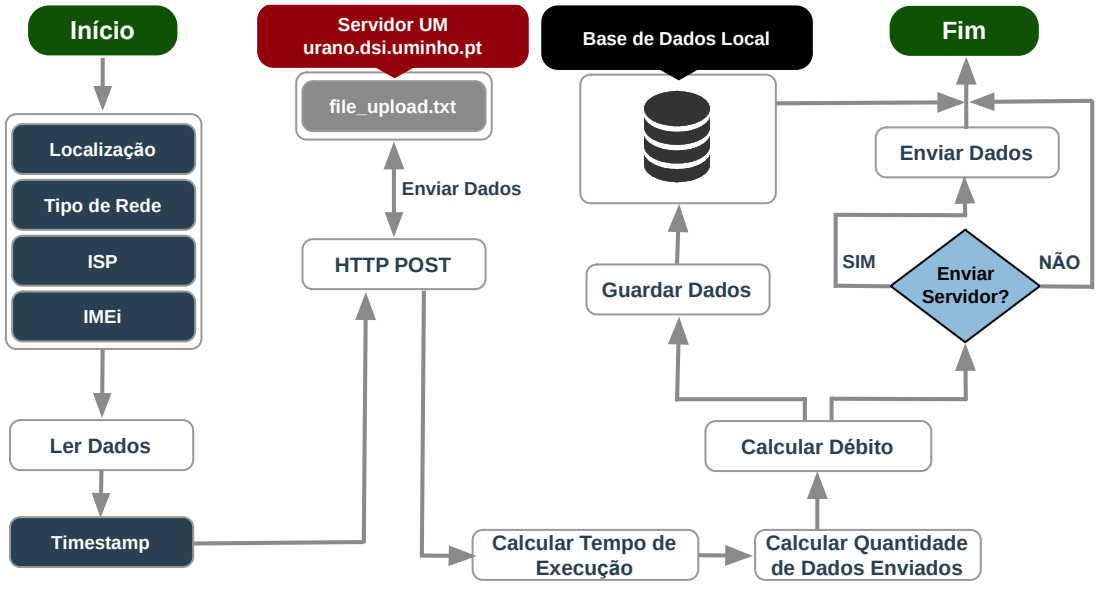


Figura 41 - Modelo de execução de um teste de *upload*.

O processo pelo qual é realizado o teste é o seguinte:

- A aplicação lança 10 *threads*;

- Cada *thread* uma executa um pedido HTTP POST ao endereço do servidor da Universidade do Minho e é registado o tempo inicial do teste;
- Após a execução de cada *thread*, é calculado o tempo final de cada pedido POST de cada uma. O tempo inicial do teste de *upload* é dado pelo tempo inicial da primeira *thread* a executar o pedido HTTP POST. O tempo final é dado pelo tempo final da última *thread* a executar o pedido. Com estes valores obtém-se o tempo total do teste de *upload*. O tempo total que cada *thread* necessita para executar o pedido é diferente, logo, a última *thread* a ser lançada não é necessariamente a última a executar (ver Fig. 42).
- Depois de obtido o tempo total de teste, contabiliza-se a quantidade de dados enviados e calcula-se a taxa de *upload*.
- O débito é calculado segundo a mesma fórmula usada no teste de *download* (ver. Eq. 2).

$$Upload\ (bits/s) = \frac{Tamanho\ Total\ dos\ Dados\ (bits)}{Tempo\ Total\ (s)} \qquad \textbf{(2)}$$

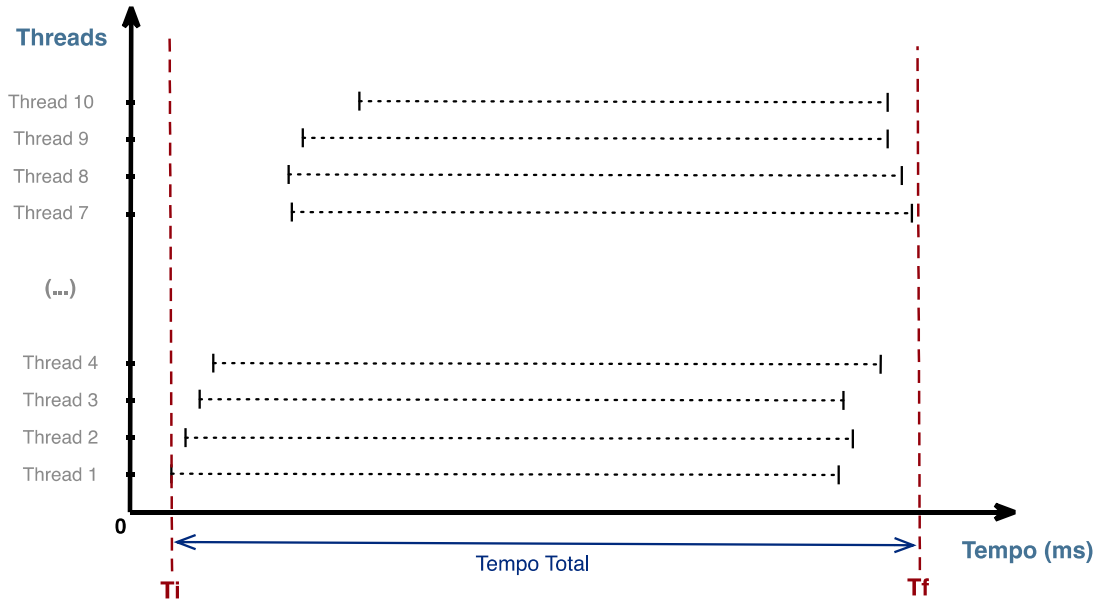


Figura 42 - Gráfico ilustrativo do teste de *upload*.

Tabela 10 - Parâmetros do teste de *upload*.

Parâmetro	Descrição
Protocolo	HTTP
Threads	10
Duração do Teste	Aprox. 16 segundos.
Total de Dados Enviados	Aprox. 3.6 Mbits
Valores Obtidos	Débito Binário (bits/s) Duração do Teste (Pedidos HTTP POST) (ms) Quantidade de Dados Enviados (bits)
Servidor de Testes	Servidor UM (urano.dsi.uminho.pt)

Foi feito um estudo para determinar o número de *threads* mais indicado e o que apresentava melhores resultados. Foi feito um conjunto de testes recorrendo a abordagens com: 10, 20 e 50 *threads*. A solução que se destacou foi a primeira. No gráfico apresentado na Fig. 45 é possível observar que com 10 *threads* foi alcançado um débito de 900 Kbits/s, aproximadamente. O resultado obtido com 20 *threads* foi ligeiramente mais baixo, na ordem dos 890 Kbits/s. Contudo, recorrendo a 50 *threads* o débito diminuiu drasticamente, aproximando-se muito dos valores recorrendo apenas a uma *thread*.

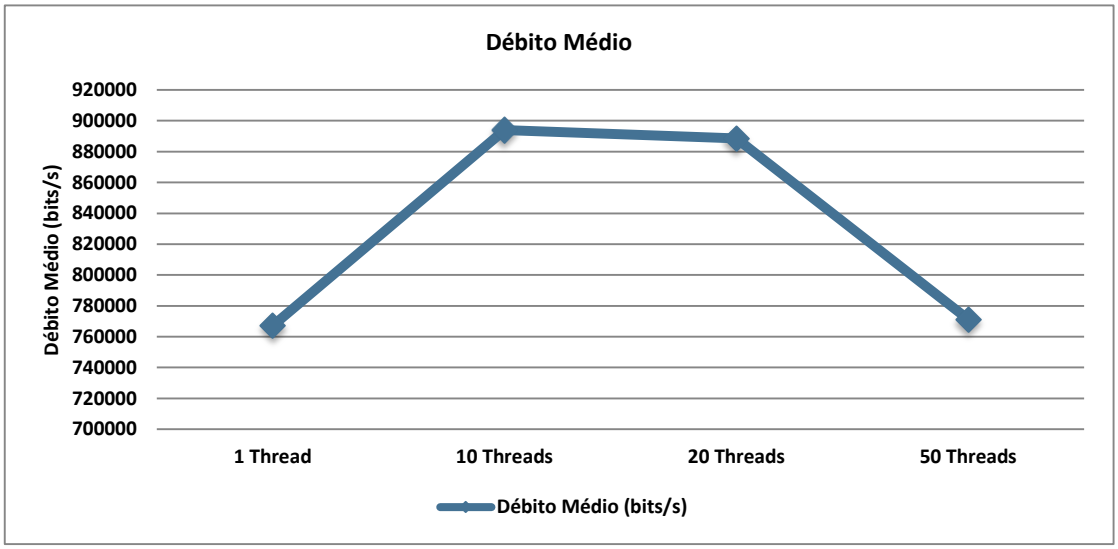


Figura 43 - Débito médio em *upload* (bits/s) para testes com diferentes *threads*.

No que toca ao tempo de execução do teste, a solução com 10 *threads* apresentou um valor médio inferior a 5 segundos (ver Fig. 44). Apesar da quantidade de dados ser maior na solução recorrendo a 50 *threads*, dado o conjunto de pedidos realizados (HTTP POST), o tempo necessário para a execução dos mesmos é muito elevado, superior a 20 segundos, explicando o fraco desempenho apresentado na Fig. 43.

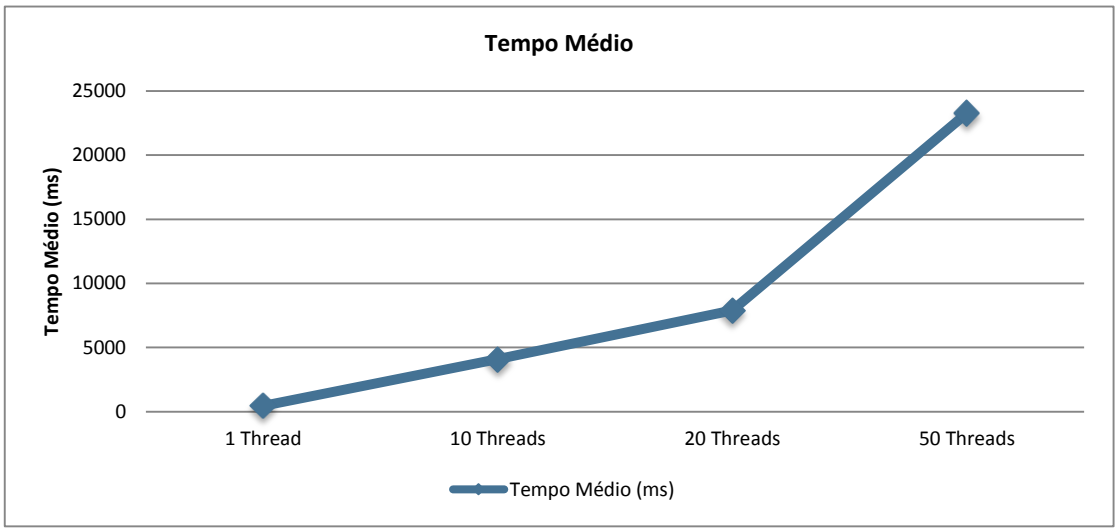


Figura 44 - Tempo médio de *upload* (ms) para testes com diferentes *threads*.

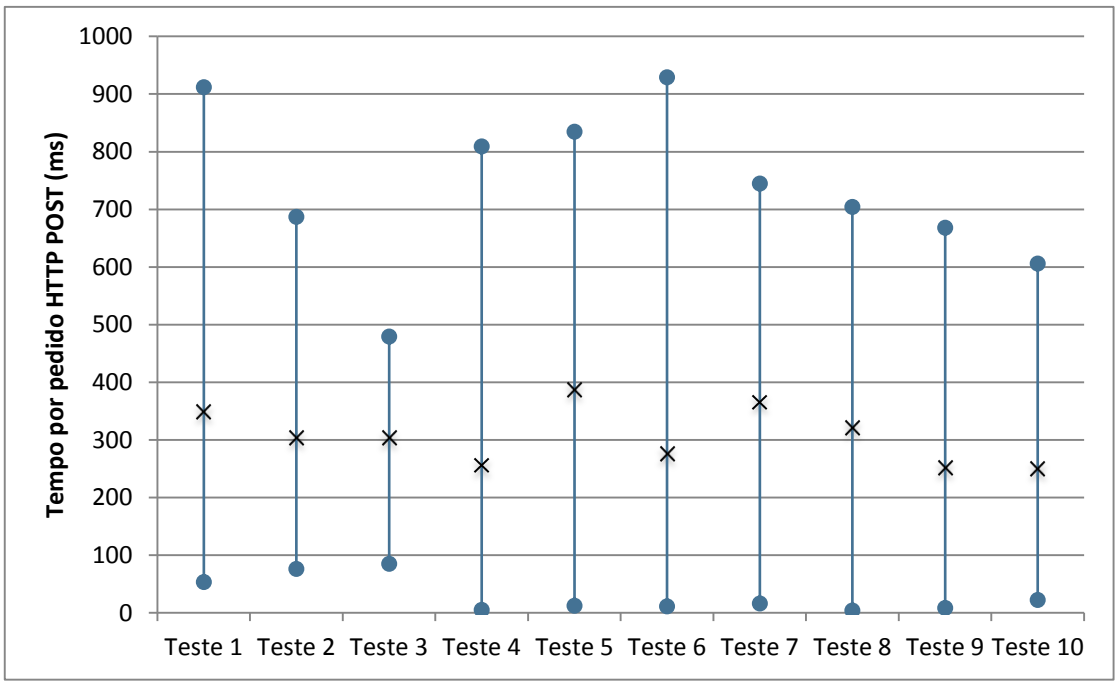


Figura 45 - Gráfico com valores máximo, mínimo e médio por pedido HTTP POST.

O gráfico apresentado na Fig. 45 permite observar os valores máximo, mínimo e médio obtido em cada um dos 10 testes realizados usando 10 *threads* em simultâneo. Tal como se tinha verificado nos testes realizados em *download*, também aqui se observa que os valores médios são bastante constantes e não sofrem grandes variações, apesar de ser observarem valores máximos bastante díspares.

Este conjunto de testes possibilitou fazer uma análise às diferentes abordagens permitindo escolher, de forma fundamentada aquela que melhores resultados apresentou.

À semelhança do teste de *download*, no teste de *upload* também foi ponderada a solução que recorria a um cálculo em tempo real, mas, pelas mesmas razões apresentadas anteriormente, a abordagem foi abandonada.

5.3.3.6 Teste de QoS – Full QoS Test

O *Full QoS Test* é um teste que avalia todas as métricas num só teste. As metodologias usadas são as mesmas usadas no testes acima descritos.

Tabela 11 - Parâmetros do Full QoS Test.

Parâmetro	Descrição
Protocolos	HTTP e ICMP
Duração do teste	Aprox. 27 segundos.
Total de Dados Enviados	Aprox. 3.6Mbits
Valores Obtidos	***Upload***: • Débito Binário (bits/s) • Duração do Teste (Pedidos HTTP POST) (ms) • Quantidade de Dados Enviados (bits) ***Download***: • Débito Binário (bits/s) • Duração do Teste (Pedidos HTTP GET) (ms) • Quantidade de Dados Recebidos (bits) **Latência**: • RTT Médio • RTT Máximo • RTT Mínimo • Desvio Padrão **Perda de Pacotes**: • Pacotes Enviados • Pacotes Recebidos • Percentagem de Perda de Pacotes
Servidores de Testes	Servidor UM (urano.dsi.uminho.pt)/Servidor Google (173.194.41.215)

5.3.3.7 Teste de QoS – Teste Periódico

O teste periódico avalia apenas as métricas da latência e perda de pacotes. Deixou-se de parte o *download* e o *upload* na medida em que estes testes são bastante pesados no que diz respeito à quantidade total de dados trocados, levando a que o impacto no serviço de dados móveis dos utilizadores fosse bastante alto. Os testes de *download* e *upload*, em termos de processamento, são bastante exigentes lançando entre os dois 130 *threads*, levando a um consumo de bateria acrescido. Por estas razões optou-se por não se avaliar estas duas métricas neste teste. Os testes de latência e a perda de pacotes são os mesmos que foram descritos anteriormente e todo o processo é igual.

Tabela 12 - Parâmetros de um teste periódico.

Parâmetro	Descrição
Protocolos	ICMP
Duração do teste	Aprox. 16 segundos.
Valores Obtidos	**Latência**: • RTT Médio • RTT Máximo • RTT Mínimo • Desvio Padrão **Perda de Pacotes**: • Pacotes Enviados • Pacotes Recebidos • Percentagem de Perda de Pacotes
Servidores de Testes	Servidor Google (173.194.41.215)

Os testes periódicos são efetuados com um intervalo que se situa entre 18 e 30 horas, permitindo ter um valor que se situa num intervalo de 12 horas em torno das 24 horas. A aleatoriedade permite obter dados com mais relevância, aumentando a probabilidade de os testes serem feitos em localizações diferentes e em diferentes circunstâncias, utilizando diferentes tipos de rede, entre outros.

Para a implementação dos testes periódicos usam-se dois sub-módulos específicos, *Alarm Management* e *Service Management*. O primeiro permite configurar um alarme na aplicação, onde se define o intervalo de tempo. O alarme permite que haja uma execução de código mesmo quando a aplicação não está a ser usada ou mesmo que o dispositivo está em repouso. Quando o alarme é ativo, o

código que é executado está no sub-módulo *Service Management*. Este, executa os testes de latência e perda de pacotes segundo os modelos apresentados anteriormente. O fluxograma apresentado na Fig. 46 apresenta o modo de execução de um teste periódico.

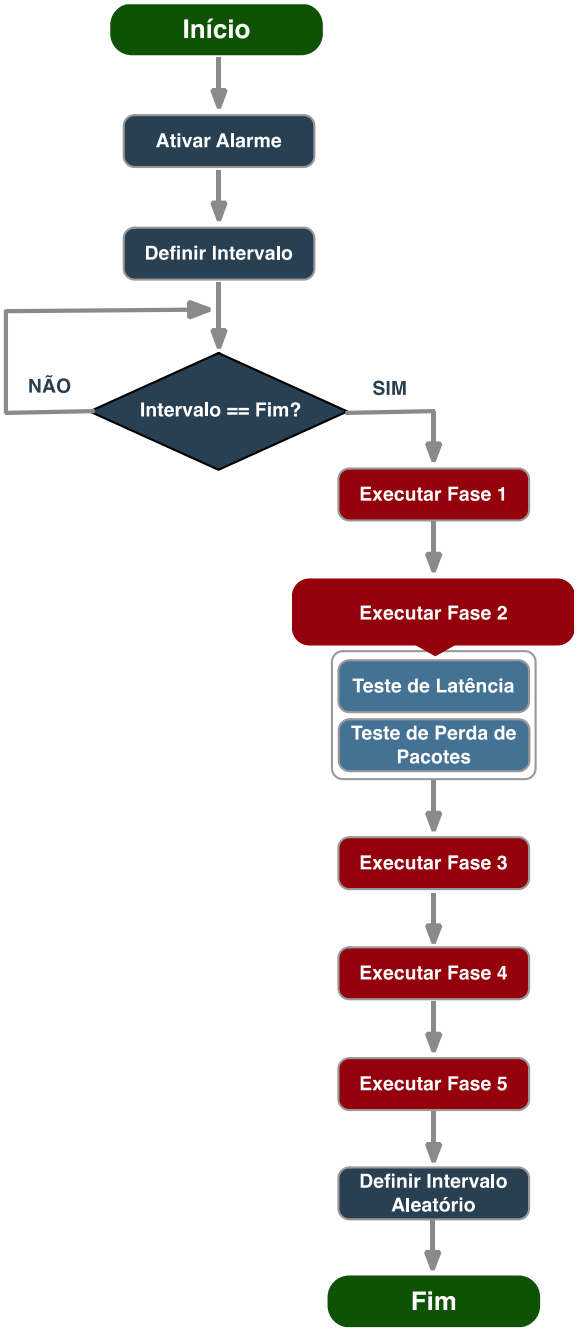


Figura 46 – Fluxograma da execução de um teste periódico.

O processo inicia-se pela ativação do alarme, depois a aplicação define o intervalo para a execução. A primeira execução ocorre sempre dois minutos após a ativação do utilizador. A anatomia de todo o teste é semelhante à apresentada anteriormente, diferindo apenas no facto de não se executar a fase 6, respetiva à apresentação dos resultados ao utilizador. Neste caso concreto os resultados deste tipo de testes são acedidos exclusivamente através do menu pessoal do utilizador. Adicionalmente ao procedimento normal, no fim da execução do teste é sempre necessário definir um novo valor aleatório para o intervalo de tempo a decorrer até ao próximo teste. Ao fim desse tempo é iniciada uma nova execução e definido um novo valor aleatório e assim sucessivamente, até que o utilizador desative a opção no seu menu pessoal.

5.3.4 Avaliação Qualitativa

No capítulo referente ao modelo de funcionamento foram descritos os resultados qualitativos que podem ser classificados segundo cinco níveis distintos: *Excellent* (Excelente), *Very Good* (Muito Bom), *Good* (Bom), *Acceptable* (Aceitável) e *Bad* (Mau). Para cada uma das métricas existe um conjunto de intervalos de valores associados a cada nível. Era necessário definir como se iria avaliar os resultados antecipadamente, para que fosse possível executar os testes e desenvolver uma solução de acordo com a perspetiva inicial, apresentada na descrição das métricas. A avaliação qualitativa é sempre feita pelos métodos do sub-módulo *QoS Management*.

O modelo de avaliação do débito de *download* tem em conta a possibilidade de serem usadas redes 3G e 4G. Por outro lado, considerou-se também que a generalidade das aplicações em equipamentos móveis ainda não tem uma necessidade muito elevada em termos de débito descendente, no entanto, à media que a capacidade de processamento dos equipamentos aumenta, a complexidade e o tipo de aplicações também evolui, que poderá levar a maiores requisitos neste domínio.

Tabela 13 - Avaliação Qualitativa do débito em *download*.

Download	Nível Qualitativo
> 75 Mbits/s	**Excelente**
75 – 20 Mbits/s	**Muito Bom**
20 – 10 Mbits/s	**Bom**
10 – 2 Mbits/s	**Aceitável**
< 2 Mbits/s	**Mau**

Estipulou-se que valores acima dos 75 Mbits/s seriam considerados "Excelentes". O segmento entre os 2 Mbits/s e 10 Mbits/s, considerado de aceitável é provavelmente o mais comum em termos de débito associados às ofertas comerciais dos utilizadores.

Relativamente à métrica de *upload*, como já foi referido, no domínio das aplicações utilizadas em dispositivos móveis, os requisitos são mínimos. Devido a este aspeto, considera-se que todos os débitos medidos acime dos 1.5 Mbits/s são considerados "Excelentes" e satisfazem em pleno as exigências da generalidade das aplicações (ver Tab. 14).

Tabela 14 - Avaliação Qualitativa do débito em *upload*.

Upload	Nível Qualitativo
> 1.5 Mbits/s	**Excelente**
1.5 Mbits/s – 1.1 Mbits/s	**Muito Bom**
1.1 Mbits/s – 950 Kbits/s	**Bom**
950 – 750 Kbits/s	**Aceitável**
< 750 Kbits/s	**Mau**

Ao nível do modelo de avaliação da taxa de pacotes perdidos (ver Tab. 15), usa-se como base os valores referenciados pelo projeto PingER desenvolvido pelo SLAC (*Stanford Linear Accelarator Laboratory*) [22]. Os níveis qualitativos foram manipulados, para estar de acordo com os níveis qualitativos definidos para as outras métricas.

Tabela 15 - Avaliação Qualitativa do taxa de perda de pacotes.

Upload	Nível Qualitativo
< 1%	**Excelente**
1 – 2.5%	**Muito Bom**
2.5 – 5%	**Bom**
5 – 12%	**Aceitável**
> 12%	**Mau**

Quanto à avaliação da latência, foi feita uma análise à realidade nacional usando o teste Ping para efetuar alguns testes. Atendendo a recomendações feitas pelo ITU-T, apresentadas anteriormente e aos resultados obtidos no relatório da ANACOM [6], estipulou-se uma escala de valores até aos 80 ms. Qualquer valor acima é considerado "Mau".

Tabela 16 - Avaliação Qualitativa da latência.

Latência	Nível Qualitativo
< 30 ms	**Excelente**
30 – 40 ms	**Muito Bom**
40 – 50 ms	**Bom**
50 – 80 ms	**Aceitável**
> 80 ms	**Mau**

Quando se desenvolveu o *Full QoS Test* criou-se uma métrica específica, global, que representa a QoS num único valor. Dado que estes testes avaliam todas as métricas, considerou-se importante encontrar uma valor que permitisse sumariar todos os resultados qualitativos referentes às quatro métricas avaliadas. Para tal é considerado um modelo de médias ponderadas para agregar os dados obtidos, sendo que as diferentes métricas de QoS apresentam diferentes pesos. Neste contexto, apenas o *Upload*, devido a razões apresentadas anteriormente neste documento possui um peso inferior relativamente às restantes métricas avaliadas.

Tabela 17 - Peso de cada métrica.

Métrica	Peso
Download	**30%**
Upload	**10%**
Perda de Pacotes	**30%**
Latência	**30%**

Em todas as métricas são considerados 5 níveis qualitativos. A cada um dos mesmos é atribuído um nível quantitativo, sendo os seguintes: 10, 20, 30, 40, 50 (do valor qualitativo mais baixo para o maior). Após ser efetuado o mapeamento dos níveis qualitativos em níveis quantitativos é possível agregar os dados e obter um valor ponderado.

Tabela 18 - Mapeamento dos valores qualitativos.

Nível Quantitativo	Nível Qualitativo
50	**Excelente**
40	**Muito Bom**
30	**Bom**
20	**Aceitável**
10	**Mau**

O cálculo da média ponderada $\overline{x_p}$, onde P_i representa o Peso e x_i o Valor Absoluto, é obtido da seguinte forma:

$$\overline{x_p} = \frac{\sum_{i=1}^{n} P_i \times x_i}{\sum_{i=1}^{n} P_i}$$

Por forma a apresentar um exemplo concreto da aplicação desta equação neste contexto considere-se o seguinte:

- Taxa de *Download*: 11.3 Mbits/s (Bom);
- Taxa de *Upload*: 2.9 Mbits/s (Excelente);
- Perda de Pacotes: 2.1% (Muito Bom);

- Latência: 34 ms (Excelente).

Efetuando o cálculo da média ponderada com estes valores obtém-se o seguinte:

$$\overline{x_p} = \frac{30 \times 30 + 10 \times 50 + 30 \times 40 + 30 \times 50}{30 + 10 + 30 + 30} = 41$$

Observando o valor obtido considera-se que o serviço que foi testado, segundo os dados obtidos das métricas que são avaliadas é "Excelente", pois é superior a 40.

5.3.5 Bases de Dados Internas

5.3.5.1 Base de Dados de Preferências

Esta tem como finalidade guardar dados pessoais e preferências do utilizador por forma a criar uma sessão no momento em que este efetua o *login*, permitindo que o mesmo saia da aplicação e use outras sem ser necessário voltar a fazer o *login* quando inicia a aplicação QoS.Test. A sessão criada não expira. Esta é considerada a base de dados de Preferências e é implementada pelo sub-módulo *Session Management*. A API utilizada já foi descrita. Os dados são armazenados associados a uma *key*, que pode ser do tipo *String*, *Integer* ou outro. Na aplicação recorre-se a *keys* do tipo *String*.

Esta base de dados permite guardar os seguintes dados:

- Key=IS_LOGIN: Permite guardar informação sobre o estado do utilizador, se está com o *login* efetuado ou não. Este valor é verificado sempre que a aplicação é iniciada;
- Key=KEY_PASS: Permite guardar a *password* do utilizador, no momento em que efetua o *login*;
- Key=Key*_EMAIL*: Permite guardar o *email* com que o utilizador efetuou o *login*;
- Key=KEY_FNAM: Guarda o primeiro nome do utilizador;
- Key=KEY_LNAM: Guarda o último nome do utilizador;
- Key=KEY_DEVID: Armazena o IMEI do dispositivo;

- Key=KEY_AUTO_*DOWNLOAD*: Guarda a informação relativa à preferência do utilizador (ativo ou não) face ao envio automático dos resultados dos testes para o servidor quando estes são efetuados;
- Key=KEY_AUTO_TEST: Guarda a informação relativo à preferência do utilizador (ativo ou não) face à execução de testes periódicos;
- Key=AUTO_TEST_PERIOD: No caso de a opção dos testes periódicos estar ativa, esta *key* permite guardar o intervalo de tempo que decorre até ao próximo teste.

Com os dados armazenados é possível verificar se a componente de teste periódicos está ativa e se os resultados devem ser enviados para o servidor no fim de cada teste. Valores como o *email* e o IMEI do dispositivo são essenciais para efetuar a comunicação com o servidor.

5.3.5.2 Base de Dados Local

À semelhança do que acontece no servidor existe uma base de dados local para armazenar os dados na aplicação. Isto permite salvaguardar os resultados dos testes que os utilizadores realizam. Por questão de melhorar o processamento e não sobrecarregar o dispositivo a base de dados só permite armazenar um máximo de 50 resultados.

Esta foi desenvolvida em SQLite, que é um sistema de gestão de bases de dados relacionais. A sua implementação e gestão é feita pelo módulo *Database Management*. O Android disponibiliza uma API nativa para o uso do SQLite, tal como já foi referido anteriormente.

O modelo de dados é constituído por apenas uma tabela, "*data_values*", possibilitando uma gestão simples, mais concretamente nas operações de inserção, leitura e substituição. Permite armazenar os dados recolhidos nas fases 1 e 2. Este conjunto de valores permite depois construir as listas de resultados que são apresentadas ao utilizador para consulta.

O armazenamento de dados nesta base de dados corresponde à fase 3, de todo o procedimento de execução de um teste. Na Fig. 47 são apresentados os dados que são armazenados.

data_values
Id (*Chave Primária*)
Último Nome
Tipo de Rede
ISP
Latitude
Longitude
Timestamp
Débito Download
Duração do Teste de Download
Tamanho de Dados Recebido em Download
Débito Upload
Duração do Teste de Upload
Tamanho de Dados Recebidos em Upload
RTT Máximo
RTT Mínimo
RTT Médio
Desvio Padrão
Percentagem de Perda de Pacotes
Pacotes Enviados
Pacotes Recebidos
Duração do Teste de Perda de Pacotes

Figura 47 - Tabela *data_values* da base de dados local.

5.3.6 Comunicação com o Servidor

Tal como foi descrito anteriormente, a comunicação com o servidor é feita através do sub-módulo *ServerCom Management*. Este recebe os dados que foram recolhidos pela aplicação nas fases 1 e 2, apresentadas anteriormente. A comunicação é feita usando o protocolo HTTP, recorrendo ao método POST. Os dados são enviados e recebidos em JSON.

A comunicação com o servidor deve ser segura, recorrendo ao HTTPS, implementado *sockets* seguros (SSL). Contudo, no contexto do projeto não foi possível recorrer a esta implementação dado que o servidor usado (urano.dsi.uminho.pt) possui apenas certificados auto-assinados, o que do ponto de

vista da criptografia não tem valor, uma vez que o certificado não foi assinado por um CA (*Certification Authority*). No entanto, dado o contexto do projeto e o tipo de dados trocados com o servidor não se considera crítico o não uso de uma conexão totalmente segura. Por outro lado, dado que a *password* é o elemento mais fundamental e que tem que ser mantido privado, tanto do ponto do vista dos administradores como terceiros, foi usado uma função de *hash* para codificar a *password* sempre que esta é trocada entre a aplicação e o servidor. Uma função de *hash* é unidirecional e é impossível de inverter. No projeto foi usado o SHA-1 (*Secure Hash Algorithm*). Este produz um mensagem com 160 bits.

Existe um conjunto total de 13 operações de comunicação com o servidor, que se dividem por três grupos de comunicação: troca de informação pessoal, envio de dados de QoS e recepção de dados de QoS.

Relativamente à troca de informação pessoal existem duas operações concretas:

- ServerUserRegistration: Comunicação que permite à aplicação executar a operação de registo de um utilizador. Os dados enviados para o servidor são: primeiro nome, último nome, *email*, *password* e IMEI;
- ServerUserLogin: Operação de execução do *login*. Os dados enviados são: *email* e *password*.

O envio de dados de QoS designa o conjunto de operações que permite à aplicação enviar para o servidor os resultados dos testes de QoS. O conjunto de dados trocado é semelhante para todas as operações diferindo apenas nos dados relativos à fase 2, que são específicos de cada um dos testes. Os dados enviados, não recolhidos na fase 2 são: *email*, ISP, IMEI, tipo de rede, latitude, longitude, *timestamp*. À exceção do *email*, que está armazenado na base de dados de preferências, todos os outros dados são recolhidos na fase 1 de um teste de QoS. As operações dentro deste grupo são as seguintes:

- Server*Download*Latency: Operação de envio de dados relativos ao teste de latência. Além dos dados mencionados anteriormente, também são enviados os seguintes: RTT médio, RTT máximo, RTT mínimo e o desvio padrão;
- Server*Download*PacketLoss: Operação de envio de dados relativos ao teste de perda de pacotes. Além dos dados mencionados anteriormente,

também são enviados os seguintes: pacotes enviados, pacotes recebidos, percentagem de perda de pacotes e a duração do teste;

- Server*Download*Throughput*Download*: Operação de envio de dados relativos ao teste de *download*. Além dos dados mencionados anteriormente, também são enviados os seguintes: débito, duração do teste e quantidade de dados recebidos;
- Server*Download*Throughput*Download*: Operação de envio de dados relativos ao teste de *upload*. Além dos dados mencionados anteriormente, também são enviados os seguintes: débito, duração do teste e a quantidade de dados enviados;
- Server*Download*FullQoS: Operação de envio de dados relativos *Full QoS Test*. Os dados enviados são relativos a todos as métricas, ou seja, todos aqueles que foram enviados nos métodos anteriores;
- Server*Download*PeriodicTest: Operação de envio de dados relativos ao teste periódico. Neste caso são enviados apenas os dados relativos às métricas da latência e perda de pacotes.

Para certos tipos de execução, nomeadamente cálculos estatísticos, é necessário que a aplicação possa aceder aos resultados armazenados na base de dados do servidor, para tal existe um conjunto de operações específicas para a recepção de dados de QoS, que são as seguintes:

- ServerLatencyResults: Conjunto de resultados de latência de um determinado utilizador. Apenas é enviado o *email* como parâmetro para o servidor e este devolve os seguintes dados: RTT médio, RTT máximo, RTT mínimo, desvio padrão, *timestamp*, latitude e longitude.
- ServerPacketLossResults: Conjunto de resultados de perda de pacotes de um determinado utilizador. Apenas é enviado o *email* como parâmetro para o servidor e este devolve os seguintes dados: pacotes enviados, pacotes recebidos, percentagem de perda de pacotes, duração do teste, *timestamp*, latitude e longitude.
- ServerThroughput*Download*Results: Conjunto de resultados de *download* de um determinado utilizador. Apenas é enviado o *email* como parâmetro para o servidor e este devolve os seguintes dados: débito, duração do teste,

quantidade de dados recebidos, duração do teste, *timestamp*, latitude e longitude.

- ServerThroughput*Download*Results: Conjunto de resultados de *upload* de um determinado utilizador. Apenas é enviado o *email* como parâmetro para o servidor e este devolve os seguintes dados: débito, duração do teste, quantidade de dados enviados, *timestamp*, latitude e longitude.

5.3.6.1 Protocolo de comunicação

Para cada uma das operações descritas acima, existe uma função no *Web Service* para atender o devido pedido.

Os dados são enviados segundo o protocolo HTTP, via POST. Os parâmetros são enviados em JSON. A cada parâmetro é associada uma etiqueta que o permite identificar. A Fig. 48 apresenta a estrutura de uma mensagem real enviada em JSON. Neste exemplo concreto é apresentada uma mensagem com dados relativos a um teste de latência. Toda a troca de dados com o servidor é efetuada segundo este modelo.


```
{
"user*Email*":"joao.rpvilela@gmail.com",
"networkInfo":"HSPA+",
"isp":"vodafone P",
"avgRTT":"123.42",
"maxRTT":"132.13",
"minRTT":"92.13",
"devRTT":"23.13",
"deviceID":"9237727351722",
"timeStamp":"2014/30/04",
"latitude":"41.2414145",
"longitude":"-9.64338267"
}
```


Figura 48 – Exemplo de mensagem em JSON (Latência).

Qualquer comunicação com o servidor é sujeita a uma resposta. A estrutura de uma resposta em JSON contém três parâmetros:

- *status*: Este parâmetro contém um código usado no protocolo HTTP. São usados três códigos distintos: 200, 400, 406 ou 412;
- *status_message*: Descrição do status. Cada código apresentado anteriormente tem uma descrição associada: "*OK*" (200), "*Bad Request*" (400), "*Not Acceptable*" (406) e "*Precondition Failed*" (412). O primeiro é usado em comunicações bem sucedidas. O segundo é usado quando um pedido é mal efetuado, podendo dever-se a um erro numa das etiquetas associadas a um parâmetro ou existir um parâmetro não preenchido (*null*), etc. O terceiro refere-se a situações em que o pedido é inválido, como por exemplo, pedir dados de um utilizador que não existe na base de dados. O último refere-se a situações onde a inserção ou leitura de valores da base de dados falha;
- *data*: No caso de haver dados a serem devolvidos ao servidor são enviados neste campo. Se a resposta não necessitar de conter dados, este parâmetro é enviado a "*null*".

Sempre que é trocada informação com o servidor, o "*status*" é o primeiro campo a ser verificado permitindo controlar a comunicação, apurando se esta foi bem sucedida ou não.

O diagrama apresentado na Fig. 49 traduz o modelo de comunicação com o servidor. A comunicação é feita de forma transparente, o modelo é homogéneo para qualquer um dos métodos usados. Como foi descrito, o *Web Service* é do tipo REST e o acesso aos recursos do servidor são feitos através de um URI específico para cada tipo de comunicação. A cada método apresentado anteriormente, relativos ao *ServerCom Management*, existe o respectivo recurso do lado do servidor.

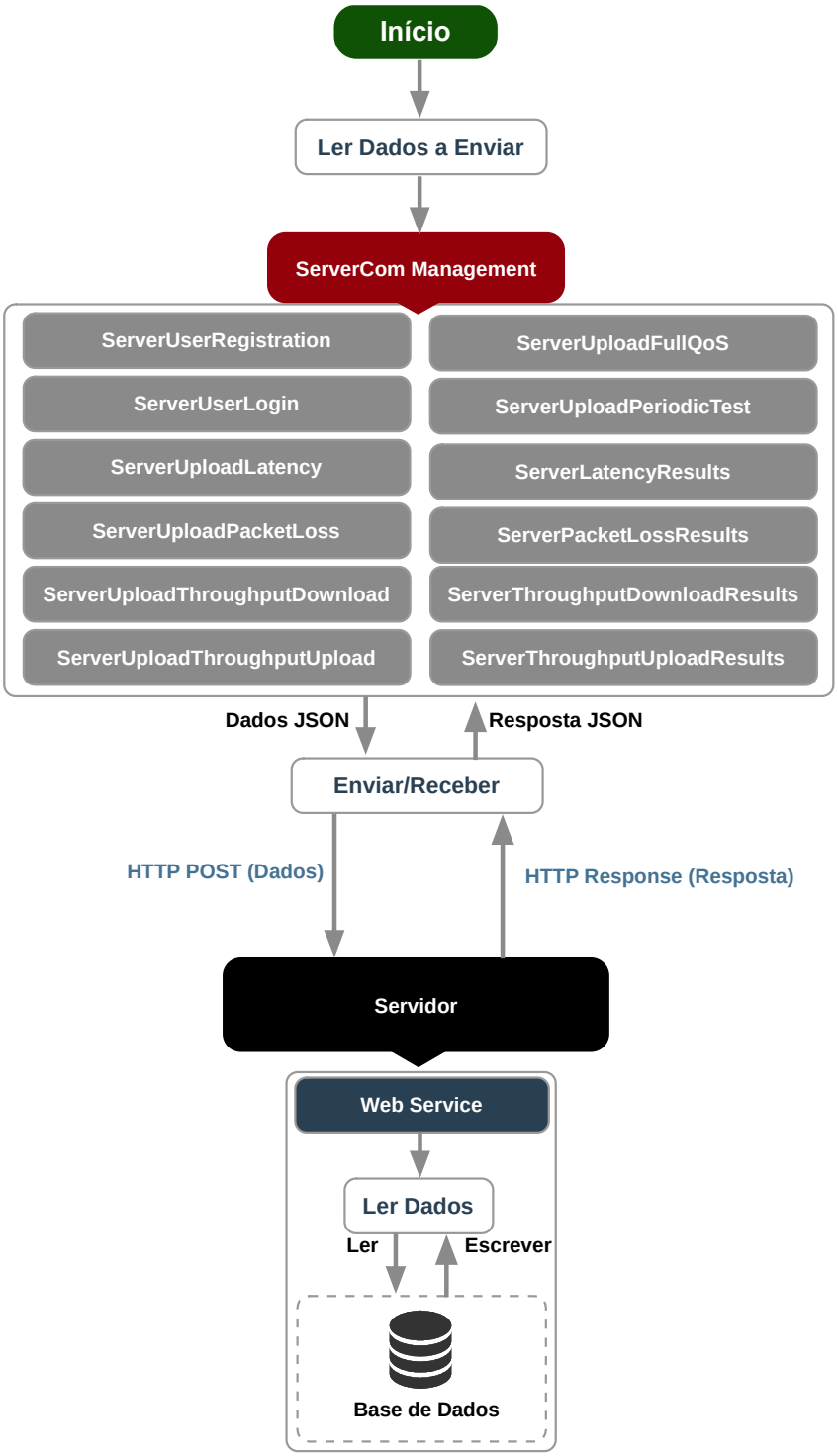


Figura 49 – Modelo de Comunicação com o Servidor

5.4 Servidor

O servidor é o componente de apoio e gestão de dados do sistema colaborativo. O servidor estende as capacidades não só da aplicação mas também de todo o sistema. Este possui quatro elementos: *Web Service*, Base de Dados, Servidor de Testes, estes ao nível da comunicação e apoio à aplicação, e por fim o *Web Site*, que é um elemento muito importante para os objetivos do desenvolvimento deste projeto.

O servidor é uma máquina localizada nos serviços da Universidade do Minho, gerido pelo suporte técnico dos serviços de informática do Departamento de Sistemas de Informação. O acesso ao servidor é feito de forma confidencial, através do protocolo SFTP (*Secure File Transfer Protocol*). Apenas os administradores possuem credenciais de acesso.

Quando se refere o servidor de testes, refere-se apenas à capacidade do servidor aceitar a recepção de um ficheiro de texto enviado pela aplicação, para a avaliação da métrica de *upload*.

Ao longo deste capítulo serão explorados e descritos os elementos do servidor bem como o papel destes no sistema.

5.4.1 Web Service

O *Web Service* é uma das principais ferramentas de suporte à aplicação. Foi desenvolvido em PHP e implementa um modelo de comunicação REST com recurso a JSON, tal como já foi abordado.

Contém dois módulos internos:

- RestClass.php: É um ficheiro PHP, que contém uma classe com várias funções que permitem fazer a recepção dos pedidos e envio de respostas, bem como o descapsular e encapsular os dados.
- WS_QoS.php: Contém todos os recursos que pode ser acedidos. Cada recurso é identificado por um função descrita em PHP. Dependendo do tipo de operação é ou não feita a comunicação com a base de dados.

A Tab. 19 apresenta os recursos disponíveis no *Web Service*, bem como a forma como estes são acedidos através dos respectivos URIs.

Tabela 19 - Recursos do *Web Service*.

URI	Descrição
urano.dsi.uminho.pt/qossystem/WSQoS/Users	Registo de utilizadores.
urano.dsi.uminho.pt/qossystem/WSQoS/Login	Login dos utilizadores registados.
urano.dsi.uminho.pt/qossystem/WSQoS/ UploadLatency	Envio de dados relativos aos testes de latência.
urano.dsi.uminho.pt/qossystem/WSQoS/ UploadPacketLoss	Envio de dados relativos aos testes de perda de pacotes.
urano.dsi.uminho.pt/qossystem/WSQoS/ UploadThroughputDw	Envio de dados relativos aos testes de *download*.
urano.dsi.uminho.pt/qossystem/WSQoS/ UploadThroughputUp	Envio de dados relativos aos testes de *upload*.
urano.dsi.uminho.pt/qossystem/WSQoS/ UploadFullQoS	Envio de dados relativos aos testes de Full QoS.
urano.dsi.uminho.pt/qossystem/WSQoS/UploadPeriodic	Envio de dados relativos aos testes periódicos.
urano.dsi.uminho.pt/qossystem/WSQoS/LatencyResults	Leitura de dados de latência da base de dados.
urano.dsi.uminho.pt/qossystem/WSQoS/PacketLossResults	Leitura de dados de perda de pacotes da base de dados.
urano.dsi.uminho.pt/qossystem/WSQoS/DownloadResults	Leitura de dados de *download* da base de dados.
urano.dsi.uminho.pt/qossystem/WSQoS/UploadResults	Leitura de dados de *upload* da base de dados.

A ligação com a base de dados é feita através do MySQLi. É uma extensão nativa do PHP que permite a gestão de bases de dados relacionais MySQL.

5.4.2 Modelo de Dados

A base de dados no servidor é do tipo relacional e foi desenvolvida em MySQL. Tem uma estrutura simples, como apenas duas tabelas. Uma denomina-se de *user*, que permite guardar a informação do utilizador e a outra denomina-se de *teste_data* e permite guardar toda a informação relativa aos testes de QoS realizados.

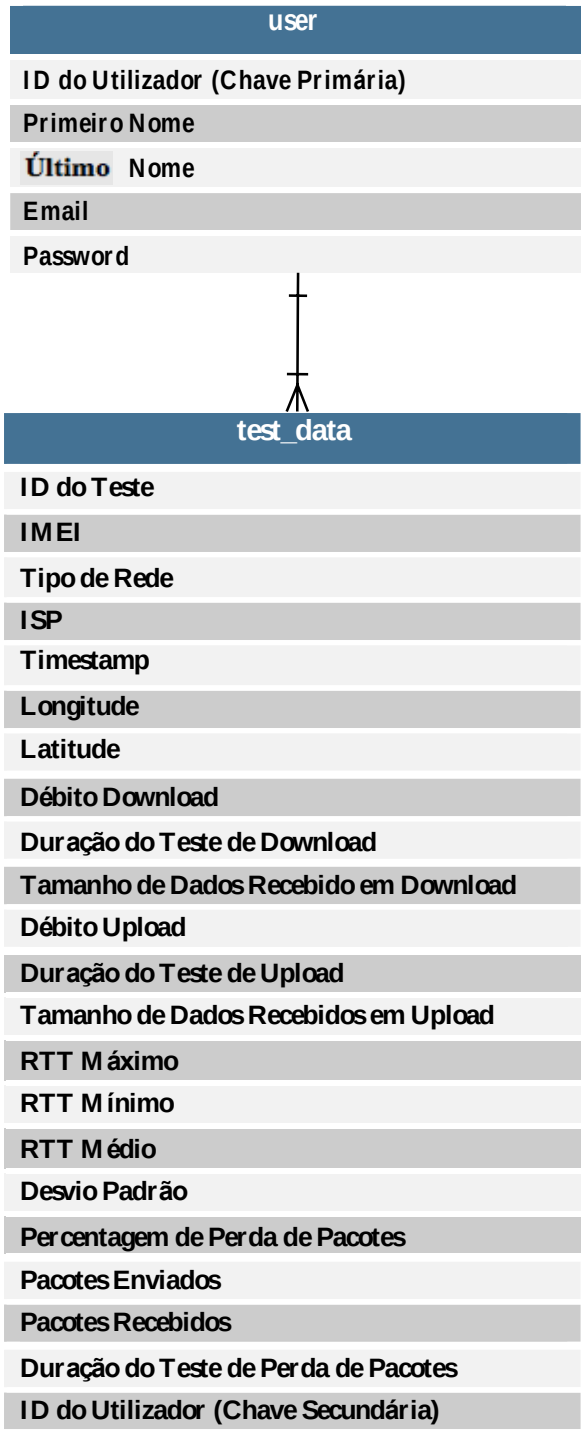


Figura 50 - DER da Base de Dados do Servidor

5.4.3 Cliente Web

O servidor *Web* permite que os utilizadores possam analisar os dados recolhidos, através de cálculos estatísticos, entre outros. Foi desenvolvido usando páginas em PHP, recorrendo ao HTML e Javascript para a componente gráfica.

Todo o *layout* do cliente *Web* foi criado usando o Bootstrap 3.2.0.

Os utilizadores registados, fazendo o *login*, têm acesso a todos os seus resultados, por outro lado, utilizadores não registados também têm acesso aos dados, no entanto são dados generalizados sem qualquer ligação aos utilizadores em que tiveram origem. O objetivo é ter uma plataforma de agregação de dados para consulta livre, por forma a que se possa observar com detalhe a qualidade de serviço de diferentes operadores em diferentes localizações geográficas.

A página principal (ver Fig. 51 e Fig. 52) permite aos utilizadores conhecerem um pouco mais sobre o sistema, introduz a aplicação móvel e apresenta as suas principais capacidades e aborda as funcionalidades do *Web Site*.

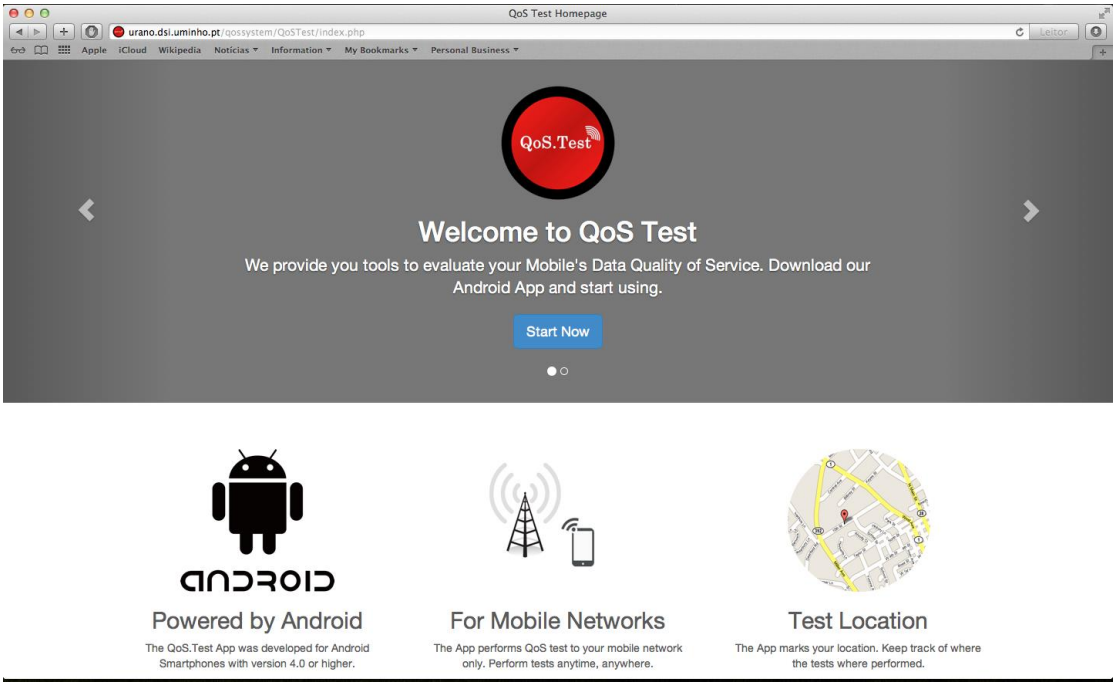


Figura 51 - Página inicial do cliente Web.

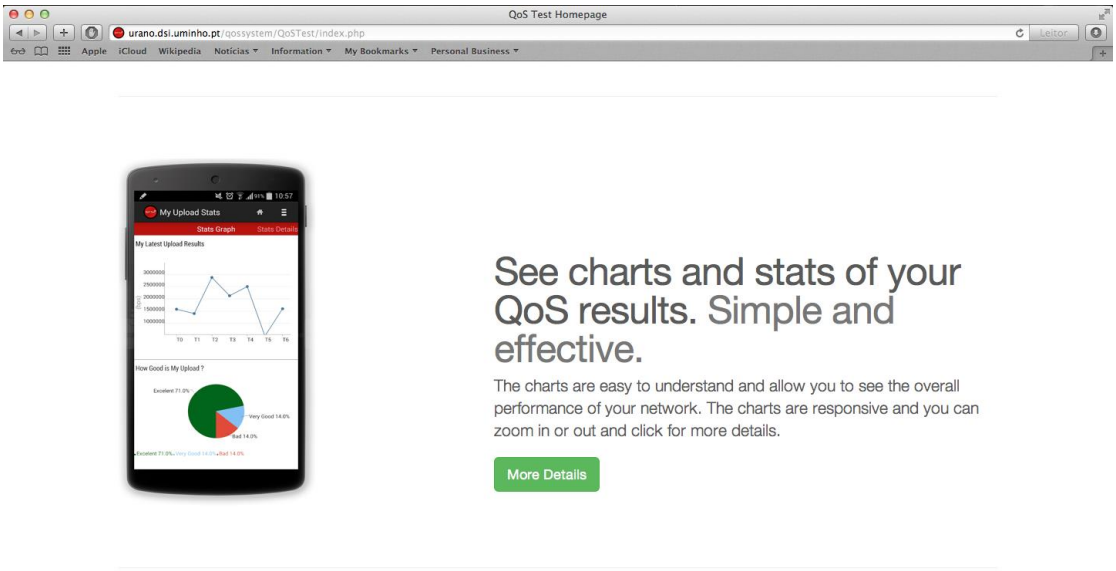


Figura 52 – Exemplo de informação apresentada no cliente Web.

Tratando-se de uma aplicação para distribuição ao público, uma das páginas do site contém uma breve descrição do projeto, explicando que é realizado no âmbito de uma tese de mestrado da Universidade do Minho. Foi criado também um *disclaimer*, que contém um conjunto de cláusulas sobre a utilização da aplicação. Desresponsabiliza a Universidade do Minho, fica estabelecido de que se trata de um projeto universitário sem qualquer fim lucrativo, a informação fornecida pelos utilizadores não é fornecida a terceiros e fica definido o modelo de avaliação qualitativa usado nos testes às diferentes métricas. A Fig. 53 apresenta a página que contém a informação sobre o projeto e a parte inicial do *disclaimer*. Esta secção tem a denominação de "*About*".

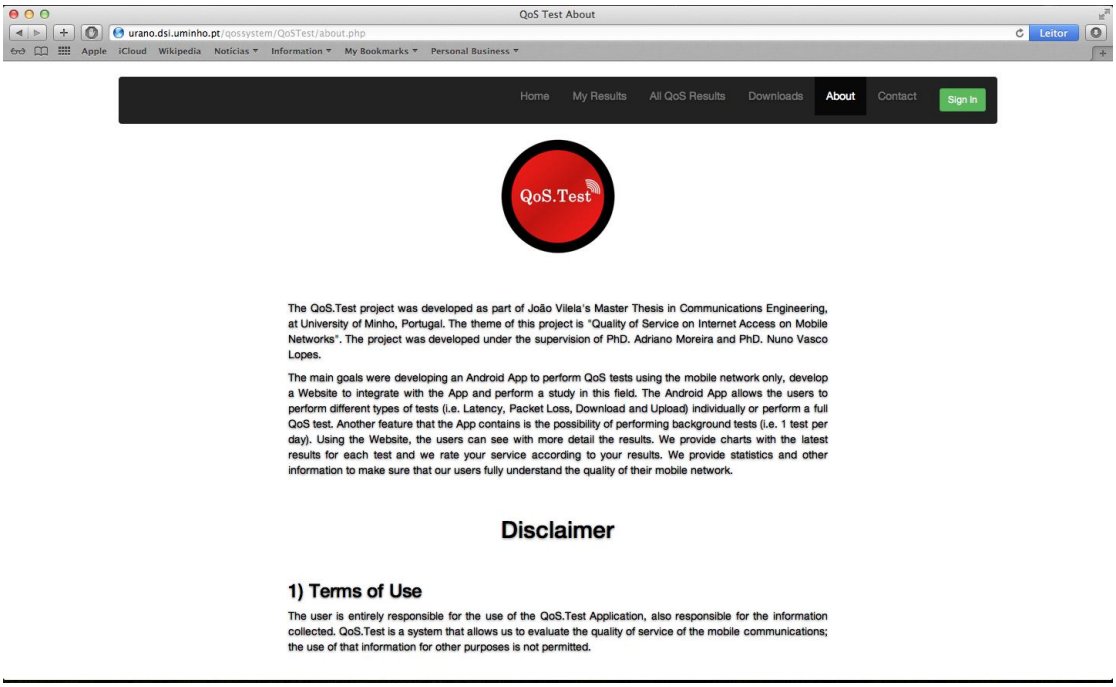


Figura 53 – Descrição do projeto e início do *disclaimer*.

Nos resultados na área pessoal é possível aceder à secção denominada de *My QoS Results Overview*. Esta apresenta um conjunto de quatro gráficos circulares com um resumo da qualidade do serviço prestado ao utilizador, onde cada um representa uma das quatro métricas avaliadas. Nestes é observável a percentagem de testes com uma determinada qualidade.

Para cada métrica existe uma secção específica: *Latency Results*, *Packet Loss Results*, *Download Results* e *Upload Results*. Independentemente de qual a métrica, é apresentado um agregado de informação:

- São exibidos cálculos estatísticos, feitos a partir de todos os dados na base de dados associados ao utilizador. Geralmente o valor médio, máximo, mínimo e o número total de testes (ver Fig. 54). Adicionalmente, para a latência é apresentada a média total do desvio padrão;
- É também apresentado o gráfico circular com a tabela de referência que permite atribuir um valor qualitativo, Excelente, Muito Bom, Bom, Aceitável ou Mau, aos valores quantitativos (ver Fig. 54);
- Pode-se observar um gráfico com os valores máximo, mínimo e médio em cada mês do ano (ver Fig. 55);

- É apresentado um gráfico com os últimos 30 resultados medidos (ver Fig. 56);
- Lista de todos os resultados na base de dados, com acesso à localização exata do local onde ocorreu cada teste, marcado num mapa da Google (ver Fig. 57);
- Para cada métrica é apresentado um *heatmap*, recorrendo às API's da Google, cujas cores permitem identificar a qualidade numa determinada área;

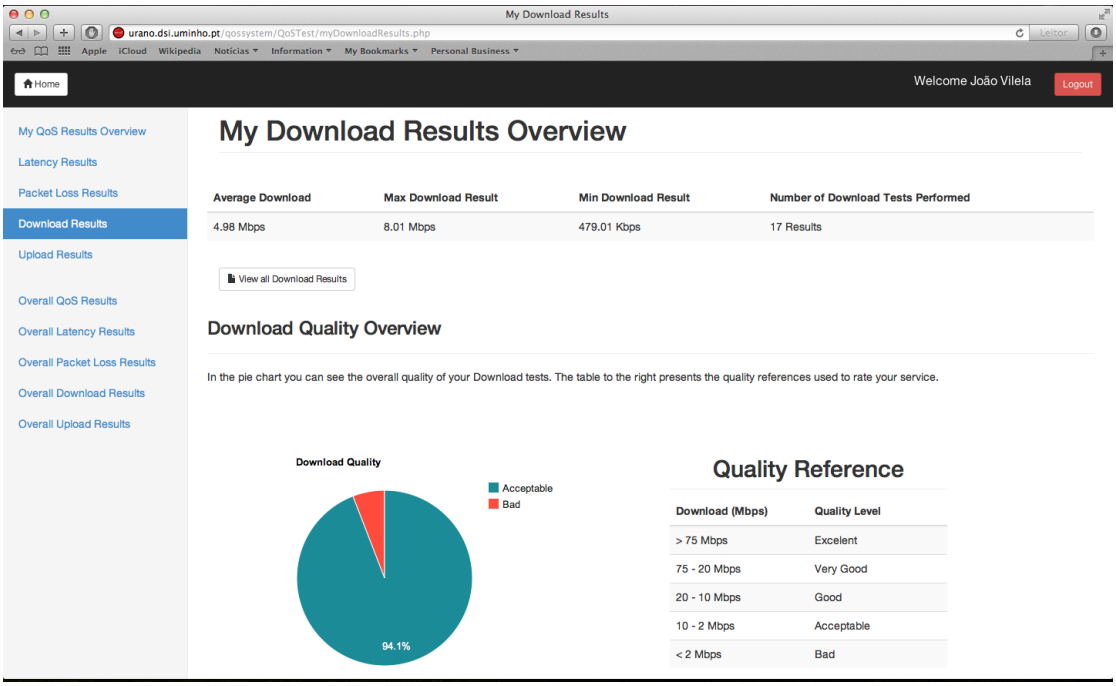


Figura 54 – Resultados de pessoais de *download* com gráfico circular e cálculos estatísticos.

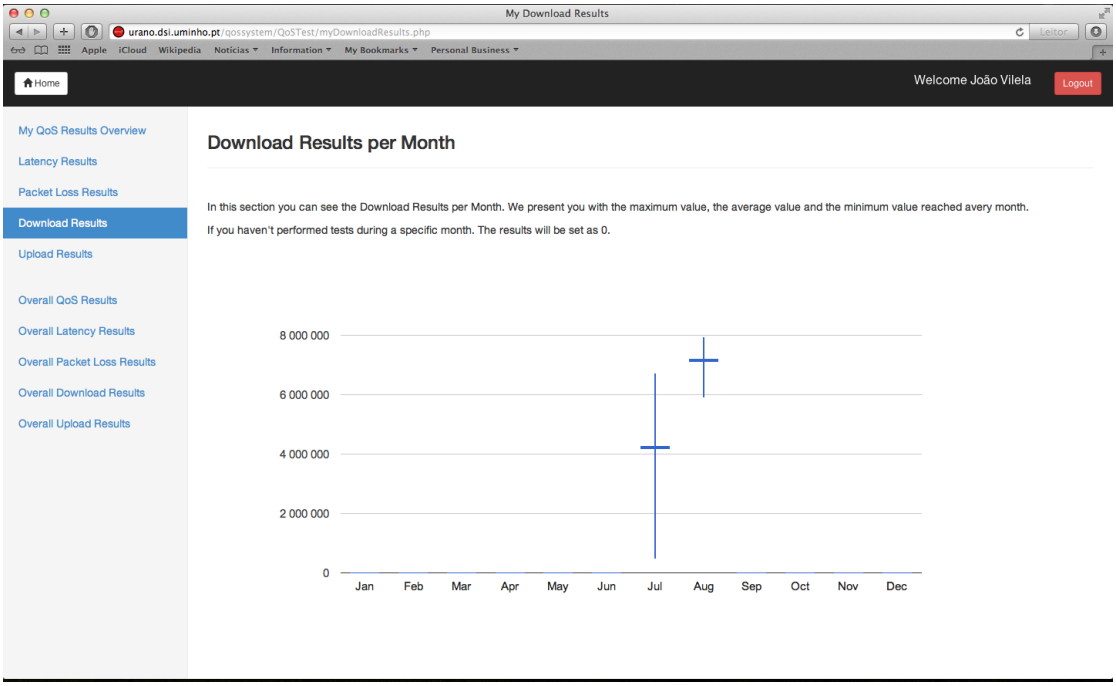


Figura 55 - Gráfico de valores médio, máximo e mínimo de *download* por mês.

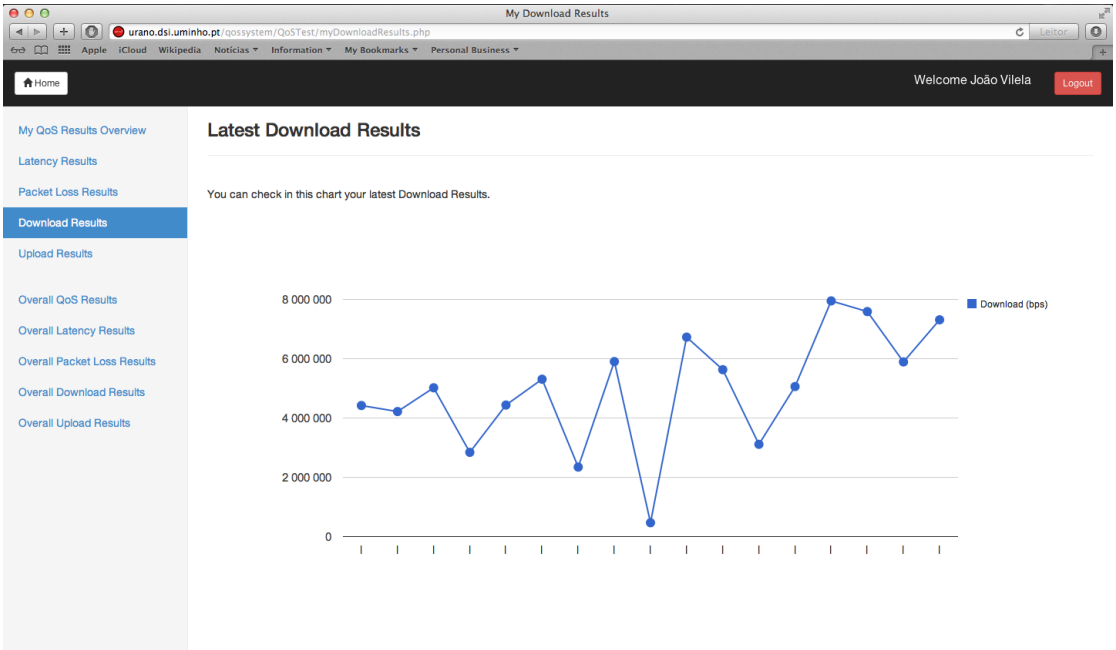


Figura 56 - Gráfico apresentando últimos 30 resultados de *download*.

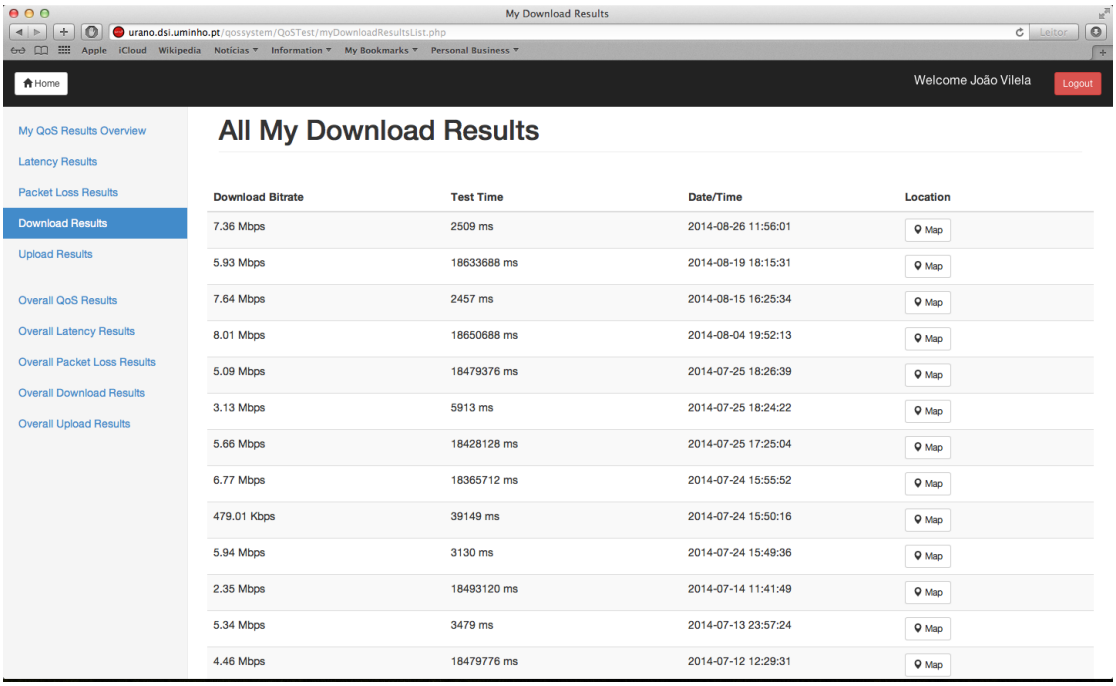


Figura 57 – Lista com todos os resultados de *download*.

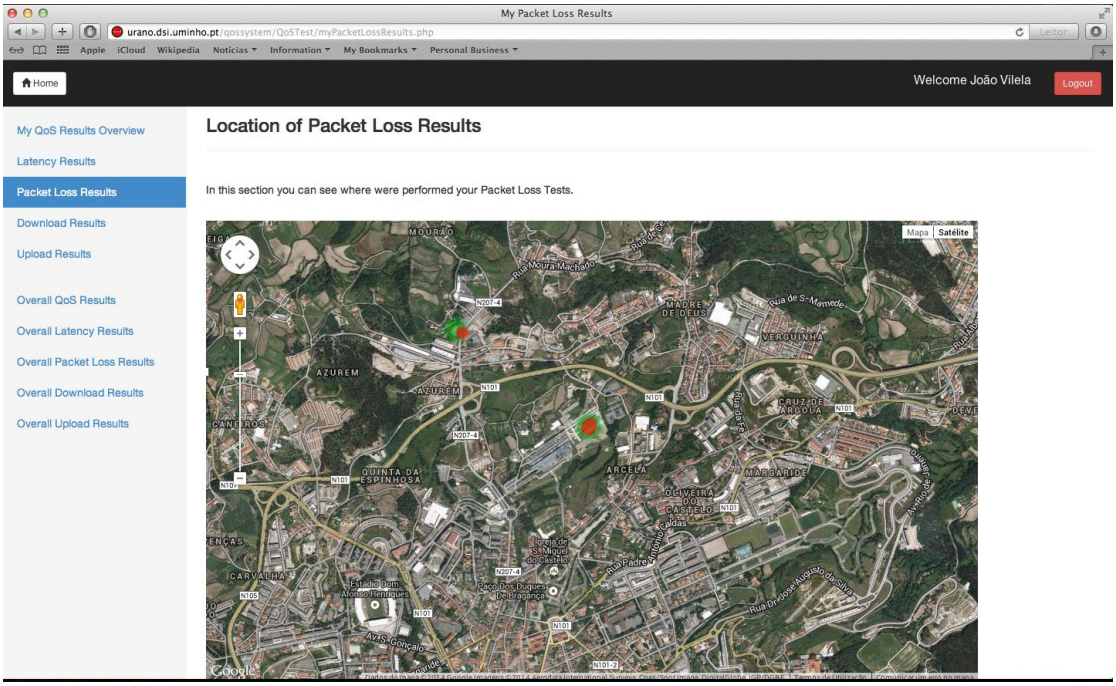


Figura 58 – *Heatmap* construído com resultados de latência.

Na área de acesso livre, não necessitando de *login*, é possível observar um conjunto de dados e resultados como os apresentados anteriormente. As únicas diferenças prendem-se pelo facto de os dados agora não serem sobre um utilizador, mas sobre todos os utilizadores do serviço. Em cada métrica os resultados são

separados em secções sobre determinados operadores nacionais: Vodafone, MEO e NOS.

6. Avaliação

6.1 Desenho da Experiência

A divulgação do sistema consiste num passo importante para o objetivo pretendido com a realização deste projeto. Tentar fazer chegar a aplicação ao maior número de utilizadores possível permitiria construir uma vasta base de dados, cuja relevância para a avaliação de QoS seria essencial. Inicialmente seria bastante positivo ter uma elevada expressão a nível nacional, para que depois se pudesse apostar numa disseminação a nível internacional.

Neste domínio são ponderadas algumas estratégias iniciais de divulgação:

- A publicação da aplicação QoS.Test no Google Play permite uma maior difusão, visto que a mesma é indexada nas pesquisas efetuadas nesta plataforma. O facto de estar ali publicada dá uma maior segurança e conforto aos utilizadores que instalam a aplicação, assegurando que houve algum controlo sobre a forma como foi publicada através da plataforma oficial para dispositivos que correm o sistema operativo Android.
- Atualmente, as redes sociais providenciam um instrumento muito poderoso para a divulgação de qualquer produto. Divulgar a aplicação recorrendo ao Facebook poderá dar uma visibilidade acrescida e ter um elevado impacto na difusão do sistema. O Facebook, no inicio do ano de 2014 possuía cerca de 1.3 mil milhões de utilizadores a nível mundial[53], 4.7 milhões em Portugal [54]. É uma rede social focada sobretudo para o *networking* pessoal, mas com grandes possibilidades no sector comercial.
- A Universidade do Minho possui um Gabinete de Comunicação, Informação e Imagem, cujas funções são a divulgação e comunicação da imagem da Universidade. Definem estratégias para garantir um contacto com a comunicação social, em diversos meios, para informar e promover atividades ou projetos desenvolvidos na Academia [56]. Recorrer a este

serviço para a divulgação do sistema é uma possibilidade que permitiria uma difusão mais clássica, recorrendo a órgãos de comunicação social, como jornais, revistas ou televisão.

- Dentro da própria Universidade e nos arredores a divulgação através de posters (ver Fig. 59) é uma possibilidade que pode ter uma elevado impacto sobretudo, tratando-se de um projeto interno. A estratégia passa por escolher locais alvo, com grande afluência e potencialmente com um público com conhecimento na área de telecomunicações ou *software*.
- Ainda, em domínio interno, recorrer à divulgação por *email* institucional é uma estratégia que permitirá fazer chegar a informação sobre a aplicação a todos os elementos da Universidade do Minho.

Este conjunto de estratégias visam a divulgação da aplicação a um nível institucional, mas, recorrendo a novas plataformas é possível uma difusão alargada do sistema, com vista a cativar os utilizadores, não só para usar, mas para consultar os dados recolhidos, através do cliente *Web*. É importante consciencializar o público para o que representa a qualidade de serviço e poder ter uma base de dados com uma elevada quantidade de informação, sobretudo a nível nacional.

6.2 Resultados Obtidos

Após a conclusão da aplicação, que data de Julho de 2014, alguns utilizadores foram efetuando diversos testes que contribuíram para o crescimento da base de dados. Desde esse período foram recolhidos:

- 264 resultados relativos a testes de latência;
- 242 resultados relativos a testes de perda de pacotes;
- 51 resultados relativos a testes de *download*;
- 45 resultados de testes de *upload*.

A divulgação pelos meios descritos acima foi iniciada no fim do mês de Setembro de 2014, com a exceção da publicação da aplicação no Google Play, que foi efectuada no início do mês de Agosto.

Até ao dia da escrita deste capítulo os resultados em termos de instalações da aplicação por dispositivo, segundo os dados fornecidos pelo Google Play Developer Console, apontam para 27 instalações no total.

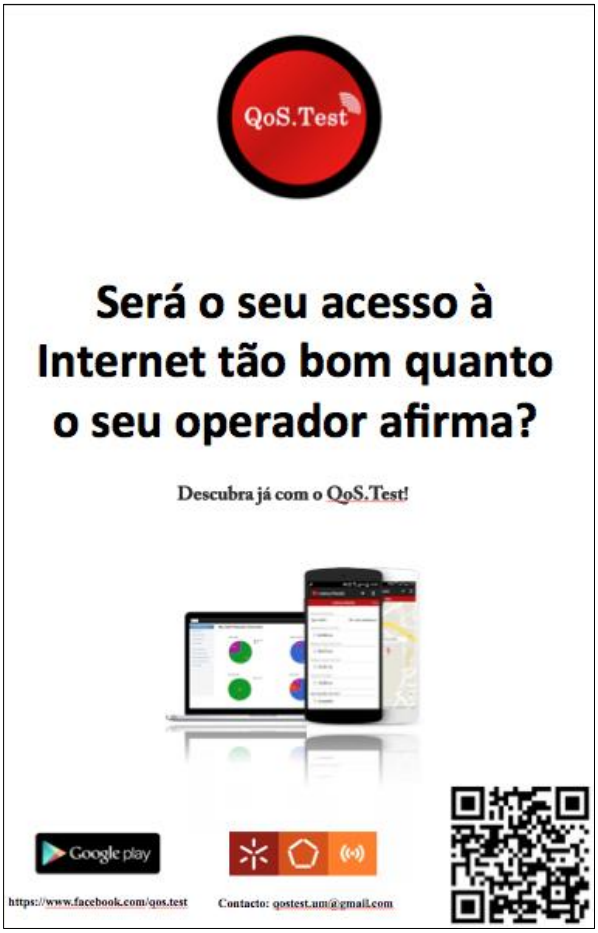


Figura 59 - Poster de divulgação da aplicação.

6.3 Análise Crítica dos Resultados

Após a realização das várias fases de divulgação da aplicação verificou-se uma fraca adesão, onde o número de instalações não ultrapassou as 27. Quando se iniciou este projeto sabia-se à partida que se estaria a entrar numa secção do mercado bastante bem estabelecida, dominada por algumas ferramentas com grande expressão, nomeadamente o SpeedTest (da Ookla). Competir com estas ferramentas seria uma tarefa difícil. No entanto, a aplicação QoS.Test oferece uma solução de avaliação de

qualidade de serviço bem diferente das outras ferramentas, disponibilizando um sistema bastante mais completo e ao mesmo tempo mais complexo. O objetivo sempre foi permitir aos utilizadores avaliarem diferentes métricas de forma conjunta ou independente, de forma instantânea ou periódica, podendo usufruir de um serviço *web* para consulta e análise detalhada dos resultados.

Um dos pontos que mais se destaca é talvez o termo QoS, onde grande maioria dos utilizadores não tem noção do que a sigla representa. Quando geralmente se pensa em avaliar uma conexão à Internet geralmente aquilo que se avalia são os débitos, *download* e *upload*, mas, como foi sendo descrito ao longo deste documento, existem outros factores que estão intrinsecamente ligados à qualidade com que um determinado serviço é prestado.

Para poder estabelecer o QoS.Test como uma referência na avaliação da QoS pelo menos em domínio nacional, para fazer face à competição de outras ferramentas, inclusive ao NETmede da ANACOM, seria necessário apostar numa divulgação mais feroz, insistindo em meios de comunicação com maior expressão, como a rádio, imprensa escrita ou televisão.

7. Conclusões e Trabalho Futuro

A área do estudo da Qualidade de Serviço é cada vez mais importante no domínio das atuais redes de comunicação. A evolução tecnológica leva a que surjam novos serviços e produtos, que requerem cada vez mais qualidade para garantir um certo nível de desempenho. Atualmente estamos perante uma revolução no paradigma das comunicações móveis, com equipamentos de alto desempenho com elevada capacidade de processamento. A ligação dos mesmo com a Internet permite todo um novo leque de possibilidades no que toca ao envio e recepção de dados. As redes celulares, desde os extremos até ao núcleo, têm que ter a capacidade de responder a estas novas exigências. Garantir que as mesmas são capazes de providenciar um serviço com elevada qualidade é extremamente importante e necessário. Estas têm que acompanhar o progresso tecnológico e requerem uma constante evolução, sendo que qualquer possibilidade de estagnação é proibitiva. Do ponto de vista comercial, os utilizadores têm cada vez mais acesso a ferramentas e mecanismos de avaliação de um serviço, fazendo com que os operadores apostem cada vez mais na qualidade dos serviços que disponibilizam para fazer face à concorrência, que neste sector é bastante elevada.

Nesta dissertação foi desenvolvido um sistema colaborativo de aferição de QoS na conexão à Internet em redes celulares. É uma plataforma aberta focada para os utilizadores. A recolha dos dados é feita de forma independente pelos utilizadores do sistema. Os dados são publicadas abertamente permitindo um acesso livre. Tem como objetivo ser uma entidade independente, sem qualquer ligação a operadores ou agências de regulamentação.

Inicialmente foi necessário encontrar um conjunto de métricas que permitisse avaliar de forma concisa uma conexão à Internet. Após alguma pesquisa foram escolhidas quatro métricas fundamentais: latência, perda de pacotes, débito em *download* e débito em *upload*. Após este passo foi pensada uma arquitetura geral do

sistema e como é que este iria ser desenvolvido, destacando quais as tecnologias e ferramentas que iriam permitir a elaboração do mesmo.

Depois de identificados os componentes essenciais foi fundamental definir quais as metodologias de teste que permitiriam levar à avaliação das métricas definidas. Usar o ICMP para os testes de latência e perda de pacotes e usar o HTTP para os testes de *download* e *upload* foram as soluções escolhidas, que permitiram desenvolver a componente da aplicação móvel. Para além dos resultados instantâneos derivados de um teste, foi-se um pouco mais além e desenvolveu-se uma componente de avaliação qualitativa e uma componente de análise estatística e gráfica dos resultados providenciando aos utilizadores uma experiência completamente diferente face a outras soluções no mercado. Além da possibilidade de executar os testes instantâneos o utilizador pode usar a componente de testes periódicos. Esta componente executa testes periodicamente entre 18 e 30 horas fornecendo informação vital na evolução da qualidade do serviço do utilizador. Este aspeto inovador é o que torna o sistema único. A aplicação foi desenvolvida usando linguagem Java e também o XML na parte gráfica, recorrendo maioritariamente a API's nativas. Todo o projeto para a plataforma Android foi desenvolvido no IDE oficial, o Eclipse.

Dado que a aplicação móvel tem uma grande dependência das capacidades providenciadas pelo servidor, desenvolver uma comunicação bem definida e organizada foi um elemento importante. A utilização de um *Web Service* REST, com dados em JSON permitiu alcançar os objetivos pretendidos. Esta solução foi analisada e comparada com outras, mas esta acabou por prevalecer, dando garantias na simplicidade e facilidade de implementação e processamento. O *Web Service* foi construído recorrendo ao PHP, uma vez que tem um suporte bastante alargado para este tipo de aplicações e a facilidade na sua implementação, bem como o acesso a documentação tornou esta abordagem a mais indicada.

O armazenamento dos resultados dos testes efetuados pelos utilizadores é o maior contributo de todo o sistema. Agregar estes dados e ter a possibilidade de os analisar de forma detalhada podendo executar um conjunto de cálculos estatísticos é o grande propósito desta dissertação. No que diz respeito à aplicação, os resultados são armazenados numa base de dados local criada em SQLite, possibilitando ao utilizador guardar até um máximo de cinquenta resultados. Do lado do servidor, a base de dados

de todo o sistema permite armazenar todos os dados enviados. Foi desenvolvida em MySQL e possui uma estrutura simples.

O elemento adicional de interação com o sistema e até o principal meio para divulgação do projeto é a interface *Web*. Esta permite aos utilizadores informarem-se um pouco mais sobre todas as capacidades da aplicação e dá a conhecer o conceito por em que se baseia esta dissertação. Para além disso possibilita a que utilizadores registados tenham um controlo total sobre os seus resultados, podendo observar com detalhe os mesmos, recorrendo a gráficos de diferentes tipos, utilizando *heatmaps*, para poder observar com detalhe o estado da sua conexão à Internet em diferentes locais. Utilizadores registados e não registados têm a possibilidade de aceder e analisar todos os dados recolhidos. Os resultados dos testes de todos os utilizadores são abertos e podem ser consultados livremente, não pondo em causa a identidade de quem os gerou. Estes valores são organizados em seções distintas por operador, NOS, Vodafone e MEO.

Após a finalização da dissertação é possível identificar pontos onde se poderia continuar a aperfeiçoar. A metodologia de testes de *download* e *upload* poderia ser melhorada apostando na abordagem alternativa apresentada como estudo neste documento. Desenvolver um algoritmo capaz de possibilitar um teste com a execução dinâmica de *threads*, evitando um valor fixo, permitiria obter outros resultados que seriam comparados com a solução atual. Esta nova metodologia permitira melhorar o desempenho da aplicação.

Visto ser uma aplicação focada para os utilizadores, poder ter uma base de dados mais completa seria uma mais valia, tanto a nível nacional como a nível internacional. Lançar a aplicação num mercado global poderia levar a um impacto no domínio dos serviços de redes móveis. Para tal seria necessário recorrer a servidores internacionais, cuja distância ao utilizador fosse a menor possível no instante do teste, por forma a que fosse possível obter os melhores resultados de QoS. Os testes teriam que ser feitos todos ao mesmo servidor, o que levaria a que o teste de *upload* deixasse de ser efetuado com recurso exclusivo ao servidor da Universidade do Minho.

Aquando dos testes, com os resultados obtidos poderiam ser apresentados mais cálculos, com informação mais detalhada. No projeto não se ficou apenas pela simples avaliação da latência, perda de pacotes, *download* e *upload*, tentou-se em cada teste fornecer informação adicional e relevante para os utilizadores.

A interface gráfica da aplicação poderia ser melhorada, tornando-a mais apelativa a quem a usa. Neste tipo de ferramentas é sempre uma mais valia poder contar com um *layout* aprimorado, podendo atrair mais utilizadores. Por vezes pequenos detalhes fazem toda a diferença.

A comunicação com o servidor não implementa SSL, mas seria uma das prioridades ao nível do melhoramento de todo o sistema. Garantir uma comunicação segura entre a aplicação e o servidor é dos pontos mais importantes. Os dados que circulam entre ambos não são críticos, mas merecem um tratamento seguro e confidencial.

Ao nível dos dados recolhidos seria importante desenvolver novos mecanismos de avaliação de QoS, ou seja, efetuar mais cálculos estatísticos, análise mais detalhadas, permitir mais interação dos utilizadores com os dados, podendo ver com mais pormenor os valores armazenados. Os resultados obtidos pelos utilizadores são a mais valia de todo o projeto. Tornar este sistema uma ferramenta de divulgação e consulta aberta seria um passo muito importante para a educação do público em geral, permitindo um distanciamento das avaliações de QoS, tanto dos operadores como das agências de regulamentação.

Referências

[1] *Series E: Overall Network Operation, Telephone Service, Service Operation And Human Factors*. ITU-T. September 2008.

[2] P. Stangierski, R. Hordyński, I. Kisilowska (2010, May). *Telecommunications: Measuring Quality of Service*. Available: http://www.atkearney.com/paper/-/asset_publisher/dVxv4Hz2h8bS/content/telecommunication-measuring-quality-of-service/10192

[3] D. Balasubramanian. (2006, April 20). *QoS In Cellular Networks*. Available: http://www.cse.wustl.edu/~jain/cse574-06/ftp/cellular_qos/

[4] P. Mohaparta, J. Li, C. Gui, *QoS in Mobile Ad Hoc Networks* in IEEE Wireless Comunnications, University of California, June 2003.

[5] J.M. Cabral. *"Qualidade de Serviço"*. Integração de Sistemas de Comunicação. Universidade do Minho, Guimarães, Portugal, 2013.

[6] (2010, June) *Estudo De Aferição Da Qualidade Do Serviço De Acesso À Internet Banda Larga*, ICP–ANACOM. Available: http://www.anacom.pt/render.jsp?contentId=1052809

[7] (2014) *Global Internet Phenomena Report*. Available: https://www.sandvine.com/*downloads*/general/global-internet-phenomena/2014/1h-2014-global-internet-phenomena-report.pdf

[8] E. M., EL-Ghandour, O. M., Jehan, M. K. *Evaluation of QoS in UMTS backbone network using differentiated services*, National Radio Science Conference, Saad, 2008, (Nrsc),1–11. doi:10.1109/NRSC.2008.4542328

[9] Islam, S. S. *Analysis the impacts of data rates and Forward Access Channel Scheduling on QoS in 3G UMTS network* . IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012. 34–38. doi:10.1109/CYBER.2012.6392522

[10] *Wideband Code Division Multiple Access.* Available: http://www.3gpp.org/ technologies/keywords-acronyms/104-w-cdma

[11] N.V.Lopes. "*Redes3G".* Redes Móveis, page 75. Universidade do Minho, Guimarães, Portugal, 2013.

[12] D. Balasubramanian. (2006, April 20). *QoS In Cellular Networks*. Available: http://www.cse.wustl.edu/~jain/cse574-06/ftp/cellular_qos/

[13] *Quality of Broadband Services in the EU* Report, F. Sam Knows, DOI:10.2759/24341, ISBN 978-92-79-30933-5, March 2012.

[14] *Map of LTE Operators.* Available: http://ltemaps.org/home/operators

[15] *Anuário do Sector das Comunicações 2012 (ANACOM). Available:* http://www.anacom.pt/streaming/anuario_sector_comunicacoes2012.pdf?contentId=1143879&field=ATTACHED_FILE

[16] (2013, September 22). *Africa Development Indicators*. Available: http://data.worldbank.org/data-catalog/ africa-development-indicators

[17] R. Les Cottrell (2013, January 29). *How Bad is Africa's Internet*. Available: http://spectrum.ieee.org/telecom/Internet/how-bad-is-africas-Internet

[18] U. Kalim, L. Cottrel. (2011, August 27). *PingER Metrics Motion Chart*. Available: http://www-iepm.slac.stanford.edu/pinger/pinger-metrics-motion-chart.html

[19] *About SamKnows* . Available: http://www.samknows.com/broadband/index.php

[20] *SamKnows Openness and Transparency.* Available: http://www.samknows.com/broadband/openness_and_transparency

[21] ANACOM (2010, October 25). *Metodologia – Acessos Móveis*. Available: http://www.anacom.pt/render.jsp?categoryId=338858

[22] ICTP-SDU (2005, March). *about PingER.* Available: http://sdu.ictp.it/pinger/pinger.html

[23] D. Coldewey (2013, August 8). *Windows Phone and Android vault ahead in smartphone market share*. Available: http://www.nbcnews.com/technology/windows-phone-android-vault-ahead-smartphone-market-share-6C10871655

[24] R. Llamas, R. Reith, M. Shirer (2013, August 7). *Apple Cedes Market Share in Smartphone Operating System Market as Android Surges and Windows Phone Gains*. Available: http://www.idc.com/getdoc.jsp?containerId=prUS24257413

[25] R. Llamas, R. Reith, M. Shirer (2014). *Smartphones Market OS Share*. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[26] A. Rubin (2007, November 5) *Where is my Gphone*. Available: http://googleblog.blogspot.pt/2007/11/wheres-my-gphone.html

[27] *Eclipse ADT*. Available: http://developer.android.com/tools/sdk/eclipse-adt.html

[28] *Android Studio*. Available: https://developer.android.com/sdk/installing/studio.html

[29] *Android Dashboards.* Available: https://developer.android.com/about/dashboards/index.html?utm_source=ausdroid.net

[30] J. Rosenberg (2012, March 6). *Introducing Google Play: All your entertainment, anywhere you go*. Available: http://googleblog.blogspot.pt/2012/03/introducing-google-play-all-your.html

[31] D. Etherington (2013, March 6). *Google Play Offers Over 5M eBooks And More Than 18M Songs, One Year After Its Rebranding*. Available: http://techcrunch.com/2013/03/06/google-play-offers-over-5m-ebooks-and-more-than-18m-songs-one-year-after-its-rebranding/

[32] *Android Architecture*. Available: http://www.tutorialspoint.com/android/android_architecture.htm

[33] *Porting Android to Devices*. Available: http://source.android.com/devices/index.html

[34] *Android Activities*. Available: http://developer.android.com/guide/components/activities.html

[35] *Android Services.* Available: http://developer.android.com/guide/components/services.html

[36] *Android Intent-Filters*. Available: http://developer.android.com/guide/components/intents-filters.html

[37] *Android Content Providers*. Available: http://developer.android.com/guide/topics/providers/content-providers.html

[38] *Android App Widgets*. Available: http://developer.android.com/guide/topics/appwidgets/index.html

[39] *Android Processes and Threads*. Available: http://developer.android.com/guide/components/processes-and-threads.html

[40] *Android UI Interface Overview*. Available: http://developer.android.com/guide/topics/ui/overview.html

[41] *Web Services Glossary*. Available: http://www.w3.org/TR/ws-gloss/

[42] (2014, October 9). *SOAP*. Available: http://en.wikipedia.org/wiki/SOAP

[43] C. Janssen. *What is Representational State Transfer (REST)?*. Available: http://www.techopedia.com/definition/1312/representational-state-transfer-rest

[44] J. Musser (2010, May 26). *Open API's : State of the Market, May 2010* - http://www.slideshare.net/jmusser/pw-glue-conmay2010

[45] L. Quin (2014, March 10). *Extensible Markup Language*. Available: http://www.w3.org/XML/

[46] *How Can XML be Used*. Available: http://www.w3schools.com/xml/xml_usedfor.asp

[47] *JSON Introduction*. Available: http://www.w3schools.com/json/json_intro.asp

[48] K. Mikoluk (2013, August 16). *JSON vs XML: How JSON is better than XML*. Available: http://www.udemy.com/blog/json-vs-xml/

[49] J. Musser (2011, May 25). *Open API's : State of the Market, May 2011*. Available: http://www.slideshare.net/jmusser/open-apis-state-of-the-market-2011?related=1

[50] Y. Shiotsu (2014, March 19). *Web Development 101: Top Web Development Languages in 2014*. Available: https://www.odesk.com/blog/2014/03/*web*-development-101-top-*web*-development-languages-2014/

[51] C. Buckler (2014, July 22). *The Best Programming Language to Learn in 2014: Mid-Year Update*. Available: http://www.sitepoint.com/best-programming-language-learn-2014-mid-year-update/

[52] K. Mikoluk (2014, May 5). *Top 10 Programming Languages to Learn in 2014*. Available: http://www.udemy.com/blog/best-programming-language/

[53] A. Sedghi (2014, February 4). *Facebook: 10 years of social networking, in numbers*. Available: http://www.theguardian.com/news/datablog/2014/feb/04/facebook-in-numbers-statistics

[54] (2014, February 4). *Facebook com 4.7 milhões de utilizadores em Portugal*. Available: http://www.dn.pt/inicio/economia/interior.aspx?content_id=3667289

[55] (2014). *Quarterly Numbers of LinkedIn Members*. Available: http://www.statista.com/statistics/274050/quarterly-numbers-of-linkedin-members/

[56] *Gabinete de Informação, Comunicação e Imagem da Universidade do Minho.* Available*:* http://www.uminho.pt/uminho/unidades/servicos#GABCOM

[57] (2010, July 28). *Mobile Network Evolution: UMTS*. Available: http://conningtech.blogspot.pt/2010_07_01_archive.html

[58] (2014, September 21). *Universal Mobile Telecommunications System*. Available:http://en.wikipedia.org/wiki/Universal_Mobile_Telecommunications_System

Anexos

Anexo A

1. Arquitetura do UMTS

A arquitetura da Rede UMTS divide-se em 3 domínios distintos: *Mobile Equipment Domain*, *Radio Access Network* e ainda *Core Network*. Cada domínio possui características próprias. Estes são distintos, mas estão interligados para proporcionar um serviço comum. Ao nível da *Radio Access Network* e *Core Network* existe uma convergência de infraestruturas, sendo que o UMTS apresenta novas tecnologias, mas permite a integração com tecnologias do GSM e GPRS.

1.1 UMTS Mobile Equipment Domain

Neste domínio insere-se todo o conjunto de funções que permitem a transmissão rádio e estabelecimento/manutenção de conexões *end-to-end*. O acesso rádio pode ser efetuado de duas formas distintas:

- W-CDMA (*Wideband Code Division Multiple Access*), onde existem duas bandas emparelhadas. Esta técnica usa FDD (*Frequency Division Duplexing*).
- TD-CDMA (*Time Division – Code Division Multiple Access*), onde não existem bandas emparelhadas. Técnica baseada em TDD (*Time Divison Duplexing*).

1.1.1 W-CDMA (FDD)

A técnica FDD separa o espectro em duas bandas distintas, onde o transmissor e o receptor operam com frequências portadoras diferentes. Esta técnica apresenta vantagens quando o tráfego é simétrico, permite também um planeamento rádio mais eficiente, já que as estações, operando em frequências distintas, não se ouvem umas às outras, não havendo interferências.

O W-CDMA é uma técnica de modulação baseada em *spread-spectrum*, onde um canal apresenta uma largura de banda muito superior às dos dados que necessita

de acomodar para transmitir. Em vez de a cada conexão ser garantido uma banda de frequência com largura suficiente para acomodar as taxas de transferência máximas, os canais W-CDMA apresentam uma largura de banda muito superior. Devido à técnica de codificação CDMA, cada canal é codificado de tal forma que o receptor sabendo o código, pode fazer aquisição do sinal pretendido dentro de uma vasta gama de sinais na mesma banda, sendo que estes se apresentam apenas como ruído ao sinal original [10].

As gamas de frequências onde o W-CDMA opera são: 1920-1980 MHz (*uplink*) e 2110-2170 MHz (*downlink*). Cada canal W-CDMA ocupa uma banda de 4.4MHz a 5 MHz.

1.1.2 TD-CDMA (TDD)

O TDD é uma aplicação da multiplexação no tempo para separar sinais enviados e recebidos. Emula uma comunicação *full-duplex* num canal *half-duplex.* Todo o espectro é usado, mas é dividido em dois intervalos de tempo, um para *downlink e* outro para *uplink.* Os intervalos de tempo alocados podem ser alterados dinamicamente no caso de uma assimetria nas taxas de transmissão (*uplink/downlink*), sendo a maior vantagem desta técnica. Contudo, é bastante complexa para implementar e requere o uso de bandas de guarda, reduzindo, assim, a eficiência espectral.

A técnica TD-CDMA separa *uplink* e *downlink* no tempo, recorrendo a um acesso ao meio CDMA (*Code Division Multiple Access*). Como consequência das suas características, a trama pode ser simétrica ou assimétrica. Para indicar a comutação entre *downlink*/*uplink* existem *Switch Points* usados para este efeito.

Esta técnica apresenta uma largura de 5MHz por canal. Apresenta também um cobertura, por célula, menor que a técnica anterior, W-CDMA (FDD).

1.2 UMTS Radio Access Network

No domínio das redes de acesso UMTS existem dois tipos distintos: GERAN (*GSM EDGE Radio Access Network*) e o UTRAN (*UMTS Terrestrial Radio Access Network*).

O UMTS especifica o UTRAN que consiste num conjunto de equipamentos com funções distintas, que altera o paradigma da rede de acesso das gerações anteriores. Os componentes da infraestrutura são distintos e apresentam diferentes características, constituindo uma evolução aos equipamentos das gerações antecedentes.

O UMTS permite a integração com redes de acesso do GSM/EDGE (GERAN). Permite uma integração com tecnologias de gerações anteriores, possibilitando uma maior cobertura diminuindo a necessidade de implementação de novas infraestruturas de raiz. Esta apresenta-se como uma das grandes vantagens da tecnologia UMTS e consequente implementação. Deste modo a rede de acesso no UMTS é também referida como UTRAN/GERAN (ver Fig. A. 1).

As tecnologias 2G e 3G podem partilhar os mesmos comutadores devido à partilha de infraestruturas e arquitetura, dado que as redes GSM e UMTS se baseiam na mesma estrutura celular.

O principal objetivo da RAN (*Radio Access Newtork*) é permitir a ligação dos terminais ao núcleo da rede. Como já foi referido existem diferentes redes de acesso para a tecnologia UMTS (ver Fig. A. 1).

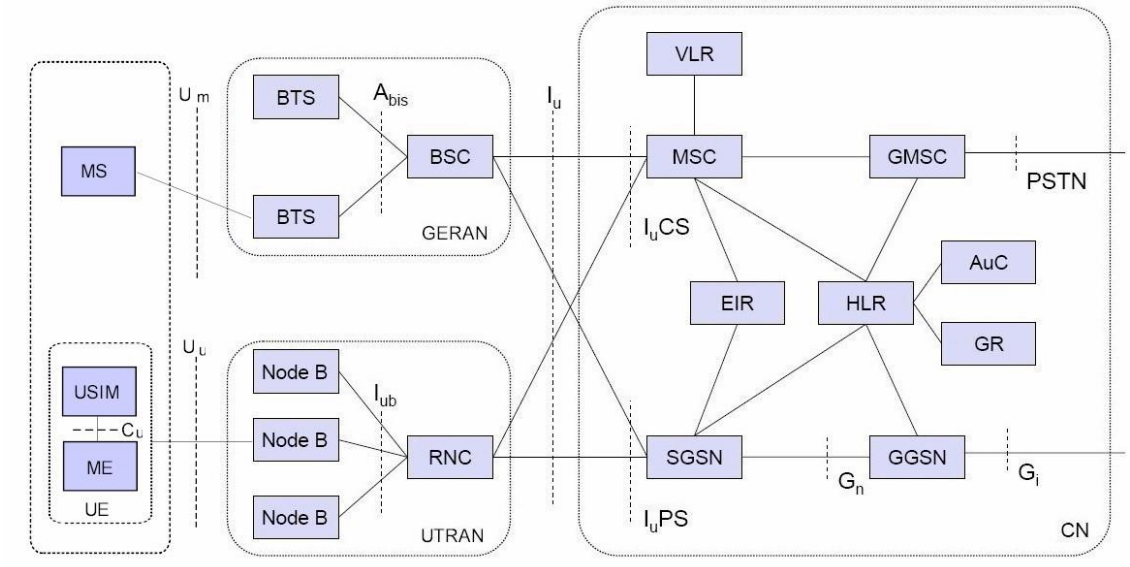


Figura A. 1 - Arquitetura da Rede UMTS [57].

1.2.1 UTRAN

Nesta rede de acesso, os componentes que a constituem são distintos dos seus homónimos nas redes de gerações anteriores, apresentando diferentes características e novas designações.

Dentro da rede UTRA são definidos vários RNS's (*Radio Network Subsystem)*. Um RNS permite fazer a ligação ao Core Network (ver Fig. A. 2). Este, é composto por um RNC *(Radio Network Controller*) e vários Nós B.

O RNC controla todas as funções da *Radio Acess Network* (RAN), assegurando a ligação entre a mesma e o núcleo da rede. Este, possui controlo total sobre os terminais que são ligados à RAN. Controla as ligações na interface rádio, as várias células e Nós B associados. É responsável pela codificação do canal, medições da qualidade do canal, funções de *handover*, entre outras. Em suma é um componente essencial de controlo deste subsistema.

O Nó B lida com as operações de transmissão e recepção entre si e o terminal na interface rádio. É controlado a partir dos RNC's. Um Nó B pode conter uma ou mais células (antenas). É responsável pelas decisões de *handover (soft-handover)* e macro-diversidade entre células. Opera em FDD, TDD, ou nos dois modos. Tem também como função o controlo de potência no *loop* interior para mitigar o *near-far effect*, medindo a qualidade da conexão e intensidade dos sinais.

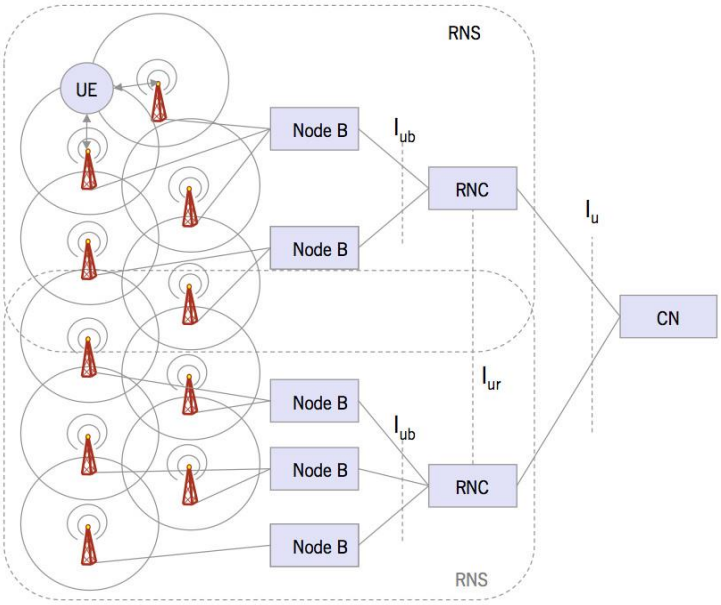


Figura A. 2 - UTRAN [11].

1.2.2 GERAN

Na rede de acesso GSM/EDGE, destaca-se o *Base Station Subsystem*, que contém dois tipos de infraestruturas:

- BTS (*Base Transceiver Station*): Esta define a dimensão das células e consequente área de cobertura e contém todos os componentes rádio afetos a uma emissão/recepção, implementando todos os protocolos de ligação. Algumas das suas funções são: codificação/descodificação dos canais, cifragem de dados, *frequency hoping*, etc.
- BSC (*Base Station Controller*): Efetua a gestão da alocação de frequências a usar e dos recursos de rede. É responsável por mapear os canais de rádio em canais terrestres. A BSC faz também a gestão do processo de *handover*, autenticação e gestão de tráfego.

1.3 UMTS Core Network

O UMTS usa o mesmo *Core Network* que o GSM/EDGE. Isto permite uma migração dos sistemas 2G para os 3G. O núcleo da rede está conectado a várias *backbones*, como a Internet, PSTN (*Public Switched Telephone Newtork*) ou a rede RDIS.

O Core Network está subdividido em dois domínios lógicos: CSD (*Circuit Switched Domain*) e PSD (*Packet Switched Domain*).

O CSD inclui os serviços clássicos de comutação de circuitos, incluindo sinalização, operando através da reserva de recursos no estabelecimento da ligação. É usado para o serviço de voz e é composto por componentes do GSM. Para se conectar ao RNS recorre a uma interface própria, I_uCS [11].

O PSD recorre à comutação de pacotes, que provém do uso dos componentes do GPRS: SGSN (*Serving GPRS Support Node*) e GGSN (*Gateway GPRS Support Node*). Este proporciona um serviço de dados orientado para as necessidades das aplicações da Internet. Suporta o conceito de "*always on",* que se refere ao facto de não existir necessidade de estabelecer ligações. Para se conectar ao RNS recorre a uma interface própria, I_uPS [11].

Estes dois domínios distintos coadunam-se para proporcionar um serviço de alta qualidade. Apesar das suas características e propósitos distintos partilham alguns componentes do núcleo. Esta integração de componentes permite um *deploy* bastante transparente e com custos bastantes reduzidos dado o uso de equipamentos já implementados e aplicados no terreno.

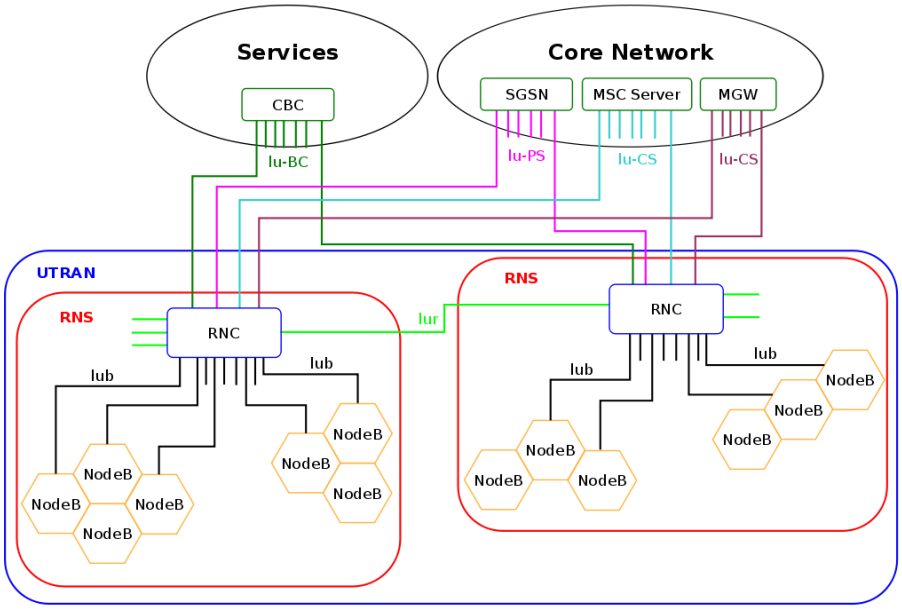


Figura A. 3 – Arquitetura da Rede UMTS (UTRAN) [58].

1.3.1 Circuit Switched Domain

Os componentes principais deste domínio são:

- MSC (*Mobile Services Swithcing Center*): Controla todas as conexões de/para um terminal móvel dentro do seu domínio. É responsável por gerir vários BSC's. Apresenta como funções: *switching*, gestão de recursos da rede, integração com bases de dados, etc. O Gateway MSC (GMSC) permite fazer a conexão com a PSTN ou a rede RDIS.
- Bases de Dados: Estas oferecem uma maior escalabilidade ao sistema. O HLR (*Home Location Register*) caracteriza-se como um base de dados mestre que contém dados permanentes/semipermanentes de todos os subscritores associados ao HLR. Um fornecedor de serviço pode possuir vários HLR. O VLR (*Visitor Location Register*) é uma base de dados

local para um subconjunto de dados de utilizador que se encontram no seu domínio de operação. Isto permite mapear a localização do utilizador e possibilitar o estabelecimento de ligações.

- Componentes de Operação: Neste subconjunto existe o AuC (*Authentication Center*) responsável pela autenticação dos assinantes, EiR (*Equipment Identify Register*) que permite efetuar o registo de terminais móveis e ainda possibilita a localização/bloqueio de terminais móveis e por fim o OMC (*Operation and Maintenance Center*) que executa operações de manutenção possuindo diferentes capacidades de controlo sobre o sistema.

1.3.2 Packet Switched Domain

Este domínio foi introduzido nas especificações do GPRS, proporcionando um serviço de dados baseado em comutação de pacotes. Isto é providenciado pela introdução de dois novos componentes:

- SGSN (*Serving GPRS Support Node*): Unidade de interoperação entre GPRS e PDN (*Public Data Newtork*).
- GGSN (*Gateway GPRS Support Node*): Suporta as funções da *Mobile Station*.

1.4 Tecnologia UMTS-HSPA

A tecnologia HSPA (*High Speed Packet Access*) é um serviço de comutação de pacotes, que permite, nas redes UMTS, débitos muito superiores no acesso à Internet, criando assim as redes 3.5G.

A tecnologia HSPA refere algumas melhorias introduzidas nas interfaces de rádio, levando a alterações, tanto no sentido ascendente como descendente:

- (HSDPA – *High Speed Downlink Packet Access*): Permite o aumento das taxas de transmissão no sentido descendente, através do uso das técnicas AMC (*Adaptive Modulation and Coding*) e HARQ (*Hybrid ARQ*). Estas

alterações permitem débitos teóricos no sentido descendente superiores a 20Mbit/s.

- (HSUPA – *High Speed Uplink Packet Access*): Possibilita o aumento de taxas no sentido ascendente. Recorre à técnica HARQ tal como no HSDPA. Apresenta também um canal de *uplink* dedicado, E-DCH.

A introdução do HSPA afeta apenas a parte rádio não havendo alterações no núcleo da rede, levando assim a uma melhoria significativa no desempenho.

Anexo B

1. Introdução ao Android

No domínio dos sistemas móveis o Android da Google posiciona-se como líder de mercado, como já foi referido anteriormente neste documento. Este destaque é potenciado por duas categorias diferentes: os utilizadores e os *developers*. Se por um lado os equipamentos são desenvolvidos com características e funcionalidades com vista a seduzir os utilizadores, o desenvolvimento de *software* para o sistema operativo assume-se como um dos factores que tornam o Android uma plataforma com um elevado potencial. O seu carácter *open-source* oferece a possibilidade de serem desenvolvidas aplicações com os mais diversos fins, podendo tirar partido de todo o potencial do *hardware* avançado dos dispositivos. Numa altura em que se assiste a uma competição feroz neste mercado tanto ao nível de *software* de proprietário ou mesmo de *hardware* com constantes processos legais sobre violação de patentes por diferentes empresas do sector móvel, o facto do Android ser livre e sem barreiras permite torná-lo líder deste mercado. A Google, nas palavras de Andy Rubin, define o Android como: "*a primeira plataforma realmente aberta e compreensiva para dispositivos móveis. Inclui um sistema operativo, uma interface gráfica e aplicações – todo o software para fazer funcionar um dispositivo móvel, sem os obstáculos de propriedade que têm impedido a inovação móvel."* [26].

O Android é um sistema operativo para dispositivos móveis baseado em Linux, onde as aplicações são normalmente desenvolvidas em linguagem Java usando o *Android Development Kit* (ADT). A principal plataforma de *hardware* é a arquitetura de processadores Advanced RISC Machines (ARM). Ao nível dos IDEs, o Eclipse ADT surge como a atual ferramenta oficial de desenvolvimento. O ADT estende as capacidades do Eclipse e é desenhado para desenvolver as aplicações. Permite criar as interfaces gráficas, os pacotes baseados na Framework do Android, fazer *debug* e exportar os ficheiros "apk" para que se possa distribuir a aplicação [27]. No entanto,

encontra-se em fase de testes (*beta*) o Android Studio. Esta é uma ferramenta de desenvolvimento baseado em IntelliJ IDEA, que providencia melhoramentos e novas funcionalidades face ao Eclipse e tornar-se-á o IDE oficial de desenvolvimento para Android assim que estiver finalizado [28].

Relativamente aos números e estatísticas da plataforma Android, a versão com maior expressão é a Jelly Bean (4.1 até 4.3), estando presente em 56,5% dos dispositivos que existem a nível mundial. A última versão, o KitKat (4.4) está presente em 17.9% dos equipamentos [29].

O Google Play é a plataforma de distribuição de conteúdo gerida pela Google. Para além das aplicações, esta plataforma oferece serviços como música, revistas, filmes, entre outros conteúdos digitais [30]. O Google Play é um serviço totalmente baseado em *cloud-computing*, que permite aos utilizadores gerir o seu conteúdo. Em Março de 2013 a Google anunciou que a sua plataforma possuía mais de 5 milhões de livros e mais 18 milhões de músicas. A estes números acrescem as mais de 700,000 aplicações que já levaram a mais de 25,000 milhões de *downloads* [31].

1.1 Android - Arquitetura de Baixo Nível

A arquitetura do Android baseia-se num conjunto de camadas que comunicam entre si (ver Fig. B. 1) [33].

A camada mais alta, denominada de Applications and Framework é o nível no qual a maioria dos *developers* trabalha. Esta camada providencia diversos serviços de alto nível para as aplicações na forma de classes Java.

Binder IPC é o mecanismo de intercomunicação de processos que permite à Applications and Framework aceder aos serviços do sistema Android. Possibilita que as API's de alto nível acedam e interajam com diferentes serviços de forma oculta ao *developer* [32][33].

A camada Android System Services disponibiliza um conjunto de serviços que permitem aceder ao *hardware* nas camadas inferiores. Esta camada está dividida em dois módulos: System Services e Media Services. O último inclui todos os serviços envolvidos na leitura e gravação de multimédia, já o primeiro envolve os serviços restantes [33].

Num nível mais abaixo encontram-se a HAL (*Hardware Abstraction Layer*) que permite ao sistema Android recorrer ao *hardware* sendo agnóstico sobre as implementações de baixo nível [33].

O Android usa uma versão especial do Kernel Linux com características adicionais, destacando-se um sistema de gestão de memória, que permite um gestão mais efetiva e mais agressiva, dados os recursos limitados. Todos os componentes de *hardware* são geridos pelo Kernel Linux [32][33].

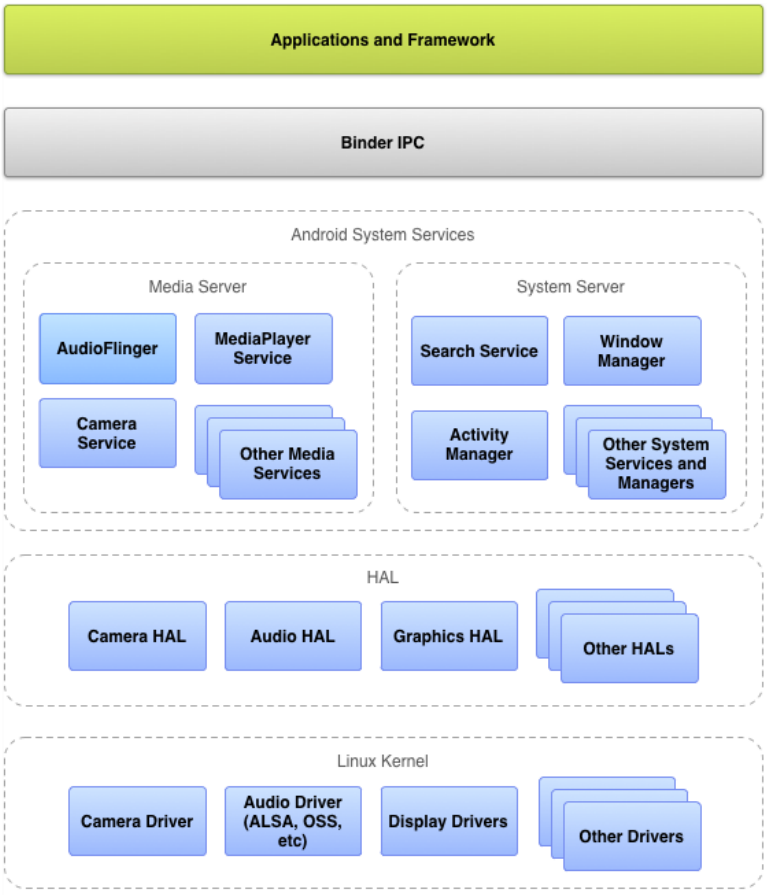


Figura B. 1– Arquitetura do Sistema Android [33]

1.2 Android – Componentes de uma Aplicação

Uma aplicação Android é constituída por diferentes componentes. Estes constituem-se como os elementos básicos da Framework. Estes componentes e a forma como interagem permite construir a fundação de uma aplicação.

O elemento que gere todos os componentes presentes numa aplicação é um manifesto, escrito em XML (AndroidManifest.xml). Todos os componentes têm que ser declarados no manifesto. Além disso, possui outras funções:

- Identificar as permissões que a aplicação requere para executar, como aceder ou alterar o estado da interface de WiFi, etc.
- Declarar o nível mínimo da API requerido pela aplicação.
- Declarar elementos de *hardware* requeridos para a execução, como: o serviço de Bluetooth ou a câmara, etc.
- Declarar as librarias de que a aplicação necessita, como por exemplo: a libraria Google Maps.

No domínio de uma aplicação existem quatro tipos de componentes:

- Activity: Uma *activity* é um componente da aplicação que providencia um ecrã com os quais os utilizadores interagem com o objetivo de realizar uma determinada tarefa, seja tirar uma foto, enviar um *email*, etc. A cada *activity* é associado um *layout* que irá conter a interface do utilizador. Geralmente, uma aplicação consiste em múltiplas *activities* que estão ligadas entre si. Uma *activity* pode ser iniciada por uma outra e sempre que isto acontece o estado da *activity* anterior é preservado numa *stack* ("*the back stack*"). Uma *activity* tem um ciclo de vida, no qual passa por um determinado conjunto de estados. Entre a transição de estados podem ser chamados métodos que permitem a execução de diversas operações (ver Fig. B. 2) [34].
- Services: Um *service* é um componente que permite a realização de operações de execução demoradas em *background* e não providencia uma interface ao utilizador. Um outro componente da aplicação pode iniciar um *service* e este irá executar em *background*, mesmo que o utilizador mude para outra qualquer aplicação no seu dispositivo. Este componente pode assumir essencialmente duas formas: *Started* e *Bound*. O primeiro refere-se a quando um componente, como uma *activity*, inicia um *service* para correr em *background* por tempo indefinido. Geralmente executa uma única operação e não retorna nenhum valor. Quando a execução

termina, o *service* também termina. Quando se trata do tipo *Bound*, refere-se a um *service* que oferece um modelo cliente-servidor, permitindo aos componentes comunicar com o *service* em execução, enviar ou receber pedidos, através de IPC (*interprocess comunication*) [35]. Tal como uma *activity*, possui um ciclo de vida, mais concretamente um para cada um dos dois tipos apresentados.

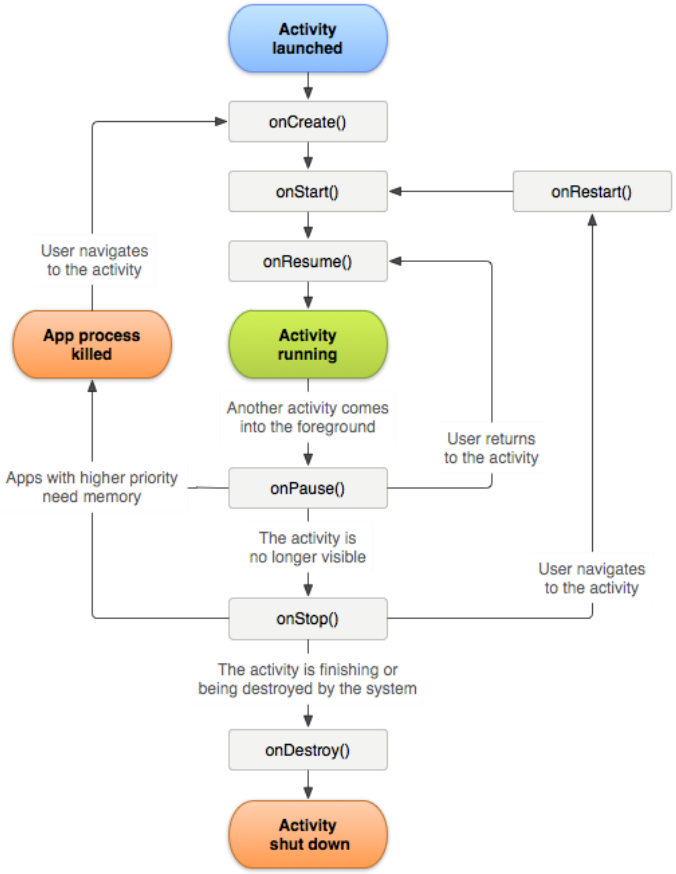


Figura B. 2 – Ciclo de vida de um *activity* [34]

- Content Providers: Permitem a gestão do acesso a um conjunto estruturado de dados. Encapsulam os dados e providenciam mecanismos para a definição de segurança dos mesmos. Os *content providers* são a interface standard que conecta os dados num processo a código que executa num outro processo [37]. Um exemplo deste componente é o *Contacts Provider*, que permite a uma aplicação o acesso aos contactos existente no dispositivo.

- Broadcast Receivers: É um componente que responde a um vasto conjunto de anúncios de broadcast feitos pelo sistema. Exemplos deste componente são: anúncio de bateria fraca, anúncio de que foi tirada uma fotografia, etc. Apesar de este componente não possuir uma interface, pode criar uma barra de progresso para alertar o utilizador quando ocorre um evento de broadcast.

Um elemento importante no domínio de uma aplicação e na forma como são geridos os componentes é o *intent*. É um objecto que permite requisitar uma ação de um componente da aplicação. Apesar de facilitar a comunicação entre componentes de diversas formas, existem três casos de uso fundamentais: iniciar uma *activity*, iniciar um *service* e entregar uma mensagem de *broadcast* [36]. A Fig. B. 3 ilustra o processo de iniciação da *Activity* B pela *Activity* A através de um *intent*. A *Activity* A inicia o processo criando o *intent* com uma descrição de uma ação e passa-o para o *startActivity()*. O *Android System* percorre todas as aplicações e filtra o *intent* respectivo. Posteriormente inicia a *Activity* B invocando o método *onCreate()* passando-lhe o *intent*.

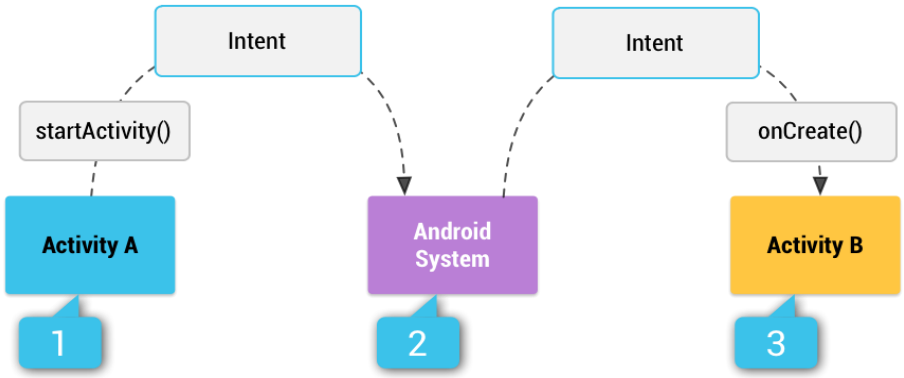


Figura B. 3 – Ilustração de iniciação de uma *activity* através de um *intent* [36]

1.3 Android – User Interface

Todos os elementos da interface do utilizador da aplicação são construídos usando elementos do tipo *View* ou *ViewGroup*. O primeiro designa um objecto que apresenta algo no ecrã, com o qual o utilizador pode interagir. O segundo refere-se a objetos que suportam outros objetos do tipo *View* ou *ViewGroup*, por forma a poder definir o

layout da interface. O *layout* é construído de forma hierárquica com estes dois grupos de objetos e define a estrutura visual da aplicação (ver Fig. B. 4) [40]. Os elementos do *layout* podem ser declarados de duas formas:

- XML: O Android providencia um vocabulário XML que permite criar classes do tipo *View* ou *ViewGroup*.
- Runtime: Os objectos *View* e *ViewGroup* podem ser manipulados e/ou criados programaticamente no código em Java.

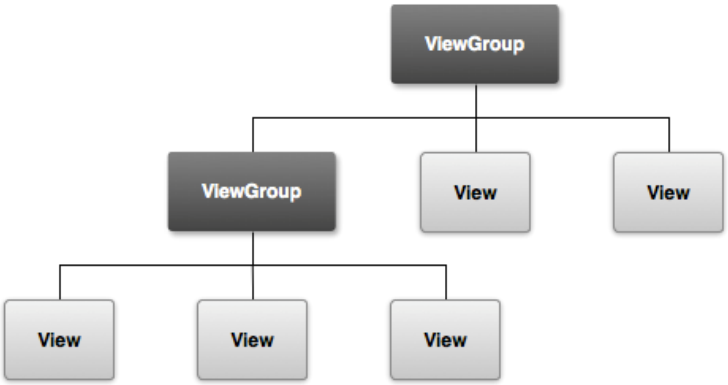


Figura B. 4 –Estrutura hierárquica que define a interface do utilizador [40]

Anexo C

1. Tecnologias para o desenvolvimento de servidores

1.1. Web Services - SOAP vs REST

Um *Web Service* é um método standard de comunicação entre aplicações baseadas em *Web* usando uma rede. É desenhado para suportar a interoperabilidade máquina-para-máquina [41]. Um *Web Service* permite disponibilizar um conjunto de recursos na rede, que são acedidos de forma estruturada usando tipos e mecanismos de comunicação bem definidos, independentemente do sistema operativo, aplicação ou linguagem utilizada que acede ao recursos (ver Fig. C. 1).

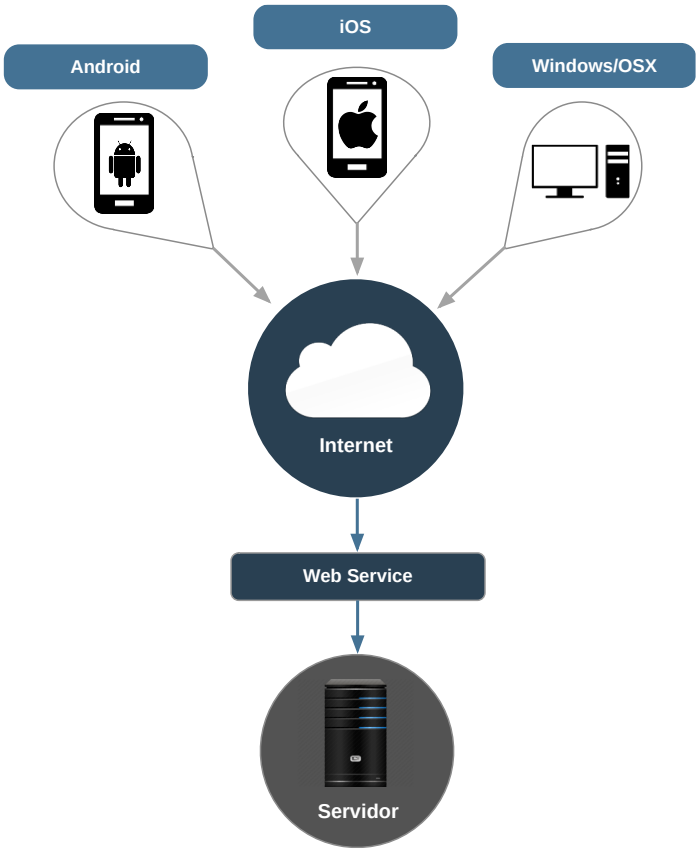


Figura C. 1 – Modelo de um *Web Service* com diferentes clientes.

Um *Web Service* permite controlar o acesso a uma base de dados, processar dados, etc. Esta capacidade de interoperabilidade possibilita o desenvolvimento de um sistema de forma quase agnóstica à implementação do servidor. A comunicação com os *Web Services* é geralmente efectuada segundo o protocolo HTTP. O método pelo qual os recursos são acedidos dita como é que os dados são trocados entre as partes. Este, não pode depender de nenhuma linguagem de programação específica. Geralmente usa-se XML ou JSON para poder fazer a troca de dados visto que grande parte do *software* atual é capaz de interpretar esta formas de representação de dados.

Quando se cria um *Web Service* é necessário estabelecer as normas de comunicação:

- Como é que um sistema efetua um pedido;
- Qual a estrutura dos dados das mensagens trocadas;
- Quais as mensagens de erro que existem quando a comunicação falha.

Um dos principais motivos para a utilização de *Web Services* é a necessidade de diminuir o processamento no lado do cliente. Desta forma é possível auxiliar nas tarefas a realizar, permitindo, neste caso concreto, que a aplicação móvel não processe tantos dados.

Para a implementação de um *Web Service* existem duas abordagens distintas: SOAP (*Simple Object Access Protocol*) e REST (*Representational State Transfer*). Estas, possuem características específicas que as distinguem.

O SOAP é um protocolo que permite a troca estruturada de informação e assenta no formato XML para a troca de mensagens. Geralmente, recorre a dois protocolos, o HTTP e o SMTP, para a negociação e transmissão de mensagens [42].

Este protocolo providencia uma *framework* de comunicação para *Web Services* e baseia-se em três componentes:

- Um envelope, que define o que está na mensagem e como é que se processa;
- Um conjunto de regras para aceder aos tipos de dados da aplicação;
- Uma convenção para a representação das mensagens de pedido e resposta.

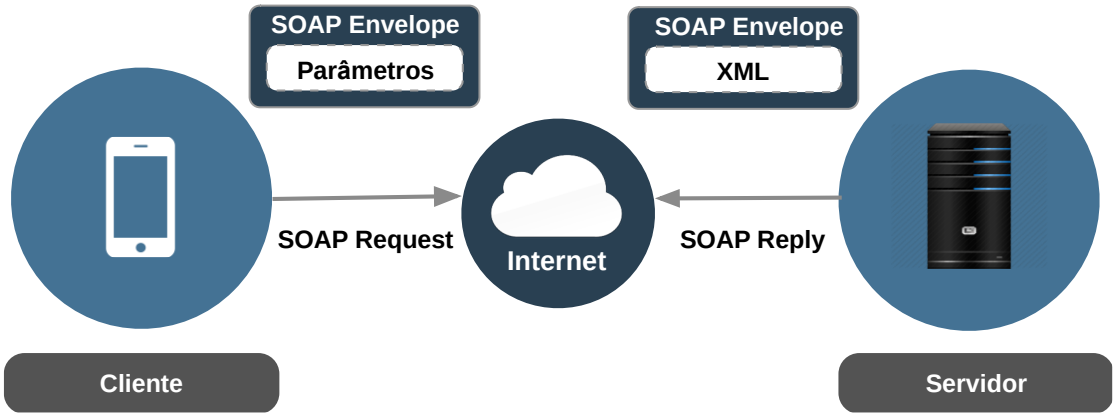


Figura C. 2 - Estrutura de uma interação através do protocolo SOAP

O REST não é um protocolo, mas uma abstração de uma arquitetura baseada na Internet. Consiste numa *framework* distribuída que usa um conjunto de protocolos e tecnologias envolvendo interações cliente-servidor para a troca de recursos.

O modelo cliente-servidor imposto pelo REST simplifica a implementação dos seus componentes, reduz a complexidade da semântica de conexão entre eles, melhora a performance de um sistema e aumenta a escalabilidade do servidor. O REST permite um processamento simples recorrendo a mensagens auto-descritivas, a uma interação *stateless* através de operações *standard*. Esta arquitetura recorre ao conjunto de operações definidas no protocolo HTTP (GET, POST, DELETE e PUT) ou outro protocolo semelhante. Um *Web Service* baseado em REST recorre ao URI (*Uniform Resource Identifier*) para identificar um recurso específico, como:

- http://exemplo.com/recursoX/valorY;

Para a troca de dados geralmente é usado o JSON (*JavaScript Object Notation*), podendo também ser usado XML ou outra.

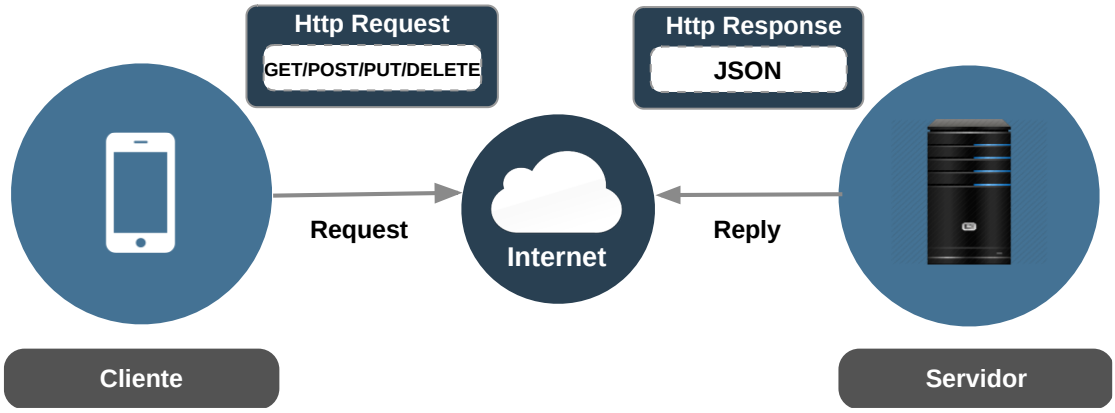


Figura C. 3 - Estrutura de uma interação através do REST

Em termos de comparação, serviços como Yahoo, Amazon e Ebay usam o REST. A Google transitou de SOAP para REST em 2006.

Do ponto de vista de implementação o REST, face ao SOAP, é mais simples uma vez que recorre ao HTTP e aos seus métodos *standard* para aceder aos recursos no servidor. O SOAP requere conhecimentos de XML e requere ferramentas e processos para criar os pedidos e analisar os resultados.

Pela Fig. C. 4 é possível observar a tendência no crescimento do REST face ao SOAP, tornando-se no modelo mais usado na concepção de *Web Services*.

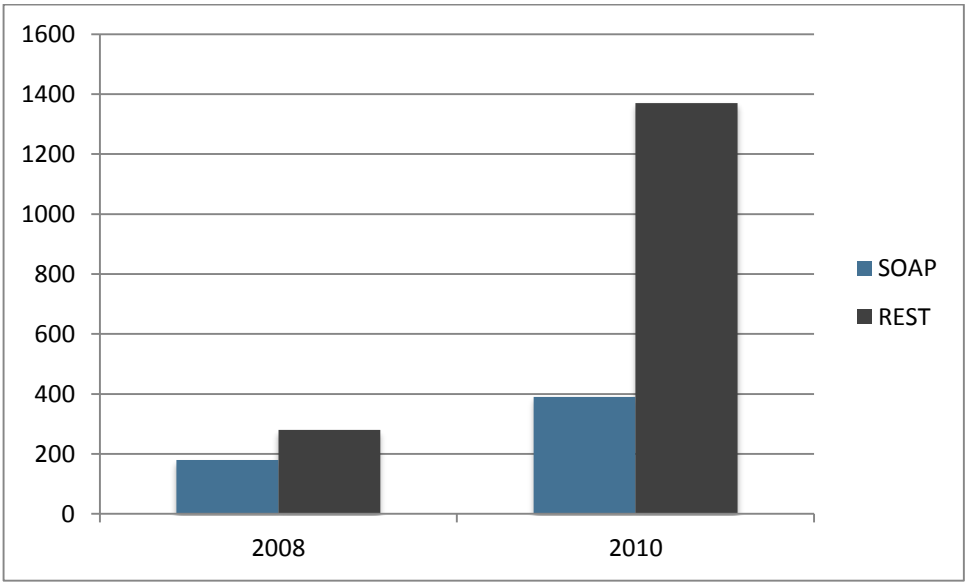


Figura C. 4 – Número total de API's baseados em protocolo e estilo [44]

1.2. Web Services – XML vs JSON

O XML é uma linguagem de marcação simples e muito flexível, recomendada pelo W3C (*World Wide Web Consortium*) em 10 de Fevereiro de 1998. Inicialmente foi desenvolvida para a publicação de documentos electrónicos em grande escala, mas foi assumindo um papel importante na troca de dados entre sistemas baseados em *Web*. É uma linguagem baseada em *tags* passíveis de ser interpretadas por humanos ou máquinas. A sua construção permite estruturar, armazenar e transportar dados. Os dados XML são armazenados em texto limpo, providenciando uma forma de os manter independentemente do *software* ou *hardware* [46].

Atualmente existe um vasto conjunto de linguagens que assentam no XML por exemplo:

- XHTML;
- WSDL;
- RSS;
- RDF.

Dada a sua versatilidade, tornou-se uma linguagem muita utilizada no domínio dos *Web Services*.

Tal como o XML, o JSON é uma linguagem de marcação baseada na sintaxe do JavaScript. É de leitura simples, tanto por máquinas como por humanos. Tem uma estrutura hierárquica. No entanto, o JSON, face ao XML, é mais rápido a ser processado, não possui *tags* de finalização, pode recorrer ao uso de *arrays* e não necessita de um *parser* específico (por exemplo, uma função em JavaScript pode fazer o *parsing* de JSON) [47].

O JSON é limitado a oito tipos de dados (como String, Number, Boolean, etc), já o XML permite armazenar qualquer tipo de dados, sendo mais flexível, mas ao mesmo tempo mais difícil de ler. O XML suporta fotografias, áudio, vídeo e até ficheiros executáveis, o que pode levar a graves falhas de segurança. A estrutura de dados do JSON é mais limitada, mas mais simples de manipular [48].

Em suma, o XML dada a sua flexibilidade e extensibilidade no que toca à partilha de documentos é a linguagem de marcação mais apropriada, dado que possui menos limitações face ao JSON, podendo o próprio documento fazer-se acompanhar pelas regras de formatação (XML *schema*, denominado de XSD). No que diz respeito a

trocas de informação não formatados, com um conjunto específico de tipos de dados tradicionais, o JSON assume-se como a melhor linguagem de marcação.

Nos últimos anos também se tem assistido a uma tendência na adopção do JSON [49].

Na Fig. C. 5 nota-se um claro crescimento no número de API's com suporte para JSON. Em 2011 cerca de 55% das novas API's possuíam suporte para JSON.

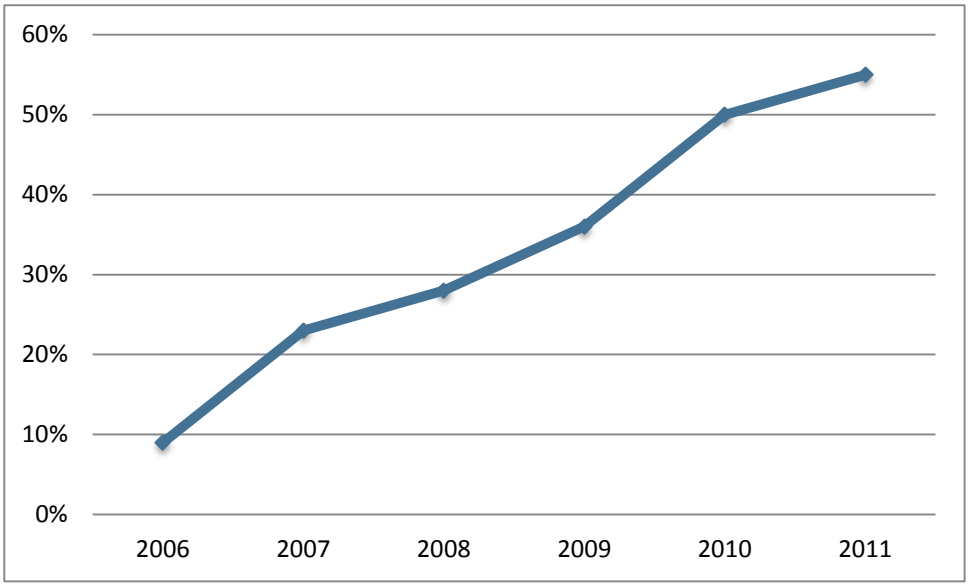


Figura C. 5 – Percentagem de novas API's com suporte para JSON [49]

1.3. Web Services – Engines

Ao nível do desenvolvimento de servidores *Web*, existem diferentes soluções, como o Perl, ASP, JSP, Java ou PHP. O Java, segundo estudos conduzidos pela W3Tech é a linguagem de desenvolvimento mais utilizada para servidores em larga escala, com elevados volumes de tráfego. A Amazon ou a Apple App Store recorrem a *frameworks* baseadas em Java [50]. Contudo, assume-se como uma linguagem de elevado grau de complexidade com elevada potencialidade no desenvolvimento de *software* em larga escala. Dadas as características do projeto, preteriu-se o Java por uma linguagem de programação mais simples com vasto suporte em termos de bibliotecas, o PHP.

O PHP é uma linguagem de *scripting* que tem vindo a crescer em termos de implementação, sendo usado em cerca de 75% dos servidores *Web* [50]. É *open source*, possuindo uma comunidade de suporte online bastante grande e é compatível com múltiplas plataformas. Possui uma sintaxe simples e bastante capaz em termos de processamento. A sua simplicidade no tratamento de dados, a sua compatibilidade com SQL ou JSON, ou ainda, com HTML, confere ao PHP um estatuto de destaque entre as linguagens de programação. Assume a sexta posição no ranking das vinte linguagens de programação mais usadas, segundo o IEEE Spectrum [51]. A plataforma LAMP (Linux Apache MySQL PHP) é usada por diversas empresas no sector do desenvolvimento *Web*, como o Facebook ou a Yahoo.