INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

SuusMDM: Gestão de Parques Informáticos de Terminais Móveis

Roberto Leal Jacinto

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e de Computadores

Júri

Presidente:	Prof. Pedro Manuel Moreira Vaz Antunes de Sousa
Orientador:	Prof. Paulo Jorge Pires Ferreira
Co-Orientador:	Prof. João Coelho Garcia
Vogal:	Prof. Artur Miguel do Amaral Arsénio

Novembro 2012

Agradecimentos

À minha mãe, pela orientação desde sempre. Ao meu pai, por me ensinar tanto sem o dizer.
Aos dois, por estarem sempre lá.

Ao Prof. Dr. Paulo Jorge Pires Ferreira sem o qual não teria conseguido.

A toda a equipa da Caixa Mágica pelo apoio.

Ao companheiro Diogo F. Cunha cuja curiosidade resulta, não raramente, em novos desafios.

Resumo

A massificação dos dispositivos móveis em todo o mundo faz com que sejam hoje encarados como importantes ferramentas de trabalho, devendo ser geridos como os restantes equipamentos presentes no parque informático.

A identificação das suas características, o controlo do software neles instalado e a atualização deste quando necessário, apresentam-se como atividades padrão a realizar.

Esta gestão apresenta, no entanto, algumas especificidades devido à natureza dos terminais móveis. Questões como a não garantia de conectividade permanente ou a grande heterogeneidade de sistemas sem base comum, requerem uma abordagem não garantida pelas atuais soluções, tipicamente direcionadas a computadores pessoais. O objetivo deste trabalho consiste na criação de uma *framework* de gestão de terminais móveis que facilite a integração e manutenção destes dispositivos de forma remota e centralizada, requerendo o mínimo de intervenção do utilizador.

Para demonstrar o impacto possível deste novo paradigma de computação, foi realizado um protótipo que permite associar perfis, compostos por várias aplicações, a utilizadores. É posteriormente possível em qualquer terminal associado ao sistema, iniciar uma sessão ficando as aplicações automaticamente disponíveis ao utilizador. Quando a sessão é terminada, estas serão removidas, permitindo a reutilização de terminais por diversos utilizadores.

Abstract

The massification of mobile devices makes them important work tools in today business world. As such, they must be managed like any other computer infrastructure equipment.

The identification of their characteristics, control the installed software and update it when they're necessary, are some of the management activities expected.

However, due to the nature of mobile devices, there are some distinct points in this management activity. Issues such as connectivity not being permanently guaranteed or highly heterogeneous systems with no common basis, require a different approach not provided by current solutions, typically targeted at personal computers. The objective of this work is the creation of a framework for mobile device management that facilitates the integration and maintenance of these devices remotely and in a centralized fashion, requiring minimal user intervention.

To demonstrate the possible impact of this new computing paradigm, a prototype was made to assign profiles, consisting of several applications, to users. It is then possible, in any mobile device associated with the system, to login and applications will become automatically available to the user. When the session is terminated, those applications are removed, allowing the reuse of the device by several users.

Palavras Chave
Keywords

Palavras Chave

Dispositivos Móveis

Gestão de Dispositivos Móveis

Parque Informático

Inventário

Distribuição

Contexto Pessoal

Keywords

Mobile Devices

Mobile Device Management

IT Infrastructure

Asset Management

Distribution

Personal Context

Índice

1 Introdução **17**
1.1 Objetivo . . . 19
1.2 Estrutura . . . 20

2 Trabalho Relacionado **21**
2.1 Conceitos . . . 21
2.1.1 Terminais Móveis . . . 21
2.1.2 *Mobile Device Management* - MDM . . . 22
2.1.3 Tecnologia *Push & Pull* . . . 22
2.1.4 Padrão *Publish/Subscribe* . . . 23
2.1.5 Single-Writer Multiple-Readers . . . 24
2.2 Sistemas de Gestão de Parques Informáticos Clássicos . . . 25
2.3 Sistemas MDM de Fabricantes dos Terminais . . . 26
2.3.1 BlackBerry® Enterprise Server . . . 26
2.3.2 Microsoft® System Center Mobile Device Manager . . . 27
2.3.3 Android® Market Webstore . . . 28
2.3.4 Apple® Enterprise Features . . . 29
2.4 Outras Soluções de MDM . . . 30
2.4.1 OMA Device Management . . . 30
2.4.2 Funambol Device Management . . . 31
2.4.3 Zenprise MDM . . . 31
2.4.4 MobileIron MDM . . . 32
2.5 Sumário . . . 33

3 Arquitetura **35**
3.1 Arquitetura Base . . . 35
3.2 Modelo de Domínio . . . 37
3.3 Casos de Uso . . . 38
3.3.1 Registo do Terminal Móvel . . . 38
3.3.2 Gestão de Perfis . . . 39
3.3.3 Início de Sessão no Terminal Móvel . . . 40
3.3.4 Atualizações Iniciadas pelo Sistema de Gestão . . . 41
3.3.5 Fim de Sessão no Terminal Móvel . . . 43
3.4 Terminais Não Alcançáveis . . . 43
3.5 Segurança e Comunicação . . . 44
3.6 Identificador Único . . . 46

4 Implementação **49**
4.1 Funcionalidades . . . 49
4.2 Cliente . . . 49
4.2.1 Instalação/Remoção silenciosa . . . 50
4.2.2 Mecanismo *Ring-home* . . . 51
4.2.3 Paradigma *Push* . . . 51
4.3 Servidor . . . 52
4.3.1 Mecanismos de Base de Dados . . . 53
4.3.2 *Deploy* do Sistema de Gestão . . . 53
4.3.3 Segurança . . . 54

5 Avaliação **55**
5.1 Avaliação Qualitativa . . . 55
5.1.1 Registo Iniciado pelo Terminal . . . 55
5.1.2 Sessões & Perfis . . . 56
5.1.3 Paradigma *Push* . . . 56
5.1.4 Testes com Utilizadores . . . 57
5.2 Avaliação Quantitativa . . . 59
5.2.1 Contexto Avaliativo . . . 59
5.2.2 Testes de Carga - Número de Terminais . . . 60
5.2.3 Testes de Carga - Recursos Móveis . . . 61
5.2.4 Análise de Tráfego Gerado . . . 63
5.2.5 Testes de Segurança . . . 64

6 Conclusão **67**
6.1 Trabalho Futuro . . . 68

A Estrutura da Base de Dados Implementada **71**

B Questionário de Testes com Utilizadores **73**

C Fluxo de Execução de Testes com Utilizadores **77**

D Resultados dos Testes com Utilizadores **79**

Lista de Figuras

1.1 Tipo de terminais e seus utilizadores ao longo do tempo 18

2.1 Modelos *Pull* e *Push* 23
2.2 Modelo Publish/Subscribe simplificado 24

3.1 Fluxograma de vida da solução 35
3.2 Arquitetura base da solução 36
3.3 Modelo de domínio simplificado 37
3.4 Registo de terminal móvel 39
3.5 Início de sessão no terminal móvel após registo 40
3.6 Envio de aplicações por iniciativa do sistema de gestão 42
3.7 Estratégia de comunicação tolerante a falhas 44

4.1 Ecrã principal e escolha de perfil - componente cliente do protótipo 50
4.2 Ecrã principal do sistema de gestão - componente servidor do protótipo 53
4.3 Ecrã principal do sistema de gestão - componente servidor do protótipo 54

5.1 Utilização de processador e memória no sistema de gestão com o aumento do número de terminais 60
5.2 Utilização de cpu e ram dos terminais móveis 62
5.3 Utilização de bateria dos terminais móveis integrados na solução 62
5.4 Captura de informação trocada no processo de início de registo de terminal e início de sessão 65

Lista de Tabelas

2.1 Comparação entre as soluções apresentadas 33
2.2 Soluções existentes e requisitos respeitados 34

3.1 Mecanismos de segurança implementados e mais valias atingidas 46

5.1 Tráfego gerado pelo SuusMDM e terminal móvel 63
5.2 Informação dos terminais utilizados para o seu registo 64

A.1 Aplicação 71
A.2 Terminal 71
A.3 Utilizador 72
A.4 Perfil 72
A.5 *Tokens* Temporários 72
A.6 Sessões Ativas 72
A.7 Log de acessos 72

Capítulo 1

Introdução

A massificação do acesso à rede global, a Internet, permite uma troca de informação de forma fácil e rápida derrubando barreiras de localização física, sendo desejável estar acessível o maior tempo possível. Mais ainda, cenários onde seja possível estar *on-line* em qualquer altura e lugar são apetecíveis oferecendo mais valias competitivas.

O aparecimento de computadores portáteis cada vez mais leves e com cada vez maior autonomia permitem uma mobilidade cada vez maior. No entanto, durante grande parte do início deste século, continuavam a não existir, de forma disseminada, infraestruturas que permitissem o acesso à rede global de forma contínua, rentável e com largura de banda suficiente para uma boa experiência de utilização. A completa realização deste cenário só foi possível no final da primeira década deste século com os avanços nas tecnologias sem fios e com a diminuição dos custos monetários para a implementação e utilização das mesmas[15, 17].

Hoje em dia é fácil encontrar pontos de acesso para redes sem fios espalhados em qualquer centro urbano - dentro e fora dos edifícios. Nos últimos anos houve ainda uma massificação de tecnologias de dados móveis, como o EDGE e mais recentemente HSPA (HSPA+ e HSUPA)[21, 15, 25] que possibilitam largura de banda suficiente para o cada vez maior fluxo de informação possível de ser acedido.

Este novo paradigma de acesso móvel à informação abriu portas a um novo tipo de dispositivos. O aumento do poder de processamento foi de tal forma acentuado e apoiado pelos avanços no tamanho de fabricação de componentes que é hoje fácil encontrar terminais com o mesmo, ou maior, poder que alguns computadores de secretária de apenas alguns anos atrás. Estes terminais móveis, muito compactos e focados na disponibilização de informação em movimento foram apelidados de dispositivos de Internet móvel.[1] Nele podemos englobar os *smartphones*, os *tablets*, e até leitores de música e consolas de jogos portáteis[5]. Devido ao seu tamanho reduzido, poder de processamento adequado a tarefas de acesso de informação, grande autonomia e cada vez menor custo, estes dispositivos móveis inundaram o mercado mundial de tecnologia, estando já a ultrapassar, no número de vendas, terminais clássicos como o estabelecido computador de secretária[17].

Outro fator que impulsionou este tipo de terminais e a sua utilização, foi a noção de informação na nuvem[2, 4]. Este paradigma tecnológico assenta na noção de que toda a informação é guardada em servidores remotos e acedida numa base de necessidade. Da mesma forma, qualquer tipo de necessidade de processamento pode ser executado a partir de um terminal com recursos medianos, sendo todo o trabalho de processamento intenso realizado remotamente quando tal for necessário. Este cenário permite-nos dispor de terminais mais compactos uma vez que os requisitos de processamento e armazenamento são passados para segundo plano.

[1] Ao longo do artigo, estes terminais serão referidos apenas como terminais móveis

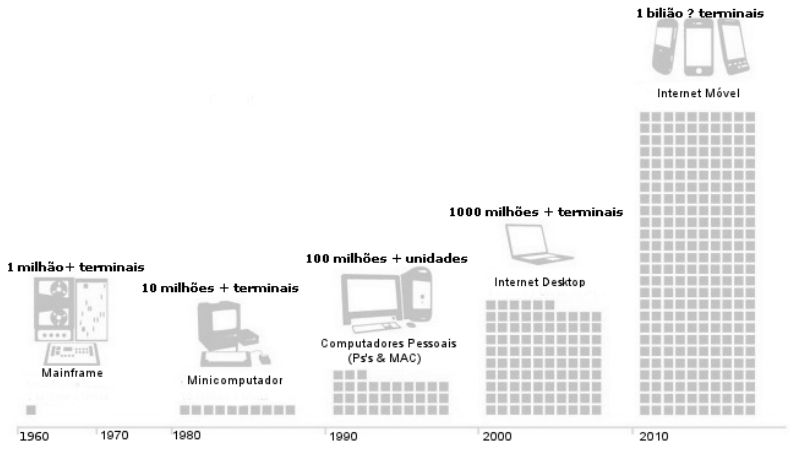


Figure 1.1: Tipo de terminais e seus utilizadores ao longo do tempo[17]

Com as vantagens referidas e preços cada vez mais acessíveis, os terminais móveis rapidamente viram o seu valor aproveitado. Mais do que um forte ícone da revolução tecnológica dos nossos tempos[17] são agora encarados como importantes ferramentas de trabalho[14, 26].

No cenário das organizações, a adoção de tecnologias de informação como ferramentas de trabalho essenciais é recente, tendo levado nos últimos anos, a um novo paradigma associado à estrutura interna das mesmas[18, 24]. A rapidez no acesso e troca de informação que se apresenta cada vez mais dispersa, fruto da globalização e internacionalização do mercado de trabalho[18] foi um fator decisivo para que dispositivos como o computador pessoal se afirmassem como indispensáveis. Da mesma forma, as atuais necessidades competitivas do mercado empresarial empurram-nos para cenários onde é necessário aceder a informação em qualquer lado, em qualquer altura e em movimento, tornando os terminais móveis uma mais valia apetecível no contexto empresarial. A sua entrada neste contexto, ainda que de grande valor, apresenta alguns desafios.

Um parque informático, deve ter mecanismos para a sua gestão. Controlar o número e as especificidades técnicas de aparelhos que dele fazem parte, assim como as suas condições de funcionamento - por exemplo avarias a nível de *hardware* - são ações típicas de gestão inventario. Sendo terminais informáticos, a gestão do software neles existente - versão do sistema operativo, versão das aplicações instaladas e a sua manutenção como atualizações, remoção e instalação dos mesmos - é também uma atividade necessária de forma a garantir o controlo e desempenho dos dispositivos enquanto ferramentas de trabalho.

De seguida, apresenta-se um exemplo destes novos cenários de parque tecnológico, constituídos também por terminais móveis.

> *O hotel Luxo&Viagem compete no negócio de hotelaria de luxo, estando sempre à procura de inovações que lhes permitam cativar novos clientes.*
>
> *A sua última novidade, consiste na oferta a cada hóspede de um tablet de forma a criar um novo canal de comunicação com o cliente, pessoal e direcionado aos seus interesses. Este tablet, que é fornecido aos hóspedes durante a sua estadia, inclui diversas aplicações que proporcionam informação ao utilizador do seu interesse - como pontos de turismo, restaurantes, farmácias, etc - bem como aplicações que lhe permitem interagir com o ambiente à sua volta - marcar excursões ou serviços do próprio hotel, enviar emails, requisitar transportes, etc.*
>
> *Ao número de terminais disponibilizados aos hóspedes, deve ainda acrescer os utilizados pelos próprios funcionários do hotel. A cada empregado é também fornecido um tablet como ferramenta de trabalho, auxiliando-o no desempenho das suas funções. Desta forma, podem trocar*

informações e manterem-se a par do que se passa nas instalações e com os visitantes, resultando numa rápida resposta às mais variadas situações.

Devido à dimensão do próprio hotel, o número de tablets existentes e em funcionamento num dado instante é considerável - mais de 300 em qualquer momento. Em todos eles, são regularmente instaladas e removidas aplicações de forma a disponibilizar ao seu utilizador um contexto mais personalizado.

No caso dos hóspedes, aplicações direcionadas aos interesses por eles expressados são automaticamente adicionadas, sem a intervenção do utilizador. Um hóspede que frequentemente consuma comida vegetariana, pode ver instalado no seu tablet uma aplicação sobre restaurantes vegetarianos na zona, ou mesmo uma aplicação de culinária.

No cenário dos funcionários do hotel, são feitas regularmente atualizações ao software por eles utilizado, assim como remoção e adição de aplicações quando estes alteram as atividades desempenhadas - por exemplo, a passagem de um funcionário da sala restaurante para o bar da piscina, pode beneficiar de um software de previsão meteorológica ou uma aplicação com a informação sobre primeiros-socorros.

Esta oferta tem tido tanta adesão que a administração do Luxo&Viagem já se encontra em negociações para a compra de novos terminais entre vários fabricantes, de forma a associar dispositivos específicos às atividades que melhor suportam.

O exemplo em cima apresenta um cenário onde vários terminais móveis são utilizados como ferramentas de trabalho, distribuídos por diversos utilizadores com necessidades específicas. Rapidamente podemos identificar uma necessidade de gestão dessa infraestrutura de forma automática. Atividades como o envio de novas aplicações ao terminal ou atualização das mesmas são vulgares e necessitam de mecanismos de controlo automatizados, sendo incomportável as necessidade de recursos humanos caso todo o processo fosse executado por um administrador informático. Da mesma forma, o grande número de terminais assim como a sua heterogeneidade proveniente de novas plataformas e fabricantes, anuncia a necessidade de uma gestão de inventário, onde a distribuição dos dispositivos, suas condições de funcionamento e informação única sejam armazenadas e acessíveis.

1.1 Objetivo

De forma a auxiliar a administração de terminais móveis em cenários como o descrito na secção anterior, pretende-se desenvolver uma solução de gestão de terminais móveis, onde estes sejam encarados como mais uma ferramenta de trabalho, quebrando a ligação terminal/proprietário existente atualmente. Alem da gestão do software nos terminais, nomeadamente com a instalação, remoção e atualização remota do mesmo, assim como uma gestão de inventário simples - *software* e *hardware* existentes - pretende-se que o sistema possibilite que diversos utilizadores partilhem os mesmos terminais tendo no entanto as suas aplicações disponíveis - e apenas a si - com o mínimo de esforço.

Esta solução deverá culminar numa *framework* que permita gerir os vários terminais e utilizadores de forma remota e com o mínimo de necessidade de interação humana.

A solução deve respeitar os requisitos indicados em seguida:

- **Escalabilidade:** O sistema deve garantir suporte ao aumento do número de terminais sem necessidade de alterações arquiteturais.

- **Configurações automatizadas:** A maioria dos processos inerentes à gestão dos terminais deve ser feita de forma automática e com o mínimo de necessidade de interação humana. Este requisito estende-se ao próprio processo de registo dos terminais móveis no sistema de gestão.

- **Minimizar a ligação à rede:** Devido aos custos económicos associados às tecnologias de ligações móveis, assim como às suas limitações a nível de disponibilidade, as necessidades de ligação à rede nos terminais móveis devem ser minimizadas. No melhor caso, apenas deve existir ligação à rede para a troca de informação entre o terminal e o sistema de gestão quando tal for estritamente necessário, quer por um pedido feito diretamente pelo utilizador ou uma ação crítica. Mesmo nessas alturas - sempre que seja necessário a existência de uma ligação - devem existir mecanismos que permitam reduzir o tamanho da informação a transferir.

- **Minimizar o uso de recursos nos terminais móveis:** Recursos como a bateria são uma das grandes limitações dos terminais alvo desta solução. Devem ser encontrados mecanismos que permitam gerir os recursos disponíveis no terminal e as ações a realizar no mesmo, podendo estas serem, por exemplo, adiadas em caso de necessidade. Outros recursos a ter em conta são o poder de processamento ou a memória.

- **Suporte gracioso a terminais não alcançáveis:** Devido às características móveis dos terminais a suportar, estes podem apresentar períodos onde com eles seja impossível comunicar. A solução a desenvolver deve ter mecanismos que suportem estes cenários mais ou menos temporários, garantindo a passagem da informação aos terminais quando estes voltarem a estar *on-line*.

- **Suporte abrangente de terminais e sistemas:** Devido à heterogeneidade atual no contexto dos terminais móveis, a solução desenvolvida deve assentar em *standards* e tecnologias fortemente disseminadas. No limite, qualquer terminal móvel deve poder ser gerido pelo sistema resultante desta solução, sem nenhuma alteração às características de ambos - terminal e solução - ou, no limite, com o mínimo de esforço de desenvolvimento.

- **Segurança:** A informação trocada entre os terminais móveis e a *framework* de gestão deve ser acessível apenas entre cada par terminal móvel/*framework*. Concretamente, devem ser garantidas características como a integridade, autenticidade e confidencialidade ponto a ponto.

1.2 Estrutura

O documento segue, após este capítulo introdutório, a seguinte estrutura: no Capítulo 2 começamos por apresentar o trabalho desenvolvido na área da computação móvel, dando a conhecer na secção 2.1 alguns conceitos base que serviram de referência ao trabalho desenvolvido e apresentando de seguida soluções relevantes.
O Capítulo 3 apresenta a nossa proposta de arquitetura para a solução da problemática sendo de seguida, no Capítulo 4, é apresentada a implementação do protótipo SuusMDM desenvolvido a fim de comprovar a análise realizada. No Capítulo 5 é apresentada a avaliação realizada ao sistema desenvolvido bem como os resultados obtidos. Por fim, no Capítulo 6, são apresentadas conclusões sobre o nosso trabalho e o paradigma móvel atual sendo discutidas ideias de melhorias para a solução, a realizar em trabalho futuro.

Capítulo 2

Trabalho Relacionado

Neste capítulo vamos dar a conhecer o trabalho realizado na área da computação móvel, em especifico no trabalho orientado à gestão de terminais móveis como *smartphones* e *tablets* que formam a base de interesse do trabalho proposto neste documento. Alguns conceitos relevantes serão abordados de início, sendo depois apresentadas e analisadas algumas das soluções existentes com maior destaque.

Este capítulo segue a seguinte estrutura:
Na secção 2.1, são apresentados os conceitos de relevo para o estudo deste documento.
Na secção 2.2, descrevemos alguns sistemas de gestão de parques informáticos *clássicos* e as suposições nas quais assentam.
Na secção 2.3 serão introduzidas soluções de MDM corporativas e orientadas a terminais com sistemas específicos.
Na secção 2.4 discutimos novas soluções de MDM com um suporte muito mais abrangente de dispositivos.
Por fim, na secção 2.5 é feito um pequeno resumo do que foi apresentado nas secções anteriores e é feita uma pequena análise à luz dos objetivos propostos neste documento.

2.1 Conceitos

Nesta secção apresentamos alguns conceitos e noções tecnológicas presentes nas soluções atuais e que oferecem grande valor em cenários de computação móvel, sendo considerados fundamentais na área por nós explorada.

2.1.1 Terminais Móveis

De forma a melhor enquadrar o tema desenvolvido neste documento, importa definir desde já o que é um *terminal móvel*.

Este é um conceito amplo que numa visão generalista engloba dois significados. O primeiro, prende-se com a possibilidade de mobilidade física do dispositivo. Nesta definição podemos contemplar vários terminais. Se é fácil indicar como terminal móvel, associado ao contexto de mobilidade física, um *smartphone* e até um *laptop*, o mesmo não pode ser dito de um servidor *mainframe*. O segundo significado prende-se com o acesso à informação a partir do dispositivo. Este remete para as características de comunicação do dispositivo no que toca a ligação à Internet. O terminal pode apenas apresentar ligação quando se encontra fisicamente estacionário, embora se possa deslocar perdendo conectividade nesses momentos, ou pode garantir conectividade durante a sua deslocação[19], normalmente apelidados de terminais de Internet móvel.

No âmbito deste documento, o foco principal serão todos os terminais que englobam o segundo significado, permitindo comunicação durante a sua deslocação física. Estes fazem já parte do quotidiano na forma de

telefones com acesso a comunicação de dados móveis como *smartphones* ou como *PDAs* e mais recentemente os *tablets*, sendo utilizados de inúmeras formas onde a interação sem barreiras de mobilidade seja desejável.

2.1.2 *Mobile Device Management* - MDM

Com o interesse crescente existente pelos terminais móveis, em especial dentro do meio empresarial, começam a surgir vários desafios no que toca à gestão dos mesmos.

Ultrapassando questões como o inventário associado ao valor financeiro dos terminais e o seu peso dentro da empresa, processos ligados à informação transmitida pelos terminais, a sua estabilidade e a manutenção de uma ligação lógica forte entre o dispositivo e a empresa começam a ganhar relevo.

O conceito de *mobile device management* - MDM - expande o conceito original de sistema de gestão, orientando-o a um paradigma móvel. Na base, MDM é um sistema orientado à gestão de terminais móveis que ofereça as seguintes funcionalidades:[1]

- **Distribuição de *Software* e/ou *firmware*:** Capacidade de gerir as aplicações para terminais móveis, quer do ponto de vista de inventário, quer implementando ações como a instalação, remoção, atualização ou proibição de execução das mesmas, diretamente no terminal. As mesmas funcionalidades podem existir para o próprio *firmware* do terminal.
- **Configurações:** Desenvolvimento, controlo e aplicação de políticas de empresariais no terminal, bem como de configurações de acesso a serviços internos da empresa.
- **Gestão de Inventário:** Alem da gestão básica de inventário, capacidade de ações como *aprovisionamento* e suporte a avarias através de mecanismos mais ou menos elaborados de diagnóstico.
-

2.1.3 Tecnologia *Push & Pull*

Uma das grandes diferenças na comunicação com terminais móveis, prende-se com a propriedade de conectividade aos mesmos. Num parque informático clássico, por exemplo, a maioria dos terminais a gerir encontram-se ligados em rede de forma estacionária sendo os seus endereços conhecidos e quase sempre imutáveis. O mesmo não se pode dizer dos terminais móveis.

Num sistema de gestão de parque informático a maioria da comunicação deve partir da *framework* que dá base ao sistema de gestão e raramente dos terminais geridos[24, 19]. De facto, uma abordagem onde o cliente pede repetidamente ao servidor informação sobre atualizações ou ações a efetuar, poderia sobrecarregar a infraestrutura de gestão e representava uma sobre-utilização da ligação à rede bem como de outros recursos, por exemplo poder e tempo de processamento. Este recurso é ainda mais importante se pensarmos em terminais móveis uma vez que por norma não apresentam grande poder de processamento e a sua utilização está diretamente ligada ao consumo de energia utilizada - tipicamente uma bateria com uma capacidade nunca desprezável contemplando as necessidades de mobilidade.

Um modelo desta natureza, onde o inicio da comunicação é inicializado pelo terminal de funções maioritariamente designadas de servidor é conhecido como modelo *push*. Este modelo apresenta vantagens face a interações do tipo pedido-resposta - também conhecido como modelo *pull*. Embora ambos sejam variações do modelo cliente/servidor, existe uma diferença fundamental: quem inicializa a comunicação[12].

Na figura 2.1, em cima, apresentam-se de forma simples a inicialização do fluxo de informação em cenários *pull* - à esquerda - e *push* - à direita. Embora a informação flua sempre no mesmo sentido, num sistema tipo

[1] De notar que um sistema de MDM completo deve garantir a totalidade das funcionalidades descritas.

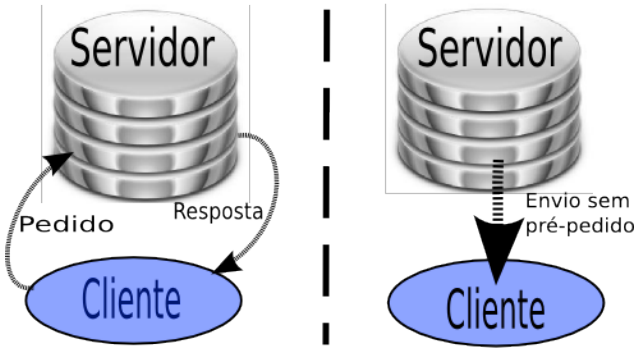


Figure 2.1: Modelos *Pull* - à esquerda - e *Push* - à direita

push, a informação é enviada diretamente sem um pedido prévio, enquanto no modelo *pull* será necessário esse primeiro contacto.

Observando cenários de grande mobilidade, esta abordagem - *push* - apresenta-se vantajosa uma vez que permite minimizar a ligação à rede por parte dos terminais móveis, assim como também permite poupar recursos como o tempo de processamento - que em terminais móveis está diretamente ligado a outros recursos como a bateria. Ambas as vantagens indicadas contemplam requisitos do sistema a desenvolver.

2.1.4 Padrão *Publish/Subscribe*

Se optarmos pelo modelo descrito em cima (modelo *push*), surge-nos outra necessidade a colmatar: se é o servidor que começa a ligação, como sabe ele quem deseja ser contactado e em que casos?

Para dar resposta a esta pergunta, podemos analisar um padrão de desenho de comunicação denominado *padrão Publish/Subscribe*. Este tem o objetivo de assistir na troca de informação entre quem a produz e quem a consome[13]. Neste paradigma de comunicação, quem quer aceder a informação, os consumidores (*subscribers*), expressa o seu interesse num tópico particular, ou num conjunto de tópicos e é notificado sempre que um novo evento atrativo é gerado (pelo(s) *publisher(s)*)[10]. Este tipo de sistema utiliza vantajosamente um modelo de troca de informação do tipo *push*, uma vez que a informação é recebida pelos clientes sem pedido constante dos mesmos. Eles apenas têm um contacto inicial por forma a demonstrarem o(s) seu(s) interesse(s), ou possíveis contactos futuros de forma a alterarem os mesmos.

Com um pouco de abstração, podemos apresentar a *framework* de gestão como o único *publisher* e os vários terminais móveis como *subscribers*. Estes associam-se e identificam todos o mesmo tópico de interesse, ou interesses do subgrupo a que pertencem. Desta forma, as possíveis ordens do sistema de gestão são passadas a todos os terminais, ou subgrupos de terminais, de maneira direcionada e sem problemas de sincronização. Neste cenário, o(s) interesse(s) podem ser introduzidos pelo utilizador do terminal ou serem pré-definidos para um determinado conjunto destes. Exemplos de interesses baseados no modelo do hotel apresentado na introdução deste artigo podem ser, por exemplo *Atualizar aplicação de ementa para clientes do hotel*, ou *Atualizar a aplicação de gestão de parque automóvel para funcionários*.

Devemos no entanto ter em atenção algumas características particulares dos terminais móveis. Como já foi referido, estes deslocam-se frequentemente tendo vários endereços de acesso durante o tempo ou até nenhum. Esta característica dificulta a aplicação deste modelo uma vez que o cliente pode apresentar

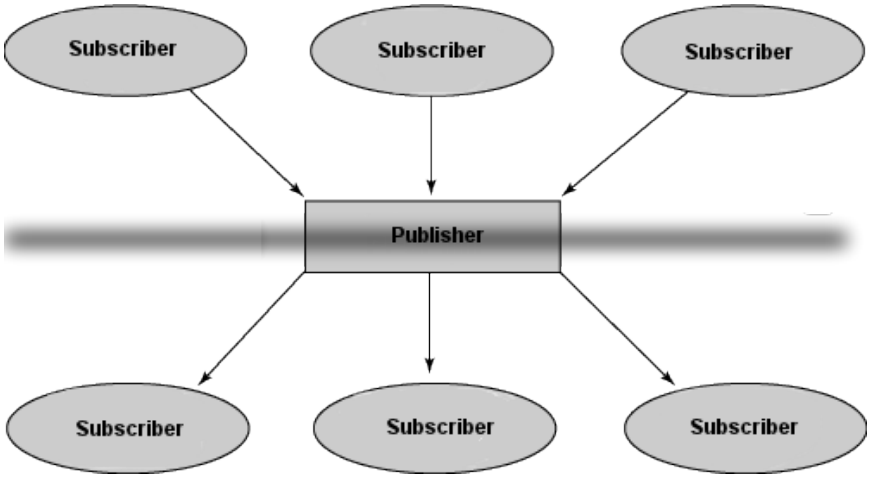


Figure 2.2: Modelo Publish/Subscribe simplificado

endereço desconhecido - ou mais uma vez, não estar contactável de todo - e é da responsabilidade do servidor de gestão iniciar o diálogo. Mais ainda, devemos ter em conta que os terminais móveis, mantêm acesso à rede em movimento o que significa que a descoberta do endereço de um terminal pode não garantir um canal de comunicação durante o tempo expectável.

2.1.5 Single-Writer Multiple-Readers

Um aspeto a ter em conta sempre que um sistema disponibiliza informação alterável ao longo do tempo, prende-se com a concorrência e consistência do acesso à mesma. As entidades que alteram a informação não o devem fazer em simultâneo, existindo o risco de informação corrompida - problemas de concorrência. Já a leitura da informação por diversas entidades pode apresenta problemas de consistência se uma leitura estiver a ser feita durante uma escrita à mesma informação.

Num sistema de gestão como os envolvidos neste documento, a mesma ideia é aplicada. Neles, podemos alocar os terminais móveis a gerir como as entidades que apenas fazem leituras da informação - *readers* - recebendo-a da *framework de gestão*. Esta sim, será a única a poder adicionar ou alterar a informação a transmitir - *writer*. Este modelo designa-se por *Single-Writer Multiple-Readers* uma vez que apenas existe uma entidade que altera informação e várias que a ela acedem[3, 23].

Embora uma primeira análise pareça descartar quaisquer problemas de concorrência num modelo deste tipo, tal não se verifica. Em concreto, podem existir problemas de acesso a informação desatualizada caso um dos *readers* aceda a um objeto de informação enquanto o *writer* único o atualiza ou remove. Tais cenários podem ser resolvidos com a ajuda de *time-stamps* nas transações ou mecanismos de troca de controlo de ação onde é a entidade com poderes para realizar alterações na informação - *writer* - a mesma que disponibiliza o poder de leitura a terceiros[3, 23].[2]

[2]http://www.itfinancialnews.com/invivoo-finance.do?action=viewarticle&menu=FINANCE&article=466

2.2 Sistemas de Gestão de Parques Informáticos Clássicos

A gestão de parques informáticos não é exclusiva ou direcionada de raiz a terminais móveis. De facto, existem diversas soluções para a gestão de parques constituídos por terminais não móveis - computadores pessoais *desktops*, portáteis e até servidores. Estas soluções não cumprem, no entanto, requisitos essenciais num paradigma de terminais móveis. Os dispositivos para os quais são direcionados apresentam endereços de rede conhecidos pelo que não existe possibilidade de não acesso.[3] Mais ainda, os dispositivos a gerir apresentam quase sempre um poder de processamento elevado e não apresentam tipicamente restrições como a necessidade de conservação de recursos como alimentação.

As soluções existentes podem ser divididas em dois grupos que caracterizam os seus objetivos principais: inventariação e gestão dos ativos.

Sobre soluções orientadas à inventariação, um dos projetos com maior sucesso é o *Open Computer and Software Inventory Next Generation*[4] (OCS-NG). Este projeto começou com o objetivo de melhorar a eficiência do processo de levantamento dos dispositivos que constituem um parque informático, permitindo realizar a tarefa da forma mais rápida e automática possível. Embora tenha sido este o objetivo proposto no inicio do projeto em 2001, o seu contínuo desenvolvimento permite atualmente muitas outras funcionalidades como a instalação de software de forma remota. Outra característica interessante desta solução é a sua natureza *open-source*, estando todo o código acessível e disponível para alteração.

Ainda assim, esta solução não pode ser diretamente implementada em cenários de terminais móveis como os apresentados até aqui. As suas funcionalidades de inventariação são bastante complexas, tornando-o demasiado abrangente para os terminais objetivo. Mais ainda, de forma a suportar todas essas funcionalidades, o desenvolvimento técnico da solução teve de ser suportado por tecnologias orientadas aos terminais onde será executado. Embora exista um grande suporte de ambientes de execução, novos ambientes não suportados necessitam de um novo desenvolvimento que acarreta alguns custos. Em todo o caso, compreende muitos dos requisitos desejados nos cenários alvo deste artigo como uma utilização reduzida de acesso à rede ou grande escalabilidade, permitindo a gestão de cerca de 1000000 máquinas com um sistema central de recursos medianos.

Um outro exemplo ainda no contexto da inventariação é o *Fusion Inventory*.[5] Este projeto que se apresenta como um *fork* do OCS-NG, tenta dar resposta a alguns dos problemas referidos anteriormente. Centrando-se apenas no cliente instalado nos terminais, não tendo qualquer objetivo de desenvolvimento a nível do servidor de gestão. As tecnologias utilizadas para o seu desenvolvimento foram o principal tópico tendo o objetivo de maximizar o número de sistemas suportados. Aquando do fim da escrita deste artigo, era apresentada uma versão orientada a terminais Android®.[6] Esta versão cliente foi desenvolvida como uma aplicação nativa de plataforma Android® utilizando no entanto bibliotecas previamente desenvolvidas e comuns às versões das restantes plataformas. A informação recolhida abrange o levantamento de características de *hardware* e *software* dos terminais, como o tamanho e resolução do ecrã, memória interna do mesmo e a versão do sistema Android® instalado, mas também de características especificas de terminais móveis como a presença de um cartão SIM, a operadora de dados móveis em que se encontra registado ou mesmo informação sobre localização física do terminal baseada em dados provenientes do GPS, caso este exista.[7]

No que diz respeito a soluções de gestão dos ativos, a sua atividade centra-se numa gestão mais orientada ao próprio terminal. Tarefas como a gestão do *software* do mesmo - atualização, instalação e remoção

[3] Não contemplando casos de falha de rede ou outros cenários erróneos
[4] http://www.ocsinventory-ng.org/
[5] http://fusioninventory.org/
[6] http://fusioninventory.org/wordpress/2011/09/13/fusioninventory-1-0-for-android-released/
[7] http://forge.fusioninventory.org/projects/android/wiki/Android_Specifications/

- backup remoto de informação e até mecanismos de agendamento de associação entre dispositivos e utilizadores. Devemos notar que a atividade de controlo de inventário também está presente nestas soluções mas muitas vezes de forma autónoma e modular, utilizando soluções como as descritas anteriormente - OCS-NG por exemplo - e integrando-as. Nestas soluções é utilizada a tecnologia *push* de forma a minimizar o uso da rede, bem como a utilização de padrões modificados do *publish/subscribe* quando se deseja criar subgrupos dentro de um mesmo parque. É ainda usual encontrar mecanismos de segurança associados à comunicação presente nestes sistemas de forma a garantir a confidencialidade da informação trocada assim como requisitos de autenticidade.

Um exemplo de grande sucesso é o projeto *Gesttion Libre de Parc Informatic* (GLPI),[8] que é distribuído como uma solução livre, de código aberto. Este apresenta-se como um *fork* do projeto Information Resource Manager (IRM), oferecendo a possibilidade de uma gestão bastante completa do hardware que constitui a infraestrutura do parque informático, utilizando como base para o processo de inventariação diversas soluções, em especial o OCS-NG. Toda a informação quer dos terminais, quer dos componentes que os constituem é gerida de forma global, permitindo várias interações sobre as mesmas, como consulta de detalhes, datas de interesse, últimas revisões de funcionalidade, entre outros.[9] Este software de gestão engloba ainda ferramentas de *tracking*, dando a possibilidade aos próprios utilizadores de requererem informação ou indicarem problemas sobre todos os aspetos do seu computador - e outros componentes.[10] Podem inclusive ser feitas reservas de recursos, sendo o processo de alocação feito com base numa visão global de toda a infraestrutura.

2.3 Sistemas MDM de Fabricantes dos Terminais

Nesta secção vão ser apresentados alguns dos sistemas MDM oferecidos pelos próprios fabricantes de terminais móveis e que são exclusivamente direcionados aos seus dispositivos. Não se pretende apresentar aqui a totalidade de soluções existentes, mas sim aquelas cuja utilização é relevante, quer pela utilização em larga escala, quer pela ligação forte aos próprios terminais. Devido à sua quota de mercado, a nível empresarial ou pessoal, foram escolhidos os terminais das marcas Blackberry®, Microsoft®, Android® e Apple®.

A partir da subsecção 2.4 apresentam-se outros sistemas de MDM sem ligação direta a um fabricante ou sistema.

2.3.1 BlackBerry® Enterprise Server

Um dos exemplos mais conhecidos do conceito de parque informático em dispositivos móveis, é o dos terminais Blackberry® da empresa Research in Motion® (RIM). A RIM começou a comercializar terminais Blackberry® em 1999 com o modelo 850. Este estava ainda longe dos atuais *smartphones* mas apresentava já algumas das principais características que estão associadas aos modelos desta marca: teclado *qwerty* completo e a capacidade de interação com servidores de *email*. De facto, estas duas são ainda hoje as principais propriedades de destaque destes terminais, muito devido à qualidade e preocupação com a segurança de que são alvo.

Atualmente, os terminais Blackberry® são importantes ferramentas de trabalho, tendo uma grande aceitação no mundo empresarial[1], permitindo aos seus utilizadores um contacto direto e permanente com a instituição da qual fazem parte, quer no acesso ao seu *email*, quer ao acesso de informação muitas vezes partilhada por canais seguros como redes privadas virtuais - Virtual Private Network (VPN).

Devido ao mercado alvo destes dispositivos e das necessidades de controlo impostas pelas organizações que os utilizam, foi desenvolvido software para a gestão e integração dos mesmos nos parques informáticos.

[8] http://www.glpi-project.org/
[9] http://www.glpi-project.org/wiki/doku.php
[10] http://www.glpi-project.org/wiki/doku.php

O Blackberry® Enterprise Server (BES)[11], como foi apelidado pela RIM®, é um software de gestão centralizado que disponibiliza, entre outras, funções de produtividade - acesso a *email* com conexão iniciada pelo servidor, sincronização de calendário e lista de contactos, suporte a assinaturas digitais centralizadas, gestão de certificados, cifra de dados, gestão do software do terminal - instalação, remoção e atualização de software remotamente - fiscalização de políticas de segurança, migração de software entre terminais, monitorização remota do terminal, entre outros.

Apesar deste sistema, por si só, ser bastante completo, a verdade é que apresenta algumas limitações ao nível do suporte e integração com restantes dispositivos possíveis de existir num parque informático. O software de código fechado, é exclusivamente orientado aos seus terminais, proporcionando reduzidas interfaces para interação com um número restrito de software. Esta falta de abertura não é uma solução para incentivar a adoção de terminais móveis como principal ferramenta de trabalho em organizações, uma vez que restringe definitivamente os produtos utilizáveis. Se pensarmos nos custos iniciais que a implementação de um sistema de gestão deste tipo pode incorporar, este cenário de ligação exclusiva pode não ser o mais desejável. Deve no entanto ser reconhecida a vantagem competitiva que esta abordagem oferece ao seu fabricante, uma vez que garante uma base de utilizadores constante e crescente de acordo com o crescimento da própria organização. Mais ainda, é fácil verificar que esta abordagem também permite alguma simplificação do lado de quem desenvolve os sistemas, uma vez que fica todo o desenvolvimento restringido a software e hardware conhecido e bem documentado. Por outro lado, a rápida mudança de necessidades que as organizações apresentam nos dias de hoje, podem inviabilizar o uso dos terminais associados sendo desejável a sua substituição. Com um sistema deste tipo, tal é impossível, sendo necessário novos investimentos para a adoção de novas soluções.

Podemos encontrar alguma informação sobre as tecnologias utilizadas pela RIM® no seu software. Embora tenham sido apontadas algumas falhas que permitam a total resposta ao problema analisado neste artigo, nomeadamente o uso exclusivo a terminais BlackBerry®, a verdade é que muitos dos problemas base da gestão de terminais móveis são transversais a todas as soluções propostas ou existentes. Podemos observar, por exemplo, o uso da tecnologia *push* para a troca de informação entre o sistema de gestão e os terminais. Desta forma, requisitos de controlo do uso de recursos são cumpridos, uma vez que o terminal não necessita de questionar periodicamente o sistema de gestão sobre nova informação. É ainda evidente o uso de mecanismos de segurança em respeito à troca de informação, sendo esta totalmente cifrada por omissão.

2.3.2 Microsoft® System Center Mobile Device Manager

Outro exemplo que apresenta melhor integração, embora a um custo, é o Microsoft® System Center Mobile Device Manager. Esta *framework* de gestão da Microsoft® segue a mesma linha de funcionalidades do sistema da RIM®, mas apresenta uma maior interação com os restantes equipamentos que possam existir no parque informático, desde que sejam do mesmo fabricante - por exemplo, computadores pessoais que usem o sistema operativo Microsoft® Windows.

De facto, este tenta interligar-se com os restantes terminais no parque informático que utilizem sistemas operativos Microsoft® Windows®, utilizando standards internos da marca como o Active Directory®.[12]

Também esta solução apresenta pouca informação técnica de relevo para este artigo. Embora anuncie um conjunto de funcionalidades bastante similar às apresentadas pelos mecanismos de gestão clássicos e tendo um suporte direcionado para os terminais móveis, é difícil retirar conclusões sobre o seu real funcionamento. O seu código fechado, informação restrita e esquema de negócio totalmente direcionado para empresas devido ao custo monetário inerente, impedem análises mais profundas.

Ainda assim, existe alguma informação sobre certos mecanismos usados. Podemos neste caso ver, mais uma vez, o uso da tecnologia *push*. Assim, tal como no sistema anterior, esta aposta prende-se com neces-

[11] http://us.blackberry.com/apps-software/business/server/full/
[12] http://www.microsoft.com/en-us/server-cloud/windows-server/active-directory.aspx

sidades de minimizar o uso de recursos dos terminais prevenindo acessos constantes por forma a aceder a eventual informação.

2.3.3 Android® Market Webstore

Atualmente, a maioria das plataformas existentes de *smartphones* ou *tablets*, disponibiliza um ponto central para obtenção e instalação de aplicações. Estas lojas de aplicações[13] vêm ampliar o negócio de aplicações desenvolvidas por terceiros para a plataforma alvo, uma vez que as juntam num só ponto de distribuição. Na perceção do utilizador que procura uma nova aplicação, é agora mais fácil encontra-lá uma vez que não necessita do adicional esforço de procura em vários canais de distribuição. Também do lado do produtor da aplicação existem vantagens, uma vez que não necessita de despender de grandes recursos, para chegar ao seu público alvo - neste caso os utilizadores da plataforma.

No caso da plataforma Android®, esta loja virtual de aplicações denomina-se Android® Market[14]. Nas primeiras iterações esta loja funcionava de forma muito semelhante às restantes. No terminal móvel existia uma aplicação cliente que acedia, via Internet, a uma aplicação servidor onde a informação relativamente às aplicações era mantida. Sempre que o utilizador desejava instalar uma aplicação, após a sua instrução, a aplicação cliente enviava um pedido ao servidor remoto respondendo este último disponibilizando o pacote da aplicação a instalar - modelo *pull*. Neste cenário não existem elementos diferenciadores face às restantes implementações. O interesse na loja de aplicações do Android® surge na sua última versão.

No inicio de Fevereiro de 2011, foi apresentado ao público uma extensão ao Android® Market, o Android® Market Webstore. Esta consiste numa implementação direcionada ao *browser* em dispositivos não móveis, tais como os computadores portáteis, vulgo *desktop*. É agora possível navegar em todo o catálogo de aplicações presente no Market a partir de qualquer sistema operativo, através do browser. Mais, é ainda possível instalar aplicações num terminal Android®, diretamente do browser, sem necessidade de software extra e a partir de qualquer sistema. Neste cenário, o utilizador apenas necessita de ter instalado no seu terminal a aplicação clássica do Android® Market com *login* feito - este *login* é obrigatoriamente uma conta Google®. O aspeto mais interessante desta solução, quando vista no contexto deste artigo, é a forma como as aplicações são enviadas para o terminal. Como referido anteriormente, nas primeiras versões desta loja, existia uma aplicação cliente no terminal que executava pedidos - modelo *pull*. Agora a informação flui no sentido contrário, sendo o servidor a ter a iniciativa de envio do item escolhido pelo utilizador - modelo *push*.

Embora a plataforma Android® seja de código livre, as aplicações diretamente desenvolvidas pela Google®, como a sua loja, não o são. Torna-se portanto difícil obter informação sobre os mecanismos que possibilitam o funcionamento desta loja. Ainda assim, baseado-nos na informação existente, podem-se tirar algumas conclusões bastante realistas.

O facto de ser necessário que o utilizador tenha instalado, no terminal, a aplicação Android® Market com *login* ativo, permite especular uma troca de informação entre essa aplicação e um servidor de suporte à loja que o utilizador visualiza no *browser*. A informação enviada pelo servidor seria reconhecida na aplicação cliente e a interação normal de instalação prosseguiria. Por outro lado, os problemas associados a um sistema do tipo *push* onde o endereço do cliente pode alterar de forma não previsível, podem ser contornados devido à associação da conta de *login*.

Como foi indicado, é necessário que quer a aplicação cliente Market quer a sessão na loja visitada pelo *browser*, estejam afetas à mesma conta Google®. Desta forma, podemos imaginar uma troca de informação - iniciada pelo cliente - onde o terminal envia o seu endereço de acesso ao servidor, que o associa ao utilizador registado. Desta forma, sempre que o servidor desejar começar uma interação, apenas necessita de consultar

[13] Comummente denominadas por *app stores*
[14] https://market.android.com/

o último endereço conhecido do cliente.

Outro ponto de interesse desta tecnologia, é a sua possibilidade de expansão para vários terminais. De facto, se o utilizador tiver mais do que um terminal Android®, todos com a mesma conta registada, é possível escolher para qual deles se quer enviar a nova aplicação. Esta possibilidade é garantida pela troca de informação associada ao contexto da conta de utilizador. Se for guardado não apenas o endereço do dispositivo, mas também um identificador único do próprio dispositivo, podemos gerir vários terminais associados a uma mesma conta.

As comparações com o objetivo proposto na gestão de terminais móveis em parques informáticos são evidentes. Se retirarmos o acesso à aplicação *web* ao utilizador, esta pode ser encarada como o software de gestão - que neste cenário não precisa de ser uma aplicação web. Os terminais continuam a ser os mesmos, mas passam a receber informação não por pedido da parte do utilizador, mas sim por ordem do administrador do software de gestão. Tal como desejado, esta solução pode ainda ser aplicada a um grande número de terminais bastando, por exemplo, implementar associações entre vários registos de terminais, formando grupos de gestão.

2.3.4 Apple® Enterprise Features

A fabricante Apple® apresenta-se atualmente como uma referência no mercado de *smartphones* e *tablets* com os seus produtos *iPhone* e *iPad*, respetivamente. No seguimento da sua quota de mercado crescente, foram também desenvolvidas soluções orientadas ao mundo empresarial.

De inicio a abordagem centrou-se num suporte a soluções já existentes de outros fabricantes, concretamente, o suporte a integração com o *Microsoft® System Center Mobile Device Manager*. Esta abordagem tinha no entanto as mesmas limitações apresentadas anteriormente por esta solução com a agravante de não ser oficialmente suportada pela Microsoft® gerando limitações ao nível de atualizações e suporte a novas funcionalidades.

Para garantir uma solução mais orientada às necessidades e terminais da marca, a Apple® optou por desenvolver uma soluções próprias das quais se destacam o *Apple® Push Notification service*[15] e o *iOS Eterprise Features*[16].

A primeira, surgiu como um extensão do sistema usado nos terminais móveis da marca de forma a permitir o desenvolvimento e suporte de aplicações utilizando tecnologia *push*. Esta não garante no, entanto, mecanismos de garantia de entrega de informação. Desta forma torna-se apenas útil para transferir informação não critica. A segunda solução já pode ser vista como um verdadeiro sistema de gestão dos terminais permitindo funcionalidades como a gestão de software de forma remota ao terminal, proteção da informação presente no terminal e configurações extra, sendo ainda utilizados mecanismos de segurança na troca de informação entre os terminais móveis e o sistema de gestão. Esta solução sofre contudo de muitas das limitações presentes nas soluções já apresentadas. Toda a solução é desenvolvida de forma fechada e orientada exclusivamente aos terminais da Apple®. Um dos resultados disso, é o fraco sistema de inventariação.

Como nos restantes fabricantes, também os terminais com o sistema da Apple® beneficiam de um ponto central de acesso a aplicações. A plataforma oferecida é denominada de Apple® App Store apresenta no entanto um modelo diferente de funcionamento não oferecendo verdadeira sincronização móvel via redes sem fios - Wifi ou GSM/UMTS.

[15] http://support.apple.com/kb/HT3576
[16] http://www.apple.com/ipad/business/software-update/

2.4 Outras Soluções de MDM

Como vimos na secção anterior, existem atualmente várias soluções para a gestão de terminais móveis. Estas são, no entanto, desenvolvidas pelos próprios fabricantes dos terminais, tentando fixar os utilizadores ao tentar oferecer o melhor serviço de forma fechada às suas tecnologias. Começam porem, a surgir outras alternativas totalmente livres de fabricante e até de plataforma.

De seguida apresentam-se algumas dessas soluções com maior visibilidade no mercado atual.

2.4.1 OMA Device Management

Com o aumento da importância dos terminais móveis, surgiu a necessidade de criar um ponto de equilíbrio entre as necessidades que este novo paradigma criou e o trabalho desenvolvido. Esta tentativa de criação de standards tem como estandarte principal na área dos telefones móveis a *Open Mobile Alliance*.[17] Este grupo surge como um organismo de normalização que, aproximando fabricantes e pesquisadores, tenta criar standards de facto - no seu caso orientados às tecnologias de telefones móveis, envolvendo *smartphones* e recentemente *tablets*.

De entre os projetos saídos deste consorcio, vamos dar atenção ao trabalho desenvolvido pelas equipas de *device management* e *Synchronization* do qual resultou a especificação do protocolo *OMA Device Management* - OMA DM.

Este protocolo foi desenvolvido como uma proposta de standard para a realização das operações mais comuns de gestão de terminais móveis como telefones, *smartphones*, *tablets* e *PDAs* entre outros. Ele suporta - na sua última versão 1.2 - os seguintes casos de uso típicos:

- Aprovisionamento do terminal - Configurações iniciais do terminal e habilitação ou desabilitação de funcionalidades.
- Configuração do terminal - Permitindo alterações das definições de configuração do terminal.
- Atualização de *software* - Sendo possível atualizar as aplicações presentes no terminal, bem como do próprio sistema.
- Gestão de falhas - Reportando falhas ocorridas no terminal ou inquirindo o mesmo sobre o seu funcionamento.

Este protocolo permite, de forma transversal, implementar os casos mais comuns da lógica de gestão de terminais móveis, abstraindo problemáticas comuns como a segurança da transferência de dados, direcionamento dos utilizadores para informação relevante e estruturas de dados que implementem a comunicação.

No geral, o protocolo tem em conta algumas das especificidades destes terminais, tendo sido arquitetado com especial atenção aos seguintes aspetos:

- Os terminais alvo não dispõem de grande capacidade de armazenamento e/ou processamento.
- A comunicação não é segura e de recursos ilimitados, devendo ser bem gerida e utilizada apenas quando necessário.
- Terminais com maior índice de insegurança quer por motivos de mobilidade, quer por motivos sociais e a fraca preocupação nesta matéria da vasta maioria de utilizadores.

[17] http://www.openmobilealliance.org/

No entanto, existem algumas desvantagens nesta solução quando revisto à luz dos requisitos deste documento.

Devemos ressalvar que as desvantagens a seguir descritas podem não o ser consideradas em outros contextos. Um exemplo será o suporte exclusivo à comunicação iniciada pelo servidor. De facto, a definição do protocolo prevê, suporta e inclusivamente incentiva ao uso d tecnologia *push*. De forma a evitar mecanismos do tipo *pooling* do lado do terminal móvel, onde este inquire o servidor sobre novos eventos, será o servidor o responsável por informar o cliente quando estes surjam. No entanto, o protocolo não prevê nenhum cenário onde o terminal inicializa a comunicação. Por exemplo, no caso de uso em que um terminal se pretenda registar num sistema de gestão, será da responsabilidade do servidor o seu registo e contacto inicial com o mesmo.

Outro ponto negativo que surge pela sobre exploração da abertura do protocolo, é o facto de existirem várias implementações do mesmo, sendo estas muitas vezes fechadas. Desta forma, um dos grande pontos positivos deste projeto, a estandardização transversal de gestão por diversos terminais independentemente do fabricante ou sistema, acaba por ser destruída. De facto, muitos dos fabricantes de terminais que suportam este protocolo, acabam por introduzir alterações sobre o pretexto de aumento de segurança, o que acaba por deturpar o conceito inicial.

Relacionado com o ponto anterior, devemos ter especial atenção ao facto do suporte ao OMA DM estar embutido no *software* de sistema do terminal. Realizar operações como instalação, /remoção e atualização de aplicações e do próprio sistema requer um acesso total ao terminal sendo necessário que o próprio terminal permita esse suporte. Ou seja, a total implementação do protocolo não pode ser assegurada pela instalação posterior de aplicações.

2.4.2 Funambol Device Management

Uma das implementações do protocolo OMA DM apresentado anteriormente é o projeto Funambol, concretamente na sua componente *Funambol Device Management*.

O projeto Funambol começou com o objetivo de oferecer mecanismos de sincronização de dados para terminais móveis. A evolução foi constante, disponibilizando atualmente soluções de *cloud storage*, sempre direcionadas a terminais móveis. Sendo a maioria do seu desenvolvimento *software* livre e de qualidade comprovada, rapidamente atraiu apoios de várias organizações ligadas a venda, produção e integração de terminais móveis.

Como vista a expandir os seus serviços, desenvolveu uma componente de gestão de terminais móveis, utilizando para isso o standard OMA DM. Esta solução é disponibilizada em duas vertentes, uma para o público em geral e outra direcionada a ambientes empresariais, onde é necessária uma grande disponibilidade de serviço e escalável.

Na prática, é mais uma implementação do protocolo OMA DM tendo todas as suas vantagens e desvantagens. Tem como ponto forte o facto de ser uma tecnologia *open-source*, mesmo na vertente empresarial, permitindo alterações de forma a direcionar a solução às necessidades existentes num grupo de gestão.

2.4.3 Zenprise MDM

A empresa *Zenprise*[18] foi fundada em 2003 com o objetivo principal de assistir empresas na gestão em larga escala de terminais BlackBerry®. Como a evolução do mercado, a sua oferta de serviços aumentou de forma a dar suporte a outros fabricantes e outros terminais - como *tablets* e PDAs. Sempre orientada a terminais móveis, desenvolveu soluções de *cloud computing* e assistência técnica para um grande número de dispositivos, tendo na sua carteira de clientes entidades governamentais, produtores tecnológicos e empresas

[18] http://www.zenprise.com/

multinacionais com necessidades especificas de computação móvel.

Do ponto de vista de soluções de MDM, a *Zenprise* apresenta uma estrutura muito semelhante a outras, baseando-se em modelos de cliente-servidor, onde um sistema central desenvolve todas as atividades de gestão habituais - instalação/remoção/atualização de *software* e/ou *firmware*, backup remoto de informação do terminal, etc - diretamente em vários terminais móveis, de forma remota e com o mínimo de interação do utilizador.

O que esta solução apresenta como mais valia e que a distancia das restantes, são as funcionalidades estendidas como a possibilidade de *login* remoto no terminal, criação de *VPNs* específicos por aplicação e distribuição segura de aplicações entre terminais, gerida de forma remota, bem como toda uma vertente de prevenção de fuga de informação dos próprios terminais, num modelo apelidado de *Mobile Data Leakage Prevention*.

Mais ainda, outras soluções da mesma empresa, que não sendo estritamente definidas como MDM, podem atuar em conjunto de forma a ajudar na atividade de gestão do parque informático de terminais móveis de uma empresa, incluem módulos de diagnóstico de rede e de configuração automatizada de serviços de email.

Esta solução apresenta no entanto um modelo de negócio estritamente fechado, sendo impossível o acesso ao seu código. Desta forma, apenas os sistemas aos quais é dado suporte podem beneficiar desta solução. Sendo ainda uma empresa com calendarizo económico, esta evolução de suporte a terminais de outros fabricantes, bem como a incorporação de outras funcionalidades, apresenta-se limitada aos interesses da própria empresa.

2.4.4 MobileIron MDM

Reconhecida como líder a nível de soluções MDM não desenvolvidas pelos fabricantes de terminais móveis, a *MobileIron* foi das primeiras empresas a desenvolver soluções deste género, ajudando a implementar a metodologia de *Bring You Own Device - BYOD* muito em voga hoje em dia.

Do ponto de vista de funcionalidades oferecidas, não destoa muito das oferecidas pelas soluções concorrentes, permitindo o a distribuição de aplicações de forma remota, implementação e execução de políticas nos terminais geridos, assim como um módulo de gestão de inventário.

Talvez o maior ponto a favor desta solução, seja a sua lógica modular e que permite interligar outras soluções para a resolução de problemas. Um exemplo, será a facilidade com que se associa a sistemas como o BIS/BES que possuem uma forte componente de gestão de email para terminais BlackBerry®, permitindo que estas soluções fiquem responsáveis por uma sub-parte do sistema de gestão, sendo no limite geridas pelo sistema da *MobileIron*.

Como ponto forte, podemos ainda apontar todas as funcionalidades de geração de relatórios baseados nos mais diversos índices, como utilização de processamento, por terminal e mesmo por aplicação, relatórios de qualidade de serviço - como estado da rede de dados móveis - segundo índices como o número de utilizadores em simultâneo, entre outros.

Ainda assim, os mesmos pontos negativos pode ser indicados a esta solução, de forma bastante semelhante às apresentadas anteriormente. Mais uma vez, esta segue um modelo de código estritamente fechado e orientado a um numero restrito de terminais que se vais adaptando segundo as margens do mercado.

2.5 Sumário

Nesta secção vamos fazer um pequeno comparativo entre as soluções apresentadas neste capítulo o os objetivos propostos na secção 1.1, analisando de forma resumida as mais valias de cada um e o porque de estas não serem uma resposta final à problemática apresentada.

De facto, as soluções apresentadas parecem suportar todas as necessidades exigidas na atividade de gestão de parques informáticos de terminais móveis. Na tabela 2.1 podemos observar um resumo entre as soluções apresentadas e os principais focos de interesse no âmbito deste documento.

	Multi Plataforma	**Registo inicializado pelo terminal**	**Fluxo da gestão**	**Atenção às restrições de terminais móveis**
Sistemas clássicos	Sim	Sim em muitos casos	Bidirecional	Não
BlackBerry Enterprise Server	Proprietário	Sim para parte das funcionalidades	Maioritariamente inicializada pelo serviço de gestão	Sim na totalidade
Microsoft SCMDM	Proprietário	Não	Inicializada pelo serviço de gestão	Sim em parte (falta boa praticas para terminais não alcançáveis)
Android Market WebStore	Proprietário	Sim embora existe necessidade de registo das credenciais	Bidirecional	Sim
Apple Enterprise Features	Proprietário	Não	Maioritariamente inicializada pelo serviço de gestão	Sim (falta boa praticas para terminais não alcançáveis)
OMA Device Management	Sim	Não Não contempla comunicação inicializada pelo terminal	Exclusivamente inicializada pelo serviço de gestão	Sim
Funambol Device Management	Sim	Não	Exclusivamente inicializada pelo serviço de gestão	Sim
Zenprise MDM	Proprietário	Não	Maioritariamente inicializada pelo serviço de gestão	Sim
MobileIron MDM	Proprietário	Não	Maioritariamente inicializada pelo serviço de gestão	Sim

Table 2.1: Comparação entre as soluções apresentadas

Como se mostra nenhuma solução cumpre na totalidade o suporte abrangente a terminais e sistemas, embora comessem a existir alguns projetos nesse sentido.

Um aspeto que merece destaque é o suporte à tecnologia *push*. Podemos observar que algumas soluções o suportam mas de forma parcial. Isto significa que, apesar de existir o conceito base onde a comunicação é inicializada pelo servidor, não sendo necessário o cliente executar um pedido prévio com o intuito de inquirir a existência dessa informação, a informação não é enviada de forma confiável. Caso o terminal não esteja

contactável ou tenha mudado de endereço e tenha omitido essa informação ao servidor, não existe qualquer garantia que o envio seja recebido. Num cenário orientado a terminais móveis, onde a falta de conexão ou a mudança dos endereços de acesso são uma constante, a garantia de receção é uma questão essencial.

Por fim, podemos observar que a comunicação é quase sempre inicializada pelo serviço se gestão, seja de forma obrigatória como em soluções baseadas em OMA DM, seja para registo inicial do terminal no sistema de gestão.

Ainda que estas soluções não cumpram na totalidade os requisitos propostos na secção 1.1 do Capítulo 1, apresentam já uma grande quantidade de funcionalidades desejáveis, estando o seu desenho orientado ao paradigma móvel. Para isso, muito contribuíram os conceitos apresentados na secção 2.1 e que vamos aqui cruzar com as soluções mencionadas como pode ser observado na tabela 2.2.

	Push	*Publish/Subscribe*	Single-write Multiple-readers	Segurança	MDM completo
Sistemas clássicos	Possível	Possível	Sim	Sim	Não
BlackBerry® Enterprise Server	Sim	Tem alguma noção de grupos	Sim	Sim	Sim - direcionado
Microsoft® SCMDM	Parcialmente. Sem garantia de entrega	Não	Sim	Sim	Sim - direcionado
Android® Market Webstore	Sim	Não	Sim	Não	Não
Apple® Enterprise Features	Parcialmente. Sem garantia de entrega	Não	Sim	Sim	Sim - direcionado
OMA Device Management	Sim	Sim	Sim	Sim	Sim
Funambol Device Management	Sim	Sim	Sim	Sim	Sim
Zenprise MDM	Sim	Sim	Sim	Sim	Sim
MobileIron MDM	Sim	Sim	Sim	Sim	Sim

Table 2.2: Soluções existentes e requisitos respeitados

Capítulo 3

Arquitetura

Nesta secção apresentamos a arquitetura da solução a implementar. O sistema SuusMDM é capaz de inventariar de forma básica terminais, utilizadores e aplicações bem como gerar e inventariar perfis que constituem grupos de aplicações. Estes perfis são associados a um ou mais utilizadores numa relação de muitos para muitos. Após o registo de um terminal no sistema, o início de sessão no mesmo permite implementar todo o perfil escolhido pelo utilizador no terminal e remove-lo após o término da sessão.

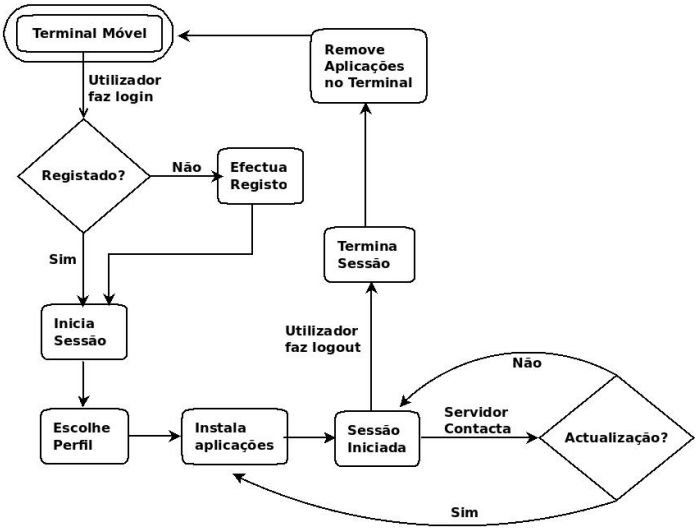


Figure 3.1: Fluxograma de vida da solução

3.1 Arquitetura Base

A solução baseia-se numa arquitetura cliente-servidor que se divide em dois componentes básicos: o sistema de gestão central e uma pequena aplicação cliente a executar nos terminais móveis geridos. De forma genérica o fluxo de execução da solução é o seguinte:

O terminal móvel é registado no sistema de gestão, quer por iniciativa própria, quer por registo direto no sistema. Após este processo, um início de sessão no terminal desencadeia a escolha de um perfil por parte do cliente que, por sua vez, instala no terminal as aplicações associadas a esse perfil. Caso haja alguma

atualização no serviço de gestão, o terminal é informado sendo contactado diretamente. Ao ser terminada a sessão no terminal, este remove todas as aplicações instaladas associadas ao perfil utilizado de início.

A análise do fluxo de execução descrito anteriormente e que pode ser visualizado na figura 3.1, não prevê uma grande necessidade de poder de processamento. Mais ainda, sendo a solução direcionada ao meio empresarial, existe uma grande necessidade de controlo sobre todo o sistema e sobre a informação presente no mesmo. Desta forma, optou-se no, sistema de gestão, por uma arquitetura centralizada com uma única entidade com poderes de alteração da informação.

A fraca necessidade de processamento leva a que não seja imprescindível uma arquitetura distribuída no sistema de gestão. O mesmo não é dizer que não será necessário um grande espaço de armazenamento, ou que este não necessite de escalar de forma a suportar um aumento do número de terminais móveis, o que pode levar a uma sobre-carga do nó de acesso. Desta forma, uma arquitetura centralizada com um balanceamento de carga inicial e várias réplicas da informação é desejável.

Por outro lado, existirão vários terminais móveis a gerir. Estes apenas poderão aceder à informação - aplicações, perfis - não tendo qualquer capacidade para a alterar diretamente no sistema. Esta arquitetura enquadra-se num modelo *single-writer multiple-readers*. De notar que poderia existir problemas de concorrência caso um leitor estivesse a aceder a informação que esteja a ser alterada no mesmo instante. No entanto, sendo o sistema de gestão - único *writer* - que contacta os terminais enviando-lhes a informação, tal cenário não poderá acontecer.

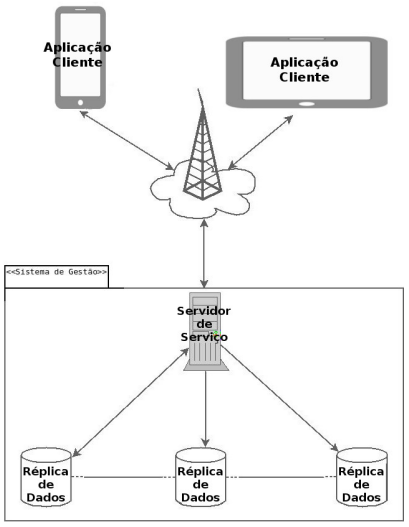


Figure 3.2: Arquitetura base da solução

A figura 3.2 mostra como os vários terminais acedem sempre ao mesmo nó do sistema de gestão, sendo o processamento executado de forma central mas replicado caso necessário, sendo também estas réplicas acedidas numa base de necessidade mas apenas pelo sistema de gestão. Desta forma, toda a informação fica guardada sobre a alçada de um sistema fisicamente central e sobre uma tutela única.

Note-se ainda que, embora se admitam cenários onde os terminais móveis se encontrem fisicamente muito longe do sistema de gestão, não existe uma necessidade crítica de resposta em tempo real, pelo que as tecnologias de transferência de dados atuais - HSPA+ / HSUPA - servem o propósito.[21, 15, 25]

Ainda na figura3.2 podemos ver a arquitetura do ponto de vista de execução técnica. Como referido, estamos presente uma arquitetura *single-writer multiple-readers* em que os terminais móveis a gerir - *readers* - se encontram dispersos acedendo todos ao mesmo sistema de gestão, o único com poder para alterar a informação disponibilizada - *writer*. Nestes terminais é executada uma aplicação cliente, a mesma instalada aquando do registo do terminal no parque informático, e que realiza toda a lógica de execução. Do lado do sistema de gestão, temos um servidor responsável pelo processo de distribuição da informação bem como pela manutenção da informação, quer dos terminais - do processo de inventário - quer a enviar aos mesmos. Ainda no contexto do sistema de gestão, temos um ou vários sistemas de base de dados que servem o propósito de preservar toda a informação do sistema, podendo escalar se necessário.

3.2 Modelo de Domínio

A informação resultante da inventariação de terminais, aplicações, perfis e utilizadores, bem como o registo gerado durante a vida ativa de uma sessão, que é guardada pelo sistema de gestão, apresenta ligações entre si. Apresentamos nesta secção uma visão geral do modelo de domínio utilizado pelo sistema, assim como as relações entre os vários nós de informação.

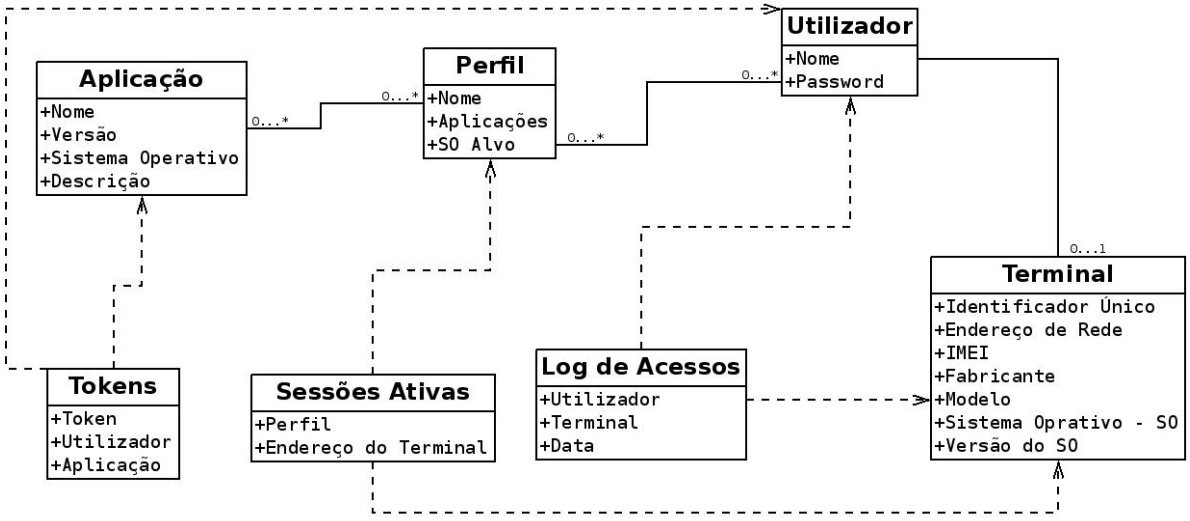


Figure 3.3: Modelo de domínio simplificado

Como se pode ver na figura 3.3 existem 7 nós no modelo de domínio da solução desenvolvida. Começando pelos mais óbvios, na medida em que apresentam conceitos utilizados diretamente pelo utilizador, temos os nós de **utilizador**, **aplicação**, **terminal** e **perfil**. Destes, os pares [utilizador; perfil] e [aplicação; perfil] apresentam relações de muitos para muitos, uma vez que um utilizador pode ter vários perfis podendo estes estar associados a vários utilizadores. Cenários idênticos no par [aplicação; perfil] uma vez que um perfil pode ser composto por várias aplicações, podendo estas estar presentes em diversos perfis. Ao mesmo tempo, um utilizador pode ter zero ou mais terminais registados em seu nome, sendo o dispositivo associado apenas a um utilizador.

A informação guardada nestes nós foi escolhida sob dois prismas: o valor identificativo da mesma e a sua presença transversal ao longo do maior número possível de terminais.

No caso da informação recolhida sobre os terminais, foca-mo-nos sobretudo no modelo e marca do mesmo, o sistema operativo que utilizam e a sua versão e, caso exista, o IMEI do mesmo. Os restantes campos são

indicativos apenas do fluxo da lógica de execução e não de características do próprio terminal. A saber: o último endereço de rede conhecido e o nome do utilizador que o registou no sistema.

Os restantes nós do modelo de domínio representam interesses da lógica interna da solução, não sendo modificados diretamente pelo utilizador. Neste subgrupo estão presentes os seguintes nós: **sessões ativas**, que fazem um uma ligação entre um perfil e o endereço de rede do terminal onde ele está ativo; ***tokens* temporários**, onde são guardadas as relações entre os *tokens* ativos, o utilizador a que pertencem e a aplicação à qual dão acesso; e **log de acessos**, nó onde é guardada a informação gerada pela ligação entre um utilizador, um identificador de um terminal e uma data e hora onde o utilizado iniciou uma sessão no dispositivo indicado pelo identificador.

3.3 Casos de Uso

Nesta secção vão ser analisados os principais casos de uso executados pela interação entre os terminais móveis a gerir e o sistema de gestão. As conclusões obtidas desta análise são as bases teóricas da implementação do protótipo final.

3.3.1 Registo do Terminal Móvel

A primeira atividade a executar é o registo do terminal móvel no sistema de gestão. De forma a oferecer maior liberdade aos utilizadores, poupando também recursos associados à execução desta atividade em casos de parques informáticos com um grande número de terminais a gerir, este processo poder executado quer diretamente no sistema de gestão, quer pelo próprio terminal móvel.

De forma abstrata, o processo de registo deve englobar duas atividades: a inventariação do terminal no sistema e a geração de um identificador único do mesmo.

A inventariação do terminal no sistema, permite dar um maior nível de controlo e personalização ao sistema de gestão. Por um lado, conhecer o número de terminais presentes no parque informático e as suas características oferece níveis de conhecimento importantes quer para uma gestão económica associada à compra, venda e atualização dos mesmos quer para orientação de políticas de desenvolvimento e orientação de negócio. Por outro lado podemos, de uma melhor forma, direcionar ações a realizar nos terminais se os conhecermos. Será mais fácil evitar fluxos de qualquer natureza que sejam à partida inválidos devido a incompatibilidades com os dispositivos finais tais como operações de controlo ou instalação de aplicações não suportadas quer pelo sistema operativo do mesmo quer pelas características técnicas deste .

Esta informação do terminal móvel - o seu sistema, características técnicas, etc - embora não seja de valor crítico, deve ser mantida privada. Neste campo o registo direto no sistema de gestão apresenta vantagens uma vez que não inclui a ação extra da passagem de informação por canais porventura pouco seguros, permitindo diversos ataques do tipo *man-in-the-middle*.[1]

Já a geração de um identificador único para o terminal a gerir apresenta uma importância elevada. É necessário garantir que é enviada, recebida e processada informação vinda de um terminal identificado de forma inequívoca. Caso contrário, o sistema bem como a informação presente nele, poderiam ser comprometidos ao permitir acesso a terminais não autorizados, ou em casos menos pessimistas, poderiam ocorrer erros de contexto não sendo enviada informação para o terminal correto. Da mesma maneira, o terminal deve ter alguma garantia de autenticidade do sistema de gestão. Aqui o caso será ainda mais crítico uma vez que o terminal responde de forma completa às instruções vidas do serviço de gestão.

[1] https://www.owasp.org/index.php/Man-in-the-middle_attack

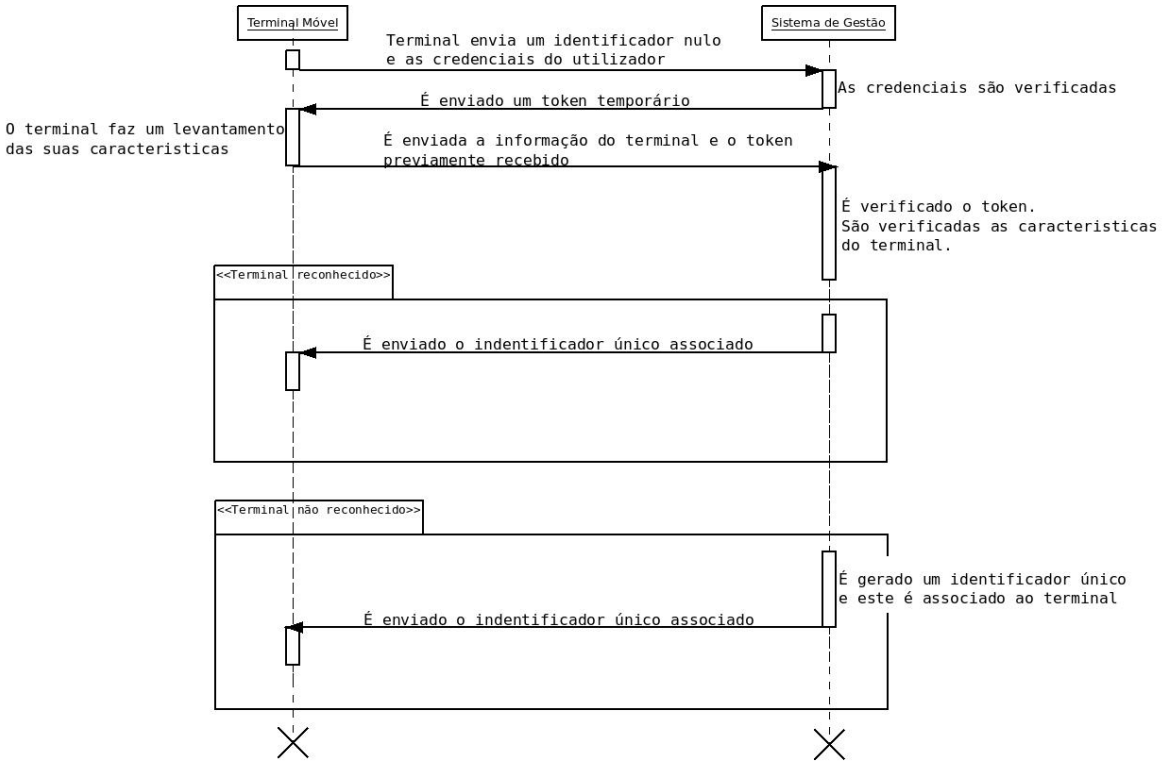


Figure 3.4: Registo de terminal móvel

Como medida de segurança, um terminal registado de forma remota, apenas o poderá fazer caso o utilizador providencie os seus dados de identificação no sistema - nome de utilizador e palavra-passe - e estes sejam válidos.

Como se pode ver na figura 3.4 este identificador é gerado caso o terminal se registe pela primeira vez no sistema de gestão ou é devolvido ao terminal móvel caso este já tivesse sido registado anteriormente mas tenha perdido o identificador - limpeza total da memória do aparelho por exemplo.

Informação sobre os mecanismos de suporte à comunicação de forma segura podem ser encontrados com mais detalhe na secção 3.5 deste capítulo. A problemática da geração e gestão dos identificadores únicos poder ser vista em detalhe na secção 3.6 deste capítulo.

3.3.2 Gestão de Perfis

Do ponto de vista do utilizador do terminal móvel o principal caso de uso será o início da sessão no dispositivo e a escolha de um dos seus perfis, o que fará com que todas as suas aplicações sejam instaladas no terminal de forma o mais transparente possível e com o mínimo de interação.

Para que isto aconteça, as entidades de **perfil** têm de ser geradas e associadas aos utilizadores - embora estas possam existir de forma independente, podendo ser gerados perfis sem associação a utilizadores assim como perfis sem aplicações embora a aplicação prática destes não sejam muita.

A gestão de perfis, especialmente a criação dos mesmos, apresenta desafios assim técnicos como humanos. O exemplo seguinte demonstra este ponto:

Na empresa Rodas&Turbo, especialista em afinação automóvel, todos os funcionários utilizam terminais smartphones de forma a desempenharem o seu trabalho de maneira mais ágil. Nos seus terminais disponibilizados pela empresa, podem existir várias aplicações conforme a especialidade do trabalhador: técnico de interiores, pintor, afinador de motores, etc. Devido ainda a questões financeiras associadas à compra de terminais, quer pela adição de novos elementos às equipas de trabalho, quer devido a avarias de terminais que necessitam de ser trocados, a Rodas&Turbo compra diversos terminais a diversos fabricantes. Estes terminais não têm, no entanto, todos o mesmo sistema, existindo versões diferentes das mesmas aplicações de forma a suporta-los na integra.

Do exemplo descrito anteriormente podemos observar a necessidade da geração de perfis para aplicações consoante as funções do trabalhador e consoante o terminal utilizado. A primeira observação remete para desafios humanos de criação de perfis associados às necessidades dos utilizadores. Podemos mesmo extrapolar a criação de *grupos virtuais*. Se todos os elementos que desempenham determinadas tarefas necessitam das mesmas aplicações, estamos perante um perfil de grupo e não individual. Já a segunda observação incorpora desafios técnicos. Podemos imaginar cenários onde para dois perfis iguais em termos de aplicações, mas onde um seja direcionado a uma plataforma móvel e o outra a uma distinta, o próprio sistema deve ser capaz de verificar em que tipo de terminal o utilizador inicia a sessão e enviar apenas os perfis relevantes.

Ambos os casos descritos são suportados pela nossa solução. Sobre o primeiro, não existe nenhum desenvolvimento especifico. Sobre a estratégia adaptativa de perfis consoante o sistema do terminal, a mesma foi implementada utilizando a informação recolhida durante o processo de registo do terminal para, em primeira instância, inventariação.

3.3.3 Início de Sessão no Terminal Móvel

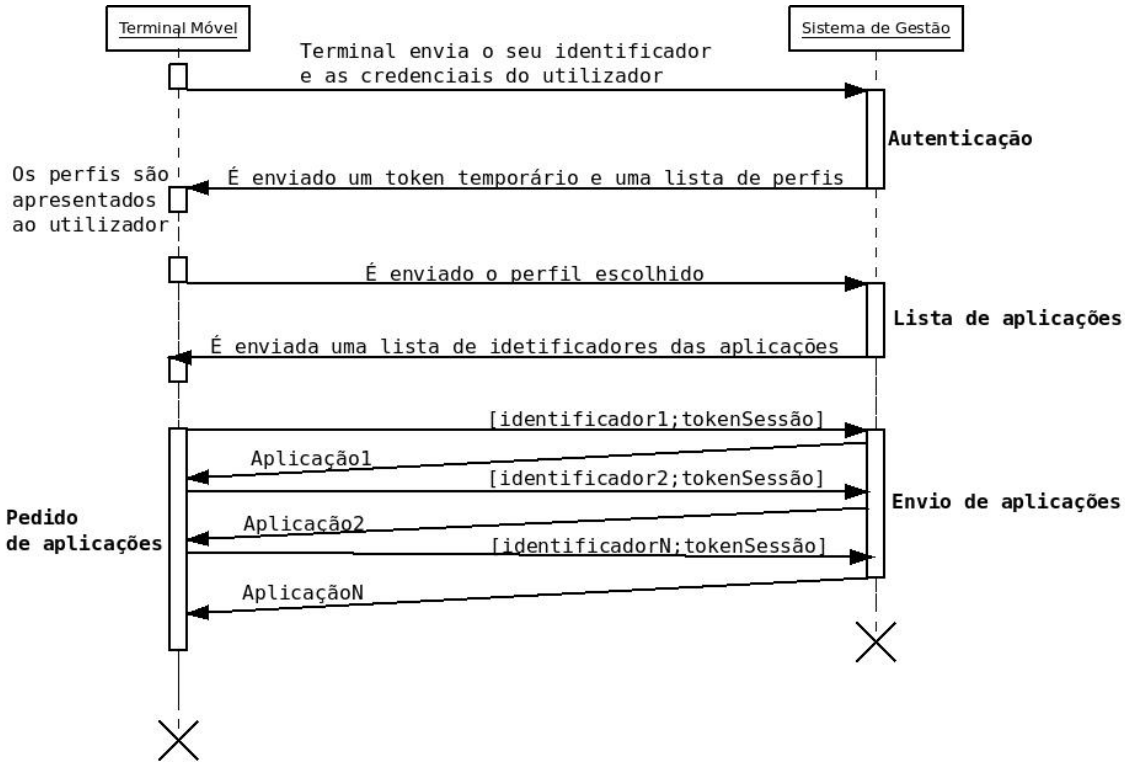


Figure 3.5: Início de sessão no terminal móvel após registo

Do ponto de vista do utilizador de um terminal móvel associado ao sistema de gestão desta solução, a única atividade a realizar é o início de sessão. De facto, a aplicação cliente apresenta uma interface muito restrita permitindo a introdução das credencias de utilizador (nome e palavra-passe) assim como o endereço do sistema de gestão. De seguida existe apenas a opção de término de sessão.

Esta ação de início de sessão engloba no entanto um outro caso de uso, caso o terminal não esteja ainda inserido no sistema de gestão. Quando o utilizador faz *login*, o terminal verifica se possui um identificador único gerado pelo sistema em interações anteriores. Caso não tenha, quer por ser a primeira execução, quer por ter sofrido alguma limpeza de memória, o processo de registo irá ser desencadeado como indicado na explicação do caso de uso de registo de terminais, presente nesta secção do documento.

Passado o processo de registo, caso ele seja necessário, e verificadas as credenciais do utilizador, é iniciada a sessão no terminal sendo o restante fluxo o apresentado na figura 3.5. Nesta altura, o sistema de gestão devolve ao terminal um lista contendo todos os perfis que o utilizador tem associados, bem como um *token* temporário associado, registando desde logo uma nova entrada no nó de *log* do modelo de domínio, associado o utilizador, o identificador do terminal e a data onde o processo de início de sessão foi inicializado.

O utilizador deve agora escolher o perfil que desejar sendo essa informação, em conjunto com o *token* recebido no passo anterior, enviada de volta para o serviço de gestão onde são desencadeados 3 acontecimentos caso o *token* e o utilizador sejam validados. Primeiro, é registado pelo modelo de domínio que o perfil escolhido está ativo no endereço de rede atual do dispositivo móvel. De seguida, para cada aplicação pertencente ao perfil indicado, deve ser gerado um *token* temporário associado. Por fim é devolvida uma lista de indicadores únicos de aplicações e os *tokens* associados. Resta agora ao terminal fazer um pedido por cada aplicação, enviando o *token* respetivo, de forma a que esta seja devolvida pelo serviço de gestão e seja instalada no terminal.

3.3.4 Atualizações Iniciadas pelo Sistema de Gestão

Os perfis criados e associados a um utilizador não devem ser vistos como entidades imutáveis. De facto, pode ser necessário adicionar novas aplicações, remover aplicações existentes ou atualizar as mesmas.

Estas mudanças devem ser automaticamente propagadas a terminais onde as sessões estejam afetas aos perfis relevantes e de forma célere. Aqui duas estratégias podem surgir: ou o terminal móvel inquire periodicamente o sistema de gestão sobre possíveis alterações ou o próprio sistema contacta diretamente o terminal móvel quando estas existam.

De forma a garantir o mínimo de ligações de rede por parte dos terminais móveis, poupando também outros recursos como processamento e bateria, é utilizada uma abordagem do modelo *push*. É o sistema de gestão que, quando existem atualizações de informação pertinente e apenas nestes casos começa a comunicação com os terminais alvo transmitindo-lhes a informação. Desta forma, não é necessário que o terminais móveis estejam constantemente a pedir atualizações até ao momento em que elas existam.

De notar que, como no processo de início de sessão, não são enviadas diretamente as aplicações para o terminal móvel - como se pode ver na figura 3.6. Como já foi referido, cenários de falha de conectividade são sempre possíveis sendo até prováveis em contextos específicos. A informação que melhor permite antecipar cenários de falha de rede, encontra-se porem, invariavelmente no lado do terminal móvel. De facto, apenas o terminal pode, quando possível, saber que uma falha na ligação vai ocorrer. Mesmo que não seja sempre possível esta antecipação, a maioria da informação encontra-se mais facilmente ao alcance deste. Por tudo isto, são enviados para o terminal os identificadores das aplicações que deverá pedir quando melhor lhe convier. Desta forma é minimizado o risco de transferências falhadas e atualizações interrompidas o que levaria a mais trocas de mensagens entre o terminal móvel e o sistema de gestão para garantir nova estabilidade. No limite, o processo de atualização pode ocorrer até de forma parcial até à sua finalização, uma vez que o

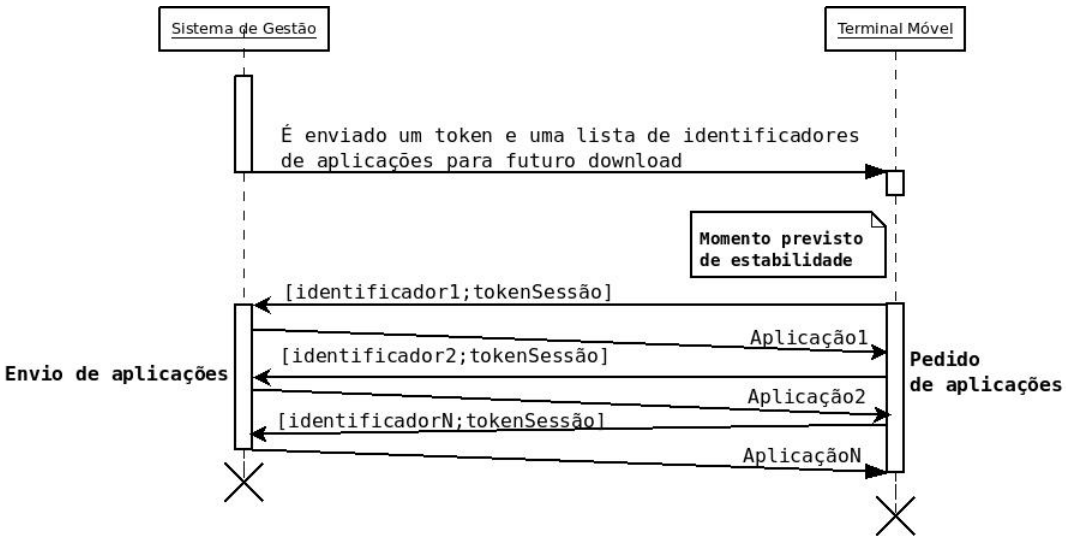


Figure 3.6: Envio de aplicações por iniciativa do sistema de gestão

terminal, quando deteta problemas de comunicação, pode interromper o processo e retomá-lo mais tarde sem limite de tentativas não sendo necessária a realização sequencial dos pedidos como é apresentado na figura 3.6. Podem assim existir várias interrupções de comunicação que resultam em várias fazes de pedido. Deve ser compreendido porém que esta estratégia, apesar de minimizar a utilização de recursos resultantes de erros por falha de comunicação, não perfeita. Na verdade, tal é praticamente impossível em cenários de mobilidade móvel. O que se pretende aqui é minimizar os efeitos colaterais ocorrentes de falhas durante o processo.

Este cenário apresenta no entanto dois potenciais problemas de segurança. O primeiro prende-se com a comunicação não protegida entre o sistema de gestão e o terminal. Sendo a comunicação inicializada pelo sistema de gestão, seria necessário uma garantia de privacidade de informação trocada que existe quando a informação circula no sentido inverso devido a utilização de certificados e comunicação sobre SSL. As soluções para este problema podem passar pela cifra direta dos dados enviados através de algoritmos orientados a terminais móveis. De facto, a privacidade é a única métrica necessária de ser salvaguardada uma vez que o pedido deve ser processado pelo cliente antes de ser pedido, o que evita ataque de repetição, e a autenticidade não é essencial pois o terminal móvel vai sempre fazer o pedido ao sistema de gestão central, com as credenciais enviadas na mensagem, independentemente de quem lhe tenha enviado a informação. A utilização de novos algoritmos de cifra especialmente aptos para terminais móveis como a cifra de curva elíptica [20][6] apresentam-se como soluções ideais até pelo tamanho reduzido de informação a trocar e os bons resultados obtidos quando realizados sobre terminais móveis.[9]

Por outro lado, como seria possível existir perdas de comunicação devido a problemas de rede, é utilizado um modelo verdadeiramente *push*, ou seja, é o próprio terminal que apresenta um porto de escuta ao mundo. Mecanismos como *long polling* baseiam-se em comunicações síncronas do modelo *pull* de forma a emular o comportamento desejado. De facto, a essência destas soluções é a realização de um pedido por parte do cliente ao servidor, pedido este que fica com a resposta bloqueada até que o servidor tenha alguma informação a devolver - ao contrario de uma resposta vazia. No nosso contexto móvel, tal iria degenerar num modelo de *pull* exaustivo sempre que a comunicação fosse interrompida por falha na rede ou uma utilização exagerada de recursos caso contrário. Esta escolha apresenta no entanto eventuais problemas de ataques de carga - *Denial of Service* **(DoS)**. Um atacante que deseje pode abrir várias conexões enviando dados aleatórios com o intuito de impossibilitar o bom funcionamento do sistema, levando em casos extremos a um bloqueio do terminal móvel. Embora este cenário não possa ser totalmente evitado, práticas comuns como o limite de conexões de um mesmo endereço ou limite temporal entre as mesmas são mecanismos de fácil

implementação e com resultados favoráveis. Estas soluções apresentam, ainda assim, algumas diferenças nos seus resultados quando comparadas a implementações em sistemas clássicos devido às especificidades de terminais móveis. Está é uma área - defesa e ataques em terminais móveis - que apresenta um grande centro de estudo atualmente sendo ainda um campo de conhecimento aberto e sem soluções gerais, como se pode observar na diversa literatura.[16, 22, 7]

3.3.5 Fim de Sessão no Terminal Móvel

Ao terminar uma sessão, o SuusMDM encarrega-se de desinstalar todas as aplicações associadas ao perfil que foram instaladas, quer no inicio da sessão quer em instalações remotas subsequentes por motivos de modificação de perfil. Será também nesta altura que as aplicações que foram descarregadas do serviço de gestão para o terminal de forma a proceder à sua instalação, serão apagadas do terminal. De facto, o instalador das mesmas reside no terminal até à finalização da sessão por questões de desempenho caso ocorram ações de remoção direta das mesmas ou por error não especificados. Em vez de existir um pedido direto ao serviço de gestão para o reenvio do instalador, este pode ser feito diretamente poupando não só o tempo e recursos associados à comunicação, bem como oferecendo a possibilidade de reinstalação mesmo quando não existe comunicação possível.

A única comunicação executada na altura do término de sessão é do terminal para o sistema de gestão de forma a comunicar a ação. Desta forma o serviço de gestão pode libertar os recursos associados ao controlo de atualizações necessárias a enviar para o terminal. Com esta informação de fecho o terminal deixa efetivamente de ser contactado em casos de atualizações a perfis, uma vez que nenhum se encontra ativo e quando uma nova sessão for iniciada, a escolha do perfil já retornará todas as atualizações executadas.

3.4 Terminais Não Alcançáveis

Embora a componente forte da nossa solução passe por um modelo de comunicação iniciada pelo terminal móvel - pedido de registo, envio de informação, pedido de perfil e aplicações - o sistema desenvolvido apresenta capacidades que beneficiam de uma ligação inicializada de forma contrária - como a atualização automática de aplicações (secção 3.3.4).

É certo que pode existir no entanto um problema de clientes não alcançáveis. Devido à sua natureza móvel, podem num dado instante não ter acesso à rede, ou existirem com um endereço desconhecido. Para resolver esta particularidade, são utilizados, nos terminais móveis, mecanismos de *call-home* fazendo com que sempre que estes obtenham um novo endereço, o transmitam para o sistema de gestão. Por outro lado, se o terminal móvel estiver de facto não alcançável, a solução apresenta mecanismos que permitam um suporte gracioso a este facto.

Aqui a abordagem da nossa solução reside na criação de filas de mensagem direcionadas ao par [perfil; terminal]. Sempre que um terminal não esteja alcançável e que tenha sessão iniciada no perfil relevante, a mensagem a enviar fica guardada de forma a ser posteriormente enviada. Na realidade, se o terminal não está ativo, quando estiver apenas duas abordagens podem ocorrer: ou envia uma mensagem a indicar o seu novo endereço ou uma mensagem a indicar o término de sessão. No primeiro caso, as mensagens que estavam em espera são enviadas para processamento. No segundo caso, o término de sessão remove o par [perfil; terminal] e todas as estruturas a ele associadas como é o caso das filas de mensagem a enviar.

Na figura 3.7 podemos observar um fluxograma da execução deste comportamento do lado do sistema de gestão. Sempre que um perfil é alterado, são encontradas as sessões que contêm este perfil ativo - observando as entradas [perfil; terminal] relevantes - sendo posteriormente gerados *tokens* para cada uma delas. Com a restante informação das entradas, é tentada a comunicação com o terminal móvel. Caso seja possível, são lhe enviadas as alterações e os *tokens*. Caso tal não seja possível, as mensagens são então agrupadas em filas

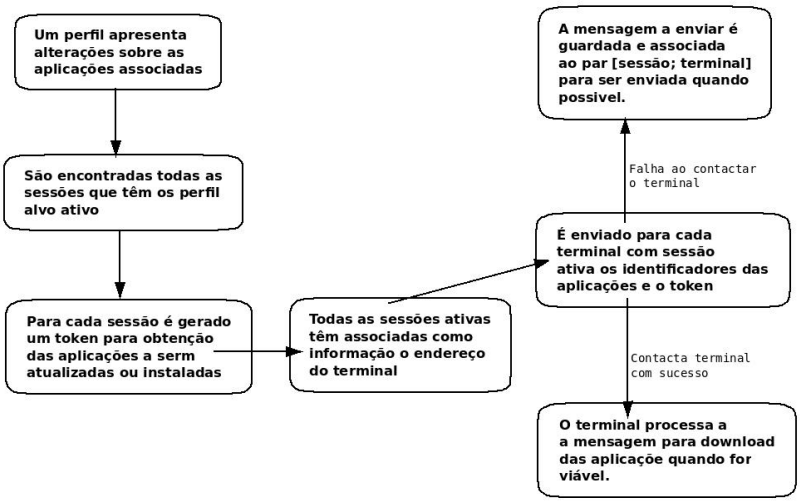


Figure 3.7: Estratégia de comunicação tolerante a falhas

até que o terminal seja de novo alcançável, ou termine a sessão, como explicado anteriormente.

Devemos indicar que a não receção de uma atualização por falta de comunicação com o terminal, apenas pode gerar problemas na atual sessão. Mesmo que o utilizador, quando voltar a ter conectividade, termine imediatamente a sessão não recebendo as atualizações do perfil, o próximo estado é obrigatoriamente um início de sessão que desencadeia uma nova instalação do perfil selecionado, perfil este totalmente atualizado pelo serviço de gestão.

3.5 Segurança e Comunicação

Independentemente da mobilidade dos terminais, uma solução que tem o intuito de troca de informação entre vários dispositivos e, neste caso, particular com possibilidade de alterações do próprio sistema deve garantir propriedades como a integridade, autenticidade e confidencialidade.

Num cenário de terminais móveis, devem ser tidas em conta duas propriedades essenciais. Em primeiro lugar devido à mobilidade dos terminais, nada garante que estes utilizem sempre um canal de comunicação seguro. Deve ser um dado adquirido a insegurança durante a comunicação entre o sistema de gestão e os terminais móveis. Por outro, lado devemos ter presente as limitações ao nível do poder computacional dos mesmos. Da primeira constatação advém a necessidade de cifrar a informação utilizada. Já a segunda propriedade surge a necessidade de uma escolha ponderada dos sistemas utilizados para a implementação da criptografia.

À partida, a resolução deste problema pode parecer simples utilizando métodos de cifra simétrica - historicamente menos pesados computacionalmente - e utilizando cifras assimétricas apenas para a assinatura da informação a transmitir garantindo propriedade de autenticidade. Devem no entanto ser analisados outros fatores. No uso de uma cifra simétrica, como é transmitida a chave entre o sistema de gestão e o terminal móvel? Que algoritmo criptográfico melhor se adequa às características computacionais dos terminais móveis? Note-se que esta última questão deve também ser posta caso sejam usados mecanismos de chave assimétrica.

Para responder à questão da troca da chave simétrica entre dois pares, algoritmos como o de *Diffie-Hellman*[8] podem ser usados. No que ao peso computacional diz respeito, podemos encontrar na literatura

novos algoritmos com o foco em terminais móveis como a criptografia de curva elíptica[6, 20] - *ECC*.

Na solução desenvolvida, sendo que toda a comunicação é efetuada com base no protocolo HTTP, optamos pela utilização de canais seguros via SSL/HTTPS de forma a garantir a privacidade da informação trocada. Existe mesmo trabalho realizado com o objetivo de incorporar o algoritmo de ECC em protocolos como SSL garantindo um melhor desempenho e viabilidade em terminais com menos recursos de processamento, em particular terminais móveis[11].

Com a comunicação a ser realizada de forma segura no que à privacidade da informação diz respeito, outras questões devem ser abordadas. A primeira prende-se com a não possibilidade de acesso direto - aceso via URI da aplicação no servidor,por exemplo - à informação presente no sistema. As aplicações, por exemplo, durante a sua transferência não permitem ataques *man-in-the-midlle* devido à sua passagem num canal seguro. No entanto, deve ser garantido que nem todos os utilizadores possam posteriormente aceder à informação de forma direta. De forma a oferecer essa possibilidade, não existem pontos de acesso diretos às aplicações fora do sistema de gestão. Um utilizador, ao pedir uma aplicação recebe a mesma como corpo binário da resposta e não um URI de acesso posterior. Desta forma, é impossível aceder posteriormente à aplicação no sistema.

Ainda assim, um outro ponto de falha pode ser encontrado: e se existir um ataque por replicação onde o mesmo pedido é executado novamente? De facto, embora não seja possível comprometer a privacidade de um canal de comunicação, nada impede que seja criado um outro canal legítimo com o servidor, sendo depois forjada uma comunicação replicada. Para impedir ataques semelhantes, a nossa solução utiliza mecanismos de *one-time tokens* que são gerados pelo sistema de gestão e enviados de forma segura para o terminal. No limite será gerado um *token* por cada pedido a efetuar pelo terminal móvel.

Por fim, a arquitetura apresenta uma medida que garante a integridade das aplicações devolvidas pelo serviço de gestão. Como foi referido aquando da explicação do fluxo de início de sessão, o utilizador deve escolher um perfil recebendo posteriormente do serviço de gestão uma lista com as aplicações que do perfil fazem parte, de forma a que no melhor momento para o terminal móvel, este faça o pedido das mesmas.

É certo que a informação trocada tem garantias de privacidade, integridade e autenticidade graças ao protocolo SSL utilizado, optamos no entanto por adicionar um outro mecanismo de segurança durante a transferência de aplicações do serviço de gestão para o terminal. O identificador de uma aplicação que é passado ao dispositivo móvel para que este a possa requisitar, não é o nome da aplicação nem um ponteiro com valor meramente interno no domínio do sistema de gestão. Na verdade, o identificador passado é o *hash* da aplicação, o mesmo que é depois calculado no terminal móvel após transferência de forma a garantir sucesso.

Mais do que um mecanismo extra de garantia de integridade da informação, esta solução garante o correto funcionamento do sistema em casos onde, por exemplo, o terminal adie o *download* de uma aplicação por falta de conectividade eminente e essa aplicação tenha sido atualizada durante o tempo que o terminal móvel permaneceu incontactável. Quando voltar a poder comunicar, a tentativa de *download* vai falhar uma vez que a aplicação, ao ser atualizada, terá um novo *hash* - certo que de seguida receberá a mensagem, que estava em fila no serviço de gestão, para atualizar uma aplicação; ainda assim são poupados recursos como comunicação, processamento e bateria.

Na tabela 3.1 é feito um breve comparativo entre os mecanismos de segurança adotados e as vantagens de segurança que estes proporcionam à solução apresentada. Podemos observar que pela utilização de um canal de comunicação seguro, utilizando o protocolo SSL, garantimos os requisitos de segurança primários apresentados no início deste documentos - secção 1.1. Para outras garantias de segurança, os mecanismos apresentados anteriormente apresentam-se como uma solução.

	SSL HTTPS	Aplicação sem acesso direto	*one-time tokens*
Privacidade da informação	X		
Integridade da informação	X		
Autenticidade da informação	X		
Previne - Ataques por repetição	X		X
Previne - Pedido direto de aplicação		X	

Table 3.1: Mecanismos de segurança implementados e mais valias atingidas

3.6 Identificador Único

Embora pareça trivial, a geração de um identificador único para terminais como *smartphones* e *tablets* apresenta algumas dificuldades. Em primeira análise podemos pensar em descartar este problema para identificadores conhecidos, como o **International Mobile Equipment Identity - IMEI**, identificadores de baixo nível como o **CPU id** ou **Media Access Control - MAC address**, ou mesmo utilizar algum identificador fornecido pelo sistema do dispositivo. Estes identificadores apresentam no entanto várias desvantagens que não permitem a sua utilização para identificação unívoca de um terminal. A saber:

- **IMEI** - A principal desvantagem deste identificador reside na sua não existência fora de terminais sem tecnologia GSM. Em terminais com esta tecnologia, que está presente na larga maioria dos terminais móveis atuais, podemos garantir uma identificação com um nível elevado de confiança. De forma geral, todos os fabricantes atribuem um IMEI diferente a cada terminal produzido, não havendo duplicação a nível global. A alteração deste número, embora possível, requer um conhecimento tecnológico acima do normal e não é garantido para todos os terminais. O maior problema reside em terminais como *tablets* que apenas possuem tecnologia de comunicação como *Bluetooth* ou *WiFi*, não tendo um IMEI atribuído. Apesar de serem um número pequeno quando comparado com a totalidade dos terminais móveis possíveis de ser geridos com uma solução deste tipo, apresentam um volume de utilização não desprezável.

- **MAC address** - De forma similar ao que acontece com o IMEI que apenas está presente em terminais com tecnologia GSM, o endereço MAC está presente em terminais com interfaces de comunicação como *Wifi* ou *Bluetooth*. Este endereço identifica a interface de rede de um dispositivo sendo teoricamente único. No entanto, ao contrário do que acontece com o IMEI, o endereço MAC pode ser facilmente alterado, muitas vezes pelo próprio sistema.

- **CPU ID** - Esta informação não está disponível na maioria dos sistemas de dispositivos móveis atuais.

- **Platform provided ID** - Embora esta informação exista de forma totalmente controlada em diversos terminais e sistemas, esse não é o caso típico. Muitos fabricantes acabam mesmo por desenvolver sistemas de geração de identificadores que no limite não são fiáveis, gerando repetições.

Como não existe nenhum identificador que permita a identificação inequívoca de um terminal, a solução adotada converge na geração de um identificador pelo próprio serviço de gestão, passando-o posteriormente

ao terminal. Este identificador não apresenta desvantagens no caso de ser comprometido ou perdido, uma vez que, para o primeiro caso, o sistema pode de forma aleatória questionar o terminal sobre outra informação relevante do mesmo. Caso o identificador seja legitimamente perdido, por limpeza do terminal por exemplo, o sistema pode optar por gerar um novo identificador e associá-lo ao terminal ou reenviar o mesmo identificador, utilizando para isso informação extra passada pelo terminal. Na solução desenvolvida assumem-se os seguintes identificadores extra, sempre que existam, para renegociação do identificador do terminal:

- IMEI
- Fabricante
- Modelo do terminal
- Sistema Operativo
- Versão do sistema operativo

Embora seja possível alterar ou emular, de forma individual, todos os campos indicados, dado que a renegociação se baseia num pedido aleatório, seria necessário obter os valores de todos os campos e alterar-los de forma a executar um ataque bem sucedido.

Capítulo 4

Implementação

Neste capítulo são apresentadas as funcionalidades do protótipo implementado, descrevendo-se posteriormente os detalhes mais relevantes da implementação.

4.1 Funcionalidades

O protótipo desenvolvido implementa um sistema de gestão de terminais móveis que permite a distribuição de aplicações agrupadas em um ou mais perfis, estando estes associados a um ou mais utilizadores.

Na componente servidor da solução, são submetidas aplicações que são agrupadas de forma a criar perfis, sendo estes posteriormente associados a um ou vários cliente(s). É ainda guardada alguma informação identificativa sobre os vários terminais e aplicações geridas pelo sistema, informação esta que permite uma inventariação básica. É ainda gerado de forma constante um *log* da atividade de 'início de sessão' onde fica registado o nome do utilizador que fez *login*, o terminal no qual o fez e a data do processo.

Na componente cliente, as aplicações do perfil indicado pelo utilizador são instaladas nos terminais onde este inicie uma sessão e são posteriormente removidas quando este terminar a mesma. Os perfis podem ser criados/removidos/atualizados em qualquer altura, sendo que um perfil pode estar associado a vários utilizadores. Da mesma forma, um utilizador pode ter vários perfis associados sendo da sua responsabilidade a escolha do desejado na altura em que inicia a sessão.

Foi ainda implementado um mecanismo de atualização automática de perfis ativos. Sempre que um perfil é modificado por adição de aplicações, estas são automaticamente enviadas e instaladas nos terminais que têm o mesmo perfil ativo. A nossa escolha em realizar atualização automática no terminal, apenas quando há adição de aplicações, deve-se à dificuldade de gerir a melhor altura para a sua remoção. Um utilizador pode estar a utilizar a aplicação ou não o estando, pode ter nela informação importante que não deve ser perdida em caso de remoção da mesma. Optamos assim por não proceder à atualização remota nestes casos.

Como referência, o protótipo resultante é direcionado a terminais móveis Android®. No entanto, tanto o componente servidor, o protocolo de comunicação utilizado e a lógica base da solução, não assentam em nenhuma especificidade da plataforma. Desta forma, seria tecnicamente viável a implementação a outras plataformas.

4.2 Cliente

Nesta secção, revemos a implementação do módulo cliente, analisando em detalhe alguns dos seus aspetos mais interessantes.

Este protótipo funcional foi desenvolvido tendo como alvo terminais com sistema operativo Android®.[1] As vantagens desta plataforma que conduziram à sua escolha são:

- Plataforma de código aberto, permitindo total controlo da mesma.
- Desenvolvimento assente na linguagem JAVA SE suportando toda a sua estrutura.
- Grande popularidade da mesma, perfazendo já 50% do mercado de sistemas operativos de *semarphones/tablets*.

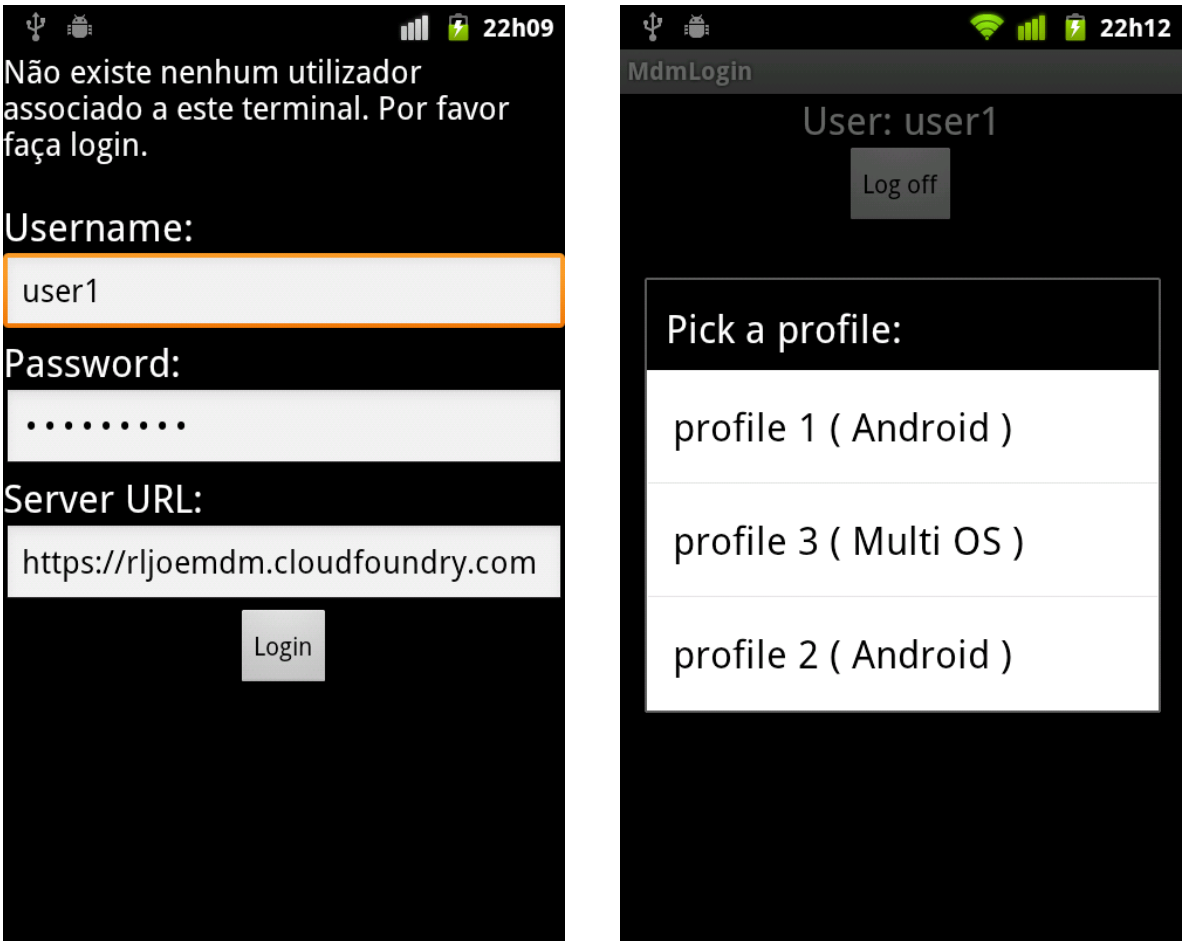


Figure 4.1: Ecrã principal e escolha de perfil - componente cliente do protótipo

Na figura 4.1 podemos observar os dois ecrãs principais do protótipo na versão cliente. À esquerda, o primeiro ecrã apresentado ao utilizador onde este insere os seus dados de utilizador - nome de utilizador e palavra passe - e o endereço do sistema de gestão (parte servidor do SuusMDM). À direita, podemos ver uma escolha de perfil após um início de sessão efetuado com sucesso.

4.2.1 Instalação/Remoção silenciosa

Para que o processo de instalação/remoção de aplicações seja invisível ao utilizador durante o início e término de sessão, é necessário que este seja silencioso. A plataforma Android® apresenta dois métodos oficiais

[1] http://developer.android.com/index.html

para este efeito, sendo possível este comportamento a aplicações instaladas de raiz no sistema ou aplicações assinadas pela Google, serviço que a mesma oferece mediante pagamento. A primeira solução, obriga a uma interação com os fabricantes de terminais, sendo necessário que a imagem de sistema seja alterada - algo usual quando uma empresa adquire vários terminais - tendo, como na segunda solução, custos monetários associados.

A solução adotada, apesar de não ser oficialmente suportada, permite o mesmo funcionamento silencioso. Sendo a plataforma Android® baseada em *linux*, existe a possibilidade de executar comandos e aplicações como *root*. Este processo, apesar de não ser oficialmente suportado, não tem qualquer impacto na garantia dos terminais e é bastante utilizado.

Para a realização desta operação de *root* do terminal foi utilizada uma versão não oficial do sistema Android®, modificada pela comunidade de utilizadores, a qual tem o nome de CyanogenMod.[2] Esta imagem de sistema, é totalmente compatível com as restantes versões, apresentando apenas melhorias ao nível do suporte de APIs de versões mais recentes do sistema operativo, muitas vezes não presentes em terminais de vários fabricantes por objetivos de *marketing*. Tem ainda a mais valia de trazer acesso *root* para as aplicações que assim o desejarem sendo da responsabilidade do utilizador a sua permissão.

Desta forma, caso a aplicação seja instalada num terminal com o sistema *vanilla*, este necessita de passar por um processo de obtenção de *root* para que a solução apresentada possa ser executada.

Foram ainda realizados alguns testes em terminais que não possuíam acesso *root* mas aos quais tínhamos contacto com o fabricante de forma a integrar a aplicação protótipo desenvolvida como parte integrante do sistema. Os resultados foram em tudo idênticos.

4.2.2 Mecanismo *Ring-home*

O mecanismo que permite dar a conhecer ao servidor de gestão qual o endereço IP do terminal, foi implementado nesta solução recorrendo à ferramenta de *intent-filters* disponibilizada pela plataforma Android®.

Esta ferramenta permite associar a qualquer aplicação mecanismos de escuta de informação indicada pelo sistema. No caso particular desta implementação, foi criado um serviço que arranca aquando da inicialização do sistema e filtra informação passada pelo *ConnectivityManager* do sistema, em concreto, chamadas *call-back* do tipo *CONNECTIVITY-ACTION*. Estas são geradas pelo próprio terminal quando:

- A interface de rede muda (WiFi, GPRS);
- Quando a mesma interface muda o seu endereço de rede.

Desta forma, uma aplicação que registe interesse nestas chamadas, é sempre alertada para mudanças de endereço de rede, sem necessidade de inquérito constante ao sistema, poupando processamento e bateria.

Na implementação realizada, a aplicação regista este interesse e sempre que recebe informação de alteração de mudança de endereço de rede, adquire-o e caso este exista - uma desconexão também gera uma chamada deste tipo, devolvendo um endereço de rede vazio - envia-o para o servidor de forma que este possa sempre comunicar com o terminal.

4.2.3 Paradigma *Push*

Para permitir uma comunicação entre o sistema de gestão e o terminal móvel, inicializada pelo servidor, foi implementado um mecanismo de escuta passivo baseado na tecnologia *Java New I/O*, em particular uti-

[2] http://www.cyanogenmod.com/

lizando a plataforma Netty.[3]

Um dos problemas do desenvolvimento para terminais móveis prende-se com o poder de processamento dos mesmos e a sua ligação direta ao consumo de energia. Neste caso particular, a tecnologia *Java New I/O* apresenta larga vantagem ao utilizar a mesma memória física para os *buffers* de I/O que o sistema operativo onde executa. Desta forma, não existe a necessidade de cópias extra para efetuar operações de escrita ou leitura não sendo necessário processamento específico pela aplicação. Da mesma forma, liga-se aos sockets de escuta do sistema onde executa, não necessitando de permissões extra ou processamento para manter uma conexão aberta. Por último, toda a lógica da mesma é assíncrona, permitindo execução paralela e não bloqueante com os restantes operações do sistema ou aplicação.

A plataforma Netty é inteiramente baseada em *Java New I/O*, implementado-a e englobando posteriormente mecanismos de comunicação com diversos protocolos (HTTP, RTSP, etc) e rotinas de uso habitual no desenvolvimento de aplicações nesses protocolos, que são oferecidas na forma de APIs de alto nível, tornando mais fácil e otimizado o desenvolvimento.

Para a implementação, foi utilizada a *framework* Netty de forma a criar um canal de escuta sempre ativo de forma a permitir a comunicação inicializada através do serviço de gestão. Na prática, este canal não recebe toda a informação mas apenas avisos da sua existência, com métodos de identificação da mesma, de forma que o próprio cliente, após verificar que a informação é legitima e relevante, realiza o pedido ao sistema de gestão.

Desta maneira conseguimos garantir alguma segurança do lado do terminal, uma vez que este não vai executar tudo o que lhe for enviado, para o canal de comunicação que está sempre aberto. Tal seria um foco importante de insegurança. E por outro lado permite implementar estratégias adaptativas de transferência de informação. O cliente pode saber que irá perder acesso à rede dentro de um limite de tempo curto e adia a transferência para depois. Será muito mais difícil, ou até impossível, que o servidor consiga inferir cenários destes.

4.3 Servidor

Nesta secção, revemos a implementação do módulo cliente, analisando em detalhe alguns dos seus aspetos mais interessantes.

No servidor foi utilizada a linguagem Groovy[4] assente na *framework* Grails.[5] As mais valias apresentadas que resultaram na sua escolha são:

- Linguagem de semântica livre assente em JAVA.
- Forte suporte à criação de *web-services*.
- Orientada ao desenvolvimento de aplicações *web*.
- Implementação nativa da interface *Model-View-Controller*.
- Assenta no paradigma 'Convenção sobre Configuração'.
- Facilidade de integração com várias tecnologias de bases de dados.
- Facilidade do desenvolvimento da interface gráfica através da técnica de *scaffolding*.

Exportado como pacote WAR - *Web application ARchive* - o servidor pode ser corrido em qualquer servidor aplicacional JAVA.

[3] https://netty.io/
[4] http://groovy.codehaus.org/
[5] http://grails.org/

4.3.1 Mecanismos de Base de Dados

Para o desenvolvimento deste protótipo funcional, foi utilizado a tecnologia *H2 Database*. Este sistema relacional de gestão de base de dados permite a não preservação da informação, funcionando sobre o paradigma de *base de dados em memória*.[6]

Foram vários os motivos que levaram à sua escolha. Por um lado, sendo um protótipo largamente em âmbito de desenvolvimento/teste, não era necessária a preservação da informação por longos períodos de tempo ou em caso de encerramento do sistema. Podemos até argumentar que o mesmo não é desejável durante a fase de constante evolução uma vez que permite um ambiente sempre limpo e sem fragmentos resultantes de execuções anteriores. Por outro lado a solução *H2 Database* permite, caso seja desejável, um esquema de salvaguarda de informação em disco.

Por outro lado, a plataforma de desenvolvimento do módulo servidor, a *framework* Grails, tem suporte nativo para este sistema de gestão de base de dados, sendo trivial uma passagem posterior para outros sistemas sem necessidade, para a maioria das solução de gestão de base de dados, de refactorização do código desenvolvido.

Sobre a informação guardada na base de dados, a implementação SuusMDM utiliza o modelo de domínio apresentado no capítulo 3. Foram geradas tabelas na base de dados de acordo com o apresentado na secção 3.2. No anexo A pode ser visto em detalhe a estrutura da base de dados utilizada bem como o tipo de dados de cada campo.

4.3.2 *Deploy* do Sistema de Gestão

A componente servidor do SuusMDM, tendo sido desenvolvida em Grails, assenta na plataforma Java suportando todos os seus mecanismos de exportação e *deploy*.

Acabamos por escolher empacotar toda a lógica do sistema de gestão num ficheiro WAR - *Web application ARchive*[7]. Desta forma garantimos simplicidade de *deploy* em qualquer servidor aplicacional Java com suporte a esta estrutura. Após alguns testes em máquinas pessoais em ambientes restritos e controlados, a versão final foi posta em funcionamento utilizando o servido de *hosting* distribuído *Cloud Foundry*.[8]

Figure 4.2: Ecrã principal do sistema de gestão - componente servidor do protótipo

A interação faz-se através de um *browser* sem nenhuma necessidade especial exceto que siga as normas do protocolo HTTP versão 1.1. Graficamente apresenta, na página principal do lado esquerdo, um menu que permite o acesso a todos os nós base da solução - terminais, aplicações, perfis e acessos como se pode

[6] De notar que este sistema também permite bases de dados guardadas em disco de forma a preservar a informação.
[7] http://java.sun.com/developer/technicalArticles/Servlets/servletapi/
[8] http://cloudfoundry.com/

ver na figura 4.2.

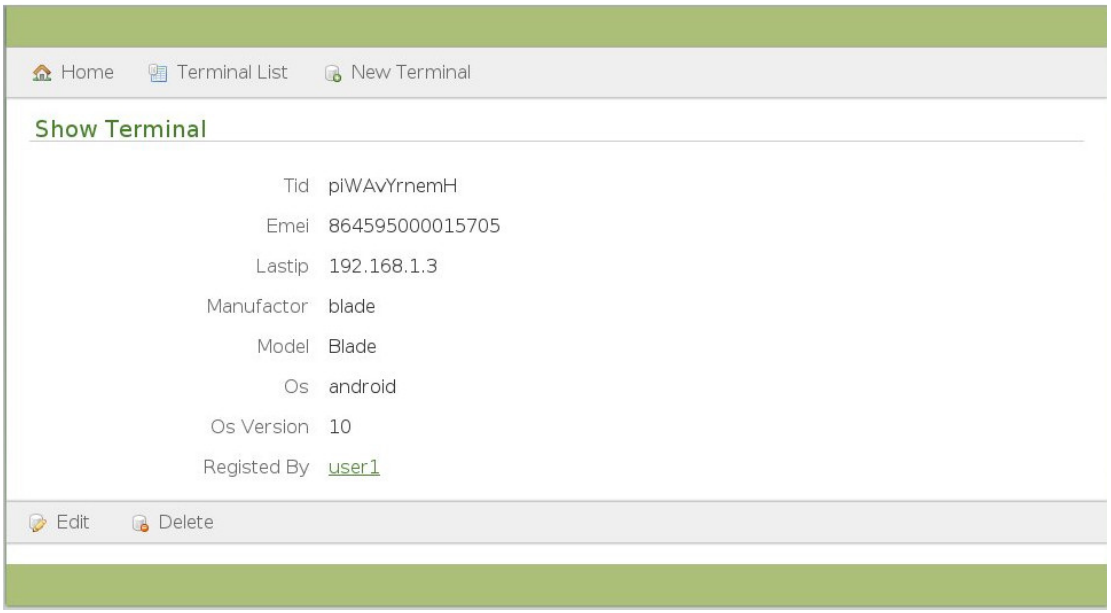


Figure 4.3: Ecrã principal do sistema de gestão - componente servidor do protótipo

Dentro de cada um destes a interface é sempre idêntica permitindo a listagem da informação de forma detalhada ou em grupo, e a criação, remoção e alteração da mesma exceto no caso da listagem de acessos que não podem ser manualmente adicionados, removidos ou alterados. Na figura 4.3 podemos ver a página de informação de um terminal sendo visíveis os acessos para a alteração da informação do mesmo.

4.3.3 Segurança

De forma a garantir a segurança da informação partilhada entre o terminal cliente e o servidor do serviço de gestão, toda a comunicação é efetuada sobre o protocolo *HTTPS* sendo a garantia de privacidade da informação oferecida pelo protocolo base *SSL/TLS*. Neste ponto, foi utilizado o próprio certificado do *Cloud Foundry* onde o serviço de gestão era executado.

Para a geração dos *tokens* de garantia de não repetição foi utilizado algoritmos de hash suportados em MD5 segundo a biblioteca *java.security.MessageDigest* disponibilizada pela Oracle na sua plataforma Java e com integração direta na *framework* Grails.

Ainda num aspeto relacionado, foram implementadas regras no que ao tamanho e criação de credenciais de utilizador e *tokens* dizem respeito. A saber, as palavras passe dos utilizadores devem ter no mínimo 5 caracteres, sendo o tamanho mínimo do nome de utilizador 3 caracteres. Já os *tokens*, são um representação de 10 caracteres alfanuméricos maiúsculos e minúsculos existindo diferenciação entre os mesmo.

Capítulo 5

Avaliação

Neste capítulo apresentamos uma análise do protótipo desenvolvido, comparando-o na secção 5.1 de forma qualitativa com outras soluções existentes. Ainda nesta secção serão apresentados os resultados dos inquéritos realizados aos utilizadores que testaram o nosso protótipo. De seguida, na secção 5.2, é apresentada uma análise quantitativa segundo vários índices de interesse.

5.1 Avaliação Qualitativa

O sistema SuusMDM é uma solução dentro do campo, em forte crescimento, que são os sistemas de gestão de terminais móveis.

Devido ao tipo de terminais e à sua relativa imaturidade como ferramenta de trabalho, as soluções existentes apoiam-se nas homologas soluções direcionadas a terminais "clássicos" - como os computadores *desktop* - não divergindo de forma relevante quer nas possibilidades apresentadas quer nos mecanismos que as suportam. Por estes motivos a avaliação seguinte não é direcionada a nenhum sistema em especifico.

O SuusMDM tenta dar um passo em frente no que respeita a funcionalidades oferecidas, tentando integrar mais o utilizador e o terminal numa gestão bidirecional. Contudo, foram tidas em atenção os requisitos não funcionais, que são avaliados na secção 5.2 de forma quantitativa, mas que do ponto de vista qualitativo não apresentam diferenças significativas.

De seguida apresenta-se uma análise qualitativa entre a solução desenvolvida neste documento e as as soluções apresentadas anteriormente - de forma global.

5.1.1 Registo Iniciado pelo Terminal

Um dos aspetos transversais em todas as soluções apresentadas anteriormente, reside na necessidade de registo do terminal no sistema de gestão, atividade efetuada do lado do servidor. Em algumas delas - todas as soluções baseadas no *OMA DM* como o projeto *Funambol* - não existe mesmo a capacidade de interação inicializada pelo terminal móvel.

Esta modalidade, embora apresente algumas medidas de segurança inerentes à interação pessoal, torna-se limitativa em cenários onde existam muitos terminais a gerir e onde os terminais e seus utilizadores não tenham contacto direto com o serviço de gestão.

O SuusMDM, permite que qualquer terminal se registe no sistema a partir dele próprio desde que para isso o utilizador que o pretende registar esteja ativo no sistema de gestão. Desta forma, diminuímos a componente humana necessária apenas para gerir uma solução deste tipo e permitindo uma maior liberdade física

na sua utilização, aspeto importante em cenários de mobilidade tecnológica.

Com esta opção a solução SuusMDM acaba por passar a segurança para o lado do utilizador ao invés da habitual centralização pelo responsável do sistema. Quando um terminal é registado, é guardado o identificador do utilizador que o registou ou um identificador genérico configurável caso tenha sido registado diretamente. Desta forma, problemas futuros causados, por exemplo, por acessos indevidos, ficam sempre associados ao utilizador que deu acesso ao terminal.

Note-se que esta abordagem permite de igual forma o registo direto no sistema e, caso seja efetuada de forma remota, oferece necessidades de gestão de políticas de utilizadores não retirando na totalidade a componente humana de gestão. Ainda assim, permite novos cenários de uso de valor acrescentado em contextos móveis onde não existe uma proximidade física adequada para um registo tradicional. Podemos pensar em cenários onde, por exemplo, um colaborador de uma instituição que utilize a solução SuusMDM esteja em viagem e adquira um novo terminal. Existindo uma distância física e não sendo desejável a transmissão oral dos dados do novo equipamento - por questões de segurança - a solução apresentada permite contornar este cenário, algo não permitido, de forma mais ou menos direta, pela totalidade das soluções apresentadas.

5.1.2 Sessões & Perfis

A opção de iniciar uma sessão num terminal móvel é algo que, até ao momento, é novo no mercado. Uma solução como a desenvolvida por nós permite quebrar de forma efetiva a ligação forte que existe entre o terminal e o seu proprietário. Permitindo que qualquer utilizador utilize qualquer terminal com o mesmo contexto de sistema pessoal, os dispositivos deixam de ser o principal foco de atenção - requerendo obviamente terminais capazes de suporte às atividades a realizar

Esta nova abordagem permite mesmo a evolução de modelos como o muito utilizado *Bring Your Own Device* - BYOD. Neste caso particular, podemos imaginar cenários onde o modelo pode ser estendido de forma a simplifica-lo ou pode mesmo destruí-lo, uma vez que os terminais podem ser "reconfigurados" segundo as exigências dos utilizadores de forma fácil e rápida.

De todas as soluções apresentadas, talvez a mais próxima deste modelo seja o Android® Market Webstore. Embora não seja uma solução MDM na sua essência, apresenta muito aspetos de sistemas análogos. No entanto, permite que a instalação remota de aplicações seja direcionada não a um terminal especifico, mas a conjuntos terminal/utilizador. O simples *login* com credenciais Google num terminal Android® desencadeia uma associação do lado do sistema de gestão da *Market Webstore* que permite posteriormente, enquanto a sessão estiver ativa, gerir aplicações no terminal alvo apenas com base no utilizador.

Os restantes sistemas seguem uma abordagem diferente onde a gestão é feita diretamente ao terminal e não ao utilizador. Embora seja possível a criação de grupos de terminais em soluções como no BlackBerry® Enterprise Server, estes apenas permitem que uma ação executada pelo serviço de gestão seja automaticamente replicada pelos vários terminais que compõem o grupo.

5.1.3 Paradigma *Push*

Uma das decisões arquiteturais da nossa solução, baseada em requisitos como a poupança de recursos como processamento, comunicação e inevitavelmente bateria, é a possibilidade de comunicação de forma inicializada pelo sistema de gestão e direcionada ao terminal móvel. Para tal, a decisão de implementar mecanismos suportados pelo paradigma *push* mostra-se bastante valiosa.

Analisando as restantes soluções apresentadas neste documento no Capítulo 2, podemos verificar que esta abordagem é algo disseminada. Todas as soluções permitem de uma maneira mais ou menos transparente

o envio direto de informação para o terminal. Existem no entanto alguns pontos nos quais a nossa solução apresenta vantagens:

- **Serviço proprietário -** em algumas soluções, o mecanismo que permite a implementação do modelo *push* é suportado por sistemas já existentes. Um exemplo prático é o sistema C2DM da Google [1] que, utilizando toda a infraestrutura já existente para os seus serviços em terminais, fornece o serviço de forma controlada a terceiros. Neste caso, dois problemas surgem imediatamente, sendo o primeiro a limitação não a terminais com sistema Android, mas a terminais dessa plataforma e com os serviços da Google suportados e ativos. Existem, por outro lado, problemas resultantes de questões de controlo. A informação é sempre processada em serviços Google o que levaria a mais mecanismos de segurança de forma a proteger a privacidade da informação. Por outro lado, caso os serviços sejam interrompidos ou cancelados, todas as soluções nele assentes deixam de ser funcionais.
  A nossa solução elimina esta ligação a mecanismos de terceiros, sendo totalmente direcionada e suportada pela entidade da qual faz parte.
- **Envio direto da informação -** Muitas das primeiras soluções a implementar o modelo *push* seguem uma lógica de envio direto da informação na sua totalidade. A nossa arquitetura assenta numa nova lógica de funcionamento que começa também a ser utilizada em projetos mais recentes.
  O envio não da informação na totalidade mas de *notificações* da existência de novidades permitem os ganhos atingidos pelo não contacto constante poupando recursos, assim como delegam algum poder de decisão ao próprio terminal para que gira as ações a desenvolver segundo o contexto onde se insere.

5.1.4 Testes com Utilizadores

Vamos aqui mostrar uma análise de resultados aos testes realizados com utilizadores de forma a perceber como estes interagem com uma solução como a apresentada bem como o seu nível de adaptação à mesma.

A seguir são apresentadas as estatísticas da análise de resultados do inquérito feito aos utilizadores, após uma execução padronizada do sistema. O inquérito pode ser encontrado no Anexo B deste documento, sendo o fluxo de teste requerido aos utilizadores disponibilizado no Anexo C. Uma representação gráfica da totalidade das repostas pode ser encontrada no anexo D.

O contexto tecnológico onde no qual os testes se inseriram é o seguinte:

- O sistema de gestão encontra-se em execução num serviço de *deploy* de aplicações na nuvem externo a nós, **CloudFoundry**.[2]
- Quando possível, técnica e autorizadamente, foram utilizados os terminais móveis dos utilizadores que realizaram os testes. Quando tal foi impossível, foi utilizado o terminal *ZTE Blade* versão Portuguesa TMN Sapo A5 com um processador ARM single-core a 600 MHz e 420Mb de ram útil. O sistema operativo instalado foi o Android 2.3 versão cyanogen Mod 7 que oferece permissões de *root* ao utilizador.
- Os testes foram sempre realizados, do lado do terminal, através das comunicação de dados móveis existentes 3G.

Análise de Resultados - Participantes

Os testes foram realizados numa amostra de 27 participantes de ambos os sexos com idades compreendidas entre os 24 e os 56 anos. Destes 14.8% utiliza terminais móveis como *smartphones* diariamente e de forma intensiva sendo a restante percentagem - 85.2% - utilizadora diária mas não intensiva. Ainda da totalidade da amostragem, 66.6% dos participantes está inserido profissional ou academicamente num contexto ligado às tecnologias, sendo que apenas 25.9% utiliza este tipo de terminais no desempenho das suas funções.

[1] https://developers.google.com/android/c2dm
[2] http://cloudfoundry.org/

Análise de Resultados - Interação Terminal

Uma das opções da realização dos testes era a possibilidade do utilizador realizar os mesmo no seu próprio terminal, caso este fosse Android, apresentasse conectividade de rede e, como explicado na secção 4.2.1, apresentasse acesso *root* às aplicações instaladas. Este cenário aconteceu apenas 4 vezes e sempre em terminais de proprietários que têm a sua atividade profissional ligada ao desenvolvimento de tecnologias para o sistema Android. Podemos concluir assim que um sistema como o desenvolvido apenas terá sucesso prático em ambientes onde a aplicação seja integrada no próprio sistema.

Um aspeto interessante para nós foi a avaliação da própria aplicação cliente no que respeita à sua utilização. A generalidade dos utilizadores não tiveram qualquer problema na sua utilização tendo reportado como mais valia a sua simplicidade. Tratando-se de um protótipo não demos muita atenção ao grafismo da aplicação. Embora achássemos que tal seria penalizador durante a interação dos utilizadores, tal não foi o caso. A interface simples mas familiar acabou por se tornar uma mais valia permitindo uma utilização fácil e com um nível de distração mínimo durante a utilização do dispositivo.

O único aspeto avaliado por nós no qual as críticas foram menos boas prende-se com o tempo necessário para a inicialização de uma sessão. Do ponto de vista técnico tal deve-se ao facto deste processo englobar o *download* e instalação das aplicações no terminal. A primeira parte deste processo será tão mais demorada consoante o número e tamanho das aplicações associadas ao perfil. A segunda parte está diretamente ligada ao número de aplicações uma vez que no sistema alvo do nosso protótipo não é possível instalar várias aplicações em simultâneo.

No geral ficamos satisfeitos com a aceitação da aplicação cliente por parte dos utilizadores tendo sido comprovadas as limitações do protótipo em sistemas reais.

Análise de Resultados - Interação Sistema de Gestão

Do ponto de vista dos utilizadores, o sistema de gestão não apresentou nenhuma dificuldade de utilização. As funcionalidades oferecidas foram facilmente identificadas e utilizadas.

Numa análise posterior podemos no entanto concluir que a maioria dos utilizadores que efetuaram os testes e que de alguma forma interagem de forma regular com sistemas informáticos móveis num contexto profissional, deu uma menor cotação à satisfação que apresentam sobre a informação dada pelo sistema sobre cada terminal. Como esta análise foi feita posteriormente, não nos foi possível confrontar os utilizadores sobre o porquê das suas respostas, no entanto podemos imaginar que tal se deva à experiência adquirida pelos mesmos durante diversos casos de uso sobre outros sistemas onde a informação por nós adquirida não é suficiente.

Análise de Resultados - Implementação Real

O último bloco de perguntas direcionadas aos utilizadores que testaram o nosso protótipo, tenta medir a validade e aceitação da solução proposta. Mais do que a nossa implementação, interessava compreender o alcance deste novo modelo associado à utilização de terminais móveis, neste caso particular familiares à maioria das pessoas, como os *smartphones*.

As respostas à pergunta **Já tinha tido alguma experiência com sistemas deste tipo?** permitiram-nos compreender que a utilização de terminais móveis como *tablets* e *smartphones* começa a ser já usual em vários cenários, sendo o seu valor reconhecido pela utilização de sistemas MDM como se pode verificar pela percentagem de 55.5% de utilizadores que referem já ter tido contacto com sistemas MDM (percentagem de pessoas que indicou um nível de confiança superior a 5 numa escala de 0 a 8).

Curiosas foram as respostas à questão seguinte **Conhece outras soluções semelhantes?** onde apenas uma percentagem mínima dos utilizadores afirmam, ainda que com um grau de certeza muito reduzido, conhecer soluções análogas. Segundo nos foi transmitido durante a realização dos testes, a nossa solução apresenta uma abordagem realmente nova à utilização de terminais geridos de forma centralizada.

Foi ainda interessante notar que a maioria dos utilizadores que contribuíram para este teste vêm o futuro da nossa solução de duas maneiras, esta as suas respostas diretamente ligadas ao seu contexto profissional. A totalidade dos utilizadores que não apresentam um contexto profissional diretamente ligado às tecnologias e em particular a terminais móveis, admite que encontrar utilidade no seu meio profissional para um sistema análogo, mas não fora deste âmbito. Já grande parte dos utilizadores que se inserem num outro espectro profissional encontram mais valias numa utilização pessoal.

Por fim podemos observar um ponto transversal a todos os participantes neste teste que assenta numa perspetiva positiva de habituação a um sistema destes embora existam ainda alguns receios ao nível da segurança oferecida.

No geral concluímos que a nossa solução apresenta uma componente de inovação forte e possível de ser aproveitada, existindo interesse geral na sua utilização. Para que tal aconteça achamos que devem no entanto existir uma evolução na forma de contacto que existe hoje com os terminais alvo, permitindo uma nova visão sobre os mesmo, as suas capacidades e especificidades com o intuito de os tornar uma efetiva ferramenta de trabalho disseminada de forma transparente.

5.2 Avaliação Quantitativa

Nesta secção vamos avaliar alguns resultados obtidos em testes de contexto quantitativo, organizados em testes realizados por nós, com o intuito de avaliar requisitos de base tecnológica e testes realizados com utilizadores de forma a avaliar a interação com o sistema SuusMDM assim como a aceitação geral da solução.

5.2.1 Contexto Avaliativo

Para avaliar o sistema qualitativamente, simulamos cenários típicos de utilização de terminais móveis registados no SuusMDM e a realizarem uma utilização esperada. Foram avaliados os índices habituais de utilização de recursos como o processamento, a memória e o tráfego de rede gerado. Estes índices foram avaliados tanto do lado do servidor como do lado do cliente - terminal móvel - tendo sido neste último avaliado o desempenho da bateria dos mesmos quando inseridos na nossa solução.

Para a realização do testes a seguir apresentados, optamos por um contexto tecnológico configurado como se descreve de seguida:

- O sistema de gestão encontra-se em execução numa máquina dedicada por um pentium i7-2760QM e 4Gb de ram, a correr o sistema GNU/Linux Ubuntu 12.04LTS (64-bit).
- Foi utilizado o terminal *ZTE Blade* versão Portuguesa TMN Sapo A5 com um processador ARM single-core a 800 MHz e 420Mb de ram útil. O sistema operativo instalado foi o Android 2.3 versão cyanogen Mod 7 que oferece permissões de *root* ao utilizador.
- Foram utilizados diversos emuladores do sistema Android com a versão 2.3 do sistema sempre que um número elevado de terminais seja necessário.
- Os testes foram realizados, do lado do terminal, através das comunicação de dados móveis 3G existentes em Portugal. Quando tal não foi o caso, o mesmo é indicado.

- As aplicações variam em tamanho de dados, sendo esta informação indicada quando relevante.
- Foi gerado um certificado auto assinado de forma a garantir comunicação segura via SSL entre o terminal e o sistema de gestão.

5.2.2 Testes de Carga - Número de Terminais

Um dos requisitos principais da nossa solução, até pelos cenários empresariais alvo, consiste numa boa escalabilidade.

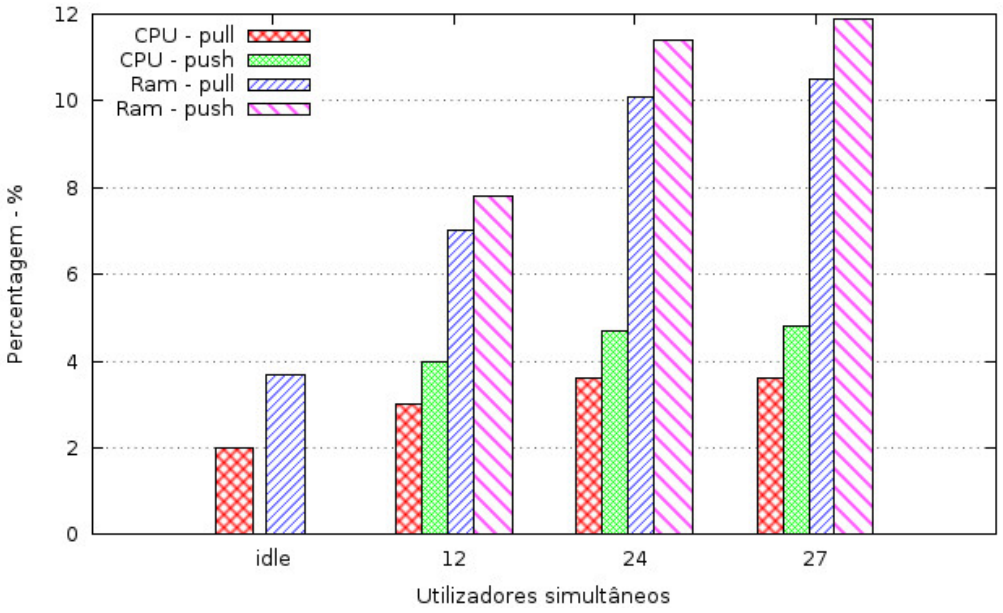


Figure 5.1: Utilização de processador e memória no sistema de gestão com o aumento do número de terminais

A solução desenvolvida é capaz de suportar um número crescente de terminais geridos sem que para isso sejam necessárias quaisquer alterações dos mecanismos internos. É, no entanto, esperado que possa ser necessário replicar a informação guardada bem com adicionar nós de mecanismos de base de dados para suportar um número crescente de dispositivos geridos.

A aquisição de vários terminais móveis para a realização destes testes não era prática quer do ponto de vista financeiro, quer do ponto de vista de execução sendo necessário um grande número de recursos humanos para assistir à realização dos testes.

Para contornar este problema, foi utilizado o emulador da plataforma. Este faz parte integrante do sistema de desenvolvimento do sistema embora permita uma execução independente. Mais anda, permite um maior controlo sendo mesmo possível automatizar as ações realizadas.

Desta forma, foram utilizadas duas máquinas de suporte compostas cada uma por um processador i7-2760QM, com 16Gb de ram a correr o sistema operativo Ubuntu 10.04, cada uma, no qual foi possível executar por máquina 12 entidades do emulador em simultâneo. Durante a execução dos testes, foram ainda adicionados três terminais reais à experiência. O fluxo do teste foi o descrito de seguida:

- Nenhum dos terminais estava registado no sistema de gestão, nunca tendo interagido com o mesmo.

- Todos os terminais foram registados simultaneamente, exceto os três terminais físicos.
- Existem dois perfis, ambos com 3 aplicações sendo uma delas repetida nos dois. Os terminais foram divididos tendo 50% escolhido o primeiro perfil e os restantes o segundo.
- Depois de todos os terminais apresentarem sessão ativa, foram atualizados ambos os perfis tendo sido adicionada uma aplicação diferente em cada um e uma aplicação partilhada.

A figura 5.1 apresenta os resultados médios da utilização de processador e memória ram da máquina descrita em cima, onde o sistema de gestão da solução apresentada estava em execução.

Podemos verificar que não existem alterações fora do esperado na utilização de ambos os recursos. De facto, o processamento realizado pelo sistema de gestão é bastante limitado tendo como principais motivos de utilização a geração de *tokens* e referências de aplicações, ambas utilizando MD5 e a gestão das ligações. Podemos assim perceber o porquê de um aumento deste recurso sempre que uma interação do tipo *push* acontece. No nosso sistema, a comunicação iniciada pelo serviço de gestão surge quando um perfil foi atualizado, sendo necessário: verificar que aplicações foram adicionadas ou atualizadas no mesmo, gerar *tokens* para as aplicações novas a instalar e por cada terminal onde exista uma sessão ativa com o perfil em causa e por fim, estabelecer contacto com os terminais.

No caso da utilização de memória devemos ter em mente um aspeto específico da nossa implementação. Como indicado no Capítulo 4, o nosso protótipo utiliza um modelo de bases de dados não persistentes, sendo toda a informação guardada em ram enquanto o serviço estiver ativo. Desta forma, percebe-se o aumento linear quando se passa de 12 para 24 terminais. É também este um fator que leva ao maior consumo quando utilizando *push* tal como no caso do consumo de processamento, devido à geração de *tokens* por exemplo, que têm de ficar em memória.

5.2.3 Testes de Carga - Recursos Móveis

Sendo que a nossa solução abrange dois módulos, o sistema de gestão e os terminais móveis, vamos aqui testar as possíveis perturbações que possam ocorrer nos dispositivos que executem a aplicação cliente.

Na figura 5.2 podemos ver os recursos de processamento e memória ram para 4 tipos de utilização que se descrevem de seguida. Em **idle-std**, o terminal não possuía qualquer ligação com o sistema de gestão da solução, não tendo mesmo a aplicação cliente instalada, tendo sido ligado com a bateria totalmente carregada e deixado ligado mas sem interação até ter apenas 10% de bateria. Uma aplicação instalada no telefone era acordada pela informação de sistema quando o nível de carga era o 10% e registava-o num ficheiro para visualização posterior. Em **idle-reg** o mesmo teste foi realizado mas com a aplicação cliente do SuusMDM instalada e com sessão iniciada, tendo sido realizadas atualizações no sistema de gestão de forma a que este interagi-se com o terminal. Os casos **work-std** e **work-reg** definem os mesmo testes mas num terminal utilizado diariamente em funções habituais do dia a dia - realização de chamadas, envio de SMS e acesso à Internet com sincronização de email. Os tempo total que a bateria demorou a descarregar até ao nível de 10% pode ser visto na figura 5.3. Os valores apresentados em ambos os gráficos referem-se à media de observações nos 3 terminais físicos utilizados durante os testes, tendo os valores da figura 5.2 sido obtidos de forma aleatória mas em simultâneo nos 3 terminais. Ainda neste último caso, as observações realizavam-se em períodos contínuos de 10 segundos.

Como esperado, os dois primeiros cenários de teste apresentam maior duração de bateria bem como um menor consumo de processamento e memória ram, no entanto podemos observar que não existem diferenças extremas a quando da inserção do dispositivo no nosso sistema de gestão em nenhum dos cenários de teste. Um dado que pode parecer curioso encontra-se na duração da bateria dos terminais no modo **work-std**, modelo onde o terminal é utilizado normal e diariamente, que apresenta um nível marginalmente inferior ao deduzido no modo **work-reg**, modo onde o terminal efetua as mesmas operações mais as resultantes

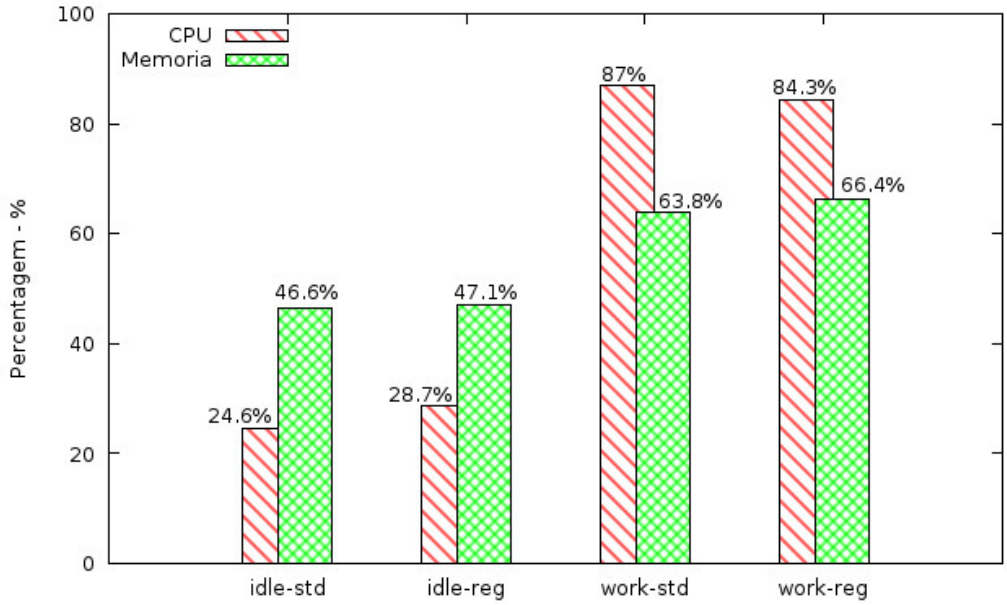


Figure 5.2: Utilização de cpu e ram dos terminais móveis. O índice é a percentagem do total utilizado

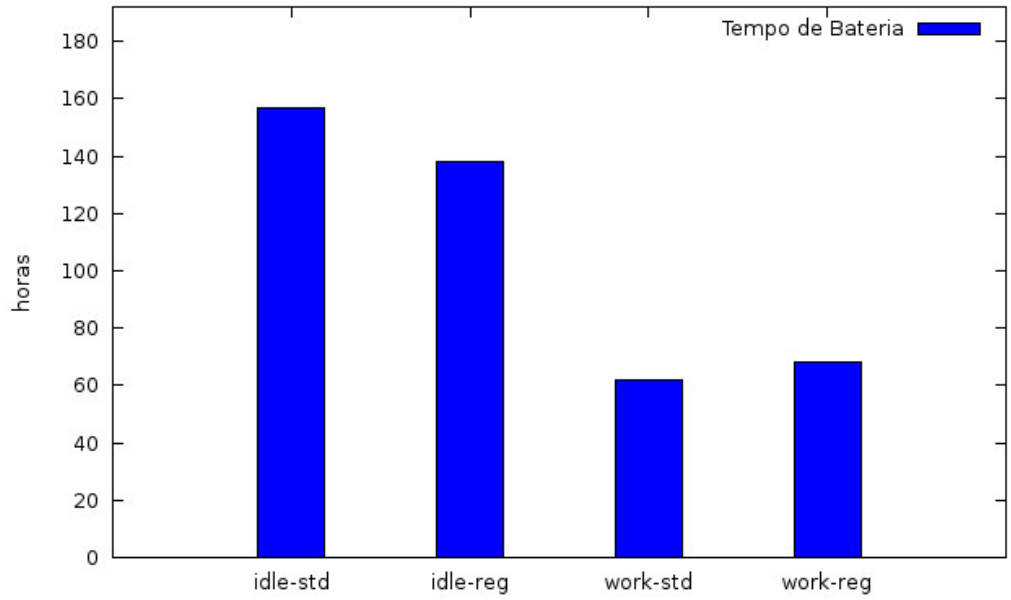


Figure 5.3: Utilização de bateria dos terminais móveis integrados na solução

da inserção no sistema de gestão. A conclusão a que chegamos deriva da subjetividade dos testes nestes dois modelos. De facto, não foi possível replicar na totalidade uma *utilização diária*. É provável que tenha existido ligeiras diferenças de utilização que alterem os dados recolhidos. Ainda assim, podemos concluir que as perturbações resultantes da utilização dos dispositivos no nosso sistema são desprezíveis.

5.2.4 Análise de Tráfego Gerado

Neste teste o objetivo é apresentar um comparativo entre o tamanho da informação trocada pela nossa solução e a informação trocada pelo sistema Android® Market Webstore. Embora não seja um sistema de gestão de terminais móveis, a loja de aplicações da Google para sistemas Android apresenta-se como um bom candidato a um teste comparativo por três motivos:

- É direcionado a terminais Android, os mesmo suportados pelo nosso protótipo.
- Tem como principal atividade o envio e instalação de aplicações em terminais móveis.
- Apresenta uma lógica *push* com o devido aprovisionamento do terminal.

Ainda assim, existem algumas limitações tecnológicas que impedem uma análise exata da informação transferida pela solução *Market Webstore*. A ideia seria capturar o tráfego utilizando uma *proxy* controlada por nós, à qual os terminais se ligariam e através da qual acederiam aos serviços do *Market Webstore*. Se na nossa solução tal é possível na totalidade, o sistema Android apresenta mecanismos de segurança extra que de alguma forma bloqueiam o acesso quando este não é feito diretamente pelo dispositivo. O único tráfego que conseguimos captar e analisar prende-se com a informação trocada pelo terminal para acesso ao serviço e pesquisa de aplicações. Apenas foi possível observar que também nesta solução todo o tráfego assenta no protocolo SSL de forma a garantir segurança na informação trocada. Mais ainda, foi observado que o terminal transmite constantemente e que pela análise dos pacotes cifrados foi possível encontrar *tags* de certificados Google o que indica uma provável comunicação com o mesmo serviço. No entanto, devido a todas as ferramentas existentes não foi possível concluir que esta informação era necessária ou mesmo direcionada ao funcionamento do serviço de obtenção e instalação de aplicações.

	App1	App2	App3
Tamanho do pacote da aplicação	42 Kb	938.7 Kb	8.5 Mb
Informação trocada para obter a aplicação (HTTP)	45.2 Kb	941.03 Kb	8.7 Mb
Informação trocada para obter a aplicação (HTTPS)	49.09 Kb	948.14	9.2 Mb

Table 5.1: Tráfego gerado pelo SuusMDM e terminal móvel

Ainda que não seja possível uma comparação direta entre soluções, podemos estudar o tráfego gerado pela nossa solução. Foram utilizadas 3 aplicações a serem instaladas no terminal. Como se pode ver na tabela 5.1, a informação trocada de forma a obter aplicação a instalar apresenta um aumento desprezável em relação ao tamanho do próprio ficheiro da aplicação. Utilizando mecanismos de compressão de dados, seria ainda possível reduzir estes valores. Quando a aplicação não é requerida pelo terminal mas apresentada pelo sistema de gestão devido a uma instalação ou atualização da mesma no perfil ativo, existe um fluxo extra de informação associado ao envio do alerta por parte do sistema de gestão sobre o modelo *push* como apresentado na secção 3.3.4.

Uma análise mais profunda revelou ainda outro aspeto curioso da implementação da solução da Google. Observando alguns cabeçalhos de informação que eram passados sem preocupações sobre privacidade, observámos o uso do *keep-alive* da ligação TCP. Não podemos de forma segura concluir que este tipo de ligações é utilizado de forma recorrente ou que dá suporte à solução *Market Webstore*, no entanto mesmo que de forma reduzida, parece-nos uma má abordagem no contexto estudado visto que ligações deste tipo aumentam o tráfego de comunicação de forma a garantir a ligação. Tal poderá no entanto estar relacionado

com os restantes mecanismos utilizados por eles na sua solução como é o caso dos *protocol buffers*.[3]

ID do terminal	"android_id" uma vez que não está registado. 10 caracteres alfanuméricos caso contrário.
EMEI	15 caracteres numéricos
Sistema operativo	"android"
Versão do SO	"10"
Fabricante	"blade"
Modelo	"Blade"

Table 5.2: Informação dos terminais utilizados para o seu registo

Na tabela 5.1 não é indicado propositadamente o tamanho da informação trocada para o registo do terminal móvel no sistema. De facto, nos testes foram utilizados terminais idênticos a nível de características como o fabricante e o modelo. Toda a outra informação a trocar apresenta valores fixos pelo que não foi possível fazer uma análise exaustiva deste campo. Podemos, no entanto, indicar que para um terminal com a informação necessária para registo igual à presente na tabela 5.2 - que é de facto a informação dos terminais *ZTE Blade* utilizados para teste - o processo de registo apresenta um tráfego de 551 bytes quando transferido de forma aberta (HTTP) ou 573 bytes quando enviada sobre o protocolo SSL (HTTPS). Com base nesta informação podemos extrapolar que o valor recolhido neste teste não deverá alterar de forma significativa para restantes terminais e fabricantes. De forma a suportar a nossa conclusão apresentamos os seguintes fatores:

- A transferência de informação é realizada recorrendo a XML. Desta forma o tamanho da informação trocada é diretamente relacionado à informação escrita (alfanumérica).
- O identificador do terminal é gerado pelo sistema de gestão tendo um tamanho sempre fixo.
- Quando disponível, o identificador IMEI apresenta sempre um tamanho fixo de 15 caracteres.
- O sistema operativo e a versão, bem como o fabricante e o modelo apresentam um tamanho restrito.

Foram ainda feitos alguns testes em relação às macro iterações do sistema nomeadamente aos casos de uso de início de sessão que engloba o envio das credenciais do utilizador e identificador do terminal seguido de um envio de lista de perfis e mais tarde uma lista de identificadores de aplicações associadas ao perfil escolhido. Neste caso, o maior impulsionador para o tamanho do tráfego trocado é a relação entre o número de perfis e o número de aplicações presentes em cada um deles. Como descrito na secção 3.3.3, a informação a passar é no entanto limitada ao estritamente necessário sendo enviados o nome para o caso do perfil e um identificador/hash representado com uma string alfanumérica de 32 caracteres para o caso das aplicações.

5.2.5 Testes de Segurança

De forma a avaliar a segurança oferecida pela nossa solução, foram realizados vários testes de segurança ao protótipo desenvolvido.

De início foram realizados testes de *sniffing* de informação trocada entre o terminal e o sistema de gestão. Foram executadas todas as operações possíveis pelo nosso protótipo - registo, início de sessão, troca de aplicações - tendo todo o tráfego de dados sido obtido e analisado. Como esperado, a privacidade da informação é garantida, não sendo possível visualizar de forma aberta quaisquer dados transferidos.

[3]https://developers.google.com/protocol-buffers/?hl=pt-PT

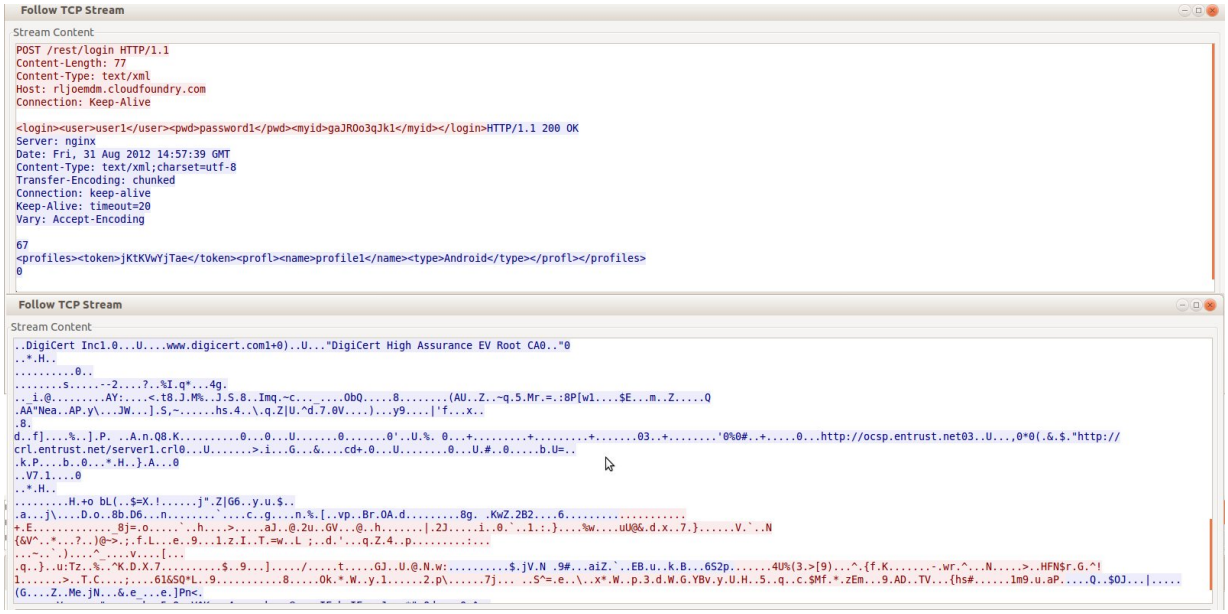


Figure 5.4: Captura de informação trocada no processo de início de registo de terminal e início de sessão. Em **cima**: comunicação não cifrada; em **baixo**: sobre protocolo SSL com certificado reconhecido.

Devemos no entanto realçar uma questão no que toca à autenticidade da informação. Uma vez que foi utilizado um certificado **self-signed**, o mesmo não consta na lista de certificados de *root* presente, por exemplo, nos terminais móveis. Neste caso, é exibido uma mensagem informativa indicando que o serviço não é reconhecido podendo o utilizador continuar aceitando a situação, ou recusar. Foi realizado o mesmo teste de forma menos exaustiva quando o sistema de gestão se encontra em execução nas infraestruturas *CloudFoundry*, estas oferecendo comunicação segura SSL baseada num certificado reconhecido e associado à *VMWare*. Parte da informação recolhida no processo de registo de um terminal e início de sessão pode ser visualizado na figura 5.4 onde é feito um comparativo gráfico entre a comunicação cifrada e a mesma comunicação não cifrada. Para a obtenção da imagem foi utilizada a ferramenta *Wireshark*.[4]

De seguida foram tentados testes de replicação de mensagens. Neste caso os testes incidiram especialmente quando a comunicação não era cifrada. De facto, o próprio protocolo SSL oferece mecanismos para prevenção deste tipo de ataques sendo renegociada uma chave de cifra temporária no primeiro contacto entre nós. Uma replicação de mensagem por outro terminal implica uma nova renegociação de chaves, tornando a mensagem a replicar inválida. Caso seja o mesmo utilizador e realizar o pedido o cenário de testes altera-se sendo a própria solução que deve garantir segurança.

Para realizar este teste, foram captados todos os pedidos inicializados pelo terminal que poderiam devolver informação privada, a saber, registo de terminal, início de sessão, escolha de perfil e *download* de aplicação - pedidos estes em canal aberto, sem cifra - sendo posteriormente reenviados quer pelo mesmo terminal quer por outros terminais utilizando ferramentas de rede presentes no sistema Android - derivado de linux - como o netcat.[5] Neste caso, o mecanismo de *tokens* temporários entra em execução tendo provado ser eficaz. Não foi possível replicar um único pedido com sucesso.

Utilizando ainda todo o equipamento em modo de comunicação não cifrada, foram executados testes de *brute-force* quer às credenciais de utilizador por forma a iniciar uma sessão válida, quer ao pedido de

[4]http://www.wireshark.org/
[5]http://netcat.sourceforge.net/

aplicações tentando adivinhar sessões válidas.

Para o primeiro teste, não existiu qualquer sucesso em adivinhar um par de credenciais de utilizador - [nome; palavra passe]. Podemos mesmo fazer um estudo baseado nas regras por nós exigidas: cada nome de utilizador tem um mínimo de 3 caracteres alfanuméricos, sendo o tamanho mínimo das palavras passe 5 caracteres. Desta forma existem 96717311574016 de possibilidades distintas. A adição de técnicas avançadas de escolha de credenciais incluindo a adição de outros caracteres permitem aumentar ainda mais a segurança. Estas alterações são suportadas de forma direta na solução apresentada, assim como mecanismos de controlo de acessos falhados repetidos.

Capítulo 6

Conclusão

Os terminais móveis foram alvo de uma grande mudança na maneira como são utilizados, sendo atualmente encarados como ferramentas de trabalho de valor imprescindível. Como tal, a necessidade de gerir estas ferramentas de forma central, englobando gestão de inventário quer de *software* quer de *hardware*, bem como oferecendo meio de acesso a informação privada de forma simples e escalável motivou o desenvolvimento de soluções de gestão de terminais móveis - MDM - centrais e orientadas a empresas.

Este documento apresenta um sistema MDM capaz de gerir terminais móveis do ponto de vista de inventário - de *software* e de *hardware* - assim como a instalação automática e não intrusiva de aplicações associadas a perfis e utilizadores, permitindo ainda um registo do terminal no sistema de gestão a partir do próprio terminal: SuusMDM.

O sistema utiliza o conceito de perfil para agrupar conjuntos de aplicações segundo qualquer métrica desejável pelos utilizadores. Esses perfis são posteriormente utilizados para que um utilizador ao iniciar sessão num terminal, escolha o desejado tendo assim todas as aplicações associadas instaladas no terminal. Quando o utilizador termina sessão no terminal, as aplicações instaladas por via do sistema são automaticamente removidas. Ainda enquanto existir uma sessão ativa num terminal, sempre que o perfil ativo for atualizado - adicionando novas aplicações, por exemplo - o terminal é contactado pelo sistema de gestão, instalando posteriormente e de forma silenciosa as aplicações, que ficam prontas a serem utilizadas.

Foi desenvolvido um protótipo orientado a sistemas Android. O protótipo consiste numa componente servidor, que pode ser executada em qualquer sistema que possua um *webserver* aplicacional Java, e uma aplicação cliente a ser instalada nos terminais móveis a gerir. O componente servidor é gerido através de qualquer *browser* permitindo pela sua interface gráfica a criação, remoção e atualização de terminais, utilizadores, aplicações e perfis. A aplicação cliente apresenta uma interface reduzida onde são inseridas as credenciais do utilizador e o URL do servidor de gestão de forma a iniciar sessão e posteriormente um botão que permite o término da mesma.

O protótipo SuusMDM foi avaliado do ponto de vista quantitativo segundo as métricas usuais em sistemas cliente/servidor. A utilização de recursos como o poder de processamento e/ou memória foram analisados assim como o tráfego de rede. Esta avaliação foi conduzida quer em sistemas servidor, quer em terminais móveis, onde também a utilização de bateria foi avaliada. Os resultados foram os esperado, sendo comparados aos encontrados nas soluções que permitem uma análise destes indicadores. Foi ainda feita uma avaliação qualitativa que englobou testes com utilizadores reais e um questionário que pretende compreender os seus sentimentos sobre o SuusMDM e soluções análogas. Os resultados foram animadores, quer do ponto de vista do protótipo desenvolvido, que se mostrou uma solução apta sendo bem recebida pelos utilizadores, quer do ponto de vista de uma análise mais geral do contexto atual dos terminais móveis que embora não apresentem uma ubiquidade total, começam a ser tão familiares para todos no quanto outras ferramentas de trabalho

indissociáveis do mundo atual. Este último aspeto foi realçado quer no número de utilizadores que utilizam terminais móveis de forma intensiva diariamente, mesmo que estes não tenham qualquer ligação profissional com os mesmos, assim como o a vontade mostrado no manuseamento dos mesmos e o conhecimentos das suas funcionalidades e características.

Como conclusão, o SuusMDM é um sistema que permite uma gestão de inventário quer do *hardware* quer do *software* de terminais móveis, permitindo a instalação e atualização de aplicações de forma remota. Mais do que um sistema de MDM clássico, o SuusMDM abre um novo paradigma de terminais móveis não associados a um utilizador, mas partilhados por vários, permitindo um ambiente de trabalho, na forma das aplicações presentes, direcionado ao utilizador que o utiliza no momento. Esta solução foi bem recebida pelos utilizadores que a testaram sendo uma mais valia forte no contexto móvel em franco crescimento.

6.1 Trabalho Futuro

São aqui apresentadas modificações e melhorias que poderão ser realizadas ao sistema desenvolvido:

- **Associar modelos de escolha de perfil baseados noutros índices.** O método atual de início de sessão apresenta-se bastante limitativo quando visto à luz da computação móvel atual. A necessidade de questionar o utilizador de forma repetida e na generalidade dos casos não tira partido das propriedades de mobilidade e ubiquidade associadas aos terminais alvo. Seria interessante a integração de mecanismos que permitissem a escolha de perfil para início de sessão - ou mesmo a troca por outro perfil quando o utilizador já se encontra com sessão ativa - baseados em índices como a localização real ou virtual do terminal. A adição de agendamento de sessões baseadas no horário de trabalho e calendário do mesmo poderia também ser vantajosa.
- **Melhorar o modelo de rota para o terminal durante a comunicação no modelo push.** Para que o serviço de gestão entre em contacto com o terminal é necessário conhecer o seu endereço de rede. Na nossa solução, a própria aplicação a executar nos dispositivos móveis apresenta um lógica de observação de alterações na camada de rede, alertando o serviço de gestão sempre que o seu endereço mudou. No entanto o terminal apresenta limitações nesta lógica onde o pedido direto ao sistema do terminal não contempla cenários onde este se encontre numa rede NAT. Por outro lado, também não pode ser utilizado o endereço do último pedido realizado ao serviço de gestão devido aos mesmos problemas.
- **Suporte abrangente de terminais.** Com o aumento do mercado de *smartphones* e *tablets* surgem cada vais mais fabricantes e os seus respetivos sistemas. A solução apresentada deve ser suficientemente abrangente dando suporte aos vários sistemas de forma transversal e totalmente transparente ao utilizador. De igual forma não deve ser necessário um grande esforço de desenvolvimento para este suporte. Um possível cenário engloba uma aplicação onde toda a lógica não se encontra implementada nu cenário online - HTML5 / CSS / JavaScript - sendo apenas alguns blocos implementados de forma nativa. A nossa implementação apresenta já um suporte básico tendo as funcionalidades de instalação e remoção no dispositivo - diretamente dependentes da plataforma - sido implementadas de forma modular do resto da aplicação.

Bibliografia

[1] It best practices: To support or not support consumer-owned smartphones. Technical report, Osterman Research, March 2009.

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical report, 2009.

[3] J. Bacon and T. L. Harris. *Operating Systems: Concurrent and Distributed Software Design*. Addison-Wesley, 2003.

[4] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6):599 – 616, 2009.

[5] Minhee Chae and Jinwoo Kim. What's so different about the mobile internet? *Commun. ACM*, 46:240–247, December 2003.

[6] Wendy Chou. Elliptic curve cryptography and its applications to mobile devices. Technical report, University of Maryland, College Park, Department of Mathematics.

[7] D. Dagon, T. Martin, and T. Starner. Mobile phones as computing devices: the viruses are coming! *Pervasive Computing, IEEE*, 3(4):11 – 15, oct.-dec. 2004.

[8] Whitfield Diffie and Martin E. Hellman. New directions in cryptography, 1976.

[9] Diaa Salama Abdul. Elminaam, Hatem M. Abdul Kader, and Mohie M. Hadhoud. Performance evaluation of symmetric encryption algorithms on power consumption for wireless devices. In *International Journal of Computer Theory and Engineering*, volume 1, pages 1793 – 8201 vol.1 N°4, oct. 2009.

[10] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35:114–131, June 2003.

[11] Vipul Gupta, Sumit Gupta, Sheueling Chang Shantz, and Douglas Stebila. Performance analysis of elliptic curve cryptography for ssl. In *Workshop on Wireless Security*, pages 87–94. ACM, 2002.

[12] Manfred Hauswirth and Mehdi Jazayeri. A component and communication model for push systems. In *Proceedings of the 7th European software engineering conference held jointly with the 7th ACM SIGSOFT international symposium on Foundations of software engineering*, ESEC/FSE-7, pages 20–38, London, UK, 1999. Springer-Verlag.

[13] Yongqiang Huang and Hector Garcia-Molina. Publish/subscribe in a mobile environment. *Wirel. Netw.*, 10:643–652, November 2004.

[14] Kathryn Huberty, Mark Lipacis, and Adam et al Holt. Tablet demand and disruption, mobile users come of age. Morgan stanley blue papper, Morgan Stanley, June 2010.

[15] K. Johansson, J. Bergman, D. Gerstenberger, M. Blomgren, and A. Wallen. Multi-carrier hspa evolution. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pages 1 –5, april 2009.

[16] T. Martin, M. Hsiao, Dong Ha, and J. Krishnaswami. Denial-of-service attacks on battery-powered mobile computers. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, pages 309 – 318, march 2004.

[17] Mary Meeker, Scott Devitt, and Liang Wu. Internet trends. Technical report, Morgan Stanley, June 2010.

[18] I.P.L. Png, B.C.Y. Tan, and Khai-Ling Wee. Dimensions of national culture and corporate adoption of it infrastructure. *Engineering Management, IEEE Transactions on*, 48(1):36 –45, feb 2001.

[19] Ivana Podnar, Manfred Hauswirth, and Mehdi Jazayeri. Mobile push: Delivering content to mobile users. In *in Proceedings of the International Workshop on Distributed Event-Based Systems (ICDCS/DEBS'02*, pages 563–570. IEEE Computer Society, 2002.

[20] G.V.S. Raju and R. Akbani. Elliptic curve cryptosystem and its applications. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 2, pages 1540 – 1543 vol.2, oct. 2003.

[21] J. Rapeli. Umts: targets, system concept, and standardization in a global framework. *Personal Communications, IEEE*, 2(1):20 –28, feb 1995.

[22] D.R. Raymond and S.F. Midkiff. Denial-of-service in wireless sensor networks: Attacks and defenses. *Pervasive Computing, IEEE*, 7(1):74 –81, jan.-march 2008.

[23] Yasushi Saito and Marc Shapiro. Optimistic replication. *ACM Comput. Surv.*, 37:42–81, 2005.

[24] S. Sirkemaa. It infrastructure management and standards. In *Information Technology: Coding and Computing, 2002. Proceedings. International Conference on*, pages 201 – 206, april 2002.

[25] S. Tenorio, K. Exadaktylos, B. McWilliams, and Y. Le Pezennec. Mobile broadband field network performance with hspa+;. In *Wireless Conference (EW), 2010 European*, pages 269 –273, april 2010.

[26] Carl Zeite, Margo Visitacion, Ellen Daley, and Kimberly Q. Dowling. The mobile enterprise: Defining your strategy. Technical report, Forrester, March 2005.

Anexo A

Estrutura da Base de Dados Implementada

Nome campo	**Tipo de Dado**	**Associações**	**Outros**
Nome	Texto	-	Obrigatório
Versão	Texto	-	Obrigatório
Sistema Operativo	Texto	-	Obrigatório
Comentários	Texto	-	-
Hash identificativo	Texto	-	Obrigatório e único
Aplicação	Binário	-	Tamanho máximo: 2Mb

Table A.1: Aplicação

Nome campo	**Tipo de Dado**	**Associações**	**Outros**
Id	Texto	-	Obrigatório e único
EMEI	Texto	-	Único
Sistema Operativo	Texto	-	Obrigatório
Versão Sistema Operativo	Texto	-	Obrigatório
Fabricante	Texto	-	-
Modelo	Texto	-	-
Último IP conhecido	Texto	-	-
Nome Utilizador	Texto	1 para 1 elemento da tabela Utilizador	Obrigatório

Table A.2: Terminal

Nome campo	**Tipo de Dado**	**Associações**	**Outros**
Nome	Texto	-	Obrigatório
Palavra Passe	Texto	-	Obrigatório
Perfis associados	Texto	1 para muitos elementos da tabela Perfil	-

Table A.3: Utilizador

Nome campo	**Tipo de Dado**	**Associações**	**Outros**
Nome	Texto	-	Obrigatório
Tipo	Texto	-	-
Nome Aplicações	Texto	1 para muitos com elementos da tabela Aplicação	-

Table A.4: Perfil

Nome campo	**Tipo de Dado**	**Associações**	**Outros**
Token	Texto	-	Obrigatório e único
Utilizador	Texto	1 para 1 com elemento da tabela Utilizador	-
Hash Aplicação	Texto	-	-

Table A.5: *Tokens* Temporários

Nome campo	**Tipo de Dado**	**Associações**	**Outros**
Endereço IP	Texto	-	Obrigatório
Perfil Ativo	Texto	1 para 1 com elemento da tabela Perfil	Obrigatório

Table A.6: Sessões Ativas

Nome campo	**Tipo de Dado**	**Associações**	**Outros**
Nome do Utilizador	Texto	1 para 1 com elemento da tabela Utilizador	Obrigatório
Id to terminal	Texto	1 para 1 com elemento da tabela Terminal	Obrigatório
Time Stamp	Date	-	Obrigatório

Table A.7: Log de acessos

Anexo B

Questionário de Testes com Utilizadores

Sexo?

M

F

Idade?

< 21

21 – 25

26 – 35

36 – 45

46 – 55

56 – 65

> 65

Quais as suas habilitações literárias?

Ensino básico

Ensino secundário

Frequência do ensino superior

Licenciado

Com que frequência utiliza/interage com terminais móveis?

Raramente

Ocasionalmente

Diariamente mas de forma não intensiva

Diariamente e de forma intensiva

O seu contacto com terminais móveis enquadra-se num cenário maioritariamente

Particular

Profissional

Com que tipo de terminais móveis tem mais contacto?

Smartphones / telemóveis

Tablets

Outros

A sua actividade profissional / académica está directamente ligada às tecnologias?

Sim

Não

Responda a às seguintes questões marcando com X a resposta numérica onde 1 indica uma avaliação muito negativa ou discordante e 8 uma avaliação muito positiva ou totalmente de acordo. Escolha 0 quando a pergunta não se aplicar ou não quiser responder.

	0	1	2	3	4	5	6	7	8
Aplicação Cliente									
O seu terminal pessoal apresentou problemas a interagir com a solução?									
Foi fácil utilizar o sistema do ponto de vista do cliente?									
Sentiu alguma alteração do normal funcionamento do terminal?									
Achou distractivo a presença da aplicação no terminal?									
O tempo de duração do processo de *login* é aceitável?									
Back-Office									
Foi fácil utilizar o sistema do ponto de vista do sistema de gestão?									
A informação apresentada sobre cada terminal era pertinente?									
Foi fácil modificar um perfil de utilizador?									
A adição de novas aplicações apresentou problemas?									
A adição de um utilizador foi fácil?									
A adição de um terminal a gerir apresentou algum problema?									

	0	1	2	3	4	5	6	7	8
Já tinha tido alguma experiência com sistemas deste tipo?									
Conhece outras soluções semelhantes?									
Encontra utilidade, no seu meio profissional, para um sistema destes?									
Fora do âmbito profissional encontra utilidade num sistema equivalente?									
Habituar-se-ia a utilizar esta solução de forma rápida e fácil?									
Receia ou encontra possíveis problemas de segurança?									

Anexo C

Fluxo de Execução de Testes com Utilizadores

Contexto Terminal Móvel

Foi contratado/a para trabalhar na *Parques & Viaturas*, uma empresa dedicada à exploração de parques de estacionamento públicos. A função que lhe foi atribuída consiste em verificar que todos os utilizadores possuem dístico de utilizador e caso não o tenham, aplicar a coima devida.

Para que possa realizar esta tarefa, a empresa utiliza várias aplicações executadas em terminais móveis que disponibiliza aos seus funcionários. Caso estes prefiram, podem também utilizar o seu terminal bastando inseri-lo no sistema de gestão.

Sendo o seu primeiro dia de trabalho, deverá realizar os seguintes passos de forma a ter uma ferramenta de trabalho que lhe permita realizar as suas tarefas:

- Instale a aplicação num terminal móvel (o seu ou um fornecido pela empresa).
- Inicie a sessão no terminal, escolhendo o seu perfil de funcionário: **FuncionarioBase** - O seu utilizador é **func1** e a sua palavra passe é **pass1**.
- Após ter iniciado a sessão, verifique que a aplicação **Multas App** se encontra instalada no terminal.

Após algumas semanas de trabalho, onde de forma recorrente teve de entrar em contacto com o departamento de recursos humanos de forma a verificar queixas de alguns utentes que asseguravam ter pago o serviço, mas que ainda não tinham recebido o dístico, decide pedir que lhe seja fornecida a aplicação **Novos Clientes App**. Após um contacto inicial, recebe a resposta do departamento de recursos técnicos que o mesmo já se encontra no seu perfil. Visualize se de facto ele se apresenta no terminal.

Algumas semanas mais tarde, devido à época de férias na empresa, acaba por acumular outra função: regularizador(a) de multas. A sua atividade consiste em regularizar processos de coimas já pagas por utilizadores mas que devido a questões técnicas necessitam de ser alterados manualmente. Como forma de prevenir falcatruas a aplicação que permite a regularização manual não se encontra no mesmo perfil *FuncionarioBase*, mas sim num outro de nome **FuncionarioPlus**. Verifique que o seguinte processo se realiza sem problemas:

- Termine a sessão de **FuncionarioBase** ativa no terminal móvel.
- Verifique que as aplicações associadas - *Multas App* e *Novos Clientes App* foram removidas do sistema.

- Inicie nova sessão e escolha o perfil **FuncionarioPlus**.
- Verifique que a aplicação **Limpa Divida App** se encontra instalada.

Contexto Sistema de Gestão

Foi contratado/a para trabalhar na *Parques & Viaturas*, uma empresa dedicada à exploração de parques de estacionamento públicos. A função que lhe foi atribuída consiste na gestão do sistema de gestão de terminais móveis existente que permite que vários funcionários partilhem dispositivos móveis tendo sempre as aplicações necessárias à execução das suas tarefas, disponíveis.

No seu primeiro dia de trabalho, um colega que acaba também de ser contratado/a necessita de ser integrado no sistema. Deve criar no sistema um utilizador para ele com as seguintes credenciais:

- Utilizador: **func1**
- Password: **pass1**

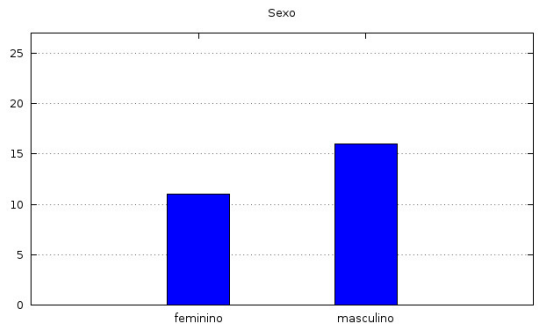
Deve também gerar um novo perfil com nome **FunionarioBase**, composto unicamente pela aplicação **Multas App**, e associa-lo ao utilizador criado.

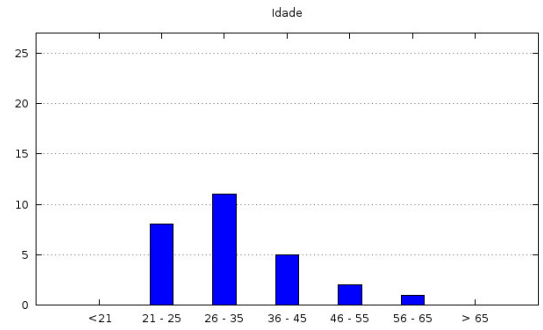
Algumas semanas mais tarde, foi-lhe indicado que atualize o perfil **FuncionarioBase** para que contenha também a aplicação **Novos Clientes App**.

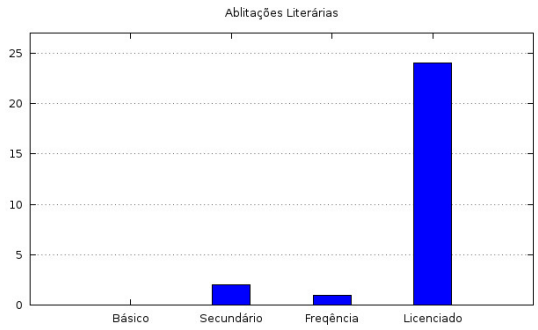
Por fim é-lhe pedido que crie um novo perfil, **FuncionarioPlus**, que contenha a aplicação **Limpa Divida App** que se deve ser inserida no sistema de gestão, e que associe este novo perfil ao funcionário **func1**.

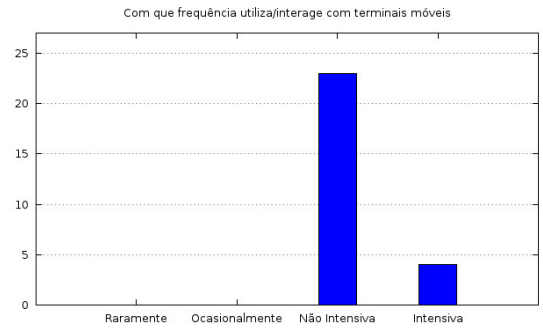
Anexo D

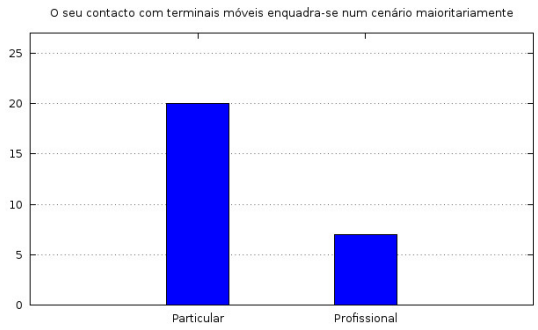
Resultados dos Testes com Utilizadores

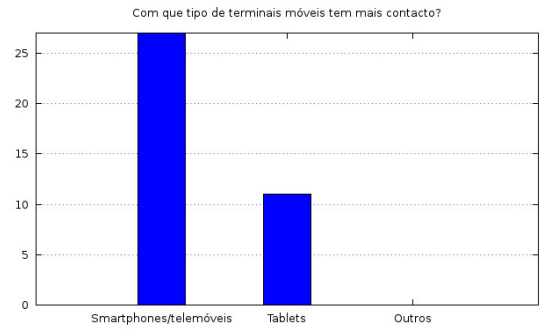


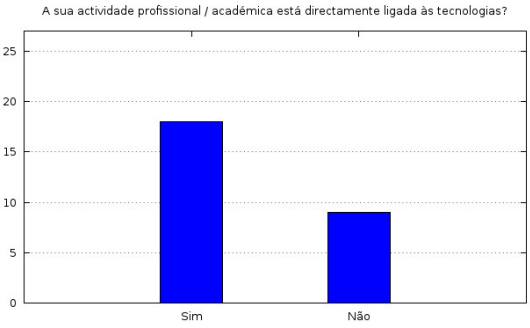


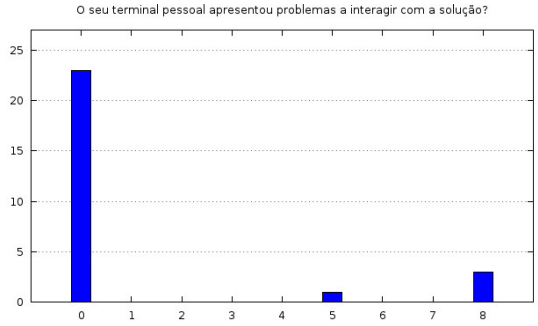


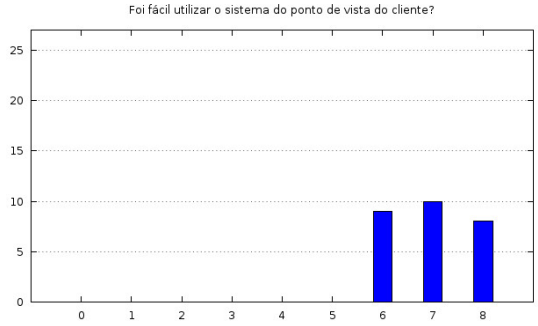


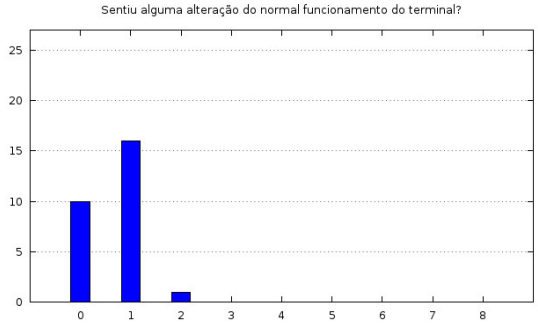


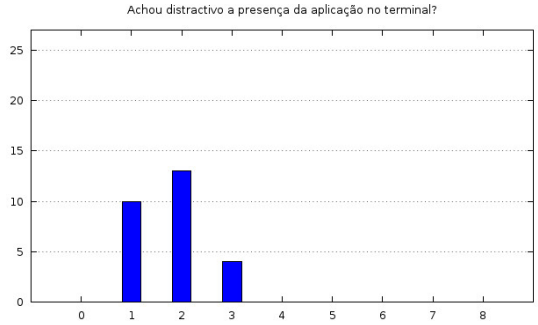


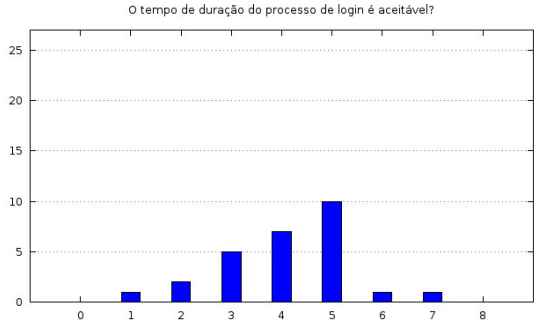

A sua actividade profissional / académica está directamente ligada às tecnologias?
25
20
15
10
5
0
Sim
Não

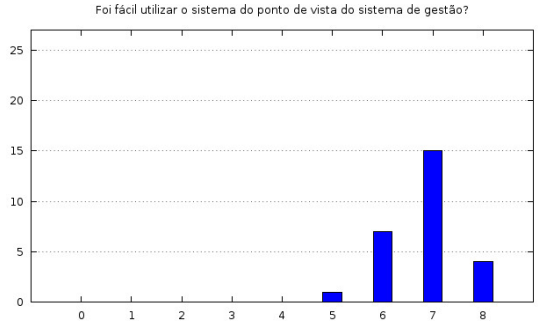

O seu terminal pessoal apresentou problemas a interagir com a solução?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8

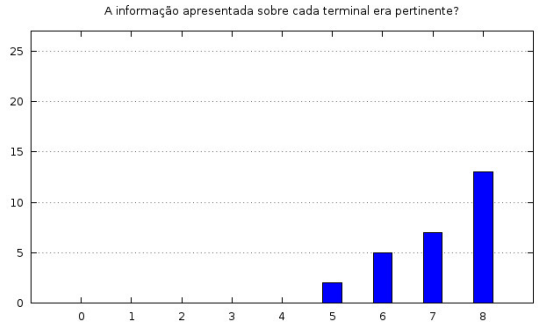

Foi fácil utilizar o sistema do ponto de vista do cliente?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8

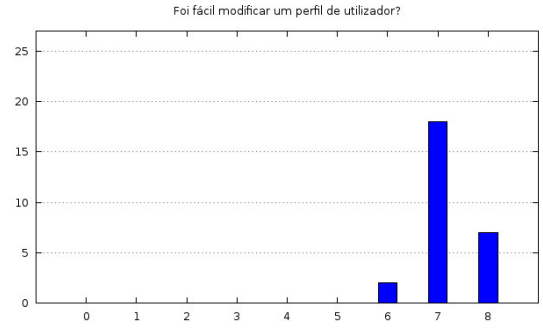

Sentiu alguma alteração do normal funcionamento do terminal?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8


Achou distractivo a presença da aplicação no terminal?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8

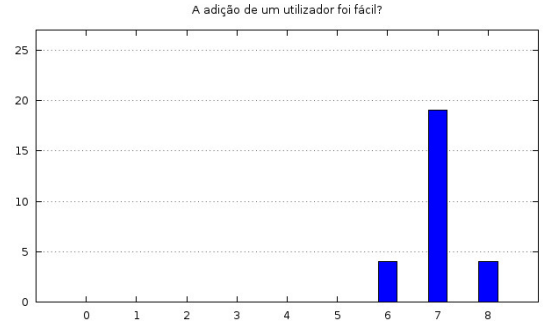

O tempo de duração do processo de login é aceitável?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8

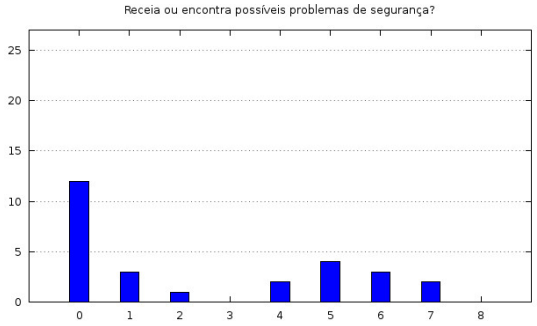

Foi fácil utilizar o sistema do ponto de vista do sistema de gestão?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8

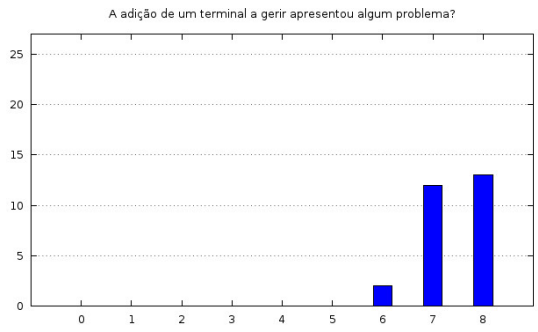

A informação apresentada sobre cada terminal era pertinente?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8


Foi fácil modificar um perfil de utilizador?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8

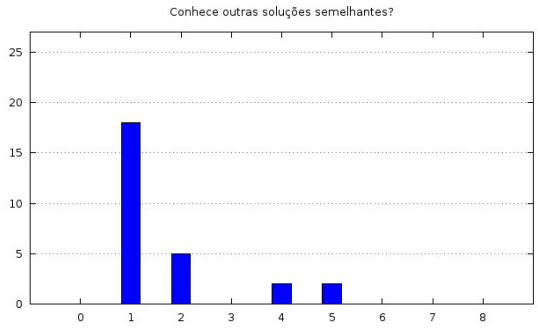

A adição de novas aplicações apresentou problemas?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8


A adição de um utilizador foi fácil?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8

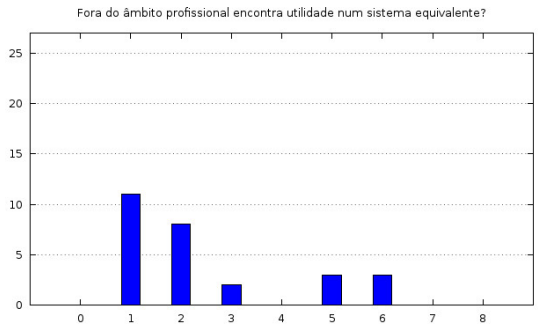

Receia ou encontra possíveis problemas de segurança?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8

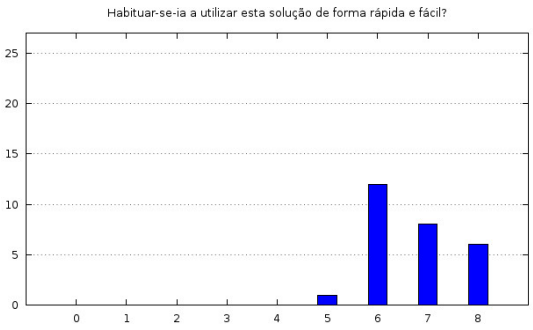

A adição de um terminal a gerir apresentou algum problema?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8


Já tinha tido alguma experiência com sistemas deste tipo?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8


Conhece outras soluções semelhantes?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8


Encontra utilidade, no seu meio profissional, para um sistema destes?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8


Fora do âmbito profissional encontra utilidade num sistema equivalente?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8


Habituar-se-ia a utilizar esta solução de forma rápida e fácil?
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8