



UNIVERSIDADE
LUSÓFONA

Android Walk Test App

Trabalho Final de Curso

Relatório final do 2.º Semestre

DEISI199

Nome do Aluno: Pedro Canaveira

Número do aluno: 21803482

Nome do Orientador: Prof. Lúcio Studer Ferreira

Co-Orientador: Eng. Nuno Valente

Trabalho Final de Curso | LEI | 23/07/2021

www.ulusofona.pt

Direitos de cópia

(Android Walk Test App), Copyright de *(Pedro Canaveira)*, ULHT.

A Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Resumo

As redes de comunicação móvel têm a responsabilidade de garantir que os seus utilizadores permanecem sempre conectados. Esta responsabilidade requer uma atenção especial por todas as operadoras responsáveis pelas mesmas. As redes de comunicação móvel têm como meio de transmissão o canal rádio e são constituídos por diferentes sistemas. As células são responsáveis pela sua cobertura e encontram-se dispersas pelo mapa, permitindo aos seus utilizadores comunicar de forma ininterrupta, mesmo em movimento. De facto, as comunicações transitam de uma célula para a outra (*handover*) de forma totalmente transparente (para o utilizador). No entanto, existem diversos problemas que podem comprometer o funcionamento da rede: interferência entre células, zonas sem cobertura, morfologia do terreno, características do equipamento, etc. Para que a comunicação nunca falhe, é necessário um planeamento e manutenção adequado. Um dos métodos utilizados para testar a rede é o *Drive Test* (DT), este permite a técnicos qualificados efetuar “capturas” (dados de potência de sinal georreferenciadas) num veículo preparado com equipamento específico. Este método apesar de obter dados pormenorizados, implica um custo elevado e encontra-se restringido aos locais de teste onde um veículo possa circular. Por esse motivo, com o presente trabalho desenvolveu-se uma aplicação para dispositivos móveis inteligentes que permita capturar medidas georreferenciadas de desempenho de uma rede e exportar para a plataforma Metric [2] da Multivision [1]. O Metric é uma plataforma capaz de receber dados provenientes de uma grande variedade de ferramentas de captura, e posteriormente analisar os mesmos, permitindo a todos os clientes da plataforma compreender o desempenho da rede. Existem soluções e aplicações móveis disponíveis no mercado capazes de realizar capturas, no entanto, estas não respondem a todos os requisitos do cliente (Multivision) para este trabalho e muitas têm um custo associado.

Abstract

Mobile communication networks have a responsibility to ensure that their users always remain connected. This responsibility requires special attention by all operators responsible for them. The mobile communication networks have the radio channel as their transmission medium and are made up of different systems. The cells are responsible for their coverage and are scattered across the map, allowing their users to communicate without interruption, even on the move. In fact, communications move from one cell to another (handover) in a totally transparent way (for the user). However, there are several problems that can compromise the functioning of the network: interference between cells, areas without coverage, terrain morphology, equipment characteristics, etc. So that communication never fails, proper planning and maintenance is necessary. One of the methods used to test the network is the Drive Test (DT), which allows qualified technicians to make “captures” (georeferenced signal strength data) in a vehicle prepared with specific equipment. This method, despite obtaining detailed data, implies a high cost and is restricted to the test sites where a vehicle can drive. For this reason, the present work developed an application for smart mobile devices that allows to capture georeferenced measures of network performance and export to Multivision's [1] Metric [2] platform. Metric is a platform capable of receiving data from a wide variety of capture tools, and subsequently analyzing them, allowing all customers of the platform to understand the performance of the network. There are mobile solutions and applications available on the market capable of making captures, however, they do not meet all the customer's requirements (Multivision) for this job and many have an associated cost.

Índice

Resumo.....	iii
Abstract	iv
Índice.....	v
Lista de Figuras	vii
Lista de Tabelas.....	viii
1 Identificação do Problema.....	1
1.1 Enquadramento	1
1.2 Motivação	2
1.3 Objetivos.....	3
2 Levantamento e Análise de Requisitos.....	4
2.1 Análise de Requisitos.....	4
2.1.1 Requisitos Funcionais	4
2.1.2 Requisitos Não-Funcionais.....	7
2.2 Casos de Uso	7
2.3 Diagramas de atividades.....	9
2.4 Modelo de Base de Dados	11
3 Viabilidade e Pertinência	13
4 Solução Desenvolvida.....	15
4.1 Arquitetura	15
4.2 Implementação da aplicação móvel.....	18
4.3 Implementação na Cloud.....	23
5 Benchmarking	25
6 Método e planeamento.....	27

7	Resultados.....	29
8	Conclusão e trabalho futuros.....	32
	Bibliografia.....	33
	Glossário.....	34

Lista de Figuras

Figura 1 - Drive Test a uma célula, obtido através da plataforma Metric. Extraído de [3].	2
Figura 2 - Estrutura para estimar a cobertura de uma nova célula baseado em nuvem. Extraído de [3].	3
Figura 3 - Caso de Uso: Iniciar sessão de captura	8
Figura 4 - Caso de Uso: Enviar histórico de capturas	9
Figura 5 - Diagrama de Atividades: Iniciar/Terminar sessão de captura	10
Figura 6 - Diagrama de Atividades: Exportar sessão para o S3 Bucket	11
Figura 7 - UML	12
Figura 8 - Estimativa da cobertura de uma célula. Integrado na plataforma Metric. Extraído de [3].	13
Figura 9 - Arquitetura	15
Figura 10 - Sessão no histórico da aplicação	17
Figura 11 - Arquitetura da aplicação Android	18
Figura 12 - Mapa com marcadores no ecrã de sessão	20
Figura 13 - Notificação durante uma sessão	21
Figura 14 - Storyboard da Aplicação	22
Figura 15 - Ficheiro de configuração Amplify	23
Figura 16 - Inicialização do serviço Amplify	23
Figura 17 - Código da exportação da sessão	24
Figura 18 - Ficheiros no Bucket S3	24
Figura 19 - Ficheiro CSV	24

Lista de Tabelas

Tabela 1 - Requisitos funcionais	5
Tabela 2 - Tabela de comparação	26
Tabela 3 - Cronograma	28
Tabela 4 - Testes realizados	30

1 Identificação do Problema

1.1 Enquadramento

As redes de comunicação móvel têm a responsabilidade de garantir que os seus utilizadores permanecem conectados mas encontram-se sujeitas a um conjunto de obstáculos e desafios que podem prejudicar o seu desempenho. O correto planeamento e operação, é essencial para garantir a qualidade de serviço e satisfação dos utilizadores.

As redes celulares têm como meio de transmissão o canal rádio, recorrendo a ondas eletromagnéticas. Este canal tem desafios enormes em termos do seu entendimento e caracterização, tornando-se difícil configurar adequadamente o sistema. São necessárias ferramentas complexas para a planificação e otimização do mesmo e existem diversos modelos teóricos para estimar a cobertura. Além disso, as redes são heterogéneas, misturam vários sistemas (2G, 3G, 4G, 5G), destinados a serviços diferentes. Por exemplo, o sistema GSM (2G) está vocacionado para voz e o LTE (4G) para dados a alta velocidade. Além disso, os sistemas são constituídos por equipamentos de diferentes fabricantes (Exemplos: Ericsson, Huawei, ZTE).

Os sistemas celulares são compostos por células. Uma célula corresponde à área iluminada por uma antena, tipicamente instalada no topo de edifícios ou mastros. Estas redes recorrem a milhares de células que, justapostas, permitem a cobertura contínua de uma área de serviço, permitindo aos seus utilizadores comunicar de forma ininterrupta, mesmo em movimento. De facto, as comunicações transitam de uma antena para a outra (*handover*) de forma totalmente transparente (para o utilizador). No entanto, estas redes enfrentam difíceis problemas: interferência entre células que se sobrepõem, zonas sem cobertura (que não são iluminadas por nenhuma antena), antenas desviadas ou com ângulos indevidamente configurados, morfologia do terreno, densidade de edifícios e materiais que os constituem, assim como as características do equipamento e potência usada. Estes problemas exigem uma resolução adequada, por forma a garantir um bom desempenho do sistema.

O planeamento e otimização são complexos do “ponto de vista da rede”, devido à sua heterogeneidade, cada sistema e fabricante têm o seu software: *Operation and Support System* (OSS). Responsável pela monitorização do desempenho e configuração de cada célula. No entanto, muito do trabalho de configuração é feito de forma manual. Requer a utilização de vários *softwares*, não sendo possível ter uma visão global. Assim, o engenheiro dificilmente detém um conhecimento realista da forma como a sua rede celular cobre a área de serviço.

Drive Test (DT) é um dos métodos utilizados do “ponto de vista do utilizador”, para avaliar uma instalação recente de uma antena, alterações de configuração ou *check-up*. Captura medidas da rede móvel com equipamento específico. O Metric [2], plataforma com a qual vamos trabalhar neste trabalho, permite a visualização de DTs como se ilustra na Figura 1.

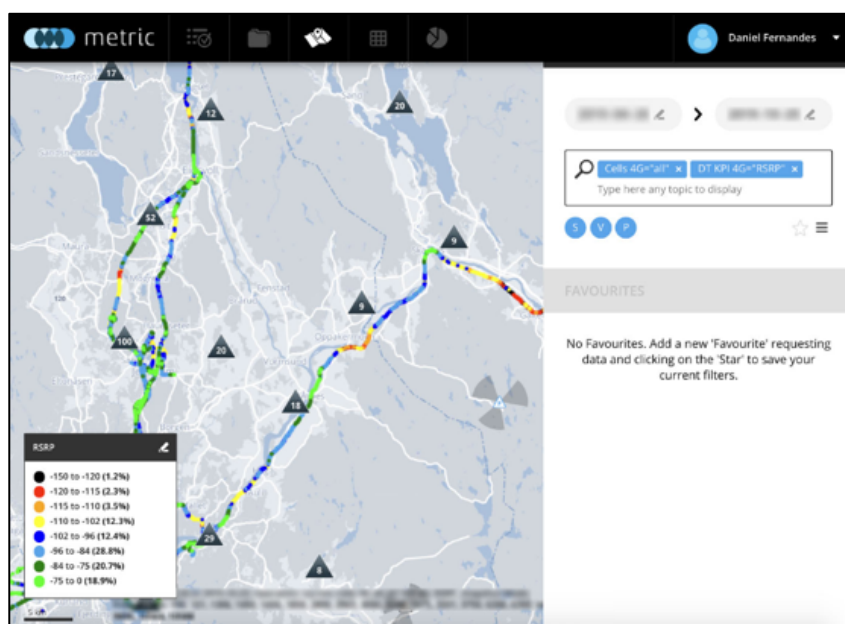


Figura 1 - Drive Test a uma célula, obtido através da plataforma Metric. Extraído de [3].

Habitualmente, para efetuar um DT, usa-se automóveis com equipamento específico e trabalhadores qualificados. Este equipamento permite capturar dados detalhados, mas tem um investimento significativo e mobilidade limitada. Muito do tráfego móvel é gerado em ambientes *indoor*, com um DT clássico, não é possível avaliar o desempenho nestas áreas.

Minimization of Drive Test (MDT) usa um dispositivo móvel com uma aplicação apropriada para “capturar” medidas da rede, cada captura inclui a potência de sinal e localização da mesma. Em comparação com o DT “tradicional”, tem como objetivo reduzir o custo e melhorar a mobilidade, funcionando em áreas inalcançáveis por um DT “tradicional”. Existem aplicações móveis que permitem efetuar capturas da rede móvel, no entanto, apresentam limitações que serão identificadas no capítulo 5 sobre o Benchmarking.

1.2 Motivação

O OptiNet, projeto P2020 da Multivision [1] financiado pela Comissão Europeia, apresentou vários resultados inovadores em algoritmos de planeamento e otimização automática destas redes. Foram desenvolvidos protótipos com estas funcionalidades e integrados no Metric [2], ferramenta de monitorização da Multivision capaz de congrega informação das redes de um operador, integrando dados vindos de diferentes sistemas e fabricantes. Recentemente, foram apresentadas publicações com investigações [3], onde é proposto um modelo, que calibra modelos teóricos de estimação de cobertura de antenas, com medidas de drive test. Prova-se que, quantas mais medidas e mais variadas, melhor se consegue estimar.

Foi desenvolvido um protótipo e produto que permite calcular de forma totalmente automática a área de cobertura de uma antena. Este modelo, ilustrado na Figura 2, recebe como "input" DTs. A aplicação que pretendesse desenvolver neste projeto, vai aumentar o número de capturas, e alcançar zonas anteriormente inalcançáveis, melhorando a precisão dos modelos.

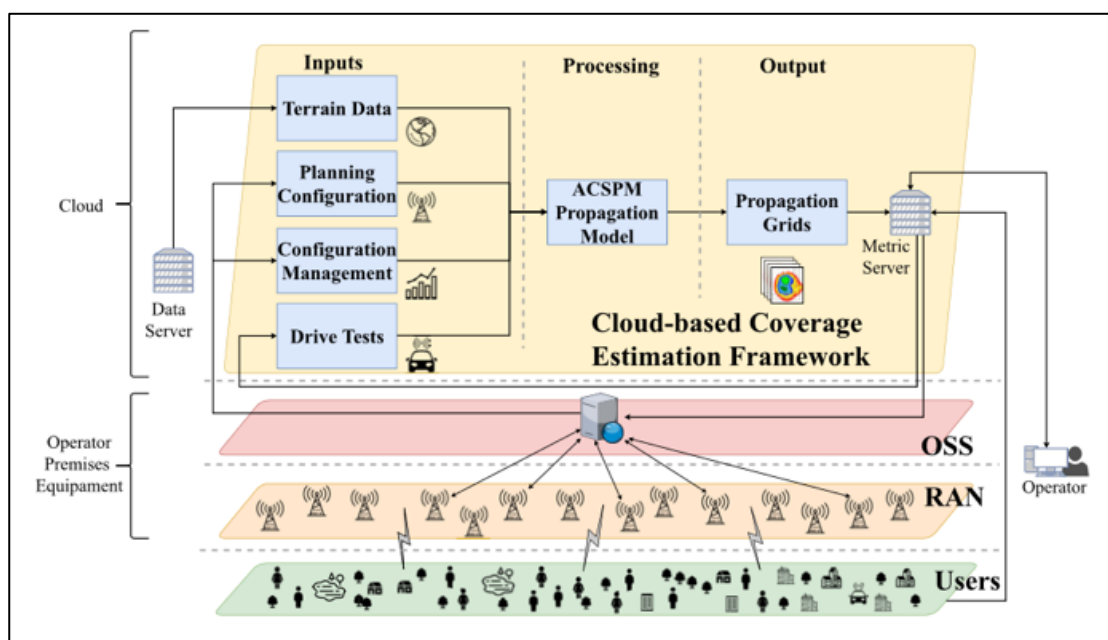


Figura 2 - Estrutura para estimar a cobertura de uma nova célula baseado em nuvem. Extraído de [3].

1.3 Objetivos

Os DTs “tradicional” são dispendiosos e limitados. Neste trabalho pretende-se desenvolver uma aplicação para dispositivos móveis que permita capturar medidas georreferenciadas de desempenho de uma rede. Esta aplicação efetua capturas e envia posteriormente para o Metric, integrando-se no modelo que se calibra com medidas. A Multivision criou um protótipo sobre o tema deste TFC em 2017. Esta aplicação Android, foi programada em Java, no entanto, não é compatível com as versões mais recentes do sistema operativo.

O presente TFC tem como objetivo criar um MVP (*Minimum Viable Product*) "Android Walk Test App" em parceria com a Multivision, capaz de capturar dados de rede móvel. Cada captura enviada é constituída pela localização onde foi efetuada e a potência do sinal de rede móvel naquele instante. Todas as capturas válidas podem ser exportadas para o Metric. São apresentadas outras funcionalidades desenvolvidas no capítulo da Solução Desenvolvida.

2 Levantamento e Análise de Requisitos

A Multivision pretende uma aplicação direcionada para os técnicos que pretendem validar um determinado “site” (local onde se encontra a célula). Foi sugerido dividir o desenvolvimento da aplicação em 4.ª fases distintas, com diferentes requisitos funcionais. Este capítulo apresenta os requisitos importantes para o cliente, assim como um “cenário de continuidade”.

2.1 Análise de Requisitos

Os requisitos a ser implementados apresentados, estão dependentes do hardware utilizado para testes. Infelizmente, alguns equipamentos Android não disponibilizam o acesso a certas informações do modem presente no chipset. Estas limitações vão influenciar a informação que é possível recolher numa captura.

2.1.1 Requisitos Funcionais

Os requisitos funcionais encontram-se divididos em 4.ª fases de desenvolvimentos distintas.

Objetivos de cada fase:

- **1.ª Fase:** Versão inicial capaz de capturar, guardar e exportar as métricas necessárias;
- **2.ª Fase:** Configurar frequência das capturas e representar o histórico das mesmas num mapa na aplicação;
- **3.ª Fase:** Funcionalidade adicional para os técnicos no terreno: validação após instalação de um site novo;
- **4.ª Fase:** “Cenário de continuidade” com *crowdsourcing* de capturas. Utilizadores da aplicação (gratuita) efetuam capturas e recebem benefícios em troca (Exemplo: dados móveis). Além disso, os utilizadores podem consultar os dados móveis, memória interna e consumo de energia necessário para a aplicação funcionar. Estas funcionalidades incentivam novos utilizadores a experimentar, assim como acrescenta transparência.

Representam-se na Tabela 1 todos os requisitos funcionais (RF) idealizados. Encontram-se presentes requisitos para “cenários de continuidade”. A coluna “Implementado” identifica os requisitos.

Todos os requisitos que foram planeados foram implementados, com exceção do requisito RF3.3 que não foi possível pois, o **acordo com uma operadora não foi estabelecido em tempo útil** para esta entrega.

Tabela 1 - Requisitos funcionais

ID	Requisito funcional	Descrição	Implementar	Implementado
RF1.1	Capturar posição geográfica	Determinar as coordenadas do local da captura.	Sim	Sim
RF1.2	Capturar a potência do sinal do dispositivo com a célula.	Determinar a potência do sinal do dispositivo com a célula em uso.	Sim	Sim
RF1.3	Memorizar capturas na memória do dispositivo.	Guardar as capturas validadas na memória do dispositivo.	Sim	Sim
RF1.4	Criar ficheiro CSV com as capturas por enviar.	Este ficheiro inclui as capturas por enviar num formato compatível com o Metric.	Sim	Sim
RF1.5	Exportar ficheiro TXT para o Metric.	Exportar capturas para o Metric.	Sim	Sim
RF2.1	Configurar o tipo de captura.	Tipos de capturas: <ul style="list-style-type: none"> • Voice (realiza uma chamada e efetua capturas durante a mesma); • Idle (efetua capturas em background). 	Sim	Sim
RF2.2	Configurar a frequência das capturas.	Utilizador define frequência das capturas durante uma sessão.	Não	Cenário de continuidade
RF2.3	Representação num mapa das capturas memorizadas.	Mapa com o trajeto de cada sessão.	Sim	Sim

ID	Requisito funcional	Descrição	Implementar	Implementado
RF3.1	Capturas no mapa são ilustradas com uma cor diferente, consoante a potência do sinal.	No mapa cada captura de uma sessão tem uma cor diferente, consoante a potência do sinal de mesma.	Não	Sim
RF3.2	Representar o percurso do utilizador no mapa, durante uma sessão de captura.	Cada captura criada com sucesso é adicionada em direto ao mapa.	Sim	Sim
RF3.3	Validação de um site.	Utilizador efetua testes no local após instalação ou manutenção.	Sim	Não
RF3.4	Importação de ficheiros com informação da célula ¹ .	Importar ficheiros provenientes de uma base de dados sobre as células no local.	Não	Cenário de continuidade
RF3.5	Disponibilização de informação sobre a célula ao utilizador.	A informação sobre a célula é demonstrada ao utilizador.	Não	Cenário de continuidade
RF4.1	“Crowdsourcing” de capturas.	Público geral recebe benefícios por usar a aplicação.	Não	Cenário de continuidade

¹ Exemplos de informação neste tipo de ficheiros: n.º de células, azimuth, etc. Este tipo de informação permite demonstrar para onde é que as células estão a apontar.

ID	Requisito funcional	Descrição	Implementar	Implementado
RF4.2	Quantificar os recursos necessários para as capturas efetuados pelo dispositivo.	Demonstra a energia consumida, memória ocupada pela aplicação e os dados móveis utilizados nas sessões anteriores.	Não	Cenário de continuidade
RF4.3	Configurar a rede em qual são exportadas as capturas.	A exportação consome dados móveis adicionais. Permite ao utilizador escolher se envia em qualquer rede ou apenas por WiFi.	Não	Cenário de continuidade
RF4.4	Página FAQ	Página com perguntas frequentes sobre a aplicação.	Não	Cenário de continuidade

2.1.2 Requisitos Não-Funcionais

Lista de requisitos não-funcionais:

- Aplicação desenvolvida em Kotlin;
- Aplicação compatível com Android 10 ou superior;
- Aplicação suportada apenas por dispositivo com modem e GPS;
- Capturas efetuadas guardadas em são enviadas para o Metric, quando o utilizador pretender;
- Metric tem que estar disponível no momento da exportação de capturas;
- Segurança e privacidade. As capturas enviadas contêm identificadores únicos, no entanto não serão associados ao utilizador;
- Utilização transparente dos dados recolhidos pelas capturas (política de privacidade). Visto estarmos a lidar com dados sensíveis, o utilizador vai ser informado de todos os dados registados e a sua finalidade.

2.2 Casos de Uso

Um utilizador na aplicação pode realizar as seguintes ações:

- Iniciar nova sessão de captura;
- Pausar, terminar ou cancelar sessão de captura (em progresso);
- Ver histórico de sessões armazenado no dispositivo;
- Eliminar sessões guardadas;
- Consultar página da ajuda, onde se encontra os valores de referência (intervalos) para o Signal Strength nas diferentes tecnologias;
- Exportar capturas da sessão para o serviço de armazenamento Cloud (S3 Bucket).

Destas ações, destaca-se a ação, **iniciar nova sessão de captura e exportar capturas da sessão para o serviço de armazenamento Cloud (S3 Bucket)** que descrevemos nos casos de uso a seguir.

Caso de uso: Iniciar sessão de captura

O utilizador para iniciar uma sessão de captura, tem que cumprir certos requisitos. O dispositivo tem que ter a localização ativa e sinal de rede móvel. Além disso, o Android pede autorização ao utilizador quando uma aplicação precisa da localização e informações do sinal do telefone, esta permissão tem que ser autorizadas anteriormente. Caso de uso ilustrado na Figura 3.

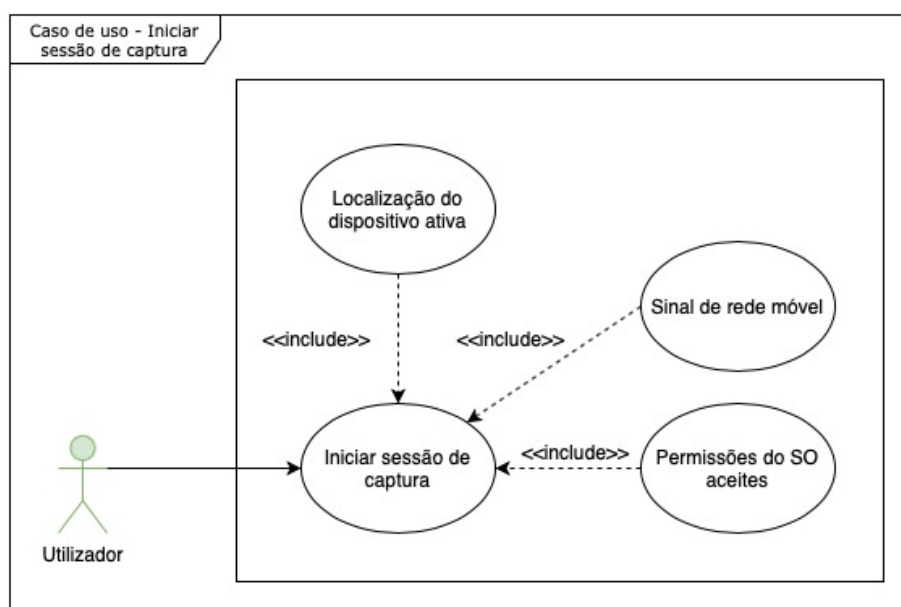


Figura 3 - Caso de Uso: Iniciar sessão de captura

Caso de uso: Enviar histórico de capturas

Quando existe pelo menos uma sessão de captura em histórico, o utilizador pode exportar a mesma, desde que o dispositivo tenha Internet. O ficheiro de OUTPUT é enviado para o S3 Bucket na AWS configurado neste projeto, ao qual o Metric pode aceder. Caso de uso ilustrado na Figura 4.

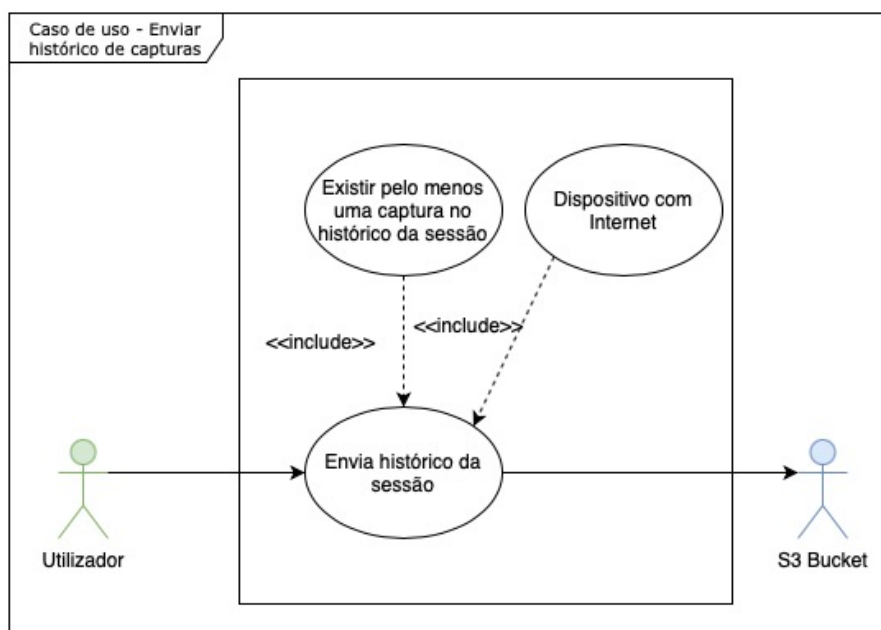


Figura 4 - Caso de Uso: Enviar histórico de capturas

2.3 Diagramas de atividades

Diagramas de atividades representam o fluxo de uma ação na aplicação. Nesta secção descrevem-se e ilustram-se os vários diagramas de atividades que identificamos para a aplicação.

Diagrama de atividades: Iniciar sessão de captura

Ilustrado na Figura 5. Demonstra o fluxo de execução quando o utilizador começa uma sessão de captura. A sessão de captura continua até o utilizador a terminar manualmente. A aplicação após iniciar uma sessão, continua a mesma em background, mesmo que o utilizador esteja fora da aplicação ou com o ecrã desligado.

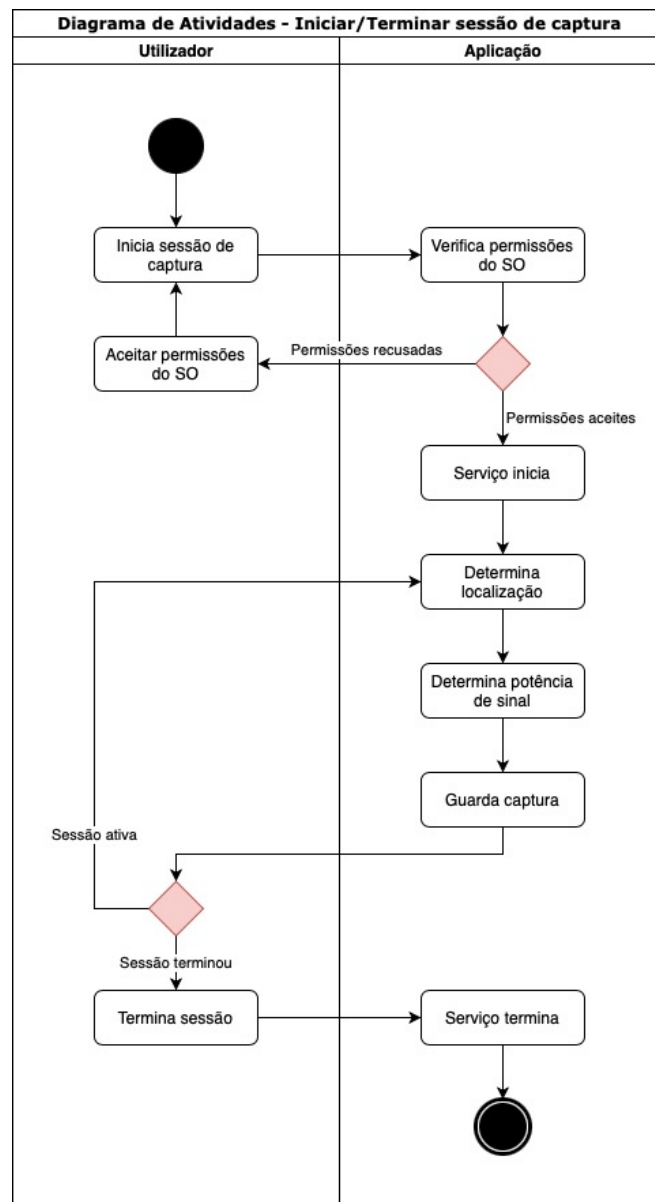


Figura 5 - Diagrama de Atividades: Iniciar/Terminar sessão de captura

Diagrama de atividades: Exportar sessão para o S3 Bucket

Neste diagrama é demonstrado o fluxo de execução quando o utilizador enviar capturas para o S3 Bucket no AWS. O utilizador escolhe a sessão que pretende enviar, a aplicação cria um novo ficheiro CSV com os dados provenientes da base de dados local sobre a sessão seleccionada e envia para o S3 Bucket. Diagrama de atividades ilustrado na Figura 6.

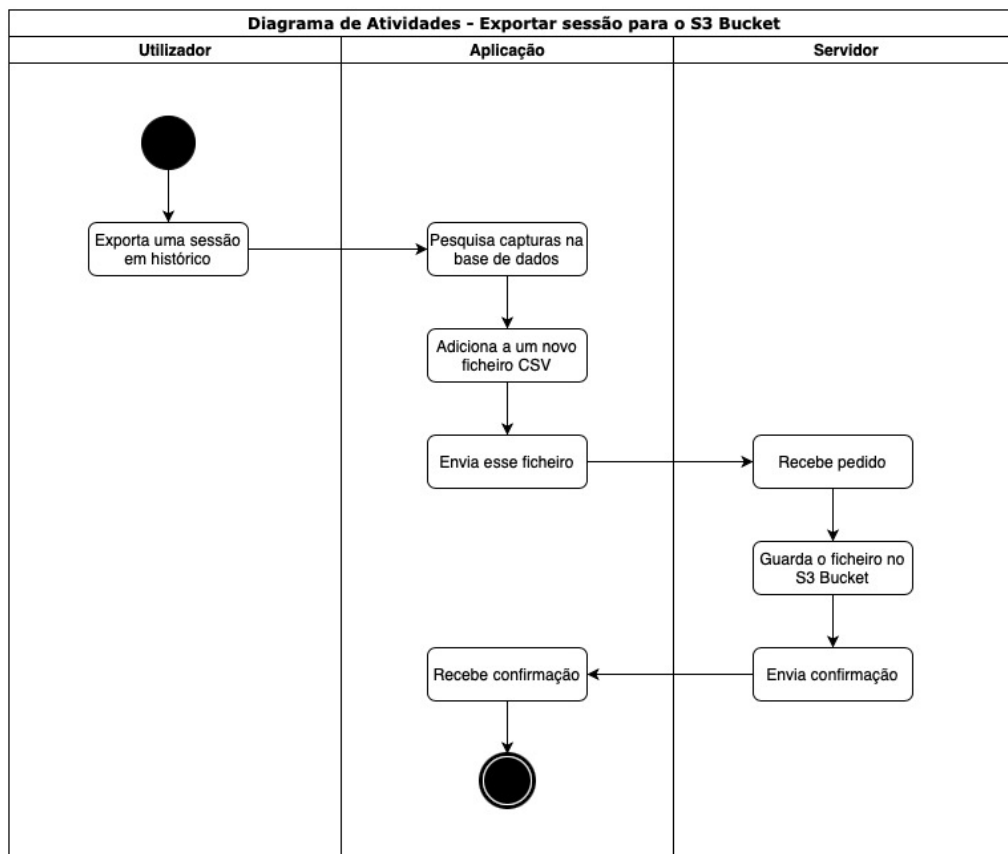


Figura 6 - Diagrama de Atividades: Exportar sessão para o S3 Bucket

2.4 Modelo de Base de Dados

Durante uma sessão, a aplicação vai efetuando capturas, cada uma é válida quanto o instante em que obtemos a localização é semelhante ao instante em que obtemos a potência de sinal. A dificuldade passa por obter ambos os dados de APIs distintas em simultâneo. Quando uma captura é validada com sucesso, esta é guardada na lista da sessão. Assim que utilizador termina a sessão, os dados da mesma são guardados na base de dados local. Apresenta-se o UML da nossa aplicação na Figura 7.

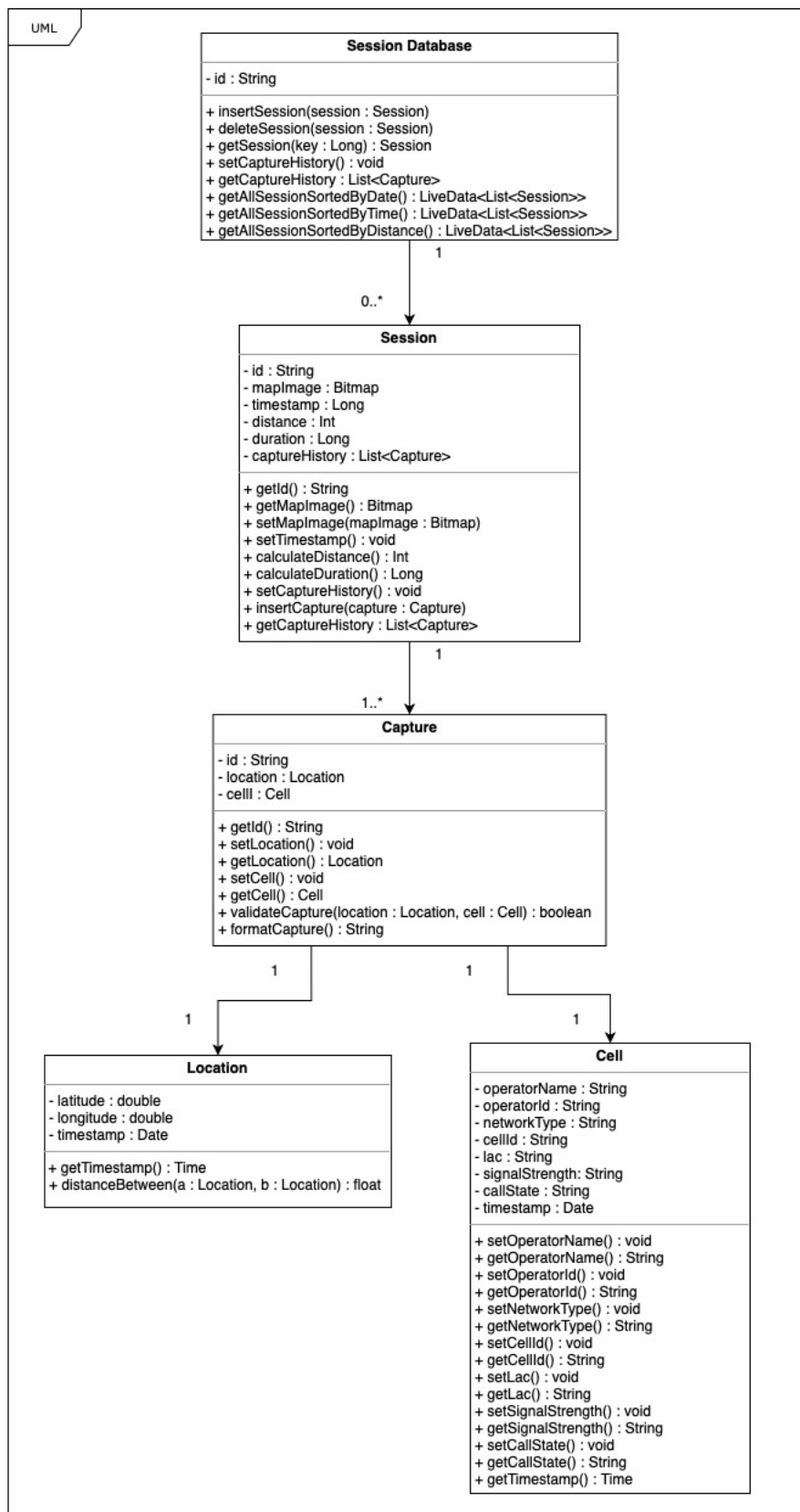


Figura 7 - UML

3 Viabilidade e Pertinência

O DT captura medidas da rede móvel com equipamento específico. Habitualmente, requer o uso de um veículo, limitando a sua área de testes. A aplicação proposta não precisa de hardware específico, reduzindo o investimento e pode ser utilizada em locais inalcançáveis por um veículo. Os dados obtidos pela aplicação são úteis para melhorar a precisão do modelo de estimação de cobertura. Permite assim viabilizar uma funcionalidade, que terá grande utilidade, pois de forma totalmente automática, permitirá melhorar a estimação de cobertura, sem custos associados (aplicação gratuita). A estimativa criada por este modelo, encontra-se ilustrado na Figura 3.

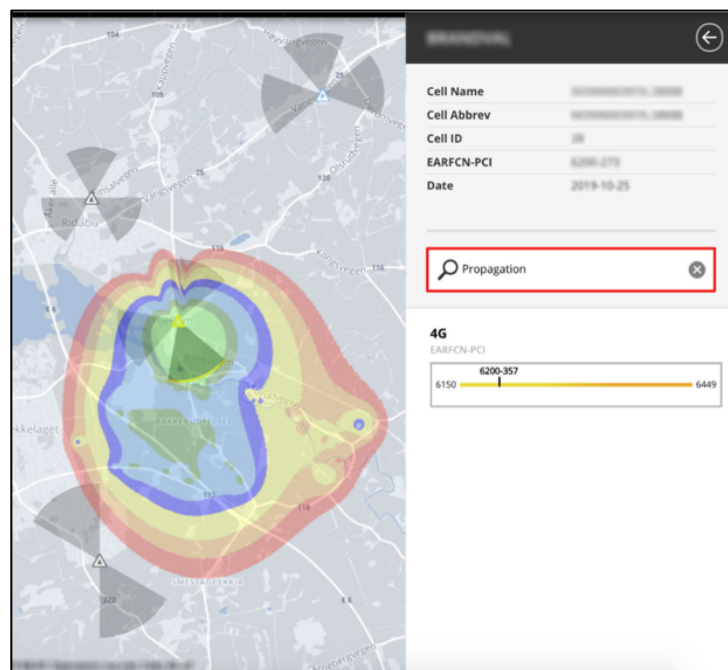


Figura 8 - Estimativa da cobertura de uma célula. Integrado na plataforma Metric. Extraído de [3].

O engenheiro com esta aplicação, pode recolher informação e efetuar alterações sustentadas pelas mesmas, melhorando a qualidade da rede. Isto realça a pertinência do presente trabalho.

Na reunião de 16/12/2020 foi discutido com o Engenheiro Nuno Valente a possibilidade de criar um questionário que permitisse aferir a viabilidade e pertinência deste projeto. Visto este projeto ser um “pedido do cliente”, não faz sentido validar se o projeto é adequado. O objetivo deste TFC é resolver os problemas demonstrados pela Multivision.

Como já referido, no passado, foi desenvolvido um protótipo sobre este tema. O Engenheiro Nuno Valente diz que, no entanto, tendo em conta que os nossos atuais clientes são operadores estabelecidos no mercado, a utilização do aplicativo tem menor relevância, visto que já todos investiram em equipamento próprio para executar *Drive Tests* ou, já foi feita a integração com ferramentas que as equipas de Engenharia acabam por usar sem supervisão do operador. A ferramenta desenvolvida acabou por ser descontinuada temporariamente, mas é pretendido retomar o trabalho na mesma por vários motivos. Apesar do que foi referido, o Metric tem como

ambição alcançar novos clientes, com contratos a médio e longo prazo, que ainda não têm equipamento dedicado e recorrem a consultoria externa, ou que apenas pretendem reduzir os custos associados à execução de *Drive Tests*. Para que a proposta comercial faça sentido, convém que seja apresentada uma solução de ponta a ponta sem requisitar investimentos extra por parte dos nossos clientes. Por outro lado, o Metric é um produto Multivision, utilizado pelos consultores de telecomunicações da empresa para executar trabalhos de planeamento, instalação e otimização da rede móvel. É importante para a equipa de gestão de produto e equipa comercial que se possa abraçar qualquer desafio de forma autónoma, sem recurso a tecnologias de fornecedores externos, vendendo serviços de consultoria contidos na Multivision.

No que toca ao ponto de crescimento em termos de clientes a médio e longo prazo, é claro que a aplicação não vai cobrir todas as necessidades de um produto de planeamento e otimização, mas serve como alavanca para a penetração no cliente, e por sua vez, fonte de divisas para investimento em tecnologia mais abrangente. Em relação a serviços de consultoria, o uso deste tipo de aplicações, permite uma agilidade comercial maior visto que apenas um consultor consegue gerir uma equipa de campo (responsável por instalar o equipamento físico) em diversos projetos que fazem parte do ciclo de negócio de um operador. Atividades como a instalação de uma nova antena ou reparação de equipamento carecem de poucos indicadores para a sua validação. O conjunto de indicadores necessário para as primeiras fases destes projetos, pode ser obtido com aplicação deste calibre. Não é necessário impor ao operador que nos forneça equipamento específico, sendo apenas necessária a adjudicação do serviço de consultoria. Como se pode compreender, a plataforma pode ser estendida a consultores que não pertençam à Multivision, mas que façam este tipo de atividade. Como serviço SaaS, o Metric permite que um consultor faça uso do mesmo pelo tempo necessário, não tendo custos extra de manutenção de equipamento. Qualquer consultor de telecomunicações com experiência suficiente necessita apenas de usar as ferramentas integradas do Metric para concluir projetos da tipologia já referida.

Finalmente, e como caso de uso de apresentação da ferramenta, pretendemos que os Engenheiros de Telecomunicações de um operador, possam usar a ferramenta “Android Walk Test” no seu dia-a-dia, tendo uma visão da performance da rede nas zonas por onde passam. Este tipo de utilização permite uma ligação emocional mais forte com o produto onde o próprio Engenheiro de Telecomunicações pode detetar falhas graves e resolver as mesmas antes que as reclamações dos clientes finais se empilhem na sua lista de tarefas.

4 Solução Desenvolvida

4.1 Arquitetura

Nesta secção apresenta-se a arquitetura da solução desenvolvida, ilustrada na Figura 9 e descrita a seguir, identifica-se os componentes desenvolvidos, componentes com que se relaciona e fluxos de dados.

Código do projeto disponível no GitHub: <https://github.com/a21803482/WalkTest>

Demo da aplicação disponível no YouTube: <https://youtu.be/nWjJtd5Lgck>

Foi enviado um ficheiro APK da aplicação desenvolvida para o Moodle. Encontra-se em Anexo o guião de instalação.

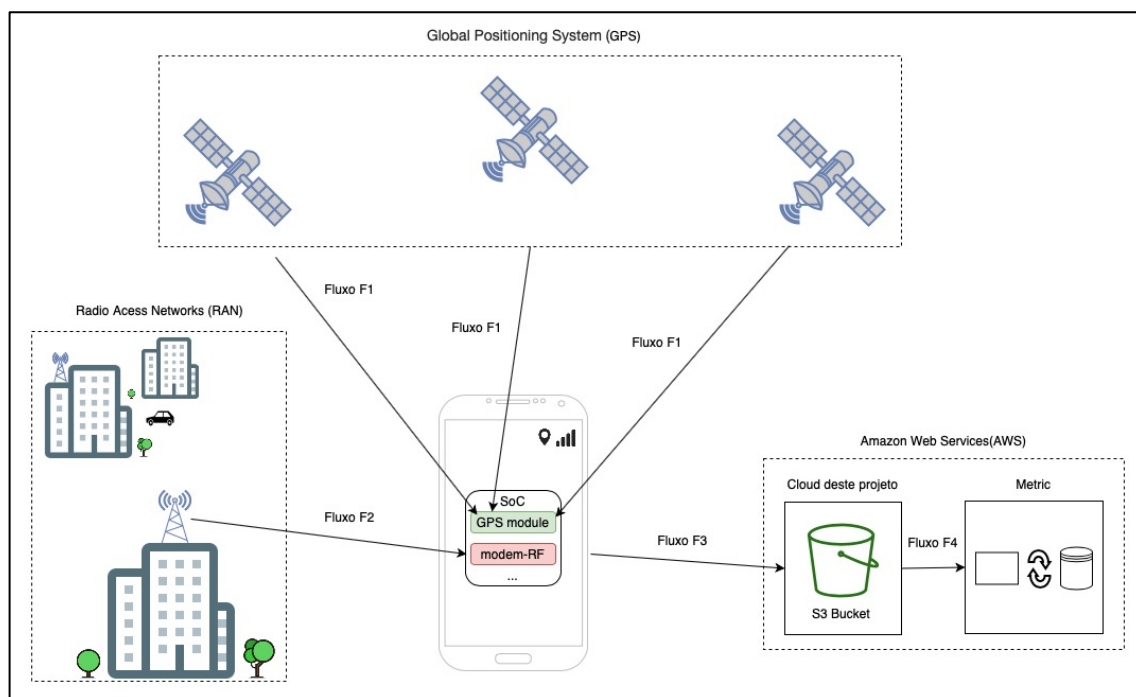


Figura 9 - Arquitetura

As componentes da solução desenvolvida e opções tecnológicas são as seguintes:

- **Aplicação móvel (Android):** Foi desenvolvida uma aplicação Android para monitorizar o sinal de RAN (Rede de Acesso Rádio).
- **Serviço de armazenamento Cloud (Amazon S3 Bucket):** Os dados recolhidos são armazenados para a Cloud, usa-se o serviço de armazenamento Amazon S3 Bucket da AWS, permitindo ao Metric aceder aos mesmos.

Outros componentes com os quais a nossa aplicação se relaciona:

- **Rede de Acesso Rádio (RAN):** Existem diferentes tipos de acesso RAN utilizados pelo smartphone: 2G, 3G, 4G e 5G. O dispositivo através da aplicação, monitoriza o nível de sinal recebido.

- **Sistemas de posicionamento global via satélite (GPS):** Permite ao smartphone obter dados de localização (coordenadas) e o tempo atual, fazendo uso dos sensores de localização do dispositivo.
- **Plataforma Metric:** Utilizada para monitorizar e otimizar as redes de comunicação móvel. Esta aplicação permite ao Metric recolher dados provenientes de um smartphone.

De seguida, explica-se os fluxos de dados desta arquitetura identificados na Figura 9:

- F1. Smartphone recebe dados provenientes do sistema GPS que permitem obter as coordenadas de posição atual e tempo.
- F2. Smartphone recebe dados provenientes da RAN com a qual tem uma ligação estabelecida.
- F3. Smartphone envia os seus dados para um serviço de armazenamento Cloud.
- F4. O Metric vai importar os dados no armazenamento Cloud, permitindo a sua visualização e processamento para outras aplicações.

Esta aplicação tem dois fluxos de entrada (F1 e F2), para o F1, os dados necessários para criar cada captura são requisitados dos sensores de localização (GPS) e para o F2, requisitamos os dados do modem presente no *System on a Chip* (SoC). Estes dados são utilizados para as seguintes funcionalidades:

- Demonstrar a posição geográfica e potência do sinal atual;
- Realizar uma sessão de capturas e memorizar as mesmas na memória interna;
- Visualização no mapa do progresso da captura;
- Enviar para a plataforma Metric as sessões de capturas.

Os dados provenientes dos fluxos de entrada F1 e F2 geram capturas para cada instante, estas capturas fazem parte de uma sessão com tempo determinado pelo utilizador. Os dados da sessão (capturas) são armazenados na memória interna do dispositivo e o seu resultado é utilizado no fluxo de saída F3, onde as sessões são exportadas para o AWS S3 Bucket.

Um DT tradicional, usa equipamento específico para efetuar capturas e apresenta algumas desvantagens, descritas na Identificação do Problema. A aplicação desenvolvida permite aumentar o volume de medidas (capturas) obtidas, assim como capturar em locais inacessíveis por um Drive Test tradicional. Durante um teste (sessão) vai efetuar capturas, estas capturas são constituídas por métricas sobre o sinal da rede naquele instante e estão associadas à posição geográfica, onde foram efetuadas. Portanto, uma sessão é constituída por uma ou mais capturas e tem uma duração determinada pelo utilizador.

Em particular, cada captura inclui os seguintes dados (por esta ordem):

- Data da captura (YYYY-MM-DD HH:MM:SS:CCC)
- Nome da operadora
- ID da operadora
- Tipo de rede

- CID (*Cell ID*)
- LAC (*Location Area Code*)
- Signal Strength
- Informação se utilizador se encontra ou não em chamada (Call State)
- Coordenadas da posição atual

As sessões incluem os seguintes dados:

- Imagem com o mapa no final da sessão (inclui todos os marcadores)
- Data do início da sessão (YYYY-MM-DD HH:MM:SS:CCC)
- Distância percorrida
- Duração total da sessão
- Lista com todas as capturas efetuadas durante a sessão

Quando uma sessão é terminada com sucesso, esta fica registada em memória e apresenta na lista de histórico. Podemos observar na Figura 10 uma sessão que tem como imagem o mapa no final da sessão e legendada com a data, tempo de duração e distância percorrida.

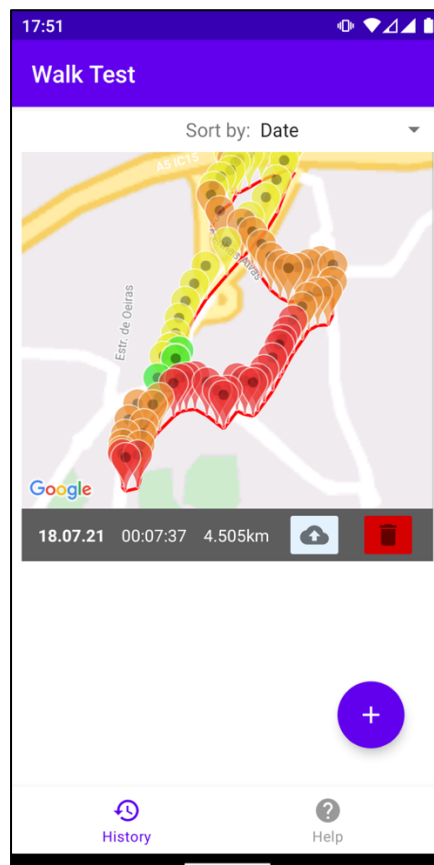


Figura 10 - Sessão no histórico da aplicação

Nas secções seguintes detalham-se os dois componentes desenvolvidos: aplicação móvel e solução de armazenamento Cloud.

4.2 Implementação da aplicação móvel

No desenvolvimento desta aplicação Kotlin para Android, utilizamos o Android Studio (SDK) para a programação e um dispositivo com Android 10 para testar no terreno. Nesta secção apresenta-se a arquitetura da aplicação Android, ilustrada na Figura 14. A aplicação tem uma arquitetura Model-View-ViewModel (MVVM).

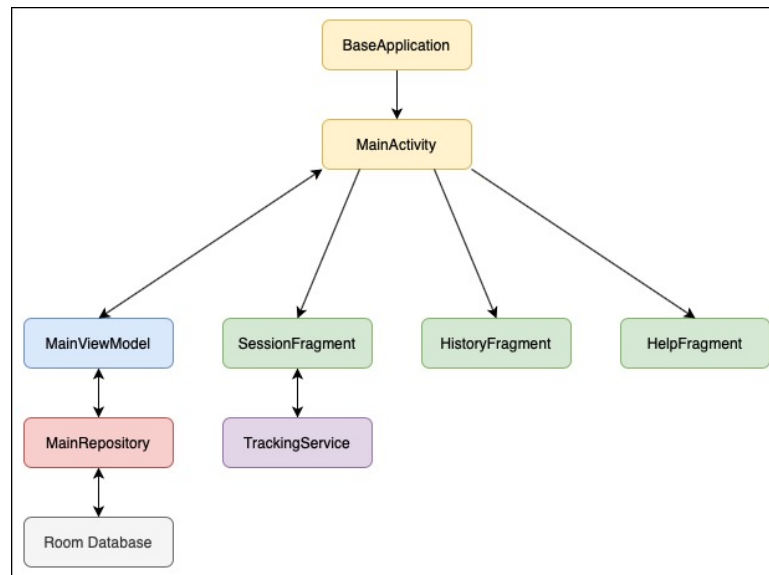


Figura 11 - Arquitetura da aplicação Android

As linguagens de programação utilizadas para o desenvolvimento da aplicação são:

- **Kotlin:** Responsável por toda a lógica e funcionalidade da aplicação.
- **XML:** Responsável pela UI apresentada ao utilizador.

Ao contrário do hardware específico, desenhado especificamente para os Drive Tests, o sistema operativo Android, tem algumas limitações. Estamos restringidos pelos dados que conseguimos obter no Android através das APIs.

A informação dos componentes com os quais a aplicação se relaciona, são obtidas através das APIs seguintes:

- **Android Telephony:** Fornece informações sobre o estado do smartphone na rede.
- **Google Fused Location:** Permite obter as coordenadas do dispositivo no instante atual.
- **Google Maps SDK:** Permite-nos adicionar mapas (do Google Maps) à aplicação Android. Utilizado para demonstrar a nossa localização atual e marcadores (com cor personalizada) com a posição (coordenadas) onde foram capturadas.
- **AWS Amplify API [12]:** Disponibiliza uma interface para efetuar pedidos ao nosso armazenamento Cloud (Back-End).

Apresenta-se as ferramentas utilizadas para facilitar o desenvolvimento do projeto:

- **Arquitetura Model-View-ViewModel (MVVM):** Esta arquitetura permite uma manutenção simples, pois o código encontra-se dividido em ficheiros diferentes, consoante o seu tipo. Neste caso, a MainViewModel sabe sobre os dados, mas não sabe nada sobre a UI. Essa responsabilidade fica ao cargo da View. Esta separação, permite reutilizar a lógica sobre os dados em diversos fragmentos (Views) da aplicação.
- **Componentes de navegação (Ficheiro XML):** Esta componente permite definir fluxos de navegação. Podemos definir ações que permitem transitar entre fragmentos, estas ações são acionadas quando necessário pela parte lógica das aplicações.
- **Room Database com Coroutines:** As sessões são guardadas numa base de dados local. Cada linha nesta base de dados corresponde a uma sessão e armazena os dados de sessão descritos anteriormente.
- **Dependency Injection com o Dagger-Hilt:** Os objetos (variáveis) muitas vezes têm dependências. Uma Dependency Injection ajuda-nos a diminuir o boilerplate code nessas situações, isto é, as dependências estão numa classe gerida pelo Dagger e posteriormente, quando precisamos de um objeto não precisamos de criar novamente todas as dependências.

A implementação da solução foi dividida em vários momentos que são:

1. **Pedir Permissões de privacidade:** Inicialmente, implementamos as permissões de privacidade para aceder à localização do utilizador e estado do telefone, só com a autorização do utilizador, podemos fazer uso das APIs, Android Telephony e Google Fused Location para correto funcionamento da aplicação.
2. **Obter dados da API Telephony e Google Fused Location:** Exploramos todos os dados que era possível obter a partir destas APIs. O Google Fused Location permite-nos obter as coordenadas atuais e o Telephony permite-nos obter as informações da captura. Cada captura é efetuado num intervalo de 3 segundos.
3. **Guardar em memória persistente capturas:** O objetivo é armazenar os dados das nossas capturas na memória interna. Utiliza-se uma Room Database para cumprir este objetivo. Consegue-se com sucesso inicializar a base de dados, adicionar e eliminar capturas.
4. **Apresentar informação das capturas durante a sessão no ecrã da mesma:** Melhoramos a UI durante uma sessão para a que seja apresentada as métricas relativas à última captura. Esta informação é atualizada à medida que são efetuadas novas capturas.
5. **Apresentar Mapa com marcadores de cada captura e trajeto:** Durante o desenrolar de uma sessão, podemos visualizar no ecrã da sessão um mapa, centrado na nossa posição atual, que vai adicionado um marcador sempre que efetuar uma captura naquelas coordenadas. Estes marcadores podem ter 4 cores diferentes consoante a intensidade do seu sinal: Verde, Amarelo, Laranja ou Vermelho. Além disso, podemos interagir com o mapa, visualizar o trajeto percorrido e sempre que tocamos num marcador, este apresenta uma label com a informação do tipo de tecnologia (Ex: LTE) no momento da captura e o Cell ID. Marcador selecionado (com label) ilustrado na Figura 12.
6. **Continuar sessão iniciada em background:** Caso o utilizador tenha iniciado uma sessão, esta prossegue, mesmo que utilizador saía da aplicação ou desligue o ecrã. Quando uma sessão se encontra em funcionamento, é apresentada uma notificação persistente na

central de notificações, a partir da qual pode parar ou ver a duração da mesma. Ilustra-se na Figura 13 a notificação demonstrada durante uma sessão.

7. **Exportar os dados de uma sessão para o AWS:** Foi utilizada a API do Amplify para configurar via *Terminal* o espaço de armazenamento (S3 Bucket), assim como as chaves de autenticação que ficam armazenados em 2 ficheiros de configuração nas pastas do nosso projeto. Estes ficheiros são verificados quando utilizamos a API nas nossas classes para inicializar o serviço Amplify, assim como efetuar o upload para a Cloud.

A solução final apresenta as seguintes funcionalidades:

- Efetua capturas das métricas pretendida durante uma sessão
- Demonstra informação sobre a captura atual no ecrã da sessão
- Apresenta mapa interativo com marcadores que representam cada captura
- Percurso percorrido (trajeto) é representado no mapa
- Sessões podem funcionar em segundo plano (background)
- Armazena as sessões na memória interna do dispositivo
- Ordenar a lista com o histórico das sessões pela: Data, Distância, Tempo de Sessão
- Calcula a distância percorrida durante uma sessão de captura
- Podemos eliminar sessões
- Exporta sessões para a Cloud quando pretendido

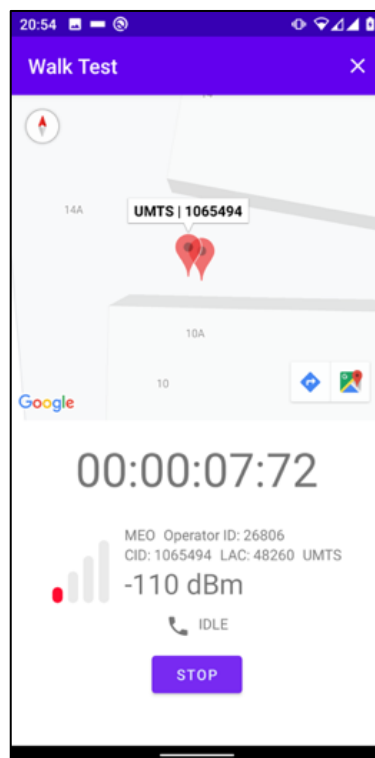


Figura 12 - Mapa com marcadores no ecrã de sessão

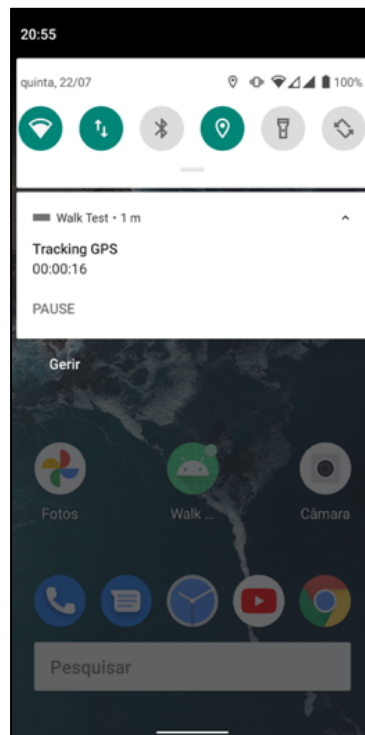


Figura 13 - Notificação durante uma sessão

A aplicação é constituída por 3 ecrãs (páginas/fragmentos) diferentes ilustrados na Figura 14:

- Histórico (página inicial): Onde se encontra uma lista com todas as sessões realizadas. Além disso, podemos começar, eliminar ou exportar uma sessão na mesma.
- Página de sessão: Onde verificamos o progresso da sessão (mapa e informação da última captura), assim como controlar a sessão. Os controlos possíveis são os seguintes: Iniciar, Pausar, Terminar e Cancelar.
- Página de ajuda: Encontra-se nesta página 3 tabelas de referência com os intervalos utilizados na lógica da aplicação para classificar o Signal Strength consoante a tecnologia (2G, 3G e 4G). Estes valores são utilizados para definir a cor do marcador.

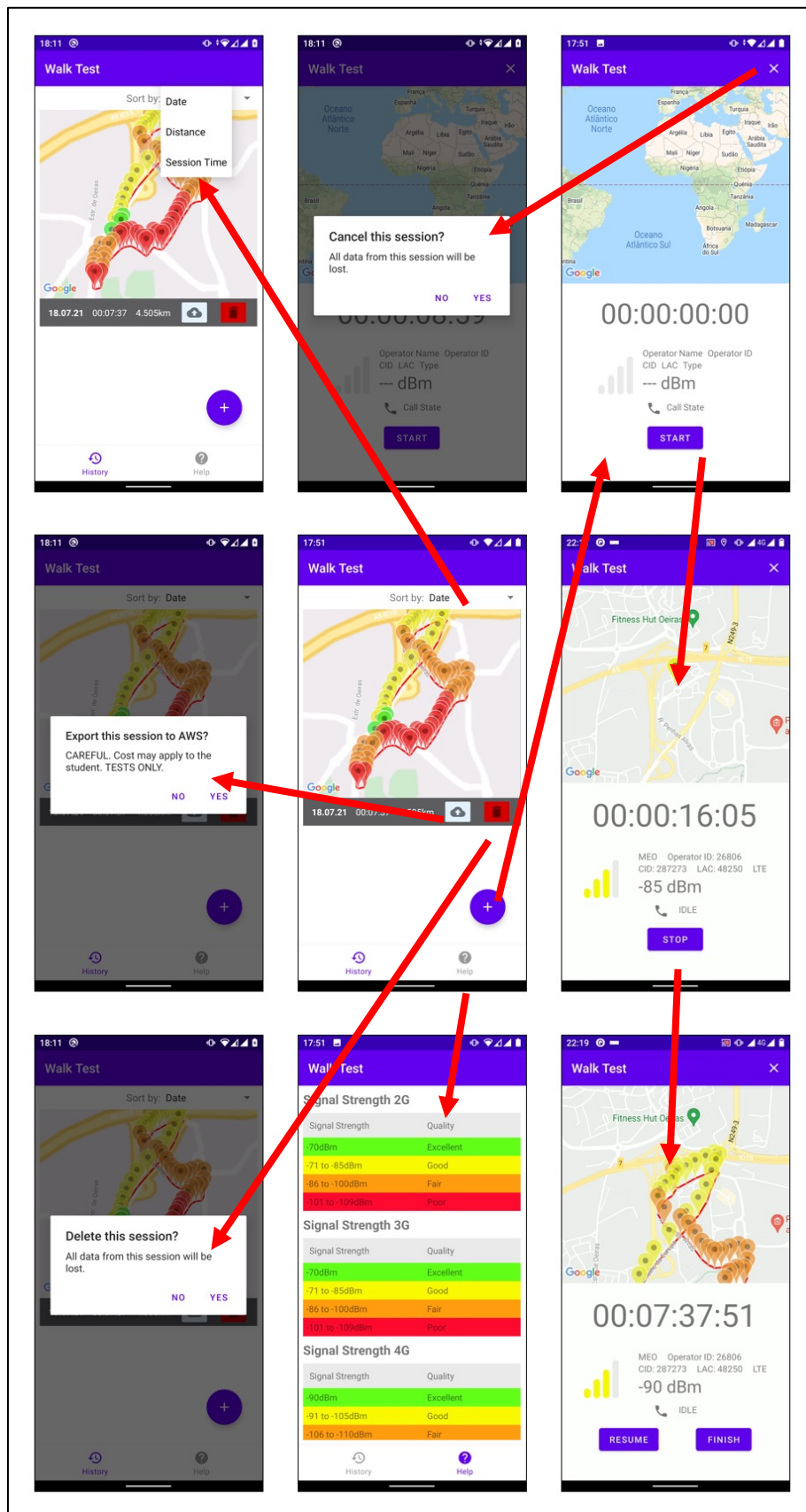


Figura 14 - Storyboard da Aplicação

4.3 Implementação na Cloud

A aplicação exporta as capturas de uma sessão num ficheiro CSV para o S3 Bucket. Cada linha deste ficheiro corresponde a uma captura da sessão. Formato da linha:

YYYY-MM-DDTHH:MM:SS:CCC, Operator Name, Operator ID, Network Type, Cell ID, LAC, Signal Strength, Call State, Latitude, Longitude

A configuração do Amplify foi efetuada via terminal, após a configuração, foi adicionado ao folder do projeto dois ficheiros de configuração com o ilustrado na Figura 15.

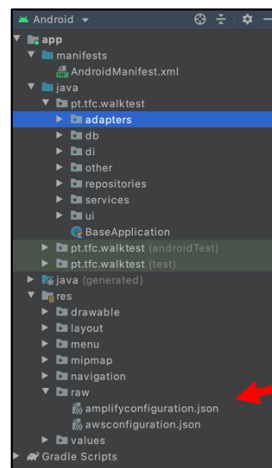


Figura 15 - Ficheiro de configuração Amplify

O serviço Amplify é inicializado na classe BaseApplication como ilustrado na Figura 16.

```
@HiltAndroidApp
class BaseApplication: Application() {
    @SuppressLint(value = "LogNotTimber")
    override fun onCreate() {
        super.onCreate()
        Timber.plant(Timber.DebugTree())

        try {
            // Add these lines to add the AWS CognitoAuthPlugin and AWSS3StoragePlugin plugins
            Amplify.addPlugin(AWSCognitoAuthPlugin())
            Amplify.addPlugin(AWSS3StoragePlugin())
            Amplify.configure(applicationContext)

            Log.i(tag: "MyAmplifyApp", msg: "Initialized Amplify")
        } catch (error: AmplifyException) {
            Log.e(tag: "MyAmplifyApp", msg: "Could not initialize Amplify", error)
        }
    }
}
```

Figura 16 - Inicialização do serviço Amplify

O Amplify pode agora ser utilizado na nossa aplicação noutras classes para efetuar as diversas ações (Ex: Upload de ficheiros). Quando o utilizador seleciona o botão para exportar uma sessão na página de histórico, é criado um ficheiro CSV local para o qual são adicionadas todas as capturas da sessão correspondente. Esse ficheiro, quando pronto, é enviado para o S3 Bucket através do Amplify. Ilustra-se na Figura 17 o código de quando esse ficheiro é criado e enviado para a Cloud. Ilustra-se na Figura 18 todas as sessões exportadas, guardadas no Bucket em ficheiro CSV, cada um deles tem um filename específico: {Data do Início da Sessão}-WalkTest.

Figura 17 - Código da exportação da sessão

Figura 18 - Ficheiros no Bucket S3

[illegible]

Figura 19 - Ficheiro CSV

5 Benchmarking

As operadoras beneficiam muito com as capturas, pois permite uma melhor compreensão da sua rede. Além disso, foi discutido no capítulo 2 dos Requisitos Funcionais a realização de uma 4.ª fase (ambicioso), na qual a aplicação que podia ser usada por todos e beneficie aqueles que a usem (Exemplo: Operadoras oferecem dados móveis a todos os utilizadores que efetuem capturas com frequência). Seria interessante no futuro, poder diagnosticar com maior eficácia problemas de rede. Existem várias opções no mercado com ferramentas que permitem obter capturas de rede, mas nem todas se integram diretamente no sistema Metric, outras são dispendiosas ou requerem equipamento específico.

Algumas ferramentas de “Drive Test” compatíveis com o Metric:

- **TEMS Network Testing Solutions** (Infovista) [4]: Utilizado por muitos operadores de telecomunicações. Tem um portfólio de soluções que permite aos seus clientes diagnosticar a qualidade da rede móvel com equipamento específico ou dispositivo móvel. Permite a exportação de dados, mas tem um custo associado;
- **XCAL-Mobile** (Accuver) [6]: Aplicação Android que permite capturar dados de rede móvel e exportar. Metric tem suporte “semiautomático” (limitações na importação) para exportações desta ferramenta. Não se encontra atualizada para versões mais recentes do Android e tem um custo associado;
- **G-NetTrack Pro** (Gyokov Solutions) [7]: Aplicação Android capaz de monitorizar a rede móvel e efetuar “Drive Tests” e exportação de dados. Tem um custo associado.

Outras ferramentas de “Drive Test”:

- **R&S TS9955** (Rohde & Schwarz) [5]: Exemplo de um sistema tradicional para “Drive Test” instalado num veículo. Permite recolher capturas detalhadas e exportação, mas tem um alcance limitado e um custo elevado. O suporte desta ferramenta no Metric foi descontinuado, visto a empresa não ter contratos recentes com operadores que usem a mesma;
- **QualiPoc Android** (Rohde & Schwarz) [11]: Aplicação Android capaz de diagnosticar a qualidade da rede móvel. Permite a exportação de dados e diferentes tipos de testes: Voice, Data, Video Streaming e “dedicated OTT apps (e.g. Facebook, WhatsApp, etc.). Tem um custo associado.

A Tabela 2 compara as ferramentas de *Drive Test* referidas anteriormente.

Tabela 2 - Tabela de comparação

Ferramenta	Gratuita	Compatível com a versão mais recente do Android	Compatível com o Metric	Portátil ²
TEMS	Não	Sim	Sim	Sim
XCAL-Mobile	Sim	Não	Sim	Sim
G-NetTrack Pro ³	Não	Sim	Sim	Sim
R&S TS9955	Não	Não	Não	Sim
QualiPoc Android	Não	Sim	Não	Sim
Aplicação anterior da Multivision ⁴	Sim	Não	Sim	Sim
Aplicação desenvolvida	Sim	Sim	Sim	Sim

² Permite a recolha de capturas em qualquer local. Ainda está limitado a um veículo equipado com equipamento específico.

³ Existe uma versão gratuita com menos funcionalidades: G-NetTrack Lite.

⁴ Multivision desenvolveu em 2017 um protótipo de uma aplicação Android que permite captura e exportar capturas.

6 Método e planeamento

Apresenta-se nesta categoria o plano de trabalho e cronograma. O plano de trabalho encontra-se dividido em duas listas: Tarefas e Entregáveis. Na lista de tarefas, encontram-se todos os desafios necessários que permitiram realizar os entregáveis com sucesso. Na Tabela 3 encontra-se o cronograma com todas as tarefas e entregáveis, assim como uma linha de progresso.

Tarefas:

- T1 Compreensão do desafio proposto;
- T2 Desenvolvimento da 1.ª Aplicação Android "Hello World" ⁵;
- T3 Levantamento e análise de requisitos;
- T4 Preparação da 1.ª apresentação oral;
- T5 Desenvolvimento do protótipo funcional "Android Walk Test App";
- T6 Desenvolvimento da funcionalidade capaz de aceder às APIs necessárias;
- T7 Desenvolvimento da base de dados na aplicação Android;
- T8 Desenvolvimento do serviço de *Tracking*;
- T9 Desenvolvimento da funcionalidade para exportar sessões; **NOVO**
- T10 Validação da arquitetura;
- T11 Testes para validação prática da solução;
- T12 Análise de resultados e conclusões;
- T13 Preparação da 2.ª apresentação oral. **NOVO**

Entregáveis:

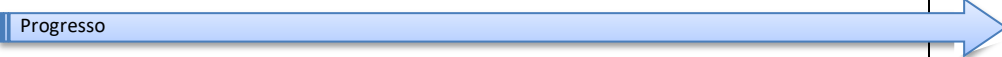
- | | | | |
|------|--------------------------------------|------------|------------------|
| • R1 | Relatório intercalar de 1.º semestre | 27/11/2020 | ENTREGUE |
| • R2 | Relatório intermédio de 1.º semestre | 22/01/2020 | ENTREGUE |
| • R3 | Relatório intercalar de 2.º semestre | 23/04/2021 | ENTREGUE |
| • R4 | Entrega em 1.ª época | 25/06/2021 | NÃO FOI ENTREGUE |
| • R5 | Entrega em 2.ª época | 23/07/2021 | ENTREGUE |

Face ao planeamento referido em entregas anterior existem alterações a destacar:

- **Introdução da tarefa T9:** Na proposta inicial a aplicação ia exportar os dados da sessão em ficheiro CSV diretamente para o Back-end (target) da Multivision. Durante do desenvolvimento, acabamos por optar por configurar o S3 Bucket, exportar para o mesmo e o Metric que tem parte da sua infraestrutura na AWS, acede aos ficheiros nesse Bucket. Na Tabela 3 destaca-se estas alterações a **amarelo**.
- A **tarefa T5 e T8** foram mais demoradas que o planeado. A arquitetura MVVM apesar de oferecer diversas vantagens, tem maior complexidade, sempre a principal causa do atraso para a entrega em 2.ª Época. Na Tabela 3 destaca-se estas alterações a **laranja**.

⁵ Criar 1.ª aplicação Android "Hello World". Esta deve ter a capacidade de demonstrar no ecrã as coordenadas e potência de sinal (da rede móvel) atual. Estas medidas devem atualizar ao longo do tempo.

Tabela 3 - Cronograma

	Novembro	Dezembro	Janeiro	Fevereiro	Março	Abril	Maio	Junho	Julho
	<div> <div>Progresso</div>  </div>								
T1	T1								
T2		T2							
T3			T3						
T4			T4						
T5				T5					
T6				T6					
T7				T7					
T8					T8				
T9							T9		
T10							T10		
T11							T11		
T12									T12
T13									T13
R1	R1								
R2			R2						
R3						R3			
R4								R4	
R5									R5

7 Resultados

Esta secção valida os assuntos detalhados nos capítulos 2, 3 e 4. Explica-se a seguir como testamos os requisitos implementados, assim como o comportamento da solução implementada no terreno. Foram utilizados dois *Smartphones* com Android (v.10) para realizar os testes. Os testes começam com duas condições essenciais, o Smartphone tem sinal de rede móvel e a localização ativa.

Apresenta-se de seguidas algumas funcionalidades testadas que merecem de destaque. Por último, a Tabela 4 identifica, descreve e apresenta o OUTCOME para cada teste que foi realizado.

Informação da rede

Quanto à operadora, facilmente conseguimos identificar se a operadora do SIM presente no dispositivo, corresponde com o da informação guardada na aplicação. O Android permite seleccionar manualmente o tipo de rede que pretendemos, podendo assim confirmar se a aplicação guarda nas capturas o tipo de rede correto, consoante a configuração do tipo de rede nas definições do dispositivo. O Signal Strength, CID e LAC presumimos que estejam corretos, pois são diretamente provenientes das API. Tentamos estabelecer um acordo com a NOS, que iria permitir obter informação junto da operadora sobre as antenas nas proximidades e assim poder comparar a informação obtida pela aplicação com a fornecida pela operadora, infelizmente, não avançou em tempo útil. Confirma-se o requisito **RF1.2**.

Coordenadas e representação no mapa

Durante uma sessão, dentro da aplicação, visualizamos um mapa com a atualização atual e o percurso realizado na sessão. Torna-se assim possível certificar-se que o Smartphone está a recolher capturas na área correta e a adicionar marcadores para cada captura com a cor correspondente ao valor do Signal Strength apresentado. Confirma-se que **RF1.1**, **RF2.3** e **RF3.2** estão devidamente implementados.

Memorizar capturas

Após uma sessão terminada com sucesso, somos redirecionados para a página de histórico onde podemos ver no topo a sessão mais recente. A sessão é apresentada, logo o armazenamento na base de dados foi bem-sucedido. Requisito **RF1.3** está devidamente implementado.

Exportar para o S3 Bucket

As capturas armazenadas podem ser exportadas para o S3 Bucket. O ficheiro CSV é gerado com o formato correto, enviado e fica acessível no S3 Bucket. Confirma-se que o requisito **RF1.4** e **RF1.5** encontram-se testados e a funcionar corretamente.

Tabela 4 - Testes realizados

ID	Título do teste	Descrição	Requisitos testados	Resultado esperado	Resultado obtido
1	Capturar medidas	Efetuar capturas válidas. Cada uma tem que conter coordenadas e informação do sinal.	RF1.1, RF1.2	Capturas efetuadas com sucesso.	Sim
2	Controlar progresso da sessão	Utilizador pode começar, pausar, terminar ou cancelar uma sessão.	RF2.1	Utilizador consegue controlar sessão sem problemas.	Sim
3	Representação no mapa	Durante uma sessão, um mapa centrado na posição atual, vai adicionando o trajeto e marcadores de cores diferentes, consoante a intensidade do sinal.	RF2.3, RF3.1, RF3.2	O mapa representa a informação corretamente.	Sim
4	Memorizar sessões	Confirmar se a sessão terminada fica em memória.	RF1.3	Sessões são memorizadas e aparecem na lista.	Sim
5	Eliminar sessões	Utilizador pode eliminar sessões da lista.	RF1.3	Sessão selecionada é eliminada.	Sim
6	Alterar a ordem do histórico	O histórico pode ser ordenado por 3 critérios: Data, Distância e Tempo.	RF1.3	Utilizador altera o ordem da lista.	Sim

7	Exportação para o S3 Bucket	Utilizador seleciona uma sessão para exportar. A aplicação gera um ficheiro com os dados da mesma e envia para o S3 Bucket.	RF1.4, RF1.5	Ficheiro é adicionado ao Bucket.	Sim
---	-----------------------------	---	--------------	----------------------------------	-----

8 Conclusão e trabalho futuros

A rede móvel é extrema importância no nosso dia-a-dia. Este projeto a melhor a compreensão da mesma, assim como contruir para o diagnóstico de um sistema com tremenda complexidade. A solução desenvolvida permite agora aos seus utilizadores recolher de capturas de forma flexível, esteja onde estiver.

Para o desenvolvimento deste projeto apliquei muitos dos conhecimentos adquiridos em diversas unidades curriculares das quais se destacam:

- Computação Móvel: A aplicação Android foi desenvolvida em Kotlin com arquitetura MVVM. Ambas foram trabalhadas nesta UC.
- Sistemas de Informação na Nuvem: A solução deste projeto usa a infraestrutura AWS, através do Amplify para enviar e armazenar os nossos dados num Bucket. A UC em questão trabalhou ao longo do semestre com diversos serviços do AWS.
- Interação Humano-Máquina: Grande parte do esforço deste projeto acaba representado nas páginas da nossa aplicação, através de resultados. A importância de uma boa UI e UX permite que tudo a dedicação, tenha como consequência uma boa usabilidade e corretor usufruto de todas as funcionalidades implementadas.

Por último, no futuro seria interessante explorar como este tipo ferramentas (aplicações) se fossem massificado (crowdsourcing), poderia contribuir para o progresso da qualidade de rede móvel.

Bibliografia

- [1] Multivision. Disponível em: www.multivision.pt.
- [2] Metric. Disponível em: www.metric.pt.
- [3] [DFCF20] D. Fernandes, D. Clemente, G. Soares, P. Sebastião, F. Cercas, R. Dinis and L. S. Ferreira, "Cloud-Based Implementation of an Automatic Coverage Estimation Methodology for Self-Organising Network", pp. 66456-66474, Volume 8, Abr. 2020.
- [4] InfoVista TEMS. Disponível em: www.infovista.com/tems.
- [5] R&S TS9955. Disponível em: www.rohde-schwarz.com/en/product/ts9955productstartpage_63493-8705.html.
- [6] XCAL-Mobile. Disponível em: accuver.com/acv_products/xcal-mobile/.
- [7] Gyokov Solutions. Disponível em: gyokovsolutions.com.
- [8] tem.REDE. Disponível: anacom.maps.arcgis.com.
- [9] android.telephony. Disponível: <https://developer.android.com/reference/android/telephony/package-summary>
- [10] com.google.android.gms.location. Disponível: <https://developers.google.com/android/reference/com/google/android/gms/location/package-summary>
- [11] QualiPoc Android (Rohde & Schwarz). Disponível: https://www.rohde-schwarz.com/pt/product/qualipoc_android-productstartpage_63493-55430.html
- [12] AWS Amplify. Disponível: <https://aws.amazon.com/amplify/>

Glossário

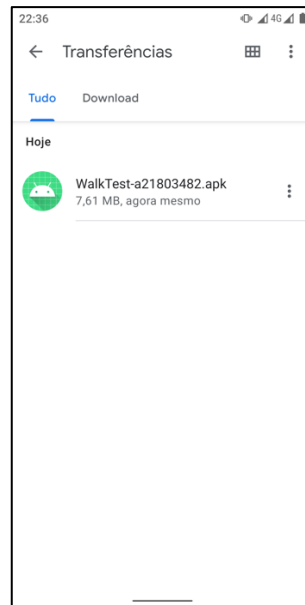
LEI	Licenciatura em Engenharia Informática
TFC	Trabalho Final de Curso
DT	Drive Test
MDT	Minimization of Drive Test
SaS	Software as a service
MVP	Minimum Viable Product
SoC	System on a Chip
RAN	Radio Access Network
GPS	Global Positioning System
AWS	Amazon Web Services
RF	Radio Frequency

9 Anexos

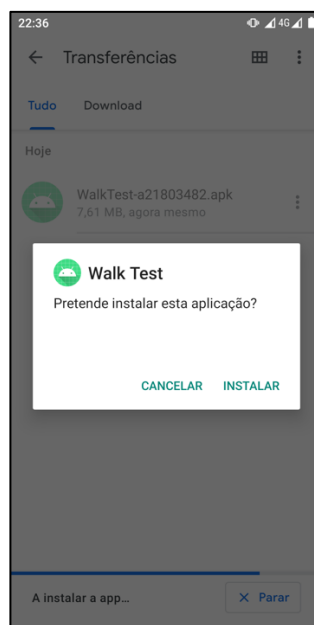
9.1 Guião de instalação

Apresenta-se o guião de instalação para a solução apresentada neste projeto.

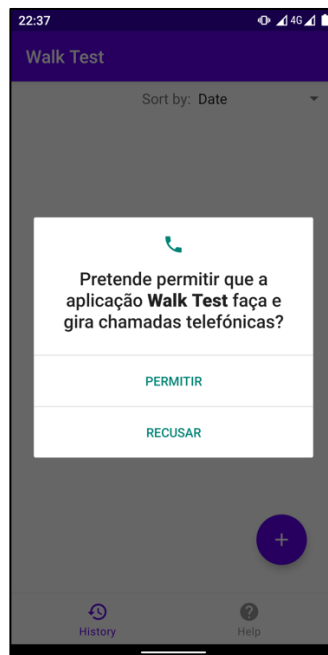
1. Após descarregar o APK do Moodle, este vai ficar disponível da pasta das Transferências (se descarregado diretamente da Web no Android). Caso contrário, o ficheiro tem que ser movido para a memória do dispositivo. Como demonstrado na figura seguinte.



2. Quando encontrarmos o ficheiro APK pretendido, selecionamos e o Android vai perguntar se pretendemos instalar. Demonstrado na figura seguinte.



3. A instalação quando concluída com sucesso deixa a aplicação disponível, junto das outras instaladas. O Walk Test está agora instalado e pode ser utilizado quando pretendido. Na 1.ª utilização, é necessário autorizar as permissões do Telefone e Localização. No caso da localização, esta tem de ser aceite SEMPRE para que a mesma possa continuar a sessão em progresso mesmo quando estamos noutra aplicação ou com o ecrã desligado.



4. A aplicação está pronta para efetuar a 1.ª de muitas sessões de capturas.

