



**Filipe Manuel
Lopes de Freitas**

**SegUA: um sistema móvel para apoio à rotina de
vigilância**



**Filipe Manuel
Lopes de Freitas**

**SegUA: um sistema móvel para apoio à rotina de
vigilância**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor João Paulo Trigueiros da Silva Cunha, Professor Associado com Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Mestre Ilídio Fernando de Castro Oliveira, Assistente Convidado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri

presidente

Prof. Dr. António Ferreira Pereira de Melo
Universidade de Aveiro

vogais

Prof. Dr. António Miguel Pontes Pimenta Monteiro
Universidade do Porto

Prof. Dr. João Paulo Trigueiros da Silva Cunha
Universidade de Aveiro

Mestre Ilídio Fernando de Castro Oliveira
Universidade de Aveiro

agradecimentos

A realização desta dissertação e de todo o meu percurso académico foi apenas possível graças ao trabalho e sacrifícios dos meus pais. Por tal, expresso a minha gratidão e espero, um dia, conseguir retribuir a confiança que depositaram em mim ao longo de todos estes anos.

Agradeço à minha namorada Sónia Moreira pelo seu apoio constante que me deu forças para continuar a lutar pelos meus objectivos nos momentos mais difíceis do curso. Agradeço também à sua família pelo apoio moral ao longo de todo este trabalho.

Um grande abraço ao meu amigo de infância Bernardo Silva, com quem passei inúmeras noites trabalhando em projectos de disciplinas, e que ficarão como boas recordações, apesar de não as terem sido na altura. Uma boa equipa conquista os maiores obstáculos.

Agradeço ao meu orientador Prof. Doutor João Paulo Cunha por considerar o este trabalho como merecedor do alto nível de qualidade esperado numa dissertação de Mestrado. Agradeço ao meu co-orientador Prof. Ilídio Oliveira por toda a sua atenção, paciência, experiência e simpatia, que sem as quais a realização desta dissertação não teria sido possível.

Agradeço aos Serviços Técnicos da Universidade de Aveiro e, em especial, ao Tiago Sousa, do Gabinete de Gestão de Informação, pela colaboração na realização desta dissertação.

palavras-chave

computação móvel, georeferenciação, sincronização de dados móveis, integração de sistemas, vigilância.

resumo

Os serviços de vigilância e segurança têm a obrigação de documentar um conjunto de incidentes observados no decorrer das rondas, e esta documentação é, sobretudo, produzida em papel, o que dificulta a sua consolidação e processamento subsequentes. Para além disso, a vigilância é uma actividade inerentemente móvel, havendo, assim, espaço para estudar a aplicação de novos dispositivos computacionais no acompanhamento do profissional para facilitar a aquisição de dados, e organização e comunicação destes.

Partindo da realidade do campus da Universidade de Aveiro e procedimentos associados, este trabalho procurou introduzir melhorias processuais nas actividades de vigilância e sua gestão.

No novo processo de trabalho, os vigilantes utilizam um módulo móvel para navegar nos mapas de uma organização e registar, no ponto de observação, os incidentes, podendo extender o relato com elementos multimédia. Durante a execução das rondas, é possível, centralmente, observar o posicionamento dos Vigilantes e obter, de forma quase imediata, os resultados das rondas. Os incidentes são imediatamente encaminhados para os pontos de serviço da instituição, que têm a responsabilidade de tomar acções subsequentes, eliminando a necessidade de execução de um processo moroso de triagem actual.

O sistema desenvolvido incorpora vistas orientadas por mapas, e integra-se com o Sistema de Informação Geográfica central para obter mapas e informações das entidades espaciais.

keywords

mobile computing, georeferencing, mobile data synchronization, systems integration, surveillance

abstract

Surveillance personnel need to report a wide range of incidents during their watches, which are mostly recorded on paper. This seriously hinders processing and archiving of such incidents for future analysis. The mobile nature of surveillance tasks presents an opportunity to study the introduction of mobile devices to assist in data acquisition, communication and organization, thus improving the quality of the data produced by the surveillance service.

Using the Universidade de Aveiro and its associated surveillance procedures as a source of requirements, this project aims at improving the existing workflow and providing an incident handling system, thus enabling easy access to data.

In the proposed workflow process, watchmen use a mobile device to report incidents, instead of paper. This mobile device allows the user to view maps and previously reported incidents, and to extend the incident facts with multimedia data. While watchmen carry out their watches, it is possible to centrally supervise their progress and current location in near real-time. In addition, reported incidents are immediately routed to their designated service points, where these incidents will be handled.

The designed system supports map visualization, and integrates with the organization's Geographic Information System to obtain spatial data and maps.

Conteúdo

Lista de figuras	3
Lista de tabelas.....	5
Dicionário de acrónimos	6
1 Introdução.....	7
1.1 Enquadramento.....	7
1.2 Objectivos.....	8
1.3 Estrutura da dissertação	9
2 Estado da Arte	11
2.1 Emergência da computação móvel.....	11
2.2 Desenvolvimento de aplicações para dispositivos móveis	13
2.3 Aplicações da computação móvel à Vigilância.....	15
3 Cenários a suportar	17
3.1 Processo de trabalho actual.....	17
3.2 Novo processo de trabalho proposto	19
3.3 Casos de utilização	21
3.4 Necessidades de integração de sistemas.....	23
4 Desenho da solução	25
4.1 Arquitectura do sistema.....	25
4.1.1 Módulo móvel	27
4.1.2 Módulo central.....	29
4.2 Comportamento adaptativo.....	31
4.2.1 Adaptação ao contexto espacial	31
4.2.2 Adaptação à variação das condições de rede	34
4.2.3 Adaptação à disponibilidade de energia.....	35
4.3 Modelo de dados	37
4.4 Integração de dados e sincronização de dados móveis.....	39
4.5 Visualização de mapas	42
5 Implementação	45

5.1	Tecnologias e modelo de programação	45
5.1.1	Implementação do módulo móvel	46
5.1.2	Implementação do módulo central.....	57
5.2	Arquitectura de instalação	68
5.3	Interacção com o utilizador.....	68
5.3.1	Desafios e soluções para aumentar a facilidade de utilização.....	69
5.4	Testes e verificação	71
6	Resultados	79
7	Conclusões.....	81
7.1	Trabalho futuro	82
8	Bibliografia	83
9	Anexos.....	89
9.1	Modelo de dados	89

Lista de figuras

Figura 1 – Message Pad vs iPhone 3GS	12
Figura 2 – Volume de vendas de smartphones e telemóveis no geral	13
Figura 3 – Exemplo da visualização de videovigilância em dispositivos móveis	16
Figura 4 – Morse Watchamn’s PowerCheck	16
Figura 5 – Fluxo actual de realização de rondas	17
Figura 6 – Fluxo actual de gestão de incidentes	18
Figura 7 – Fluxo proposto para realização das rondas.....	20
Figura 8 – Fluxo de trabalho proposto para o Pivot.....	21
Figura 9 – Casos de utilização no módulo centralista	22
Figura 10 – Casos de utilização no módulo móvel	23
Figura 11 – Integração de bases de dados	24
Figura 12 – Arquitectura inicial do sistema SegUA	25
Figura 13 – Arquitectura revista do sistema SegUA.....	27
Figura 14 – Arquitectura de 3 níveis	28
Figura 15 – Arquitectura da SegUAMóvel.....	29
Figura 16 – Arquitectura do SegUAService	30
Figura 17 – Arquitectura da SegUABD	30
Figura 18 – Comportamento autónomo de realização de ronda.....	32
Figura 19 – Comportamento autónomo de apresentação de mapas.....	33
Figura 20 – Comportamento no envio de dados	34
Figura 21 – Comportamento energético durante a ronda.....	36
Figura 22 – Modelo de dados dos utilizadores	37
Figura 23 – Modelo de dados das rondas	37
Figura 24 – Modelo de dados dos incidentes.....	38
Figura 25 – Modelo de dados dos edifícios.....	39
Figura 26 – Comparação entre os referenciais geográfico e do ecrã.....	43
Figura 27 – Correspondência dos parâmetros no ecrã	43
Figura 28 – Arquitectura implementada do sistema SegUA	45
Figura 29 – Arquitectura implementada da SegUAMóvel	46
Figura 30 – Método de georeferenciação através de áreas	48
Figura 31 – Exemplo do método de georeferenciação através de áreas.....	48
Figura 32 – Exemplo de associação de várias áreas a um edifício	49
Figura 33 – Multithreading na realização de uma ronda	52
Figura 34 – Ecrã de boas vindas e selecção de utilizador	52
Figura 35 – Autenticação do utilizador	53
Figura 36 – Pré-visualização e início ou retoma de rondas.....	53
Figura 37 – Visão do progresso da ronda.....	54
Figura 38 – Início do registo de um novo incidente	54
Figura 39 – Visualização de um incidente	55

Figura 40 – Selecção da família de situações	55
Figura 41 – Especificação da localização no interior de edifícios.....	56
Figura 42 – Anexação de conteúdo e verificação do incidente	56
Figura 43 – Envio de alarme	57
Figura 44 – Arquitectura do SegUAService	58
Figura 45 – Integração dos dados do WebSIG	59
Figura 46 – Exemplo de Período-Válido	59
Figura 47 – Fase de limpeza de registos.....	60
Figura 48 – Fase de inserção	61
Figura 49 – Algoritmo de inserção	62
Figura 50 – Sincronização dos dados	63
Figura 51 – Processo global de sincronização de dados	64
Figura 52 - Visualização das rondas em curso.....	65
Figura 53 – Relatórios diários.....	66
Figura 54 – Categorização de incidentes.....	67
Figura 55 – Arquitectura de instalação do sistema SegUA	68
Figura 56 – Fluxo de navegação da interface.....	69
Figura 57 – Comparação entre os controlos com utilizado do dedo	70
Figura 58 – Simplificação da introdução de dados.....	71
Figura 59 – Percurso registado vs percurso real	72
Figura 60 – Posição registada vs posição real	72
Figura 61 – Comparação entre a direcção na georeferenciação	73
Figura 62 – Teste da georeferenciação	74
Figura 63 – Localização obtida das equações de tradução	75
Figura 64 – Mapa de calor da potência do sinal	77
Figura 65 – Comparação entre o tempo de espera na visualização de incidentes nos processos ..	79

Lista de tabelas

Tabela 1 – Comparação das características entre o MessagePad e o iPhone 3GS	12
Tabela 2 – Comparação entre os vários ambientes de desenvolvimento	14
Tabela 3 – Medição de tempos de aquisição de sinal.....	73
Tabela 4 – Medição de tempos de sincronização	75
Tabela 5 – Comparação single threading vs multithreading.....	76
Tabela 6 – Estimativas de tempo dispendido nas tarefas.....	79

Dicionário de acrónimos

AGPS	Assisted GPS
GPS	Global Positioning System
IIS	Internet Information Services
NETCF	.NET Compact Framework
PACO	Portal Académico
PDA	Personal Digital Assistant
RAM	Random Access Memory
REST	Representational Stateful Transfer
RPC	Remote Procedure Call
SGBD	Sistema de gestão de bases de dados
SIG	Sistema de Informação de Geográfica
SOAP	Simple Object Access Protocol
SQL	Simple Query Language
SQLSCE	SQL Server CE
STecUA	Serviços Técnicos da Universidade de Aveiro
UA	Universidade de Aveiro
UA	Universidade de Aveiro
XML	Extensible Markup Language

1 Introdução

1.1 Enquadramento

A Universidade de Aveiro (UA) tem investido na informatização dos seus processos no sentido de reduzir a burocracia e, assim, melhorar o seu desempenho e, conseqüentemente, a qualidade de serviço prestada aos alunos. Um exemplo sólido desse investimento é o Portal Académico¹ (PACO), que permite ao aluno consultar notas e realizar várias operações administrativas, como o pagamento de propinas ou escolha de horários, que de outra forma teriam que ser realizadas presencialmente nos Serviços Académicos da UA, permitindo, assim, libertar funcionários dos serviços de atendimento e evitar que os alunos desperdiçam tempo de estudo enquanto esperam nas filas de espera (informação sobre estas as filas de espera estão também disponíveis online). Outro exemplo da informatização dos processos, a nível departamental, é o da distribuição de temas de dissertação de mestrado que é feita através de um *site*, onde está disponível uma lista com os temas que os alunos deverão indicar a sua escolha e, posteriormente, os professores eleger os alunos para os respectivos temas.

Os Serviços Técnicos da UA (STecUA) são um serviço cuja missão é a de zelar pelos espaços físicos da Universidade, e encarrega-se de executar a vigilância dos mesmos. Este Serviço já beneficia de alguma informatização dos seus processos, onde todo o espaço físico do campus está mapeado num Sistema de Informação Geográfica² (SIG), o que significa que os funcionários do Serviço podem facilmente gerir e localizar objectos.

A vigilância é executada por Vigilantes que realizam rondas frequentes em todo o campus e interior de edifícios com o objectivo de detectar quaisquer anomalias, como uma janela partida ou uma porta aberta. Após cada ronda, deverá ser redigido um relatório com as anomalias detectadas e, da combinação de todos esses relatórios, é gerado um relatório diário, que é analisado e processado manualmente para reencaminhar as anomalias para o responsável do local onde anomalia foi detectada, que deverá corrigir a situação. Para realizar uma ronda, o Vigilante deverá ter conhecimento prévio do percurso a efectuar ou guiar-se por um suporte em papel.

O controlo do percurso efectuado pelo Vigilante durante a ronda é feito através da validação de um dispositivo móvel rudimentar em marcadores instalados em paredes por toda a Universidade, que representam pontos de controlo por onde o Vigilante deve transitar. A verificação administrativa de uma ronda é feita manualmente e, como um ponto de controlo é representado internamente por um código, é uma tarefa morosa e susceptível a erros humanos.

Dado que o Serviço de Vigilância é realizado por vários Vigilantes e existem várias rondas que poderão possuir locais comuns, existe o risco de que uma mesma anomalia possa ser

¹ <http://paco.ua.pt/>

² Geographic Information System

registada várias vezes, criando, assim, redundância de dados o que complica e atrasa a filtragem e processamento das anomalias.

Além de ser um processo muito burocrático e envolvendo muitos funcionários, todo este processo é, de momento, realizado em suporte de papel, o que dificulta o controlo do Serviço e a gestão da informação gerada. O processo padece, assim, de vários problemas que poderão ser resolvidos através da informatização.

Dada a mobilidade inerente da tarefa, os vigilantes poderiam estar dotados de um dispositivo móvel moderno com o qual registariam anomalias que, por sua vez, seriam processadas por um sistema automatizado e encaminhadas para o responsável correspondente, tudo isto sem o envolvimento de qualquer intermediário. Assim, teoricamente, este processo seria consideravelmente mais rápido que o processo actual, e permitiria a extracção fácil de informação.

A natureza móvel da tarefa representa, assim, uma oportunidade para o estudo e aplicação de conceitos de Computação Móvel, e a existência de um sistema SIG potencializa a Integração de Dados no serviço de Vigilância.

Espera-se, então, introduzir melhorias de processo e de organização no Serviço de Vigilância através da reorganização dos fluxos de trabalho com introdução de novas tecnologias e, assim, explorar conceitos de Computação Móvel no contexto do trabalho.

1.2 Objectivos

Com colaboração dos Serviços Técnicos, pretende-se analisar o processo actual do registo e processamento de anomalias, doravante conhecidas por incidentes, e propor alterações que aumentem a eficiência recorrendo à interacção de sistemas com a mobilidade.

Pretende-se desenvolver um sistema – SegUA – composto por um módulo central e um módulo para dispositivos móveis – SegUAMóvel – que permita o registo de incidentes, facilite a realização e controlo das rondas, tanto por parte dos Vigilantes como administrativamente, e permita um acesso rápido e organizado à informação gerada. Pretende-se prototipar os serviços de back-end que suportam o módulo móvel e permitem gerir e visualizar a informação gerada por ele.

Pretende-se explorar a aplicação de conceitos de Computação Móvel e Integração de Dados na resolução de um problema real com utilizadores reais e detectar e tentar ultrapassar possíveis obstáculos, através de uma aplicação que seja executada num dispositivo móvel que acompanhe o Vigilante na sua ronda, utilizando as capacidades (multimédia, localização, etc.) do dispositivo, tentando aproveitar a disponibilidade de dados geográficos fornecidos pelo sistema SIG.

1.3 Estrutura da dissertação

Esta dissertação está dividida nas seguintes secções: Introdução, Estado da arte, Cenários a suportar, Desenho da solução, Implementação, Resultados, e Conclusão.

Na Introdução, são apresentados o contexto e o problema sobre os quais incidem esta dissertação, e são definidos objectivos para implementar uma possível solução.

No Estado da arte, o leitor é introduzido à Computação Móvel através da evolução dos dispositivos móveis e tecnologias desta área, e através da análise do impacto que a Computação Móvel teve. São apresentadas, também, aplicações da Computação Móvel na área da Vigilância.

Nos Cenários a suportar, são descritos, em detalhe, o cenário actual e o cenário proposto, seus fluxos de trabalho e papéis envolvidos, assim como as necessidades de integração de sistemas.

No Desenho da solução, é apresentada uma proposta de arquitectura para o sistema.

Na Implementação, é descrita a forma como a proposta de arquitectura foi implementada, quais os problemas encontrados e como foram ultrapassados.

Nos Resultados, discutem-se os resultados observados da aplicação do sistema face ao problema, tanto a nível técnico como organizacional.

Na Conclusão, verifica-se o cumprimento dos objectivos propostos para esta dissertação e apresenta-se o trabalho futuro.

2 Estado da Arte

2.1 Emergência da computação móvel

A transformação da sociedade actual para uma sociedade constantemente ligada e em movimento tem impulsionado a indústria para a criação de soluções computacionais com o máximo de capacidades móveis possíveis. A melhoria do processo de fabrico de componentes e o desenvolvimento de novas tecnologias, especialmente na área das baterias, tem permitido reduzir o tamanho e consumo de energia dos componentes, permitindo criar dispositivos móveis cada vez mais pequenos, mais potentes, mais baratos e com mais funcionalidades. Assim, tem-se observado uma convergência da telefonia com a computação, na forma de dispositivos que possuem funcionalidades de telefonia móvel e, também, funcionalidades de assistentes digitais³, tendo os dispositivos, assim, cada vez uma maior parte na vida das pessoas [1].

No sentido de se separarem da concorrência e de apresentarem um produto mais completo e com mais funcionalidades para o utilizador, os construtores embutem um número cada vez maior de componentes, o que aumenta o leque de aplicações que os dispositivos móveis têm, o que significa que, hoje em dia, os dispositivos podem possuir receptores de GPS, placas de rede 802.11, sensores fotográficos com vários megapixéis, etc. Os ecrãs têm verificado melhorias como aumento da resolução e sensibilidade ao toque humano (em ecrãs capacitivos, ao invés da pressão em ecrãs resistivos).

A melhor forma para compreender a evolução da tecnologia é comparar um dos primeiros dispositivos móveis com um dos últimos, o *MessagePad* e o *iPhone 3GS*, dois dispositivos móveis da *Apple* comercializados em 1993 e 2009 respectivamente, ilustrados pela Figura 1. Analisando as características descritas na Tabela 1, observa-se que o *iPhone 3GS* é 30 vezes mais rápido, possui 400 vezes mais memória RAM⁴, 512 vezes mais capacidade de armazenamento, e pesa um quarto do peso do *MessagePad*, tudo isto num espaço de 16 anos. Ao contrário do *MessagePad*, o *iPhone 3GS* possui ainda GPS, câmara fotográfica e telefonia.

³ Personal Digital Assistant (PDA)

⁴ Random Access Memory



Figura 1 – Message Pad vs iPhone 3GS⁵

	<i>MessagePad</i> [2]	<i>iPhone 3GS</i> [3]
Ano	1993	2009
CPU	20 MHz	600
RAM	640KB	256MB
Ecrã	Resistivo, preto e branco	Capacitivo, multi-toque
Resolução	336x240 (ppi desconhecido)	320x480 (163ppi)
Capacidade	64MB (cartão PCMCIA)	32 GB
Bateria	5-10 horas em uso contínuo	30 horas em uso contínuo (música)
Dimensões	7.25x4.2x0.75 polegadas	4.5x2.4x0.48 polegadas
Peso	410g	135g
Preço no lançamento	700\$	300\$ (com contrato)

Tabela 1 – Comparação das características entre o MessagePad e o iPhone 3GS

Os sistemas operativos acompanharam a evolução física dos dispositivos, possibilitando, hoje, funções de telefonia, organização, navegação na Internet, reprodução de conteúdo multimédia, jogos e aplicações, etc., basicamente imitando um computador de secretária, excepto que o computador está na palma da mão.

Muitas empresas reconhecem o valor da mobilidade associado ao consumo de conteúdos tem para os consumidores, existindo assim cada vez mais concorrência neste mercado, tanto ao nível de construtores de equipamentos como ao nível de sistemas operativos e serviços. Apesar das potencialidades dos dispositivos móveis, o mercado de dispositivos smartphone⁶ ainda é algo residual comparado ao mercado de telemóveis no geral⁷ [4][5][6][7][8] [9][10][11][12][13], como ilustra a Figura 2 .

⁵ Fotografia de Blakespot, <http://www.flickr.com/photos/blakespot/2379207825/>

⁶ Dispositivos com capacidades de telefonia e de execução de aplicações

⁷ Inclui smartphones.

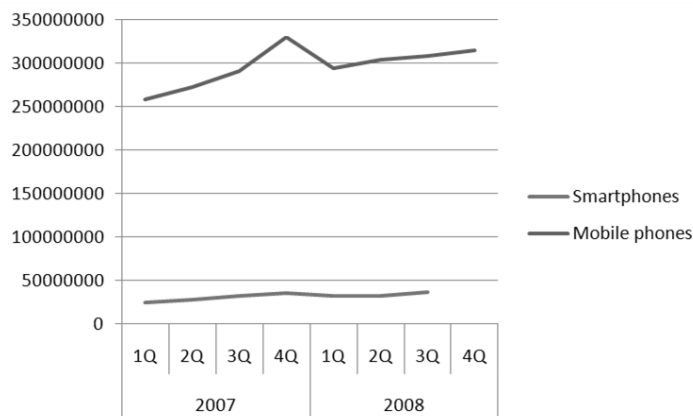


Figura 2 – Volume de vendas de smartphones e telemóveis no geral

O aumento da cobertura da rede móvel tem permitido um acesso mais ubíquo aos serviços online, tornando a mobilidade um conceito atraente aos consumidores, onde os dispositivos móveis não são só como produtos de entretenimento pessoal mas também ferramentas de comunicação e trabalho.

2.2 Desenvolvimento de aplicações para dispositivos móveis

Os dispositivos móveis possuem sistemas operativos otimizados para operarem tendo em atenção as limitações que os dispositivos possuem dada a sua natureza compacta, e também os problemas inerentes à mobilidade. Em 2008, o sistema operativo com maior quota de mercado foi o *Symbian*⁸, dividindo-se maioritariamente o resto do mercado pelo *Windows Mobile*⁹, *iPhone OS*¹⁰ e outros baseados em *Linux* (que inclui o recente *Android*¹¹) [14].

Dada a necessidade de ter em conta vários aspectos específicos a dispositivos móveis como, por exemplo, capacidades de desempenho e armazenamento reduzidos, ou ciclos de vida diferentes das aplicações devido à gestão de energia agressiva por parte dos sistemas operativos móveis, o desenvolvimento de aplicações para estes dispositivos é realizado em ambientes de programação otimizados para este fim. Estes ambientes de desenvolvimento assemelham-se aos ambientes normais, com os mesmos paradigmas de programação, no entanto, expõem funcionalidades que permitem que uma aplicação se integre com os dispositivos. Como o que acontece com os sistemas operativos, o mercado de desenvolvimento é repartido por vários ambientes. E existe uma associação directa entre estes ambientes e os sistemas operativos, isto é, o ambiente de desenvolvimento onde é programada uma aplicação dita em qual sistema operativo a aplicação irá ser executada. A Tabela 2 apresenta os ambientes disponíveis mais

⁸ <http://www.symbian.org/>

⁹ <http://www.microsoft.com/windowsmobile/>

¹⁰ <http://www.apple.com/iphone/>

¹¹ <http://www.android.com/>

relevantes, assim como sistemas operativos respectivos, linguagem de programação e quota de mercado[5].

Ambiente	Sistema operativo	Linguagem	Quota de mercado (2º quarto de 2009) [5]	Promotor principal
.NET Compact Framework[15]	Windows Mobile	C# ou VB .NET	9.0%	Microsoft
Symbian SDK[16]	Symbian	C++	50.2%	Nokia
Java Micro Edition[17]	BlackBerry OS	Java	20.9%	Sun
iPhone SDK[18]	iPhone OS	Objective-C	13.7%	Apple
Android SDK[19]	Android	Java	2.8%	Google

Tabela 2 – Comparação entre os vários ambientes de desenvolvimento

Conforme se pode observar, o *Symbian SDK* é o ambiente mais popular, muito provavelmente, devido à quota de mercado semelhante que a *Nokia* possui no mercado de venda de dispositivos móveis [20]. Note-se que o Java ME é uma plataforma interoperável suportada por vários sistemas operativos móveis, incluindo Windows Mobile, no entanto, apenas é um ambiente de desenvolvimento principal nos dispositivos BlackBerry.

O ambiente *.NET Compact Framework* (NETCF) é uma versão reduzida do ambiente *.NET Framework*, que procura atingir um equilíbrio entre o número de funcionalidades suportadas e o desempenho das aplicações nos dispositivos. No entanto, a NETCF é extensível, o que significa que é possível criar bibliotecas reutilizáveis em várias aplicações e, assim, criar funcionalidades que satisfaçam necessidades específicas, ou diminuir o tempo necessário para o desenvolvimento de uma aplicação. Esta extensibilidade da NETCF permitiu criar um mercado de bibliotecas (comerciais ou livres) como, por exemplo, a *GPS.NET*¹², que permite melhor integração com o GPS do dispositivo, ou a *Smart Device Framework*¹³, que implementa funcionalidades diversas.

A ubiquidade de redes móveis e 802.11 e a crescente popularidade de *web services* permitem criar aplicações que obtenham facilmente dados na Internet. Um *web service* é um serviço instalado num servidor de páginas dinâmicas, que permite uma transmissão de dados bidireccional, fácil, estruturada e interoperável entre o servidor e as aplicações [21][22]. Através do paradigma de invocação remota de métodos¹⁴, a integração com *web services* é quase transparente para o programador, o que significa que o consumo de dados com *web services* pode ser facilmente implementado em aplicações. Actualmente, muitos sites do tipo *Web 2.0*¹⁵ disponibilizam um *web service* que permite a adição e descarga de conteúdos que possuem, e alguns motores de busca também disponibilizam serviços diversos como serviços de pesquisas, serviços geográficos ou serviços de conteúdos multimédia. A utilização de *web services* não está

¹² Geoframeworks, <http://www.geoframeworks.com>

¹³ OpenNetCF, <http://www.opennetcf.com>

¹⁴ Remote Procedure Call (RPC)

¹⁵ <http://oreilly.com/web2/archive/what-is-web-20.html>

limitada a sites orientados ao utilizador e seu conteúdo, mas também a outros sites, por exemplo, comerciais, que pretendam aumentar as formas de acesso aos seus serviços, como o Ebay¹⁶ que, entre outras funcionalidades, permite o acesso ao catálogo dos produtos à venda e a compra destes produtos por parte de utilizadores de aplicações (ou sites) que implementem estes web services.

Assim, é possível criar aplicações dinâmicas, poderosas, inteligentes, visualmente atraentes, com informação interessante, o que permite valorizar a experiência do utilizador num dispositivo e criar um maior leque de cenários de utilização de dispositivos dentro e fora do emprego.

2.3 Aplicações da computação móvel à Vigilância

A Vigilância é uma área que tem especial interesse na aplicação de novas tecnologias, pois estas permitem aumentar o nível de inteligência dos sistemas e, conseqüentemente, o grau de automatização destes, tendo-se verificado um grande interesse da comunidade académica pelas áreas de análise de vídeo e de redes de sensores.

Através da análise do conteúdo de vídeo, com o intuito de ser realizada em tempo-real, é possível dotar os sistemas de Vigilância com capacidades de detecção de objectos específicos como, por exemplo, matrículas de veículos ou itens abandonados, e com capacidades de identificação facial e, também, capacidades de detecção de comportamentos anormais de indivíduos [23][24][25]. As redes de sensores consistem na disposição de sensores pelo espaço que se pretende monitorizar, existindo comunicação com um nó central, com o objectivo de detectar remotamente incidentes e alertar imediatamente os responsáveis da situação, ou, através de sistemas cognitivos, tratar os incidentes de forma automática[26].

Apesar da grande componente móvel associada às actividades de Vigilância, os desenvolvimentos verificados na área da mobilidade são escassos, centrando-se na codificação de vídeo para *streaming* para visualização de videovigilância em dispositivos móveis, ou então analisando problemas muito específicos. Assim, as soluções restringem-se à apresentação de imagens provenientes de videovigilância, possuindo alguma capacidade de comunicação, onde a Figura 3 ilustra um exemplo baseado em aplicação móvel, podendo, também, existir capacidade para tomada de medidas [27] [28][29][30] [31].

¹⁶ <http://developer.ebay.com/products/overview/>



Figura 3 – Exemplo da visualização de videovigilância em dispositivos móveis¹⁷

No contexto da realização de rondas e registo de incidentes, a realização da Vigilância é facilitada e otimizada com a assistência de dispositivos móveis e sistemas de suporte à decisão, pois estes permitem novos cenários de comunicação e, assim, tomar decisões mais informadas, como indica um estudo sobre a capacidade de resposta de uma equipa de Vigilantes na realização de rondas, e tratamento de alarmes e anomalias[32]. Ao nível comercial, a *Morse Watchmans* possui o *Tour-Pro*¹⁸ que, em conjunto com o seu dispositivo móvel *PowerCheck*, ilustrado pela Figura 4, permite gerar relatórios de rondas de vigilância, e informações sobre a ronda realizada. O dispositivo móvel é algo rudimentar, apesar de indicar ao Vigilante qual o próximo ponto de controlo, pois exibe pouca informação acerca da ronda, e os incidentes são registados através de códigos.



Figura 4 – Morse Watchamn’s PowerCheck¹⁹

Desta forma, observa-se uma tendência para uma maior assistência e automatização das tarefas realizadas na Vigilância através da introdução da tecnologia, no entanto, no que diz respeito à Vigilância móvel, esta área centra-se muito na extensão da videovigilância para dispositivos móveis.

¹⁷ Figura proveniente do *paper* “PDA Watch for Mobile Surveillance Services” [26].

¹⁸ PowerCheck, <http://www.keywatcher.com/products/powercheck.shtml>

¹⁹ Fotografia de Morse Watchmans, <http://www.morsewatchmans.com/>

3 Cenários a suportar

3.1 Processo de trabalho actual

O processo de registo de incidentes em vigor na Universidade de Aveiro é descrito a seguir. O Supervisor define rondas, em todo o campus e edifícios, as quais são realizadas por Vigilantes, com o objectivo de detectar quaisquer anomalias nos equipamentos e outros bens. Durante a ronda, o Vigilante regista os incidentes e outra informação relevante num bloco de notas com uma caneta e, no final de cada ronda, o Vigilante transcreve os seus apontamentos para o computador, como ilustra a Figura 5.

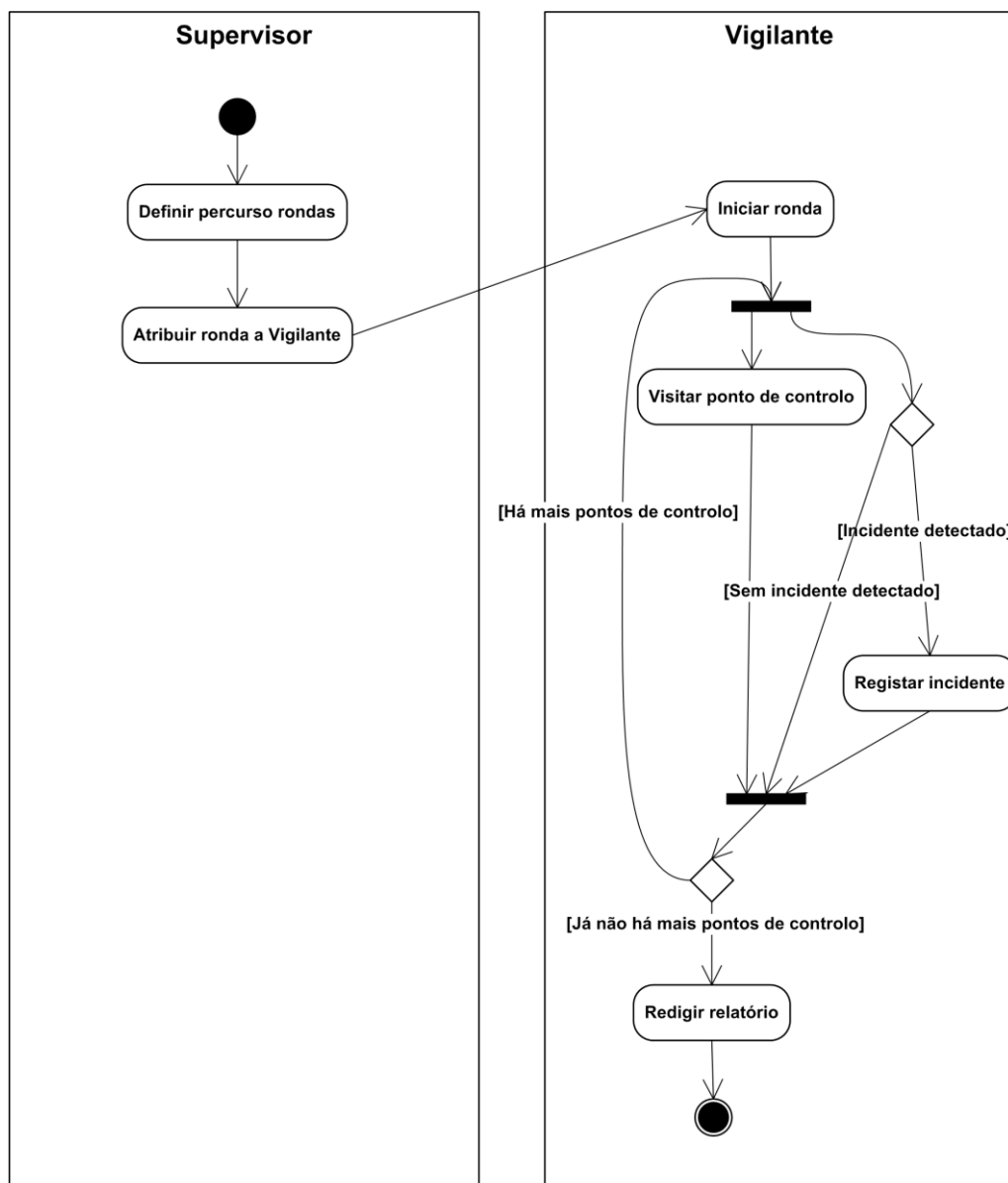


Figura 5 – Fluxo actual de realização de rondas

O Vigilante toma conhecimento do percurso da ronda através de uma descrição textual dos pontos de controlo, o que é uma forma pouco clara e subjectiva para o Vigilante.

Durante a sua ronda, o Vigilante não tem conhecimento quais incidentes foram registados anteriormente, portanto, existe a possibilidade de que um mesmo incidente seja registado várias vezes, aumenta a carga de trabalho necessária para processar os incidentes registados.

Após a realização das rondas, o Supervisor analisa os relatórios redigidos pelos Vigilantes, e redige, por sua vez, um relatório diário com a informação que considere relevante dos relatórios, encaminhando o seu relatório aos Técnicos que, por sua, vez analisam os relatórios e encaminham os incidentes ao responsável – Pivot – do departamento onde o incidente ocorreu, como ilustra a Figura 6.

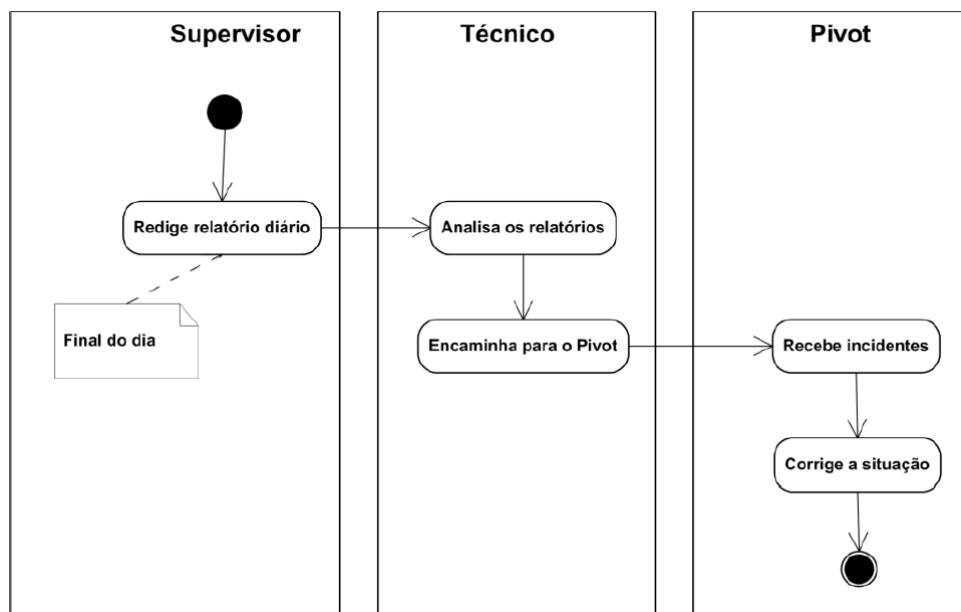


Figura 6 – Fluxo actual de gestão de incidentes

Todo este processo é realizado, inicialmente, em suporte de papel e, finalmente, misto com papel ou correio electrónico, o que significa que este é um processo muito burocrático e que não existe uma organização central dos incidentes. O conteúdo dos relatórios depende dos Vigilantes, podendo introduzir problemas de interpretação.

Apenas no final da ronda os incidentes são submetidos aos responsáveis, tendo que alguns incidentes ainda ser encaminhados para o responsável, o que significa que este processo possui baixa capacidade de resposta.

O Técnico toma conhecimento da localização do incidente através da descrição textual feita pelo Vigilante, portanto, além da descrição ser pouco clara, está sujeita a interpretação subjectiva.

3.2 Novo processo de trabalho proposto

Com recurso à tecnologia, propõe-se um novo processo que desmaterialize o registo de incidentes e o circuito de notificações destes, eliminando a necessidade de redacção dos relatórios e encaminhamento manual dos incidentes para os Pivots, de forma a diminuir o tempo de resposta do processo.

Através de uma interface ao sistema, o Supervisor define as rondas e seus percursos através de pontos de controlo exactos e arbitrários geograficamente, e atribui estas rondas aos Vigilantes respectivos que as realizem como parte dos seus deveres.

O Vigilante realiza a sua ronda através de uma aplicação executada num dispositivo móvel, que utiliza para registar incidentes, controlar a sua ronda e visualizar informações sobre ela, podendo anexar conteúdo multimédia aos registos. Por sua vez e de forma autónoma, durante a realização da ronda, a aplicação adquire a localização do Vigilante e informa periodicamente o Supervisor onde o Vigilante se encontra num dado momento durante a realização da ronda. Esta comunicação é feita de forma imediata caso existam condições de rede para tal, ou em diferido, quando as condições estejam novamente reunidas e, desta forma, o Supervisor pode observar a realização das rondas e detectar eventuais problemas.

O registo de incidentes é assistido pelo dispositivo móvel, onde o Vigilante apenas tem de seleccionar, ao invés de introduzir, a informação necessária para o registo, e este registo é enviado imediatamente para o sistema, onde fica imediatamente visível para o Supervisor e para o Técnico. No final da ronda do Vigilante, apesar de desnecessários para o novo processo, o sistema redige automaticamente o relatório da ronda, assim como o relatório diário, apenas para efeitos de visualização pelos funcionários. Esta fase do fluxo de trabalho é ilustrada pela Figura 7.

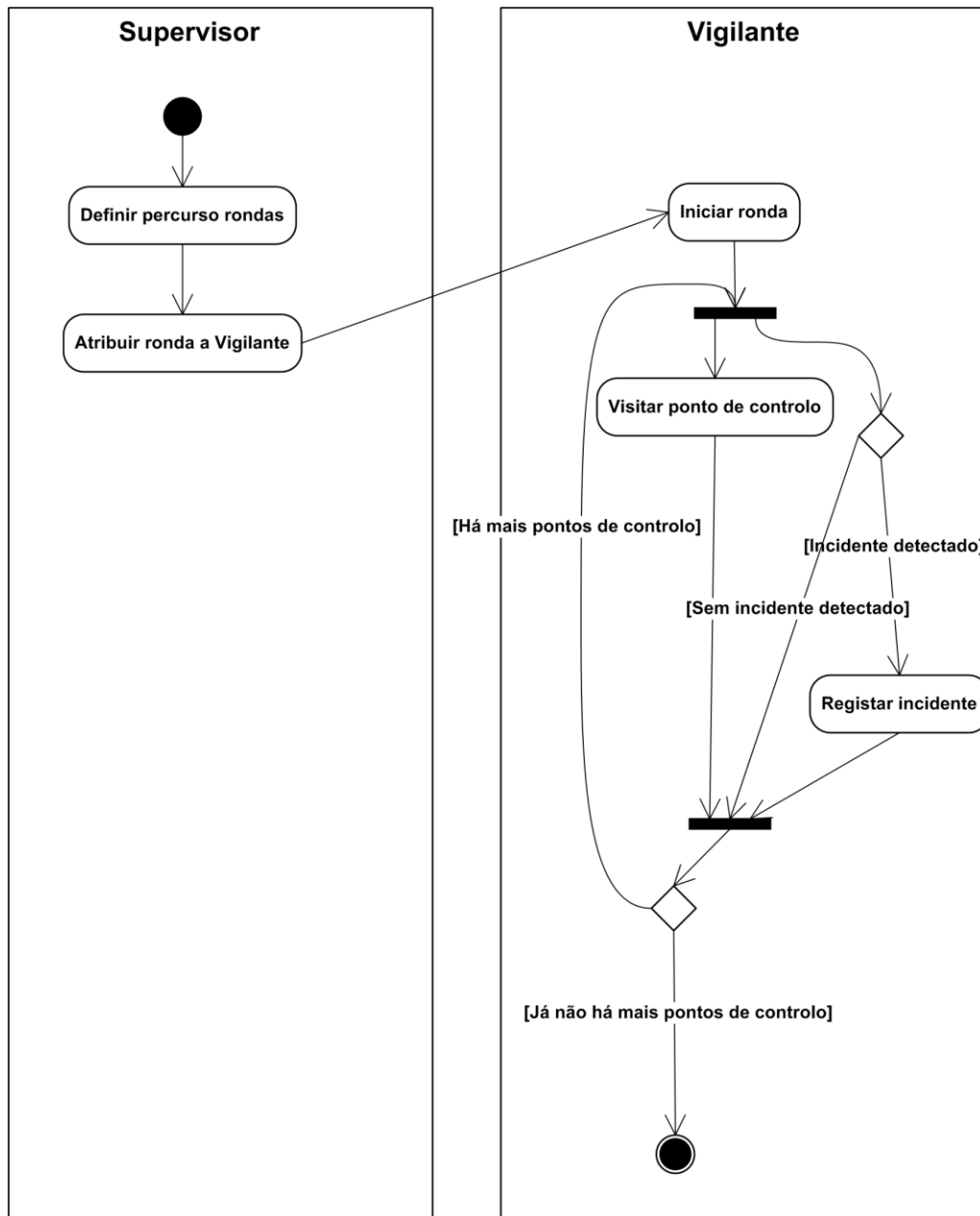


Figura 7 – Fluxo proposto para realização das rondas

Sem qualquer intermediário, o Pivot recebe imediatamente os incidentes registados pelo Vigilante, como ilustra a Figura 8.

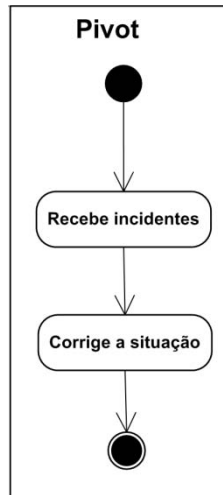


Figura 8 – Fluxo de trabalho proposto para o Pivot

Com este processo, o Supervisor apenas tem que definir e atribuir rondas e os Vigilantes apenas têm que as realizar e registar incidentes, visualizáveis imediatamente pelo Pivot. Desta forma, o processo deixa de ser burocrático, e os incidentes são imediatamente conhecidos.

3.3 Casos de utilização

O sistema conta com os seguintes actores: Vigilante, Supervisor, Técnico e Pivot. O actor Vigilante interage com o sistema através de um módulo móvel (SegUAMóvel), enquanto os actores Supervisor e Pivot interagem através de um módulo centralista (SegUAWeb). O actor Técnico pode interagir com o sistema através dos módulos centralista e móvel.

O Supervisor tem o papel de gerir a realização de rondas, o que inclui criar rondas (“Editar rondas”) e seus pontos de controlo associados (“Editar pontos de controlo”). Pode, também, visualizar a lista de rondas em curso (“Visualizar rondas em curso”), o arquivo de rondas realizadas no passado (“Visualizar rondas realizadas”), e visualizar em detalhe estas rondas. O detalhe destas rondas inclui os incidentes registados nelas, os pontos de controlo verificados e o percurso real do Vigilante. Pode, também, criar novos utilizadores do tipo Vigilante (“Editar utilizadores”). O Técnico pode criar novos utilizadores do tipo Técnico (“Editar utilizadores”) e popular a lista de incidentes possíveis (“Categorizar incidentes”) e visualizar os incidentes registados (“Monitorizar incidentes registados”). O Pivot pode apenas visualizar a lista de incidentes ocorridos no seu departamento (“Monitorizar incidentes registados”). A Figura 9 ilustra o caso de utilização dos actores Supervisor, Técnico e Pivot no módulo centralista.

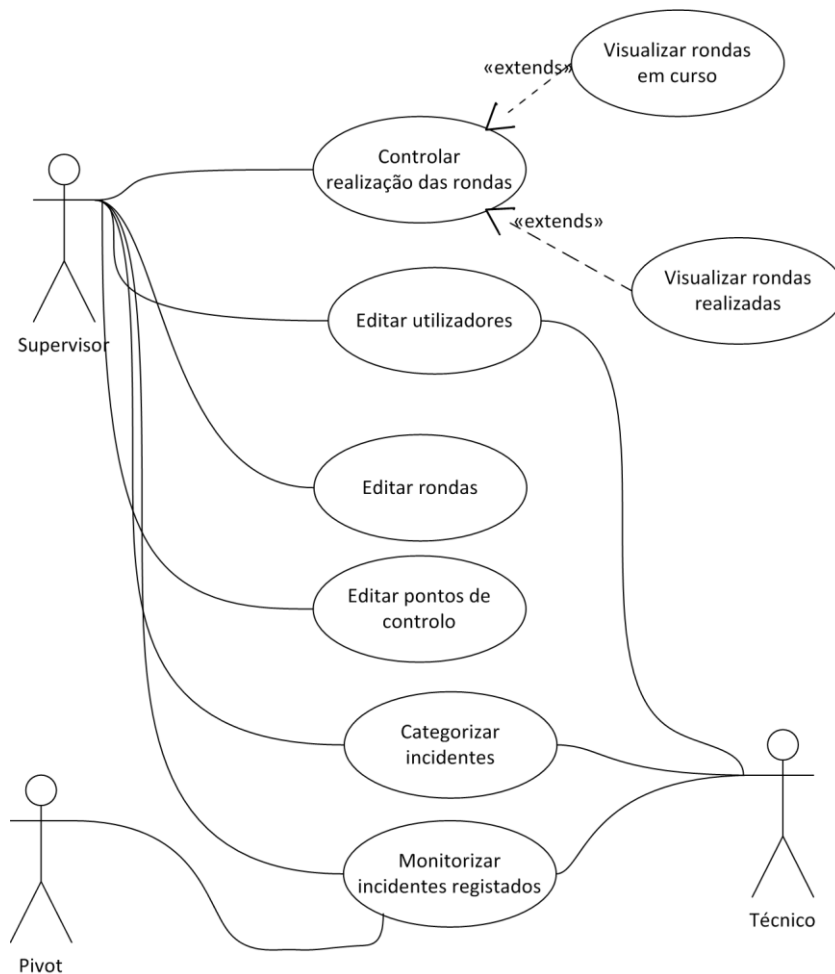


Figura 9 – Casos de utilização no módulo centralista

O Vigilante pode pré-visualizar as rondas (“Pré-visualizar rondas”) e escolher qual a ronda que pretende realizar (“Realizar ronda”), podendo continuar uma ronda que não tenha terminado anteriormente. Pode verificar o percurso da ronda através de uma lista de pontos de controlos ou através de uma vista geográfica. Durante esta ronda, pode visualizar que incidentes foram registados no local onde se encontra através de uma lista ou através de uma vista geográfica (“Visualizar incidentes”), e caso o incidente que pretende registar ainda não tenha sido registado anteriormente (cabendo a escolha ao Vigilante), pode registar o incidente (“Registar incidente”). Para este registo, pode escolher a família de incidentes, a situação e a causa do incidente, assim como associar uma localização geográfica (interior ou exterior) e anexar conteúdo multimédia (“Anexar conteúdo multimédia”). Durante a ronda, o Vigilante pode indicar manualmente que transitou por um ponto de controlo (“Indicar ponto de controlo transitado”), ou emitir um alarme em caso de emergência ou necessidade de assistência por parte de outros Vigilantes ou Supervisor.

O Técnico tem o mesmo papel que o Vigilante no que diz respeito ao registo de incidentes no módulo móvel, podendo visualizar que incidentes foram registados no local onde se encontra (“Visualizar incidentes”), através de uma lista ou através de uma vista geográfica, registar um

novo incidente (“Registrar incidente”) e anexar multimédia (“Anexar conteúdo multimédia”). A Figura 10 ilustra os casos de utilização dos actores Vigilante e Técnico, no módulo móvel.

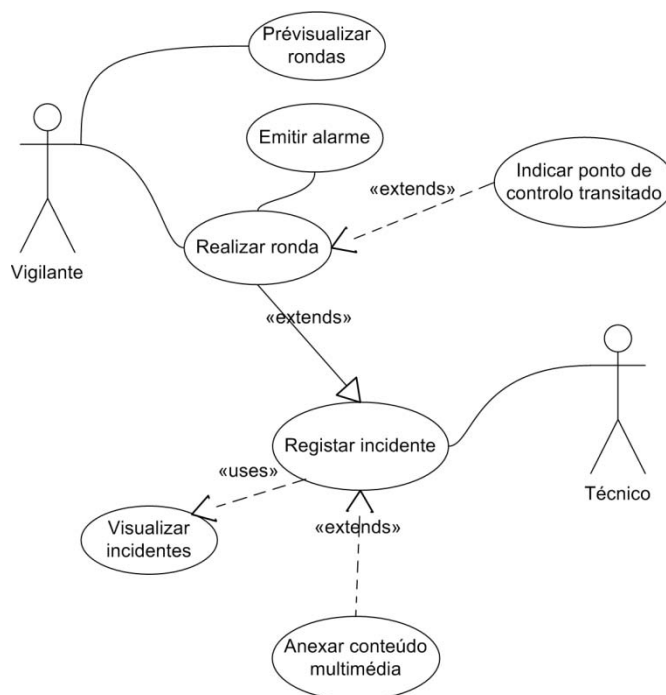


Figura 10 – Casos de utilização no módulo móvel

3.4 Necessidades de integração de sistemas

Os Serviços Técnicos possuem um sistema de informação geográfica – WebSIG – onde gerem todo o tipo de instalações, como canalização ou rede eléctrica, assim como todos os objectos que existem dentro dos edifícios como, por exemplo, lâmpadas. A partir deste sistema, é possível visualizar plantas de edifícios de acordo com várias vistas como, por exemplo, arquitectónica, eléctrica, etc., e identificar e localizar exactamente um objecto específico, existindo, assim, oportunidade para assistir e melhorar o registo de incidentes através da indicação do objecto em questão no incidente e através da visualização de plantas para especificar exactamente a localização do incidente, caso não se trate de um objecto referenciado.

Assim, a integração de sistemas é uma mais-valia para este projecto, pretendendo-se que os actores tenham acesso às plantas e às listas de objectos sempre que for necessário registar ou visualizar um incidente. No contexto do registo de incidentes, consideram-se os seguintes dados como relevantes para serem integrados no sistema: edifícios, pisos, suas plantas arquitectónicas, salas e objectos existentes. Na prática, a integração destes dados é feita desde a base de dados WebSIG e a base de dados do sistema – SegUABD – que, posteriormente, se propaga para a base de dados da SegUAMóvel – SegUAMóvelBD, como ilustra a Figura 11.

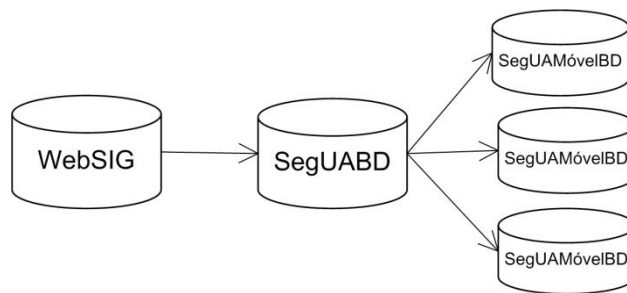


Figura 11 – Integração de bases de dados

Dada a pouca frequência de alterações de dados que se verifica no WebSIG e o elevado volume de dados existentes, uma integração periódica poderá ser suficiente, se não mais conveniente.

4 Desenho da solução

4.1 Arquitectura do sistema

O sistema SegUA deverá ser capaz de armazenar dados para arquivo e permitir a consulta através de uma interface *web*; e possibilitar a transmissão desses dados para os terminais móveis. Deverá, também, ser capaz de integrar os dados geográficos provenientes do sistema SIG em uso pela Universidade. Desta forma, o sistema é composto por uma base de dados central – SegUABD – que integra e armazena os dados necessários para realização de rondas e registo de incidentes, uma aplicação móvel – SegUAMóvel – que assiste o registo de incidentes, uma interface *web* que permite a edição e visualização dos dados – SegUAWeb – e o sistema de informação geográfica da Universidade de Aveiro (WebSIG).

Dado que a utilização do sistema, *a priori*, não será excessiva e não existindo necessidade de que as várias SegUAMóveis comuniquem entre si, o modelo de comunicação entre as SegUAMóveis e a SegUABD pode ser do tipo cliente-servidor. Este modelo é simples de implementar pois os cenários de comunicação são conhecidos e a centralização dos dados reduz a complexidade de implementação, comparando, por exemplo, a uma arquitectura distribuída. A centralização inerente ao modelo cliente-servidor pode ser inconveniente pois a SegUABD será um ponto de falha único, no entanto, este problema pode ser minimizado tanto do lado do servidor, como do lado do cliente. Do lado do servidor através de virtualização e através de políticas de replicação de dados; do lado do cliente através de armazenamento local e através de algoritmos inteligentes, por exemplo, envio retardado de informação. Desta forma, a comunicação entre a SegUABD e a SegUAMóvel não é mandatária para a operação da SegUAMóvel. A Figura 12 ilustra o sistema.

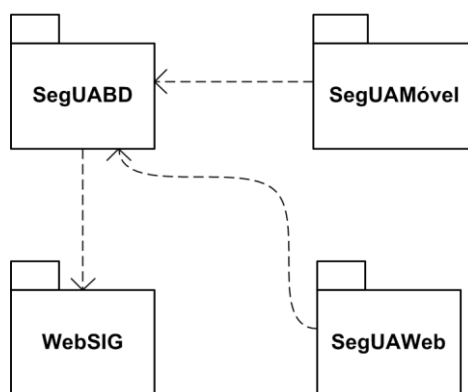


Figura 12 – Arquitectura inicial do sistema SegUA

A SegUAMóvel pode ser implementada de duas formas: aplicação no dispositivo ou site otimizado para navegador de páginas móvel, alojado num servidor. Estas duas aproximações devem ser analisadas em detalhe, pois possuem características que ditam o modo de funcionamento e funcionalidades da SegUAMóvel.

Num cenário de implementação em site otimizado, as funcionalidades estão centralizadas e, portanto, este cenário é mais fácil de implementar e gerir. É necessário que exista conectividade e largura de banda suficiente sempre que o utilizador deseje executar uma operação ou visualizar informações. Toda a visualização é feita através de páginas web, o que significa que poderá ser dispendido muito tempo à espera que as páginas descarreguem totalmente, especialmente se estas possuírem imagens. Não é possível usufruir das possibilidades da adaptação ao contexto espacial, pois não existe meio de aceder às funcionalidades nativas de localização que existem no dispositivo.

A execução de uma aplicação no dispositivo permite criar uma solução mais autónoma e, assim, lidar melhor com os problemas apresentados anteriormente, nomeadamente de largura de banda e intermitência de rede através da transferência de actualizações ao invés de descargas contínuas, e do armazenamento de dados temporários localmente [19]; e da obtenção da localização através das funcionalidades nativas do dispositivo [33]. Apesar de exigir dispositivos com maior desempenho, uma aplicação permite satisfazer os requisitos deste projecto sendo, então, a aproximação escolhida.

Existem questões técnicas que devem ser tidas em conta, nomeadamente de interoperabilidade entre os vários elementos do sistema. Tecnologias da mesma empresa, geralmente, funcionam entre si sem quaisquer problemas, no entanto, entre tecnologias de diferentes empresas o mesmo poderá não acontecer. A interoperabilidade é uma questão importante pois, numa arquitectura com vários elementos de diferente natureza, como esta, é importante que seja possível escolher a melhor tecnologia específica para cada elemento e seus objectivos, sem que isso afecte os outros elementos.

Os *web services* permitem a troca de dados de uma forma simples e interoperável, pois implementam normas públicas que muitas tecnologias, hoje em dia, suportam. Os *web services* garantem interoperabilidade pois podem ser auto-descritos através da norma WDSL e os dados são trocados em formato XML, cuja estrutura segue a norma SOAP, e, assim, reconhecidos por qualquer sistema ou ambiente de programação que suporte *web services* [21] [34] [35] [36] [37].

Assim, é introduzido um novo elemento no sistema - SegUAService – que implementa um *web service*, e funcionará como ponte entre os vários elementos. Além da interoperabilidade, este novo elemento na arquitectura introduz várias vantagens, nomeadamente, a centralização do fluxo de informação. A centralização do fluxo é de extrema relevância, pois permite esconder a proveniência dos dados, isto é, tornar a fonte de dados transparente ao receptor de informação; permite estruturar adequadamente a informação para enviar ao elemento receptor; e permite conter a propagação de alterações na arquitectura. Desta forma, poderia existir um número arbitrário de fontes externas que o SegUAMóvel apenas veria uma[38]. O SegUAService poderá, também, efectuar validação de dados e controlo de acessos, o que poderá ser um aspecto importante no que diz respeito à segurança e integridade do sistema.

O SegUAService permite, assim, modularizar a arquitectura do sistema. Assim, a arquitectura ideal será a arquitectura ilustrada pela Figura 13.

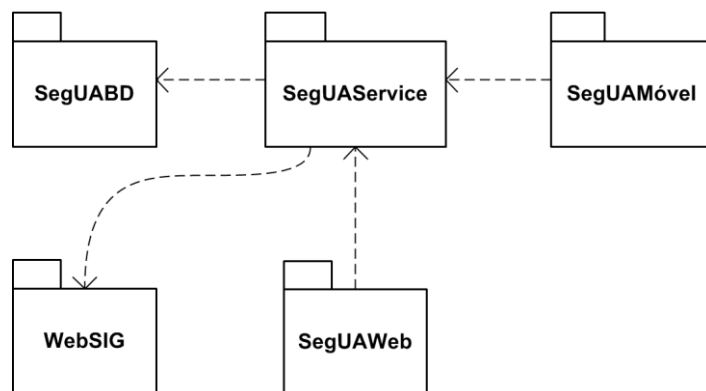


Figura 13 – Arquitectura revista do sistema SegUA

4.1.1 Módulo móvel

Existem vários obstáculos que dificultam a implementação de aplicações [39][40], nomeadamente:

- Heterogeneidade das redes,
- Gestão da energia,
- Desempenho do dispositivo,
- Variedade de características físicas dos dispositivos,
- Existência de vários ambientes de desenvolvimento incompatíveis entre si,
- Dificuldade do desenho de uma interface devido à dimensão reduzida dos ecrãs,
- Grande número de cenários possíveis inerentes à mobilidade.

A solução para lidar com estes obstáculos passa primeiro por boas práticas de programação, nomeadamente através de uma arquitectura robusta. A arquitectura da aplicação móvel (SegUAMóvel) é multi-camada com base na arquitectura dos 3 níveis²⁰: Acesso a dados, Lógica e Apresentação [41][42][43]. A Figura 14 ilustra a arquitectura mencionada.

²⁰ Do inglês “three tier architecture”.

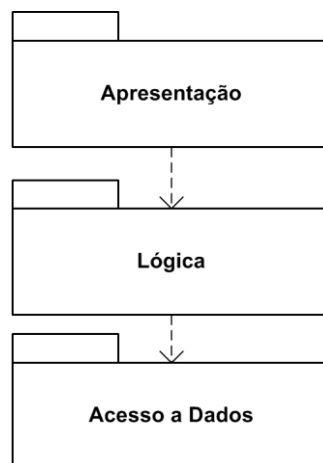


Figura 14 – Arquitectura de 3 níveis

Uma arquitectura multi-camada permite encapsular as tarefas e, assim, facilitar o desenvolvimento da aplicação, e permite minimizar a propagação de alterações quando uma camada necessita de ser alterada.

A camada Acesso a Dados é responsável pela transacção de dados entre a aplicação e o meio de persistência, por exemplo, o sistema de ficheiros ou base de dados móvel. É uma camada de tradução entre o modelo de dados e o meio de persistência, isto é, é a camada que obtém os dados de um meio de persistência e instancia os objectos com significado semântico que representam esses dados, e envia os objectos para a camada de Lógica; e efectua a operação inversa.

Dado que os dispositivos móveis possuem várias tecnologias de transferência de dados incorporadas, como *Bluetooth* ou Infravermelhos, a natureza modular desta arquitectura é claramente uma vantagem, pois a camada Acesso a Dados permite integrar facilmente estas tecnologias sem que as camadas superiores se apercebam que existem várias fontes de dados.

A camada Lógica implementa as funcionalidades da aplicação e interpreta os dados provenientes das camadas adjacentes. É a camada nuclear da aplicação, pois, como se verá mais à frente, centraliza todas as camadas e gere aspectos do dispositivo, caso isso seja necessário.

A camada de Apresentação é responsável por apresentar a informação ao utilizador, interagir com este e enviar a informação introduzida pelo utilizador à camada de Lógica. A Apresentação é a materialização do fluxo de trabalho da aplicação.

Porém, esta é uma arquitectura genérica, por isso tem que ser desenvolvida para se adaptar à realidade da aplicação ser executada num dispositivo, isto é, é necessário ter em conta as funcionalidades específicas a um dispositivo móvel como, por exemplo, telefonia; e ter em conta requisitos da aplicação para com o dispositivo como, por exemplo, o funcionamento contínuo, para controlar a ronda. Assim, apresentam-se, a seguir, as camadas necessárias para suportar a realidade mencionada.

A camada Geolocalização fornece acesso aos dados do GPS embutido no dispositivo, fornecendo condições para implementar sensibilidade espacial na aplicação.

A camada Energia permite obter e controlar o estado de energia do dispositivo. A sensibilidade ao estado de energia do dispositivo é uma funcionalidade importante em aplicações móveis, pois o consumo de energia é uma dura realidade e o esgotamento de energia poderá ser um evento que ponha em risco o trabalho do utilizador.

A camada Telefonía permite aceder às funcionalidades de telefonía do dispositivo, como enviar mensagens²¹.

A camada Outros implementa pequenas funcionalidades como vibração, detecção de teclado físico, entre outros.

Esta arquitectura é ilustrada pela Figura 15.

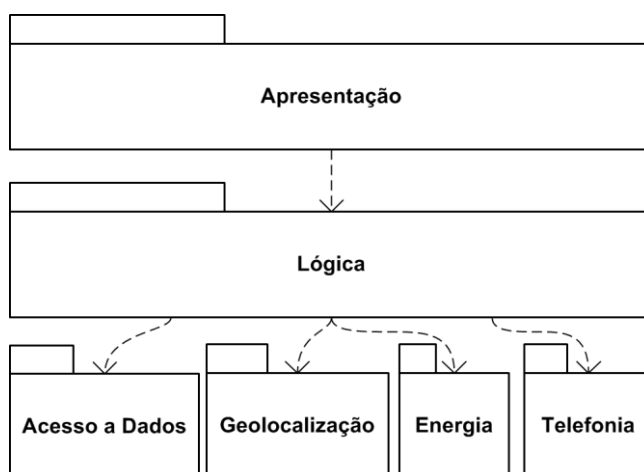


Figura 15 – Arquitectura da SegUAMóvel

4.1.2 Módulo central

O SegUAService é o elemento da arquitectura que centraliza as comunicações, tanto com os dispositivos móveis como com os sistemas externos integrados no sistema, convertendo e encaminhando dados, sendo um elemento fundamental para a interoperabilidade entre os vários elementos da arquitectura.

A arquitectura do SegUAService deve ser multi-camada e modular, de forma a abstrair os serviços expostos com os serviços de dados integrados ao SegUAService e permitir fácil integração com novos serviços de dados. Mais uma vez, a arquitectura do SegUAService baseia-se na arquitectura dos três níveis, estando ilustrada pela Figura 16.

²¹ Short Message Service (SMS)

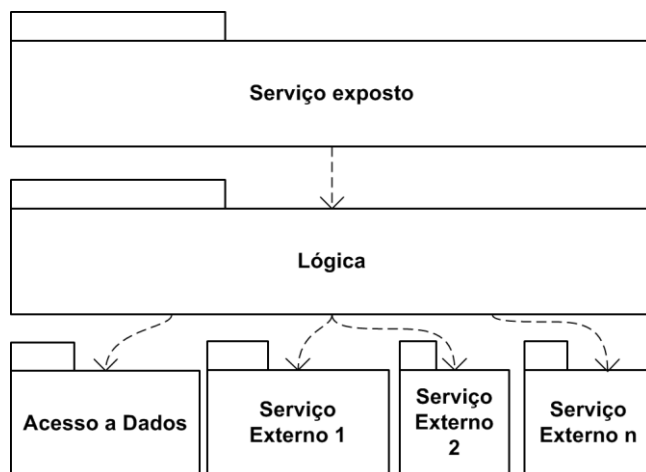


Figura 16 – Arquitectura do SegUAService

A camada Serviço Exposto é a camada que é visível às aplicações e com que estas interagem, sendo nesta camada onde devem ser aplicadas as medidas de segurança do SegUAService, nomeadamente o controlo de acessos. A camada de Controlo é a camada que possui a lógica de negócio, que se encarrega de reunir os dados necessários das várias fontes, e executar as operações estabelecidas. A camada de Acesso à SegUABD encapsula o acesso à SegUABD, convertendo os dados recebidos em objectos e vice-versa. A camada Serviço Externo encapsula o acesso e o formato de dados dos serviços externos aos quais o sistema está integrado, devendo esta camada ser modular para permitir a integração de novos serviços.

No sentido de abstrair o SegUAService da modelação de dados e de conter alterações nesta, a SegUABD também deve implementar uma arquitectura multi-camada. O SegUAService comunica com a SegUABD através de uma interface de procedimentos²² ou vistas²³, e, desta forma, o SegUAService apenas necessita de saber os parâmetros esperados pelos procedimentos. Assim, a SegUABD deverá implementar a arquitectura ilustrada pela Figura 17.

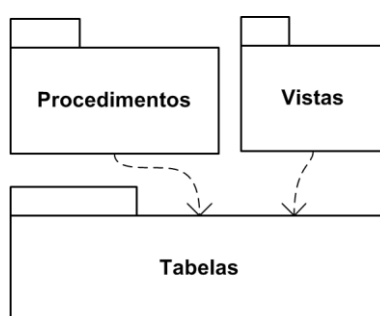


Figura 17 – Arquitectura da SegUABD

²² Stored procedure

²³ SQL view

4.2 Comportamento adaptativo

4.2.1 Adaptação ao contexto espacial

Face à mobilidade da sua utilização, a SegUAMóvel deve ser sensível ao contexto espacial, não só para satisfazer os requisitos da aplicação, mas também para melhorar a qualidade de informação apresentada ao utilizador, tentando apresentar apenas informação relevante ao local onde o utilizador se encontra.

Assim, a aplicação deve ser sensível a:

- Passagem do Vigilante por um ponto de controlo;
- Entrada ou saída do utilizador de um edifício, com salvaguarda de dados inseridos manualmente, relativos ao local do utilizador;
- Apresentação de incidentes ocorridos num local específico;
- Percurso efectuado pelo Vigilante.

A aplicação deve, autónoma e periodicamente, determinar se a localização actual do Vigilante corresponde a um ponto de controlo definido na ronda que o Vigilante está a realizar. Caso exista alguma correspondência, o ponto de controlo será anunciado como visitado ao SegUAService e ao utilizador; caso este ponto de controlo seja o último da ronda, o Vigilante é notificado de que terminou o seu percurso. Caso seja possível obter uma localização válida, que significa que existe recepção de sinal GPS, a aplicação assume que o utilizador se encontra no exterior e, portanto, a localização será comparada com as localizações de pontos de controlo do tipo exterior, cada um definido por uma coordenada GPS. Caso não seja possível obter uma localização, que significa que o componente GPS não está em linha de vista com satélites, a aplicação assume que o Vigilante se encontra no interior de um edifício, e portanto, é incapaz de determinar se um ponto de controlo interior foi visitado, sendo necessário que o Vigilante indique, manualmente, através da interface da aplicação que o ponto de controlo foi visitado. Nesta situação, a aplicação notifica o Vigilante de que é necessário que o Vigilante indique manualmente os pontos de controlo. Desta forma, o acto de realizar o percurso de uma ronda é automatizado, se todos os pontos de controlo forem do tipo exterior, parcialmente automatizado se existirem alguns do tipo interior, e totalmente manual se todos forem do tipo interior. A Figura 18 ilustra este comportamento.

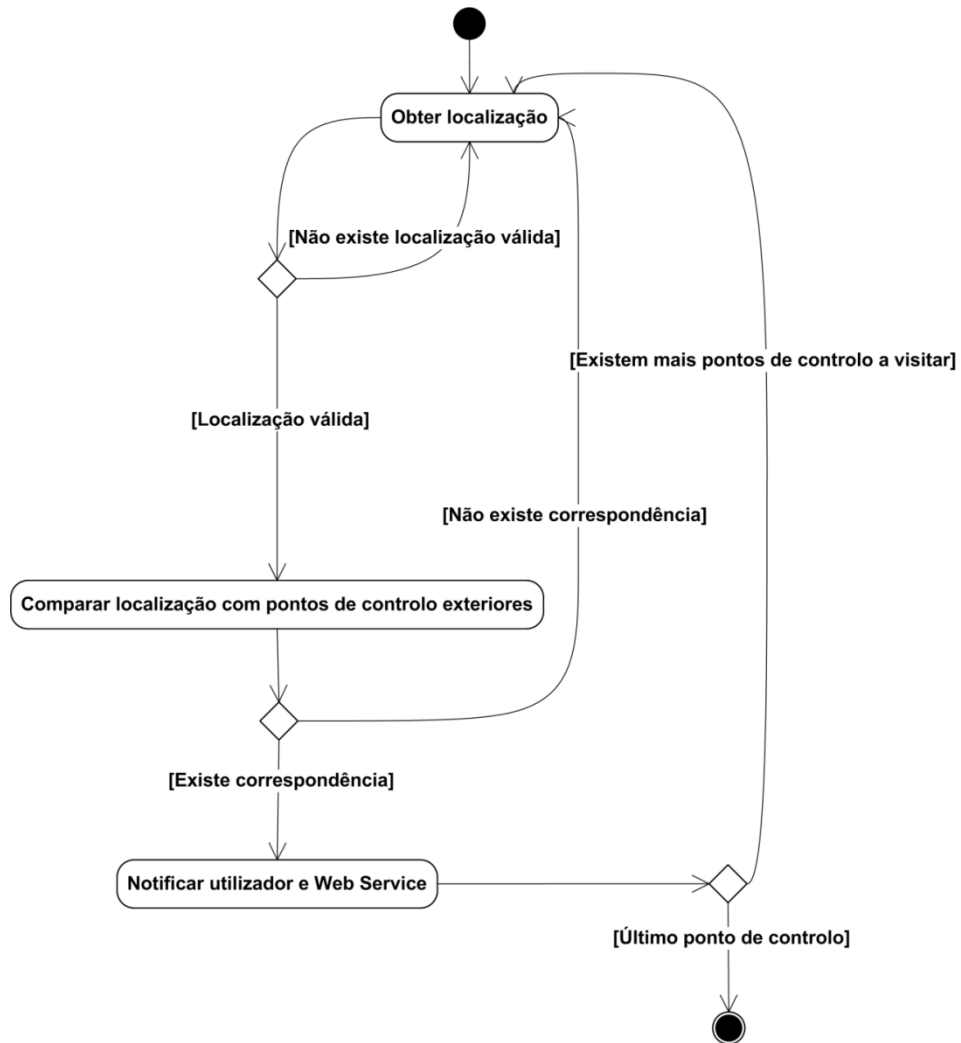


Figura 18 – Comportamento autónomo de realização de ronda

Com conhecimento da área geográfica ocupada pelos edifícios do campus, a aplicação é capaz de identificar o edifício que se encontra nas imediações do utilizador, e, caso a aplicação deixe de ter recepção de sinal GPS, esta assume que o utilizador entrou no edifício ao qual corresponda a última localização válida. Assim que a aplicação encontrar uma correspondência, a identificação do edifício é armazenada. Assim que perder o sinal GPS, a aplicação tem conhecimento de qual edifício o utilizador se encontrava perto, e assim assumir de que o utilizador entrou no edifício identificado. O utilizador pode, ainda assim, seleccionar outras localizações, incluindo exteriores. Através deste processo, a aplicação pode alternar automaticamente o mapa apresentado ao utilizador, entre o mapa do campus, quando existe recepção de sinal GPS, e o mapa do primeiro piso do edifício identificado. Desta forma, não é necessário que o utilizador defina manualmente em qual edifício se encontra. A Figura 19 ilustra este comportamento.

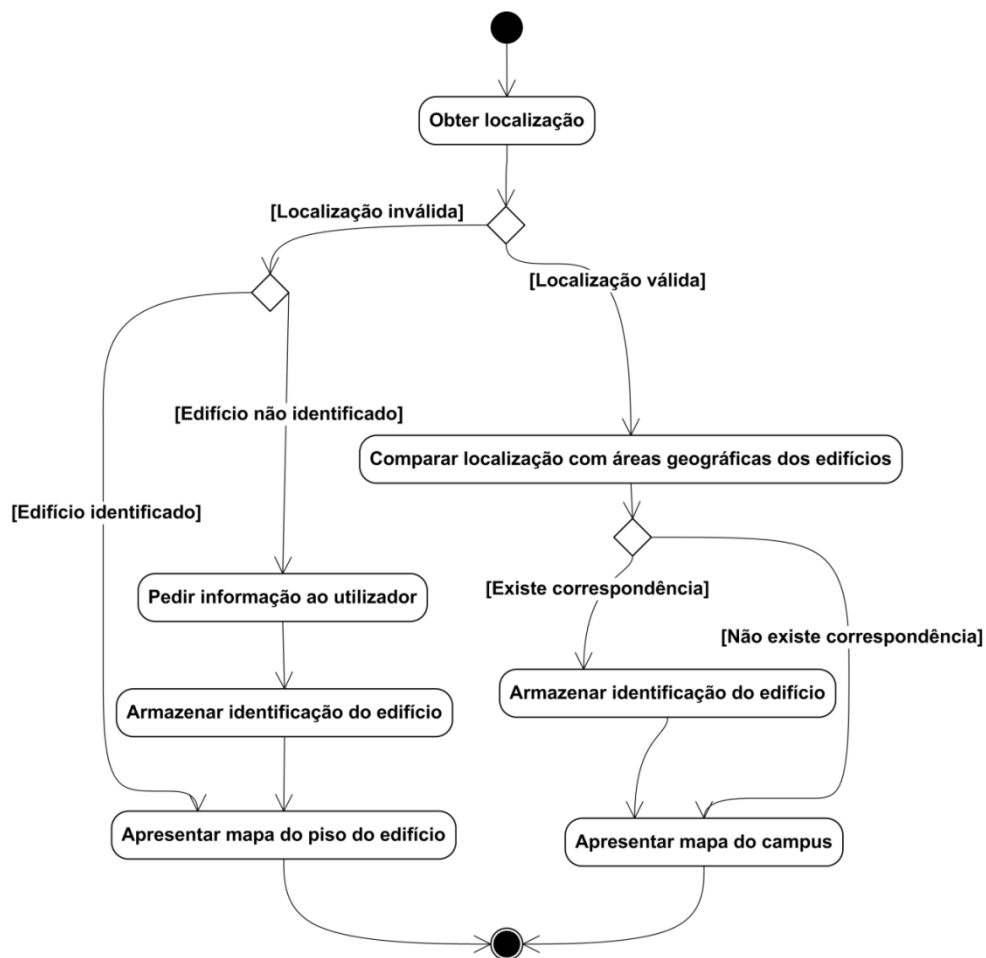


Figura 19 – Comportamento autónomo de apresentação de mapas

Através da localização, obtida pelo componente GPS no caso exterior ou pela identificação do edifício no caso interior, é possível efectuar uma selecção dos incidentes apresentados ao utilizador, apresentando apenas aqueles que ocorreram na localização actual do utilizador e, assim apresentar apenas informação relevante.

Durante a ronda do Vigilante, a aplicação não controla activamente o percurso efectuado pois, além do caminho entre os pontos de controlo não estar especificado, é muito complexo implementar uma forma de analisar o terreno, que também depende da qualidade de representação pelos dados do WebSIG, como também implementar formas de cálculo de rotas. Desta forma, a aplicação apenas regista o percurso efectuado com leitura periódica de localizações, enviando-as ao SegUAService se existir conectividade.

4.2.2 Adaptação à variação das condições de rede

A aplicação não pode assumir de que as condições de rede no campus ou edifícios são estáticas, pois existem vários factores que interferem com a recepção de um sinal de rede como, por exemplo, a distância ou obstáculos entre os pontos de acesso e a SegUAMóvel, e mesmo a passagem²⁴ entre pontos de acesso. Face a esta incerteza, a aplicação deve estar preparada para lidar com problemas, principalmente a falta de conectividade. Estando a informação armazenada na memória volátil, existe um risco elevado de perda de dados em caso de falha do dispositivo como, por exemplo, esgotamento de energia.

Assim, a utilização de um meio de persistência local permite minimizar o risco de perda de informação, pois existirá uma cópia da informação que poderá ser facilmente recuperada e, assim, continuar o trabalho posteriormente, quando a falha do dispositivo estiver resolvida.

No caso de falha de conectividade, a aplicação pode recuperar os dados e enviá-los quando a conectividade estiver restabelecida, ou a pedido do utilizador. A Figura 20 ilustra este comportamento.

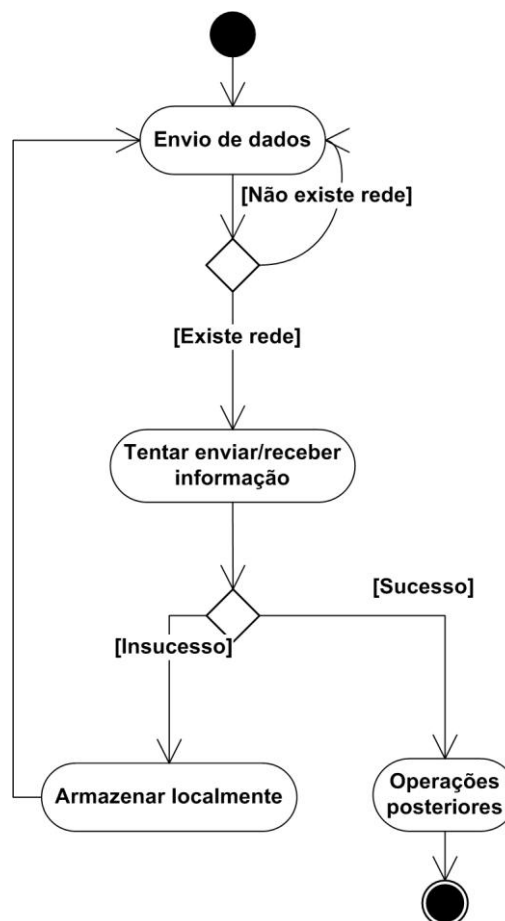


Figura 20 – Comportamento no envio de dados

²⁴ Hand-over

Assim, no que diz respeito a informação criada pela aplicação, é necessário adoptar uma estratégia de persistência, e desta depende o equilíbrio entre salvaguarda da informação e desempenho da aplicação. Para maximizar a salvaguarda de informação, a aplicação pode armazenar primeiro a informação no meio de persistência local, e após tentar o envia pela rede, posteriormente eliminando a informação enviada do meio de persistência local. Desta forma, ao armazenar o mais cedo possível, existirá menor probabilidade de perda de dados.

4.2.3 Adaptação à disponibilidade de energia

O consumo de energia é uma realidade que afecta os dispositivos móveis, e a perda de energia pode implicar perda de informação. Dado que parte da aplicação funciona em modo automático e com acessos frequentes ao componente GPS e rede sem fios, o consumo de energia é substancial. Assim, é necessário que a aplicação seja sensível ao nível de energia, adoptando um comportamento variável conforme este nível.

O comportamento energético deve ter em conta não só dos requisitos da aplicação, mas também a tarefa em curso, pois, se a tarefa em curso for uma acção longa ou crítica para o utilizador ou aplicação, uma falha do dispositivo pode ter consequências graves. Uma acção longa é uma acção que demore muito tempo (na ordem dos minutos) a ser realizada; uma acção crítica é uma acção interna à aplicação onde uma falha pode corromper os dados ou a própria aplicação.

A única acção longa que o utilizador pode fazer é a realização de uma ronda, por parte do Vigilante sendo, portanto, uma acção vital para este utilizador, e a aplicação deve então controlar o nível de energia e modificar o seu comportamento, caso necessário, para evitar interromper a ronda. A Figura 21 ilustra este comportamento genericamente. Durante outras acções (curtas) do utilizador, a aplicação apenas necessita de notificar o utilizador sobre o baixo nível de energia, cabendo a este tomar as medidas que julgue necessárias.

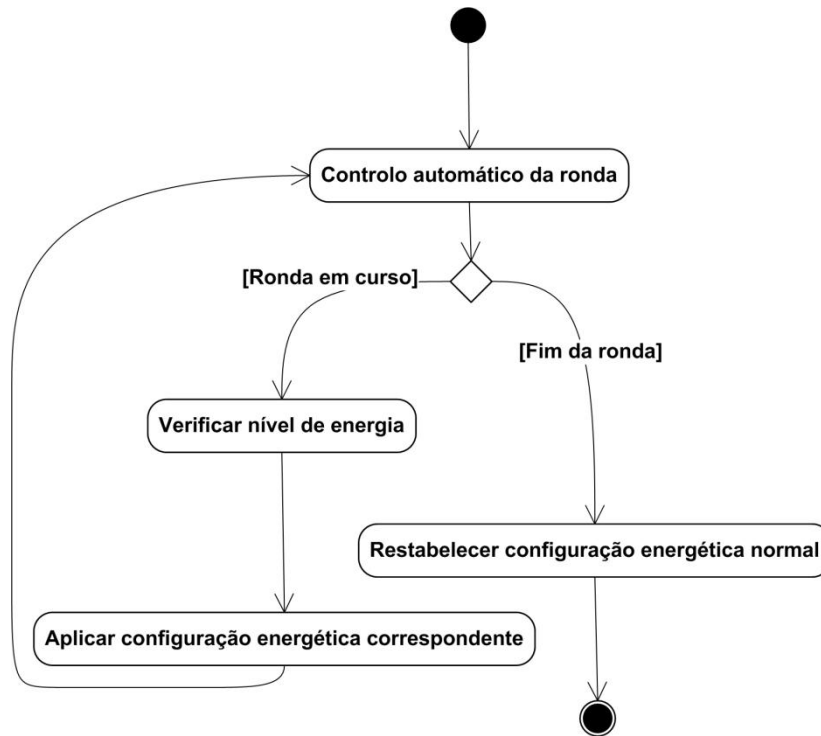


Figura 21 – Comportamento energético durante a ronda

Quanto a acções críticas da aplicação, por exemplo, a auto-actualização da aplicação, este tipo de acções não deverá ser realizada quando o nível de energia do dispositivo estiver baixo, para impedir corrupção de dados.

O comportamento energético consiste numa estratégia de modificação da configuração energética do dispositivo, isto é, consiste na gestão do estado eléctrico de componentes do dispositivo, desactivando componentes quando é atingido um certo nível de energia. Assim, com um menor consumo de energia, o dispositivo poderá permanecer mais tempo em funcionamento e, desta forma, permitir ao utilizador que termine a sua tarefa ou que tome medidas como, por exemplo, trocar a bateria do dispositivo ou reduzir a luminosidade do ecrã.

Caso a rede sem fios esteja desligada, não será possível enviar dados para o SegUAService, naturalmente. Como não é desejável que a informação fique “fechada” no dispositivo indeterminadamente enquanto não se atingir um nível de energia confortável para activar os componentes físicos, a aplicação pode tentar enviar a informação armazenada apenas quando o utilizador terminar a sua sessão na aplicação e, desta forma, a tarefa do utilizador não é interrompida pela possível falha do dispositivo devido a um esgotamento de energia. Modelo de dados

4.3 Modelo de dados

Existem quatro tipos de utilizadores: Vigilante, Técnico, Supervisor e Pivot, ilustrados pela Figura 22, que derivam de um Utilizador.

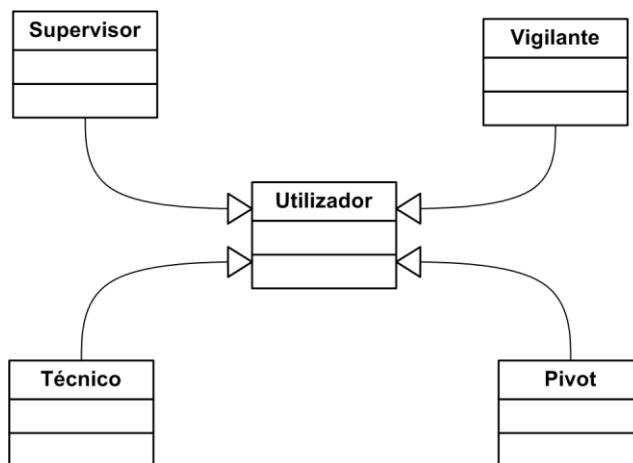


Figura 22 – Modelo de dados dos utilizadores

O Vigilante pode realizar (RondaRealizada) rondas (Ronda), cujos percursos são definidos por pontos de controlos (PontoControlo), ordenados (OrdemPontoControlo). Um ponto de controlo pode ser interior (PontoControloInterior), isto é, uma localização específica no piso (Piso); ou pode ser exterior (PontoControloExterior), isto é, uma localização específica no campus. O percurso real é registado através do registo periódico de localizações obtidas (LocalizaçãoRegistada) da SegUAMóvel. Ilustrado pela Figura 23.

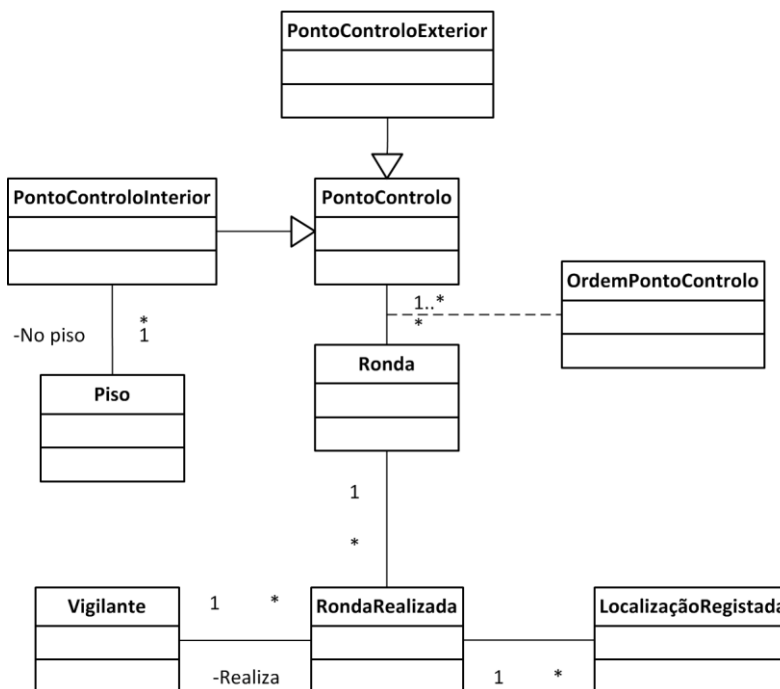


Figura 23 – Modelo de dados das rondas

O Técnico e o Vigilante (durante a sua ronda) podem registar incidentes (Incidente), que são um registo de uma família de situações (Familia), uma situação (Situação) e uma Causa (Cause). Os incidentes podem ser incidentes interiores (IncidenteInterior), associados a uma localização específica num piso ou sala; ou exteriores (IncidenteExterior), associados a uma localização específica no campus. Podem possuir anexos multimédia (Multimedia), que podem ser de texto (NotaTexto), vídeo (Video), de áudio (NotaVoz) e fotografia (Fotografia). Ilustrado pela Figura 24.

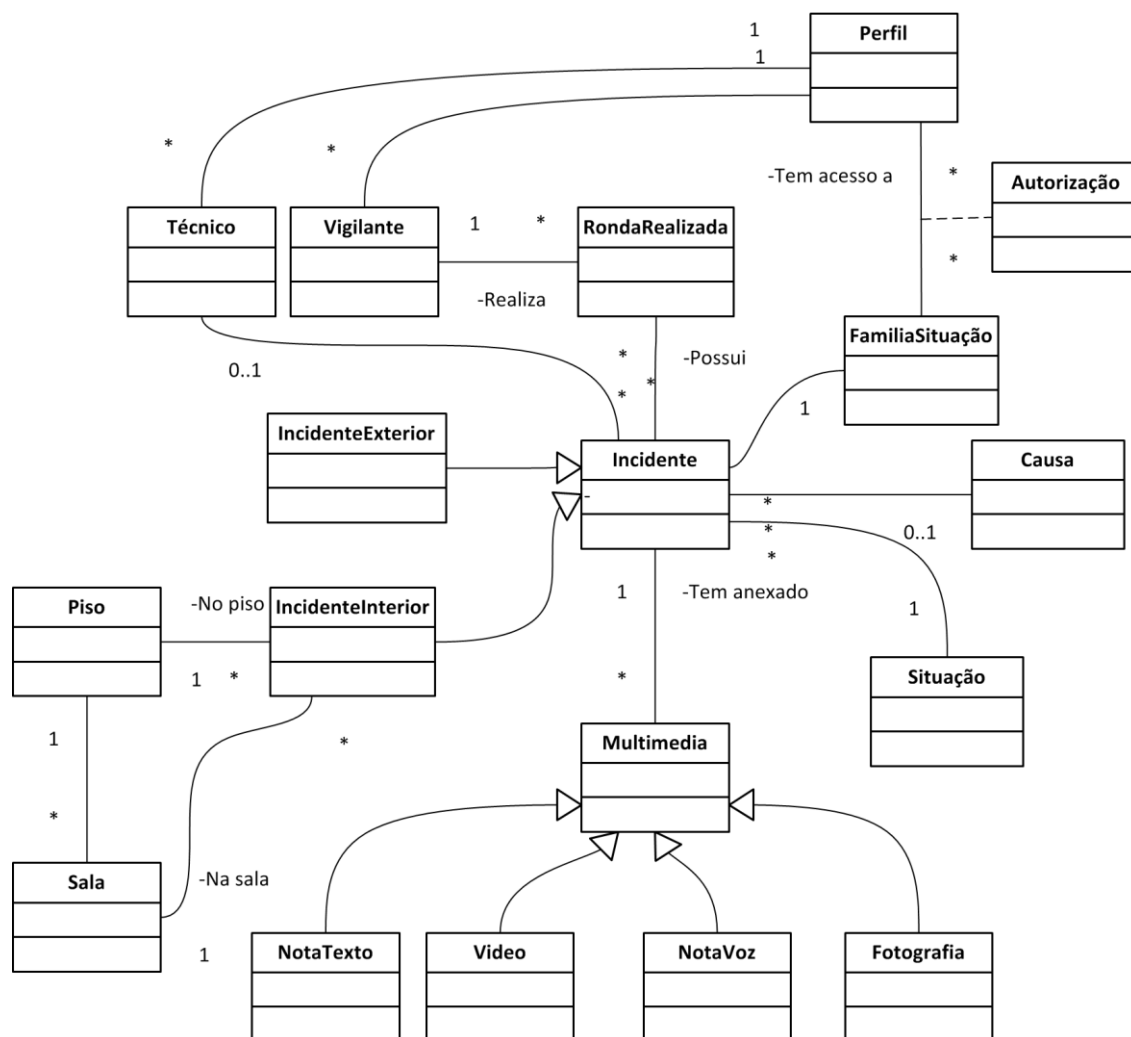


Figura 24 – Modelo de dados dos incidentes

Os edifícios (Edifício) possuem vários pisos (Piso), que por sua vez possuem várias salas (Sala). Os pisos possuem uma planta (PlantaPiso) associada sobre a qual os incidentes interiores e os pontos de controlo interiores incidem. Os edifícios podem possuir áreas para o método de georeferenciação, descrito num capítulo posterior. Ilustrada pela Figura 25.

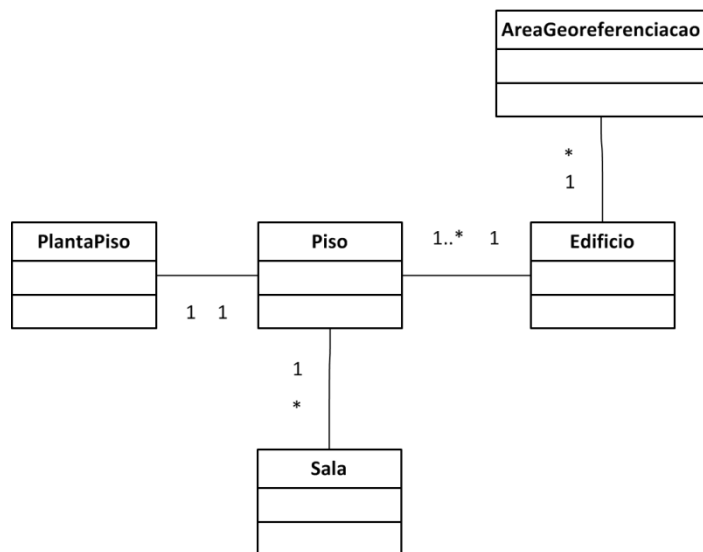


Figura 25 – Modelo de dados dos edifícios

O modelo de dados completo pode ser consultado no anexo 9.1.

4.4 Integração de dados e sincronização de dados móveis

A integração de dados é o problema de criar uma vista combinada de dados armazenados em fontes diferentes, existindo dois problemas fundamentais associados:

Devido à distribuição e heterogeneidade dos dados e das tecnologias em uso, é necessário que existam interfaces conhecidas para consulta, que exista um processo de transformação para traduzir os dados entre os diferentes contextos, e, também, que exista uma forma de relacionar os dados entre os diferentes contextos.

Para lidar com a diversidade dos dados e suas estruturas, assim como semântica, é necessária uma estratégia de conversão entre os diferentes serviços (contextos), que pode ser implementada através de mediadores, que traduzem o contexto da fonte para o contexto do receptor, utilizando conhecimentos de conversão de contexto [22][44]. Devido à tradução de contexto, os elementos os quais o mediador media não conhecem a identidade com quem estão a comunicar os dados nem os detalhes que permitem esta comunicação, o que permite, assim, desacoplar estes elementos.

Os sistemas de gestão de bases de dados são facilmente consultados através de protocolos, muitas vezes abertos, como o *Tabular Data Stream Protocol*, que permitem consultar tabelas e executar procedimentos e outras operações [45]. Os sistemas GIS que, basicamente, são sistemas de gestão de bases de dados orientados para um fim geográfico, também disponibilizam interfaces de consulta.

Devido à volatilidade dos dados em outros sistemas e de estes funcionarem de forma autónoma, os dados nestes sistemas podem ser alterados ou eliminados a qualquer momento, o que significa que os dados integrados na SegUABD poderão não existir no WebSIG a partir de certo período temporal, existindo o risco de serem eliminados numa actualização posterior no sentido de manter os dados sincronizados.

Assim, é necessário implementar uma estratégia de integração de dados e uma estratégia de arquivo dos dados armazenados na SegUABD. A estratégia implementada está descrita no capítulo 5.1.2.

Como foi referido no capítulo 0, a existência de um meio de persistência local, nomeadamente uma base de dados móvel - SegUAMóvelBD - na aplicação móvel permite lidar com problemas de conectividade, pois os dados necessários para a aplicação podem ser armazenados localmente e, assim, estarem disponíveis quando necessário em caso de falta de conectividade, e permite, também, armazenar os dados para envio posterior quando não for possível enviá-los ao SegUAService. No entanto, o armazenamento local implica uma forma de sincronização para reflectir actualizações na fonte dos dados.

Assim, o armazenamento local e sincronização de dados levantam várias questões, nomeadamente:

- Qual o modelo de dados usar para armazenar os dados na SegUAMóvelBD?
- Como relacionar os dados da SegUABD e os dados da SegUAMóvelBD?
- Como manter sincronizados os dados armazenados na SegUAMóvelBD com os dados armazenados na SegUABD?

Dado que a aplicação trabalha no mesmo contexto que a SegUABD, onde esta tem a função de apenas armazenar a história do contexto, o modelo de dados da informação armazenada na SegUAMóvelBD deve ser igual ao modelo de dados da SegUABD pois, assim, será mais fácil corresponder um objecto do modelo de dados entre as duas entidades. Caso não se use o mesmo modelo, será necessário implementar um meio de tradução entre os modelos, o que aumenta a complexidade de implementação.

Com o objectivo de armazenar a informação gerada na aplicação na SegUABD, é necessário existir uma forma de corresponder um objecto do modelo de dados criado na SegUAMóvelBD com um objecto do modelo de dados da SegUABD. Dado que, em bases de dados, os dados armazenados na mesma classe são distinguidos por identificadores únicos (chaves primárias), estes identificadores poderão ser utilizados, também, no modelo de dados da aplicação e, desta forma, é possível relacionar os objectos entre as duas entidades.

Dado que a realidade sobre a qual o sistema opera está em constante actualização, é necessária uma forma de actualizar, também, os dados na aplicação móvel. Actualizações na SegUABD devem ser verificadas o mais cedo possível na aplicação móvel para impedir o uso de dados incorrectos nesta. Por exemplo, na aplicação, o Vigilante deve visualizar uma lista de família

de incidentes que deve ser igual à lista existente na SegUABD, caso contrário, os dados gerados pelo Vigilante serão incorrectos.

De uma forma genérica, existem duas formas de disparar uma actualização em aplicações: notificação pelo servidor às aplicações; ou verificação periódica por parte das aplicações móveis. A notificação pelo servidor tem a vantagem de informar imediatamente as aplicações de que existem actualizações e de que estas devem efectuar as devidas actualizações locais. No entanto, para receber notificações, a aplicação móvel deverá estar à escuta de notificações o que implica uma disponibilidade constante de rede, que por sua vez implica um estado ligado constante do componente de rede sem fios no dispositivo. Tais condições não são realistas, devido à infra-estrutura de rede no campus, e impedem que a aplicação desactive a rede sem fios para gestão de energia. Além disso, esta hipótese não é escalável pois, um grande número de Terminais em actualização simultânea pode originar níveis excessivos de utilização da rede e do servidor que aloja o SegUAService.

A verificação periódica por parte de aplicações é mais realista, pois o momento de verificação é arbitrário para a aplicação, permitindo a esta que determine o melhor momento para a verificação. Ao permitir que as aplicações determinem o momento para actualizar, os momentos de descarga de actualizações serão aleatórios no SegUAService, o que reduz a probabilidade de um grande número de Terminais actualizar em simultâneo, diminuindo, assim, a utilização de rede e do servidor.

Existem várias estratégias para efectuar actualizações no que diz respeito à forma de como a informação é descarregada, nomeadamente: descarga de todas as tabelas; descarga das tabelas modificadas; e descarga dos registos alterados ou inseridos²⁵.

A descarga de todas as tabelas implica descarregar todo o modelo de dados relevante à aplicação, independentemente de já existirem os mesmos dados no meio de persistência local. Esta estratégia é segura pois garante que toda a informação está presente e consistente, no entanto, é também muita lenta e pesada computacionalmente devido ao grande volume de dados.

A descarga das tabelas alteradas envolve apenas a descarga completa das tabelas actualizadas, relativamente aos dados existentes na SegUAMóvelBD em questão. Esta estratégia é mais eficiente que a descarga de todas as tabelas pois descarrega um volume menor de dados, sendo mais rápida e mais leve computacionalmente.

A descarga de apenas alguns registos envolve apenas a descarga de registos inseridos, ou alterados e, portanto, a utilização de rede e o peso computacional serão os mínimos possíveis. Apesar de mais eficiente, esta estratégia é também a de implementação mais complexa, tanto do lado da SegUABD e/ou SegUAService, como do lado do SegUAMóvel/SegUAMóvelBD, pois é necessário uma forma de identificar quais dados na fonte de dados foram eliminados e quais foram actualizados.

²⁵ Também é descarregado algo que identifique os registos removidos.

No entanto, estas estratégias apenas têm em mente a descarga de actualizações que ocorreram na base de dados central, sendo necessária uma forma para aplicações actualizarem a própria base de dados central com os dados que produziram.

Um padrão comum, que permite actualizações bidireccionais, é o Publicação-Subscrição, onde se mantém um registo das alterações no sistema de gestão de bases de dados (SGBD) e nos terminais móveis [46][47][48]. O SGBD começa por aplicar uma versão inicial da base de dados a todos os clientes, denominados de subscritores e, a partir deste momento, regista todas as alterações da base de dados, onde este registo é denominado de publicação. Após receberem a versão inicial da base de dados, através de um agente instalado no subscritor, este regista todas as alterações que ocorram na sua base de dados local. Para sincronizar os dados, o subscritor envia para o SGBD o seu registo de alterações, onde este compara o seu registo com o do subscritor. De acordo com regras definidas pelo administrador de sistema, o SGBD resolve conflito de dados (actualizações, eliminações, etc.) entre as duas versões das bases de dados, e executa as alterações na publicação e envia os resultados ao subscritor, onde este os aplica na sua base de dados local. Desta forma, permitindo actualizações parciais e aplicáveis a qualquer versão temporal de uma base de dados móvel, este padrão adapta-se especialmente bem ao ambiente desconexo de terminais móveis com um número variável de terminais.

Devido à minimização do volume de dados, o padrão Publicação-Subscrição é a forma ideal de sincronização entre bases de dados centrais e bases de dados móveis, tenta-se adoptar este padrão no SegUA, estando a sua implementação descrita no capítulo 5.1.2.

4.5 Visualização de mapas

A componente de mapas teve grande ênfase na interface, onde teve que ser implementado de raiz um controlo que facilitasse a visualização de mapas, permitindo o arraste, zoom, interposição e interacção com itens no mapa, e comportamento adaptativo ao contexto do utilizador.

Quando o utilizador indica uma localização no mapa pressionando com o dedo ou *stylus*, está, na realidade, a seleccionar uma coordenada em pixéis no ecrã, que terá que ser traduzida para coordenadas GPS. A operação inversa também é verdade, isto é, coordenadas GPS têm de ser traduzidas para coordenadas de pixéis de forma a poderem ser interpostas correctamente no mapa.

É necessário compreender como funcionam os referenciais envolvidos na tradução, isto é, qual o comportamento do referencial geográfico WGS 84 [49], para a região de Portugal, e do referencial do ecrã, ilustrados pela Figura 26.

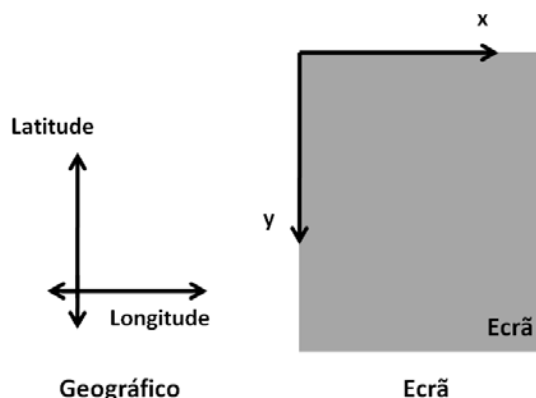


Figura 26 – Comparação entre os referenciais geográfico e do ecrã

A tradução de coordenadas é feita com recurso à associação de coordenadas GPS em graus decimais aos cantos superior esquerdo e inferior direito da imagem, de forma a saber a quais pontos geográficos correspondem estes extremos da imagem (em pixel). Estes cantos foram escolhidos de forma a que, com apenas estas duas coordenadas, seja possível saber as outras duas coordenadas dos outros cantos. Sabendo isto, é possível criar equações matemáticas que convertam as coordenadas.

Como se pode observar, os comportamentos dos referenciais são diferentes. Enquanto que o referencial geográfico evolui de forma semelhante ao referencial cartesiano, o referencial do ecrã evolui com o eixo do Y invertido, com origem no canto superior esquerdo do ecrã.

Para traduzir uma coordenada em pixel para uma coordenada GPS, calcula-se primeiro o rácio que a coordenada do pixel representa na dimensão da imagem, e depois aplica-se este rácio à diferença entre as coordenadas GPS, sendo o resultado somado ou subtraído à origem do referencial geográfico. A Figura 27 ilustra os dados envolvidos na tradução.

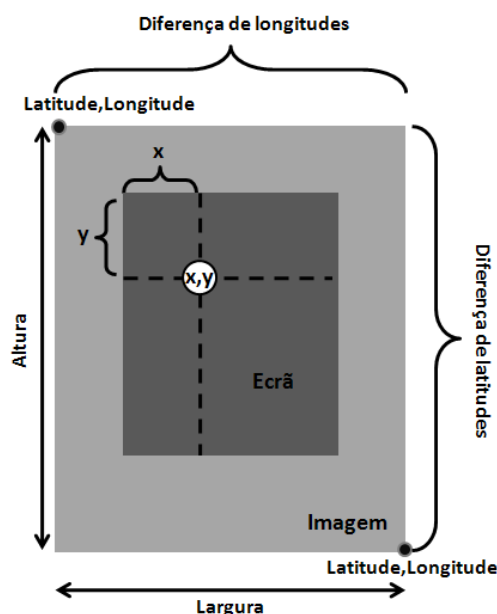


Figura 27 – Correspondência dos parâmetros no ecrã

As equações de tradução de coordenadas pixel (x,y) para coordenadas GPS são as seguintes:

$$Longitude = LongitudeEsquerda + \frac{x}{Largura} * Abs(LongitudeEsquerda - LongitudeDireita)$$

$$Latitude = LatitudeSuperiorEsquerda - \frac{y}{Altura} * (Abs(LatitudeSuperior) - Abs(LatitudeInferior))$$

As equações de tradução de coordenadas GPS (lat,long) para coordenadas pixel são as seguintes (resolução das equações anteriores em ordem a y e x):

$$Y = \frac{AlturaImagem * (LatitudeSuperior - lat)}{LatitudeSuperior - LatitudeInferior}$$

$$X = \frac{LarguraImagem * (long - LongitudeEsquerda)}{Abs(LongitudeEsquerda - LongitudeDireita)}$$

É de salientar que estas equações envolvem as coordenadas XY normalizadas, isto é, as coordenadas que correspondem aos pixéis quando a imagem está na sua dimensão original (factor zoom de 1). Assim, antes de efectuar estes cálculos, é necessário normalizar as coordenadas, o que é feito através da multiplicação/divisão das coordenadas X e Y pelo factor zoom.

As equações apresentadas assumem que as coordenadas progridem linearmente e que o mapa cobrirá apenas uma ínfima parte do planeta, sendo assim, a curvatura do planeta insignificante. Para que as equações apresentadas produzam valores correctos, é fundamental que o mapa seja escolhido de forma a que as latitudes dos cantos superiores não variem entre si, e que as longitudes dos cantos superior e inferior também não variem entre si. Para eliminar estas restrições, é necessário criar equações mais complexas, envolvendo cálculos trigonométricos.

5 Implementação

5.1 Tecnologias e modelo de programação

As tecnologias e ferramentas usadas foram escolhidas de acordo com os desejos do cliente (os Serviços Técnicos) sendo, por isso, todas da *Microsoft*.

A SegUABD foi implementada em *SQL Server 2008*²⁶; a SegUAWeb em *ASP .NET*²⁷; o SegUAService na plataforma *.NET Framework 3.5*; e o SegUAMóvel na plataforma *.NET Compact Framework 3.5*. O desenvolvimento do SegUAMóvel e SegUAService foi efectuado no *Visual Studio 2008*²⁸.

A arquitectura implementada do sistema difere ligeiramente da arquitectura desenhada no capítulo 4.1, pois na arquitectura implementada, a SegUAWeb comunica directamente com a SegUABD no sentido de usufruir da excelente integração que existe entre o *SQL Server 2008*, o *ASP .NET*, o *IIS 7* e o *Visual Studio 2008*, e assim permitir uma rápida implementação da SegUAWeb, sobretudo para efeitos de validação do projecto. O SegUAService e a SegUAWeb, alojados pelo *IIS 7*, comunicam com a SegUABD através do protocolo nativo, o *Tabular Data Stream Protocol (TDSP)*; o WebSIG está implementado em *ESRI ArcGIS Server*²⁹, e o SegUAService comunica com este através de um web service REST disponível no WebSIG. O SegUAMóvel, executado em *Windows Mobile 6.1*, comunica com o SegUAService através do *web service XML-RPC* disponível no SegUAService. A Figura 28 ilustra a arquitectura implementada.

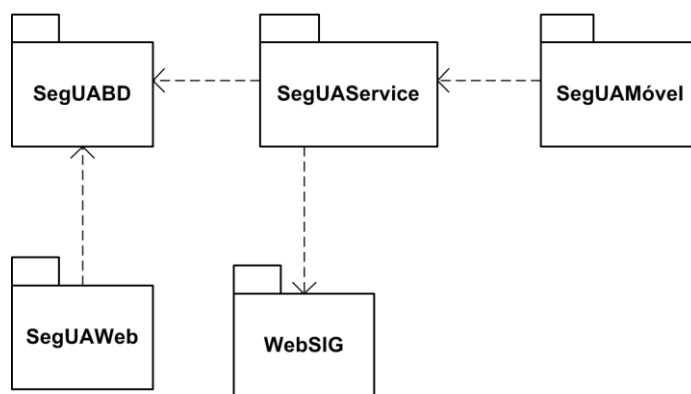


Figura 28 – Arquitectura implementada do sistema SegUA

²⁶ <http://www.microsoft.com/sqlserver/2008/en/us/default.aspx>

²⁷ <http://www.asp.net/>

²⁸ <http://www.microsoft.com/visualstudio/en-gb/default.msp>

²⁹ <http://www.esri.com/software/arcgis/arcgisserver/index.html>

5.1.1 Implementação do módulo móvel

5.1.1.1 Arquitectura

A arquitectura implementada da SegUAMóvel está ilustrada pela Figura 29.

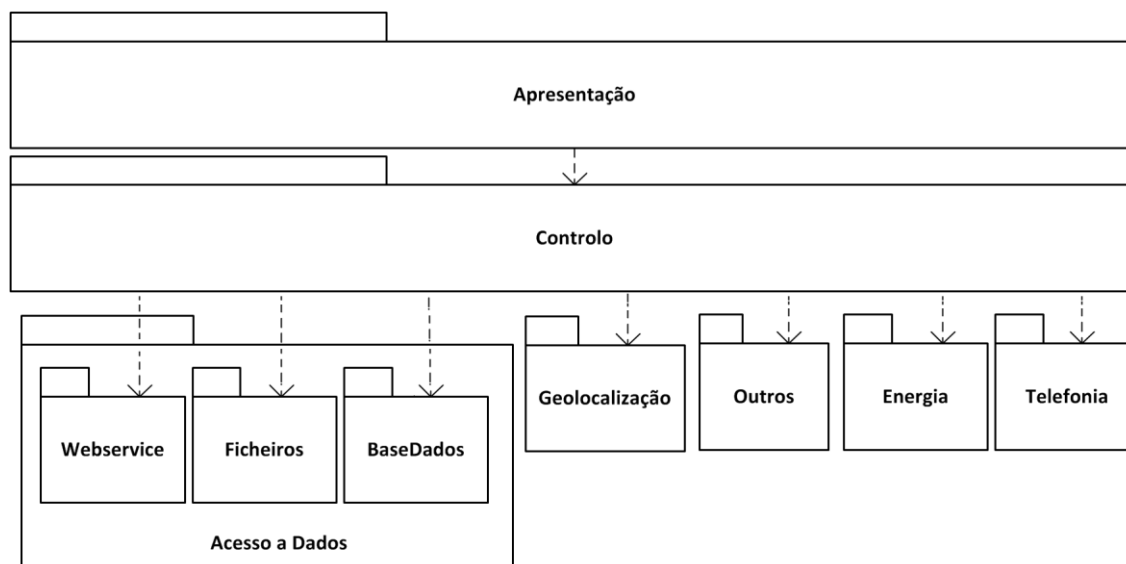


Figura 29 – Arquitectura implementada da SegUAMóvel

O meio de persistência local implementado, SegUAMóvelBD, é um combinado entre uma base de dados *SQL Server CE (SQLSCE)*, representada pela camada BaseDados, e o sistema de ficheiros, representado pela camada Ficheiros, pois o SQLSCE impõe um limite de 8000 bytes nas colunas, não sendo suficiente para o armazenamento de conteúdo pesado na base de dados, portanto, é necessário usar o sistema de ficheiros para armazenar este conteúdo multimédia e, também, os mapas dos pisos. Dado que a capacidade interna é, muitas vezes, reduzida, a aplicação procura armazenar os dados num cartão de dados. A camada Webservice implementa o acesso ao SegUAService e é insensível quanto ao tipo de rede utilizado para a comunicação, cabendo ao *Windows Mobile* fornecer conectividade, seja por redes 802.11 ou pela rede móvel.

Implementando o padrão Publicador-Subscritor, descrito no capítulo 4.4, a sincronização de dados poderia ser feita através da funcionalidade nativa ao SQLSCE, a *Merge Replication*, não tendo a sincronização de dados que ser, assim, implementada pelo programador. No entanto, esta funcionalidade requer tecnologias proprietárias Microsoft em ambas as partes cliente e servidor, nomeadamente um servidor de páginas IIS e a SegUABD deve estar implementada em *SQL Server*, o que impede interoperabilidade entre os vários elementos da arquitectura [15]. A Oracle comercializa soluções interoperáveis de bases de dados, incluindo móveis³⁰, e ferramentas de sincronização, incluindo ao nível de desenvolvimento de aplicações, no entanto, não são gratuitas e, portanto, não se teve acesso a estes produtos. Desta forma, não foi utilizada uma

³⁰ Oracle Database Lite 10g,

funcionalidade proprietária de sincronização, tendo sido implementado um método descrito no capítulo 5.1.2.

O SQLSCE armazena cada base de dados num ficheiro com extensão SDF, característica que pode ser vantajosa pois, por exemplo, para popular inicialmente uma base de dados no SegUAMóvel na implantação da aplicação, basta copiar o ficheiro, não sendo necessário executar a aplicação para sincronizar.

A camada Energia está implementada através de funções nativas ao *Windows Mobile* pois a NETCF não possui suporte para gestão de energia do dispositivo.

A camada Outros está parcialmente implementada através de funções nativas e com recurso à biblioteca *OpenNET Smart Device Framework*. Esta camada detecta a existência de um teclado físico, permite a vibração do dispositivo, e permite a gravação de áudio através do microfone.

A camada Telefonía implementa o envio de SMS.

A camada GeoLocalização permite o acesso ao componente GPS e está implementada com recurso à biblioteca *Microsoft GPS Intermediate Driver*³¹, pois não existe suporte para componentes GPS na NETCF. O cálculo da distância entre dois pontos geográficos é feito com recurso à biblioteca comercial *Geoframeworks GPS.NET*.

5.1.1.2 Georeferenciação

A georeferenciação foi um tema que exigiu mais trabalho e inovação, pois, do *web service* do WebSIG apenas se obtêm listas de dados sem qualquer georeferenciação, nomeadamente:

- Lista de edifícios;
- Lista de pisos;
- Lista de salas;
- Plantas dos pisos,
- Lista de objectos.

Optou-se por não implementar suporte para a lista de objectos pois existe um número de objectos demasiado elevado, e não existe forma de filtrar os resultados de uma forma mais granular, o que problematiza o desenho de uma interface que permita a interacção do utilizador com estes objectos.

A ESRI disponibiliza a biblioteca *ArcGIS Mobile*, uma biblioteca geográfica para NETCF, que poderia, potencialmente, integrar o SegUAMóvel com o WebSIG, no entanto, não se obteve acesso a esta biblioteca.

³¹ <http://msdn.microsoft.com/en-us/library/bb158708.aspx>

Uma forma de fornecer georeferenciação de estruturas é através do uso de shapefiles[50], ficheiros que armazenam dados geográficos que podem incluir coordenadas GPS, no entanto, existem apenas produtos comerciais e suporte para estes ficheiros é difícil de implementar.

Para possibilitar a georeferenciação, foi criado um método que, a partir de uma simples coordenada GPS, é possível determinar se essa coordenada corresponde a um ponto de uma estrutura previamente definida na SegUABD.

Inspirado pelo sistema de coordenadas planas³²[50], onde um mapa é dividido por células identificadas por coordenadas X (colunas) e Y (linhas), o método consiste em definir uma área com uma certa forma geométrica regular que corresponda à área geográfica que o edifício ocupa, onde os limites da área geométrica são coordenadas GPS, e associar essa área ao edifício que pretendemos georeferenciar. A Figura 30 ilustra a associação de uma área rectangular a um objecto.

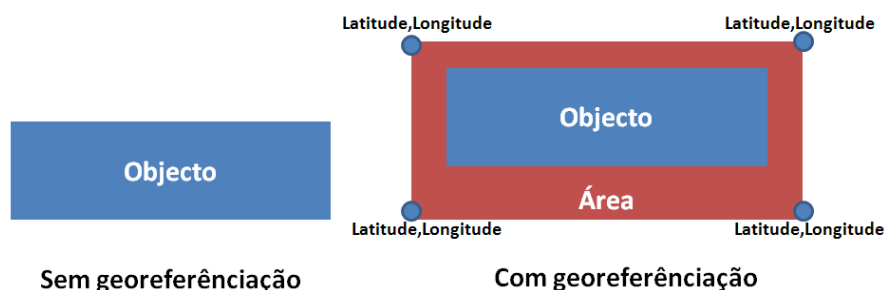


Figura 30 – Método de georeferenciação através de áreas

Conhecendo o referencial geográfico que rege as coordenadas na região em questão, para a SegUAMóvel determinar se se encontra nas imediações de um edifício, basta que este verifique se as suas coordenadas GPS actuais se encontram dentro da área definida pela área geométrica. Por exemplo, supõe-se que o SegUAMóvel se encontra numa localização com latitude de 5 e longitude de 5 e deseja verificar se se encontra na área do Departamento A, área ilustrada pela Figura 31.

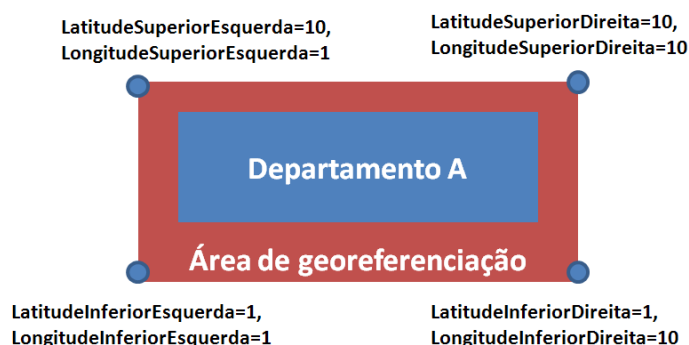


Figura 31 – Exemplo do método de georeferenciação através de áreas

³² Raster coordinate system

Sabendo que a área geométrica é rectangular, a SegUAMóvel irá, então, executar as seguintes verificações:

LatitudeSuperiorEsquerda > Latitude actual => 10 > 5 => verdade
LongitudeSuperiorEsquerda < Longitude actual => 1 < 5 => verdade

LatitudeSuperiorDireita > Latitude actual => 10 > 5 => verdade
LongitudeSuperiorDireita > Longitude actual => 10 > 5 => verdade

LatitudeInferiorEsquerda < Latitude actual => 1 < 5 => verdade
LongitudeInferiorEsquerda < Longitude actual => 1 < 5 => verdade

LatitudeInferiorDireita < Latitude actual => 1 < 5 => verdade
LongitudeInferiorDireita > Longitude actual => 10 > 5 => verdade

Verificando-se todas as condições, a SegUAMóvel determina que se encontra dentro da área do Departamento A. Note-se que estas condições devem ter em conta o hemisfério no qual a SegUAMóvel se encontra, pois o referencial não é o mesmo para qualquer ponto do planeta.

Porém, poucas estruturas são perfeitamente rectangulares, podendo originar intersecções indesejadas com áreas de outros edifícios sendo, portanto, necessário modelar a área com maior precisão. Com este método, existem duas formas para melhor descrever uma área geográfica que um edifício possui: utilização de outras formas geométricas, associação de várias áreas rectangulares a um mesmo edifício.

Enquanto que, com uma forma rectangular, as condições a serem verificadas são simples, a utilização de outras formas geométricas implica um algoritmo mais complexo pois as condições a serem verificados devem ter em conta a forma geométrica. Por exemplo, o círculo possui uma infinidade de pontos que representam o seu limite, o que dificulta uma definição prática do limite. Com outras formas, especialmente as irregulares, o algoritmo torna-se ainda mais complexo pois não é evidente onde se encontram os pontos que definem as formas.

Desta forma, a associação de várias áreas rectangulares, sem intersecção, a um edifício permite modelar de uma forma mais precisa que com apenas 1 área rectangular, e não aumenta, de forma significativa, o algoritmo de verificação de condições. A Figura 32 exemplifica a associação de várias áreas rectangulares ao Departamento A (azul), existindo 3 áreas (verde, vermelho, cinzento) que, juntas, representam a área geográfica do edifício.



Figura 32 – Exemplo de associação de várias áreas a um edifício

Como se pode observar pela Figura 32, a área total representada pelas áreas geométricas é maior do que a área real do edifício, o que se deve ao facto de que, no interior de edifícios, não existe recepção de sinal GPS e, portanto, a SegUAMóvel não possui recepção de sinal GPS. Sem sinal GPS, não é possível obter uma localização actual e sem localização actual, não é possível comparar a localização actual com as coordenadas das áreas geométricas. Assim, no seu todo, a área geométrica deve ser superior à área real, no sentido de permitir que o SegUAMóvel tenha leituras do sinal GPS antes de perder o sinal quando já estiver no interior do edifício, porque está-se a observar se se está dentro ou fora de uma certa área.

Para implementar a funcionalidade de vista geográfica, optou-se pela apresentação de uma imagem estática. Apesar de existir um certo número de serviços na Internet que permitem criar aplicações móveis com integração de mapas como, por exemplo, o *Google Maps*³³, estes serviços exigem conectividade para descarregar os dados necessários, e as imagens disponibilizadas por eles podem não possuir resolução suficiente para uma dada zona. Assim, com uma imagem estática, fornecida pelo próprio cliente, a imagem apresentada é uma imagem arbitrária, dando liberdade de escolha ao cliente em termos de resolução, caso se trate de uma imagem aérea, ou em termos do conteúdo apresentado, podendo ser uma planta do campus ao invés de uma fotografia.

A apresentação de uma imagem estática exigiu o desenvolvimento de um controlo da interface do utilizador, para possibilitar funcionalidades associadas à visualização de mapas, como zoom, arraste do mapa, etc. O desenvolvimento deste controlo foi abordado no capítulo 4.5.

5.1.1.3 Gestão de energia

O *Windows Mobile* é agressivo quanto à minimização do consumo de energia, pausando a execução de aplicações e desligando componentes do dispositivo, assim que detectar que este não está em uso pelo utilizador. Tal comportamento impede que a aplicação controle autonomamente a ronda do Vigilante, portanto, foi necessário contrariar o comportamento energético normal do *Windows Mobile* e, para tal, foi necessário compreender o esquema energético sobre o qual o *Windows Mobile* opera. Para o *Windows Mobile*, existem vários estados energéticos[51][52], os quais se apresentam os mais relevantes:

- Ligado – todo o dispositivo está operacional e as aplicações executam normalmente.
- Ecrã desligado – após algum tempo sem interacção com o utilizador, o *Windows Mobile* desliga o ecrã.
- Autónomo – o *Windows Mobile* utiliza este estado para executar operações que não necessitem de interacção com o utilizador como, por exemplo, actualizar a caixa de correio.

³³ <http://maps.google.com/>

- Suspenso – a maior parte dos componentes estão desligados, esperando por um evento. Este estado é o estado em que se encontra quando o dispositivo não está a ser utilizado.
- Desligado – o dispositivo está completamente desligado.

Analisando os estados energéticos, observa-se que apenas os estados Ligado, Ecrã desligado e Autónomo permitem, potencialmente, que a aplicação mantenha a sua execução sem interação com o utilizador. O estado mais conveniente será aquele no qual o dispositivo consome o mínimo de energia possível mas que permita a execução da aplicação, isto é, o estado Autónomo, portanto, a aplicação impede que o dispositivo transite para o estado Suspenso, forçando a transição para o estado Autónomo. No entanto, neste estado, os componentes físicos não são automaticamente ligados, sendo necessário fazê-lo manualmente.

A NETCF não possui suporte para gestão de energia do dispositivo, no entanto, a integração da aplicação com o dispositivo foi implementada através de chamadas nativas ao sistema. Devido a tal situação, a integração energética não foi totalmente conseguida, onde apenas se conseguiu integração com o ecrã e o componente da rede sem fios 802.11. Conseguiu-se ligar e desligar o ecrã e a rede sem fios de forma arbitrária mas sem integração com o componente GPS do dispositivo, não é possível efectuar controlo autónomo da ronda, nem registo periódico da localização. Para permitir tais funcionalidades, foi necessário forçar que o dispositivo se mantenha no modo Ligado, o que implica um grande consumo de energia, mesmo forçando uma luminosidade reduzida do ecrã. Por outro lado, a SegUAMóvel permite a realização das tarefas num cenário desconexo e sem localização, não impedindo, assim, o trabalho do utilizador, perdendo-se, apenas, na automatização do processo.

Devido ao desempenho reduzido dos dispositivos móveis, comparando com os computadores de secretária, assim como uma menor largura de banda disponível na comunicação com o exterior, as operações executadas na aplicação poderão demorar um tempo significativo bloqueando a aplicação enquanto a operação não tiver terminado, o que perturba, assim, a experiência do utilizador [53].

A delegação de tarefas para *threads* diferentes da da *thread* que controla a interface permite manter a interface com capacidade de resposta e executar operações em paralelo possibilitando, desta forma, que o utilizador realize o seu trabalho sem interferência. Nem todas as operações foram delegadas para uma *thread* dedicada, tendo sido apenas delegadas as *threads* que sejam de registo e envio de dados, de forma a possibilitar a continuação da tarefa do utilizador enquanto a aplicação se encarrega da operação de forma invisível. O controlo autónomo da ronda do Vigilante executa, também, numa *thread* dedicada, estando em execução permanente enquanto a ronda é realizada. No início da ronda, a *thread* da interface inicia a *thread* de controlo, e esta permanecerá em execução cíclica esperando pela notificação da *thread* da interface. Durante a sua execução cíclica, a *thread* de controlo regista periodicamente a localização geográfica actual da SegUAMóvel, determina se este se encontra na zona de um ponto de controlo, notificando o Vigilante em caso afirmativo, e verifica o estado de energia do dispositivo. O controlo autónomo notifica o Vigilante quando este atingir um ponto de controlo

ou terminar a ronda, assim como notificações acerca do estado de energia do dispositivo. Na computação da localização, caso a aplicação determine que o Vigilante não se encontra em movimento após um determinado período de tempo, emite um alarme automático, enviado através de SMS, para o Supervisor. A Figura 33 ilustra esta dinâmica entre as *threads* em questão.

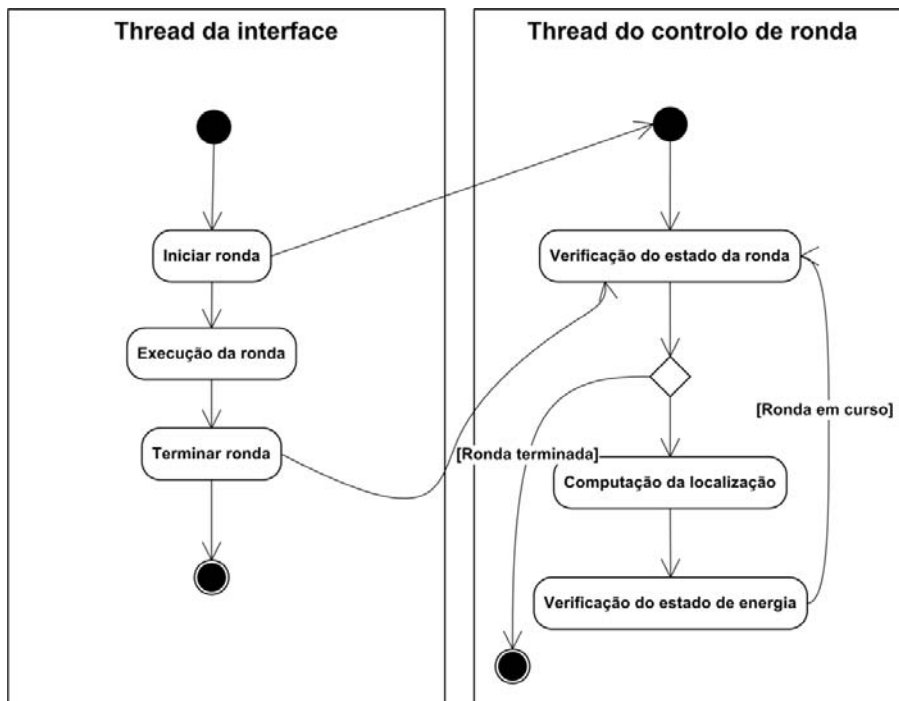


Figura 33 – Multithreading na realização de uma ronda

5.1.1.4 Funcionalidades

O utilizador é recebido pelo ecrã de boas vindas. Clicando em 'Autenticar', o utilizador poderá autenticar-se no sistema, tendo que seleccionar primeiro a sua identidade, como ilustrado pela Figura 34.



Figura 34 – Ecrã de boas vindas e selecção de utilizador

Tendo a identidade sido escolhida, o utilizador pode confirmar os seus dados, e insere a sua palavra-chave numérica para se autenticar, como ilustra a Figura 35.

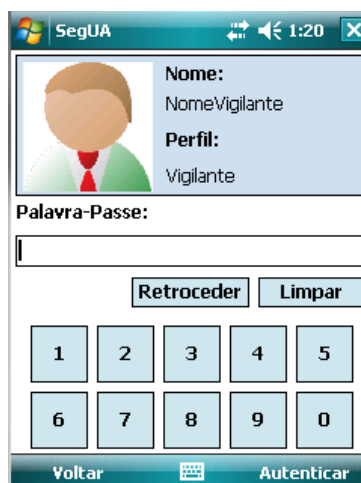


Figura 35 – Autenticação do utilizador

Caso o utilizador possua um perfil de Vigilante, os próximos ecrãs permitem pré-visualizar rondas, resumir uma ronda incompleta, ou iniciar uma nova ronda, com ilustra a Figura 36.



Figura 36 – Pré-visualização e início ou retoma de rondas

Retomando ou iniciando uma nova ronda, o Vigilante tem acesso ao ecrã principal da ronda, onde pode verificar e controlar o progresso dela, e registar novos incidentes. Existe, também, uma versão deste ecrã em modo geográfico, onde o controlo da ronda e registo de incidentes é iniciado através da interação com o mapa, como ilustra a Figura 37.



Figura 37 – Visão do progresso da ronda

Para registar um incidente, os passos seguintes são comuns ao Vigilante e Técnico. Antes de poder registar um incidente, o utilizador visualiza a lista de incidentes que ocorreram no local, podendo filtrar os resultados em termos de dias, como ilustra a Figura 38. Este ecrã é, também, o ecrã principal do Técnico.



Figura 38 – Início do registo de um novo incidente

Seleccionando um incidente, o utilizador pode visualizar os detalhes do incidente, incluindo a sua localização (interior ou exterior) e anexos, ilustrado pela Figura 39.

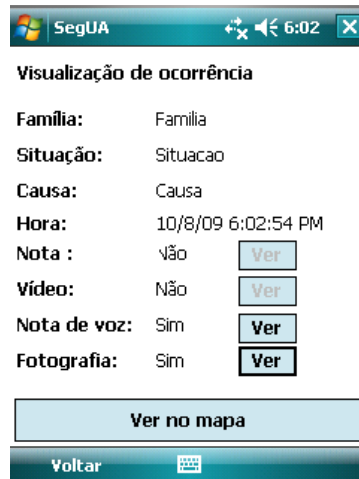


Figura 39 – Visualização de um incidente

Para registar um incidente, o utilizador necessita de percorrer uma série de passos seleccionando ou inserindo os dados necessários: Família de Situação, Situação, Causa, Confirmar local, Inserir Anexo, Confirmar. Na selecção da Família de Situação, ilustrada pela Figura 40 – Selecção da família de situações, o utilizador apenas visualizar as famílias a que o seu perfil tem acesso.

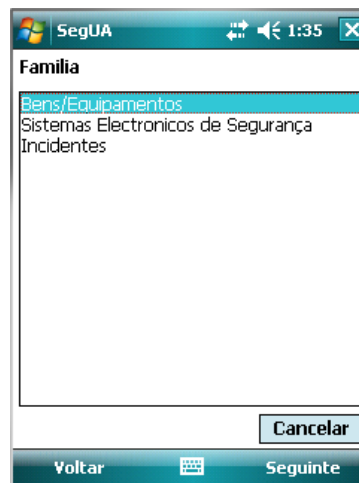


Figura 40 – Selecção da família de situações

Caso a localização do utilizador seja exterior, esta localização pode ser confirmada pelo mapa. Caso a localização seja interior, o utilizador poderá seleccionar os detalhes necessários, incluindo a localização no piso. Ilustrada pela Figura 41.



Figura 41 – Especificação da localização no interior de edifícios

O utilizador pode anexar conteúdo multimédia e, por fim, verificar os detalhes do incidente antes de este ser enviado, como ilustrado pela Figura 42.



Figura 42 – Anexação de conteúdo e verificação do incidente

Em caso de emergência, o Vigilante pode accionar um alarme, onde é enviada uma SMS com informações acerca da localização do Vigilante, como ilustrado pela Figura 43.



Figura 43 – Envio de alarme

5.1.2 Implementação do módulo central

O SegUAService foi implementado com a filosofia XML-RPC no sentido de usufruir da integração que a plataforma .NET possui com este tipo de *web services* [54][55]. Outra variante popular de implementação de *web services* é através de *Representational Stateful Transfer* (REST), no entanto, esta variante exige que os clientes construam endereços³⁴ identificando a operação pretendida e seus parâmetros, o que não é prático enquanto que, devido ao conceito de chamada de procedimentos remotos³⁵, a integração programática de um *web service* XML-RPC na NETCF é transparente para o programador pois a chamada de métodos do *web service* não difere da chamada de um método local, e existe verificação de tipos dos parâmetros.

A integração de dados provenientes do WebSIG foi implementada com recurso a um mediador, na forma de um componente do SegUAService, ficando, assim, apenas este componente acoplado ao WebSIG, o que permite centralizar toda a lógica operacional do sistema no SegUAService. A arquitectura do SegUAService está ilustrada pela Figura 44.

³⁴ URL

³⁵ RPC – Remote Procedure Call

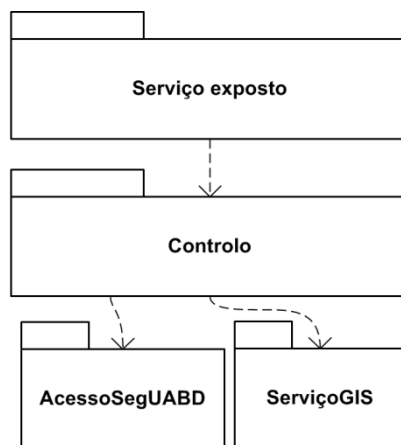


Figura 44 – Arquitectura do SegUAService

Idealmente, a integração com o WebSIG é feita simultaneamente com o pedido de informação por parte dos SegUAMóveis, no entanto, este cenário requer conectividade para efectuar o pedido e, se não existir conectividade, o SegUAMóvel não poderá efectuar o pedido e, desta forma, não terá acesso aos dados, perturbando ou impedindo o trabalho do utilizador. Além disso, o atraso de processamento tanto no WebSIG como no SegUAService, acrescendo a latência e largura de banda da rede, pode originar uma latência global demasiado significativa e que perturbe o trabalho do utilizador.

Por outro lado, se a SegUAMóvel armazenar localmente os dados que necessita, estes estarão sempre disponíveis independentemente do estado de conectividade. Este cenário requer que a integração de dados do WebSIG seja realizada anteriormente e de uma forma massiva no sentido de obter todos os dados necessários, o que significa que os dados integrados não serão os dados mais actuais e que será necessário manter actualizados os dados da SegUAMóvel, no entanto, os dados no WebSIG são alterados apenas esporadicamente, existindo, assim, pouca probabilidade que a SegUAMóvel não possua os dados mais actuais num dado momento.

Um cenário intermédio entre os cenários anteriormente apresentados, seria a integração de dados em simultâneo com os pedidos quando a SegUAMóvel possuisse conectividade, e uso dos dados armazenados localmente quando não possuisse, no entanto, este cenário estaria dependente da largura de banda disponível para a SegUAMóvel.

Desta forma, optou-se pelo cenário onde a SegUAMóvel armazena localmente os dados que necessita e o acesso a eles dependerá apenas do desempenho do dispositivo. Com o objectivo de possuir os dados mais actuais, a SegUAMóvel verifica, no início das sessões de utilização, se existem actualizações para os seus dados e, assim, o utilizador terá sempre os dados mais actuais, caso estes não sejam alterados durante a sua sessão de utilização.

Para permitir este cenário, os vários elementos da arquitectura devem colaborar tendo, cada um, o seu papel. O SegUAService e a SegUABD trabalham em conjunto para manter a SegUABD sincronizada com o WebSIG, onde o SegUAService descarrega do WebSIG, periodicamente e de forma massiva, a informação que a SegUAMóvel necessita e envia-a para a

SegUABD, onde esta analisa os seus dados e actualiza-os se for necessário, como ilustra a Figura 45.

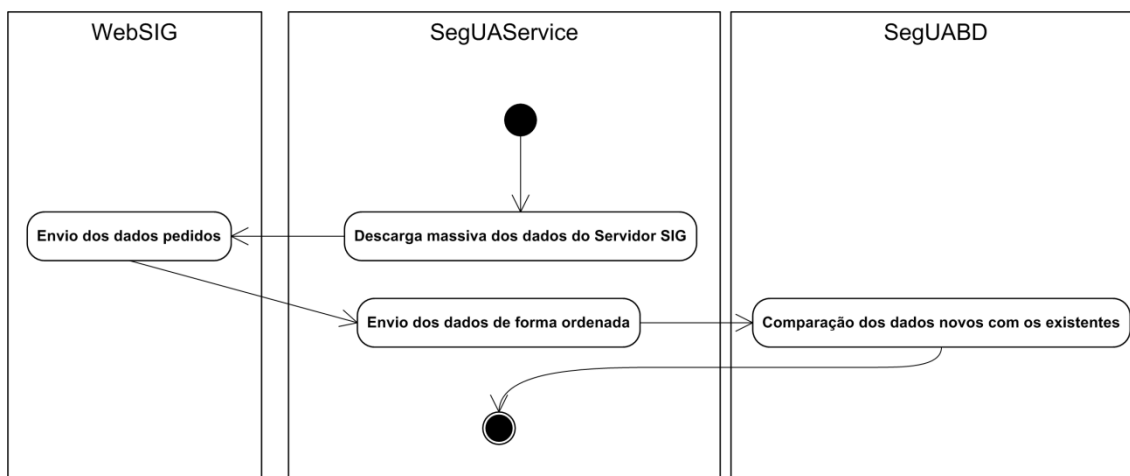


Figura 45 – Integração dos dados do WebSIG

A actualização da SegUABD é um processo cuidadoso pois, além de ser necessário manter os dados iguais aos dados do WebSIG, é necessário, também, manter os dados iguais no período em que foram usados, para efeitos de arquivo. Assim, a SegUABD é implementada com conceitos de base de dados temporais, nomeadamente, Período-Válido³⁶, onde cada registo possui campos que indicam as datas entre as quais o registo é válido [56][57]. A Figura 46 ilustra um exemplo, sobre a classe que representa uma planta de um piso. Se forem efectuadas obras num piso e se juntarem duas salas, a planta do edifício irá ser alterada. Caso a imagem seja simplesmente substituída, as localizações de incidentes ocorridos que os utilizadores indicaram deixarão de ser válidas, pois a imagem já não é a mesma. Através do Período-Válido, a imagem pode ser arquivada, mantendo as localizações correctas, sendo criado um novo registo para a nova imagem.

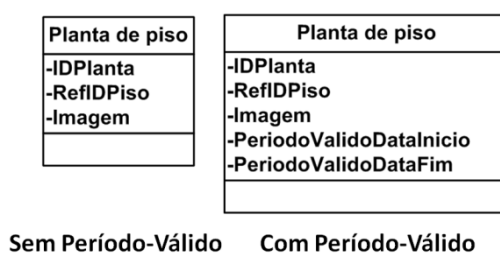


Figura 46 – Exemplo de Período-Válido

Quando um registo é criado, a data de início do Período-Válido é a data actual e a data do fim é uma data infinita. Quando um objecto deixa de existir no WebSIG, apenas será eliminado da SegUABD caso não referenciado, caso contrário, a data de fim do Período-Válido será a data actual. Numa primeira fase, após descarregar a informação do WebSIG, o processo de actualização é iniciado pelo SegUAService e este informa a SegUABD que irá iniciar o processo.

³⁶ Do inglês “valid-time”

Neste momento, a SegUABD executa uma limpeza de registos que não estão referenciados, ou seja, não foram usados anteriormente, e invalida temporalmente os registos que permanecem e que não tinham sido invalidados anteriormente. As tabelas afectadas por este processo são as tabelas que representam os edifícios, pisos, salas e plantas. Esta fase é ilustrada pela Figura 47.

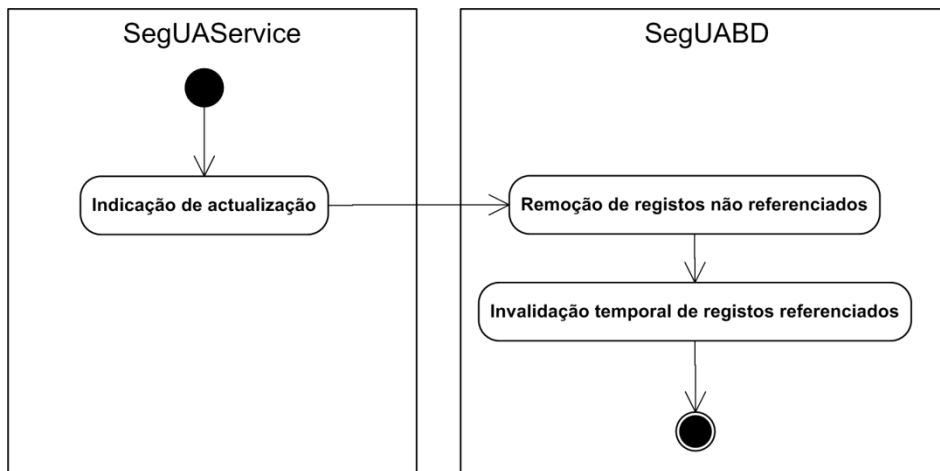


Figura 47 – Fase de limpeza de registos

Numa segunda fase, o SegUAService executa procedimentos da SegUABD, nomeadamente de inserção singular de dados (por exemplo, inserir os pisos um a um) pois é necessário que a SegUABD devolva o identificador único do registo que criou para cada inserção. Este comportamento é necessário pois, por exemplo, é preciso indicar à SegUABD a qual edifício se pretende associar os pisos correspondentes e, para tal, é necessário o identificador do edifício. A Figura 48 ilustra esta fase.

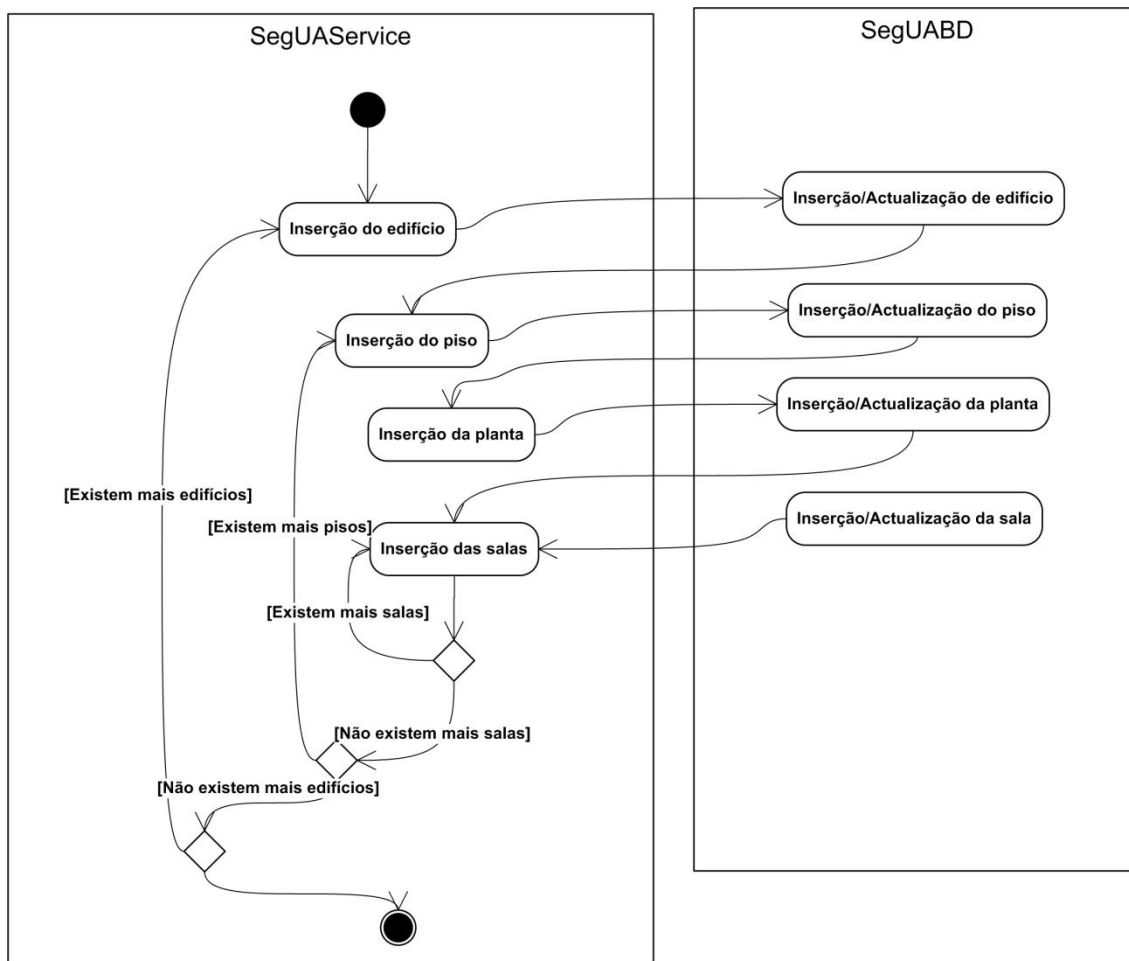


Figura 48 – Fase de inserção

A SegUABD analisa cada objecto recebido para inserção e verifica se este objecto já existe. A prova de existência é feita através de um identificador que o objecto possui: alfanumérico para o caso dos edifícios, e numérico para o caso dos pisos e salas. Se o objecto não existir, é criado um registo para este objecto da forma descrita anteriormente; se já existir, a SegUABD verifica se os dados que descrevem este objecto correspondem e, caso correspondam, o registo respectivo é validado temporalmente sem alteração aos campos do registo; caso não correspondam, o registo respectivo permanece invalidado temporalmente e é criado um novo registo para o objecto da forma descrita anteriormente. Este algoritmo é ilustrado pela Figura 49.

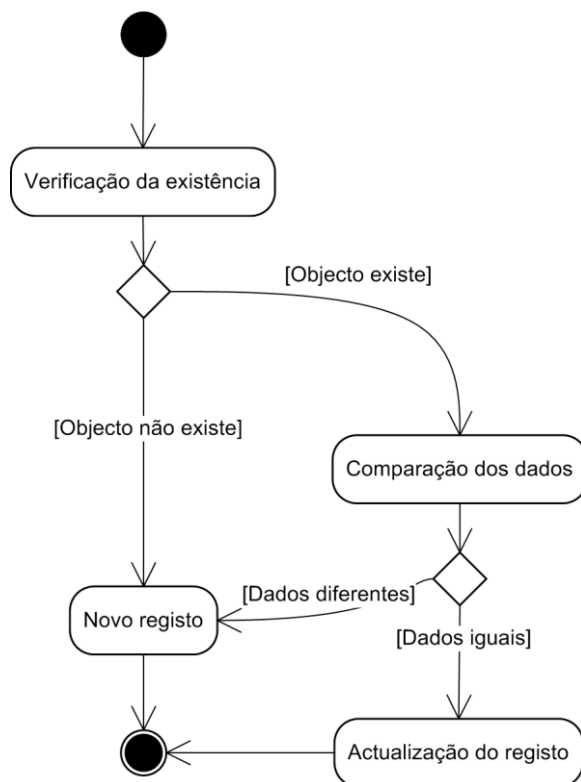


Figura 49 – Algoritmo de inserção

Apenas a SegUAWeb tem acesso aos dados cujo período é inválido; o SegUAService e, consequentemente, a SegUAMóvel recebem apenas dados cujo período é válido e, desta forma, consegue-se manter um arquivo de dados ao mesmo tempo que se consegue manter os dados sincronizados com o WebSIG.

Para manter os dados sincronizados entre a SegUAMóvel e a SegUABD, foi implementado uma estratégia semelhante ao padrão Publicador-Subscritor mas mais simples, nomeadamente o cenário de descarga semi-parcial, descrito no capítulo 4.4. No início da sessão do utilizador, a SegUAMóvel descarrega as tabelas que sofreram actualizações na SegUABD, elimina os registos das tabelas correspondentes na sua base de dados local, e insere os registos recebidos. Para saber quais tabelas sofreram alterações, a SegUABD mantém actualizada uma tabela que possui contadores incrementais³⁷ para cada tabela a qual a SegUAMóvel deve manter sincronização, e estes contadores são incrementados em cada inserção, actualização e remoção de registos na SegUABD. No processo de sincronização inicial, a SegUAMóvel descarrega estes contadores, mas não os incrementa durante as sessões de utilização. Em cada sincronização posterior, a SegUAMóvel envia os seus contadores, que são comparados pela SegUABD e esta envia o conteúdo válido temporalmente de cada tabela cujos contadores não sejam iguais. Recebendo informação para ser sincronizada, a SegUAMóvel elimina todos os registos das tabelas respectivas (este processo é, assim, destrutivo), e insere os novos registos. No fim da actualização da sua base de dados, a SegUAMóvel armazena os novos contadores (os contadores actuais da SegUABD). A integridade referencial na SegUAMóvelBD é mantida pois, a SegUABD garante a sua própria

³⁷ Timestamps

integridade, o que significa que quando um registo na SegUABD for removido, o contador da sua tabela será incrementado e, assim, será descarregada pela SegUAMóvel como parte do pacote de actualizações. Caso a SegUABD não sofra alterações até à próxima verificação da SegUAMóvel, os contadores irão corresponder, não sendo necessário sincronizar informação. A Figura 50 ilustra este processo.

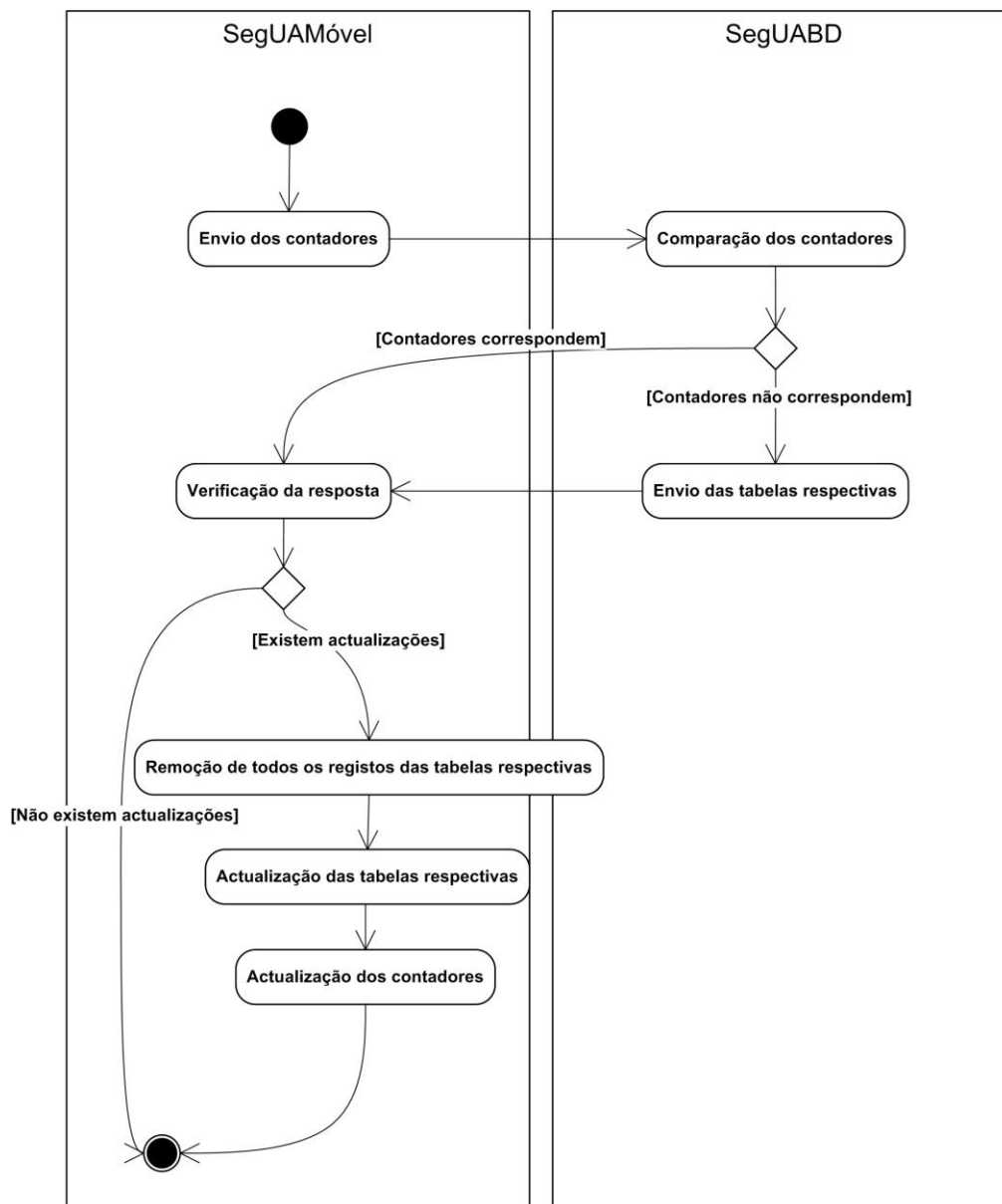


Figura 50 – Sincronização dos dados

É importante salientar que apenas os dados com que a SegUAMóvel trabalha são sincronizados por este processo, enquanto que os dados produzidos pela SegUAMóvel têm que ser enviados através dos métodos existentes no SegUAService. Antes de iniciar a sincronização, devido ao processo de sincronização na SegUAMóvel ser destrutivo, é necessário que este envie primeiro os dados que, possivelmente, possui armazenados devido a alguma falta de conectividade anterior e que ainda não tenham sido enviados. O processo de sincronização é,

assim, composto por duas fases: envio de dados produzidos, e sincronização de dados de trabalho. O envio pela SegUAMóvel dos seus dados produzidos é feito da mesma forma que o envio em normal, reutilizando, assim, a infra-estrutura existente no SegUAService; relativamente à sincronização de dados de trabalhos, o SegUAService encaminha os contadores da SegUAMóvel à SegUABD onde esta envia o conteúdo das tabelas respectivas, que é todo agrupado num só pacote pelo SegUAService. A Figura 51 ilustra as duas fases envolvidas no processo de sincronização.

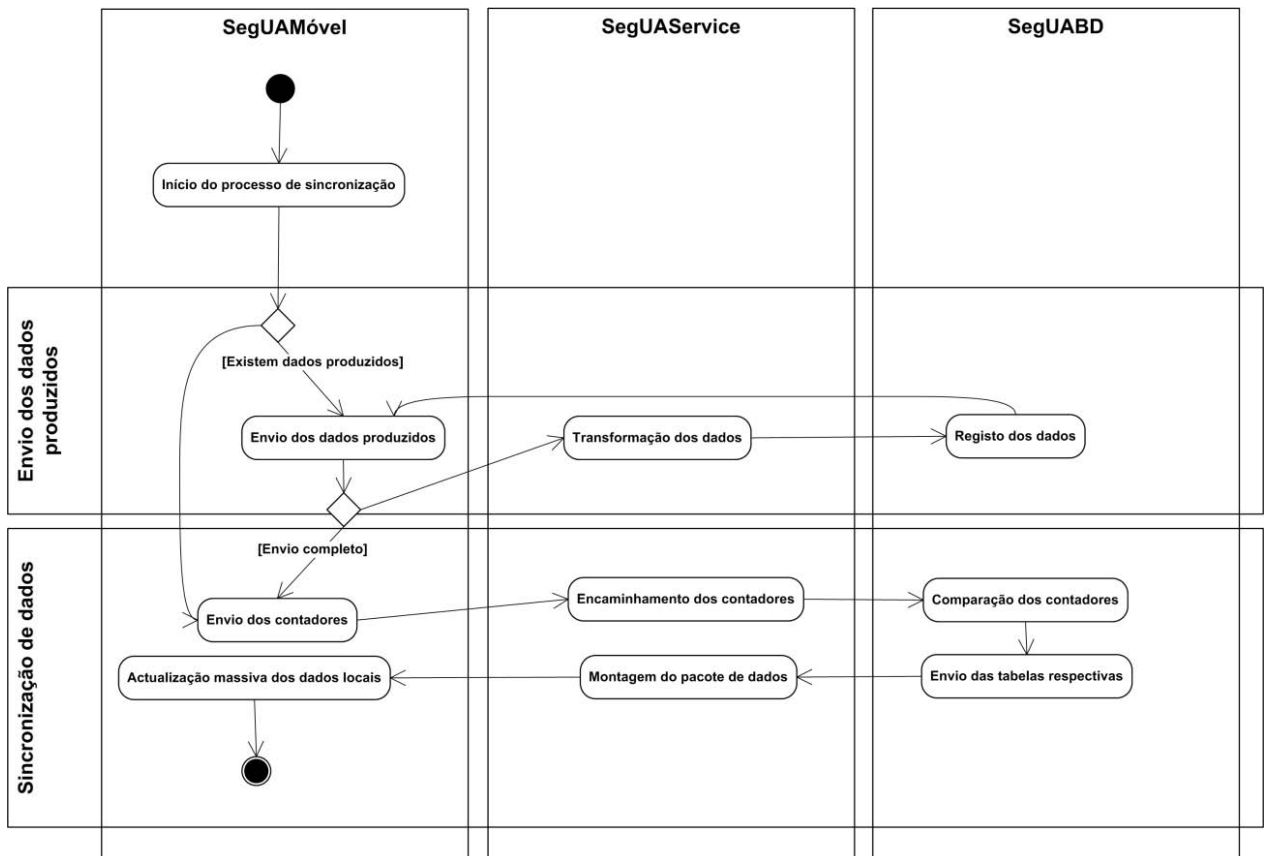


Figura 51 – Processo global de sincronização de dados

Desta forma, todos os Terminais mantêm-se sincronizados sem que seja necessário a SegUAMóvel ou à SegUABD manter algum tipo de histórico de alterações para o efeito e, dado que em cada sincronização não é enviado o conteúdo integral da SegUABD, permite-se, assim, poupar largura de banda e tempo.

5.1.2.1 Funcionalidades


A SegUAWeb oferece apenas funções básicas de edição e visualização dos dados do sistema. O Supervisor tem uma visão global das rondas em curso, incluindo visualização geográfica da localização actual dos Vigilantes, num dado momento, como ilustra a Figura 52.

Rondas em curso:

PerformedRoundID	StartDate	CompletionDate	RealName	Name	DateOnDB	LastCheckPoint
1	09-10-2009 01:38:45	09-10-2109 01:39:11	dude	ronda 1	09-10-2009 01:39:11	
2	09-10-2009 01:46:44	09-10-2109 01:47:10	dude	ronda 1	09-10-2009 01:47:10	
3	09-10-2009 01:51:43	09-10-2109 01:52:08	dude	ronda 1	09-10-2009 01:52:08	
4	21-10-2009 23:19:13	21-10-2109 23:41:00	dude	ronda 1	21-10-2009 23:41:00	EEETA
5	22-10-2009 16:38:04	22-10-2109 17:00:05	dude	ronda 1	22-10-2009 17:00:06	DETI
6	25-10-2009 02:23:09	25-10-2109 03:45:40	dude	ronda 1	25-10-2009 03:45:41	

Últimas localizações:

Latitude	Longitude	Name	RealName
40,6324376642547	-8,65889489650726	ronda 1	dude



Ver ronda em detalhe:

1 ▲

2 ▢

3 ▾

4 ▼

Ver

Figura 52 - Visualização das rondas em curso

Pode ser gerado um relatório diário com a lista de rondas realizadas e os incidentes registados num dia especificado, como ilustra a Figura 53.

Relatórios diários:

Dia:

Outubro de 2009								
≤	seg	ter	qua	qui	sex	sáb	dom	≥
>	28	29	30	1	2	3	4	
>	5	6	7	8	9	10	11	
>	12	13	14	15	16	17	18	
>	19	20	21	22	23	24	25	
>	26	27	28	29	30	31	1	
>	2	3	4	5	6	7	8	

Rondas realizadas:

Ocorrências:

OccurrenceID	OccurrenceDate	DateOnDB	SituationFamilyName	SituationName	CauseName	Tipo	nPhoto	nVideo	nText	nAudio
1	20-10-2009 21:37:20	15-10-2009 00:12:58	Bens/Equipamentos	Elevadores	Alarme inoperacional	Exterior	0	0	0	0
2	20-10-2009 21:37:20	16-10-2009 23:22:33	Bens/Equipamentos	Elevadores	Alarme inoperacional	Exterior	0	0	0	0
3	20-10-2009 21:37:20	18-10-2009 15:52:46	Bens/Equipamentos	Elevadores	Alarme inoperacional	Exterior	0	0	0	0
4	20-10-2009 21:37:20	18-10-2009 16:04:52	Bens/Equipamentos	Elevadores	Alarme inoperacional	Exterior	0	0	0	0
5	20-10-2009 21:37:20	18-10-2009 16:08:00	Bens/Equipamentos	Elevadores	Alarme inoperacional	Exterior	0	0	0	0
6	20-10-2009 21:37:20	18-10-2009 16:15:52	Bens/Equipamentos	Elevadores	Alarme inoperacional	Exterior	0	0	0	0
7	20-10-2009 21:37:20	18-10-2009 16:38:45	Bens/Equipamentos	Elevadores	Alarme inoperacional	Exterior	0	0	0	0
8	20-10-2009 21:37:20	18-10-2009 16:48:59	Bens/Equipamentos	Elevadores	Alarme inoperacional	Exterior	0	0	0	0
9	20-10-2009 21:37:20	18-10-2009 16:49:51	Bens/Equipamentos	Elevadores	Alarme inoperacional	Exterior	0	0	0	0
10	20-10-2009 22:11:01	20-10-2009 22:32:50	Bens/Equipamentos	Elevadores	Alarme inoperacional	Exterior	0	0	0	0
11	20-10-2009 22:29:39	20-10-2009 22:51:27	Bens/Equipamentos	Elevadores	Alarme inoperacional	Exterior	0	0	0	0

Figura 53 – Relatórios diários

A categorização de incidentes, ilustrada pela Figura 54.

Conteúdo prédefinido:

Famílias:

	SituationFamilyID	SituationFamilyName	Active
Edit Select	1	Bens/Equipamentos	<input checked="" type="checkbox"/>
Edit Select	2	Sistemas Electronicos de Segurança	<input checked="" type="checkbox"/>
Edit Select	3	Incidentes	<input checked="" type="checkbox"/>

Situações:

	SituationID	RefSituationFamilyID	SituationName	Active
Edit Select	1	1	Janela Aberta	<input checked="" type="checkbox"/>
Edit Select	2	1	Janela Danificada	<input checked="" type="checkbox"/>
Edit Select	3	1	Janela Vidro Partido	<input checked="" type="checkbox"/>
Edit Select	4	1	Porta Aberta	<input checked="" type="checkbox"/>
Edit Select	5	1	Porta Danificada	<input checked="" type="checkbox"/>
Edit Select	6	1	Porta Vidro Partido	<input checked="" type="checkbox"/>
Edit Select	7	1	Elevadores	<input checked="" type="checkbox"/>
Edit Select	8	1	Equipamento Urbano Danificado	<input checked="" type="checkbox"/>
Edit Select	9	1	Furtos	<input checked="" type="checkbox"/>

- ▣ Rondas
 - Gerir rondas
 - Rondas em curso
 - Rondas realizadas
 - Pontos de controlo
 - Mapa de calor
- ▣ Relatórios
 - Relatórios diários
- ▣ Ocorrências
 - Gerir conteúdo pré-definido
 - Lista de ocorrências
 - Mapa de calor
- ▣ Gerir utilizadores
 - Utilizadores
- ▣ Avançado
 - Integração SIG
 - Gestão de dispositivos

Causas:

	CauseID	CauseName	RefSituationID	Active
Edit Select	1	Avariado	7	<input checked="" type="checkbox"/>
Edit Select	2	Iluminação deficiente	7	<input checked="" type="checkbox"/>
Edit Select	3	Alarme inoperacional	7	<input checked="" type="checkbox"/>
Edit Select	4	Falha de energia de 220V	7	<input checked="" type="checkbox"/>
Edit Select	5	Pessoal retido no seu interior	7	<input checked="" type="checkbox"/>

Adicionar novo conteúdo:

Nova família:

Nova situação:

Associar a : Bens/Equipamentos

Nova causa:

Associar a : Bens/Equipamentos

Associar a : Janela Aberta

Figura 54 – Categorização de incidentes

5.2 Arquitectura de instalação

A SegUABD necessita de um sistema de gestão de bases de dados que suporte procedimentos e vistas. O SegUAService e a SegUAWeb necessitam de um servidor web capaz de servir páginas dinâmicas. Para a SegUAMóvel, é necessário um dispositivo que tenha como sistema operativo o Windows Mobile 6, e que possua componente GPS, ecrã sensível ao toque, e conectividade 802.11 ou móvel. A Figura 55 ilustra a arquitectura de instalação.

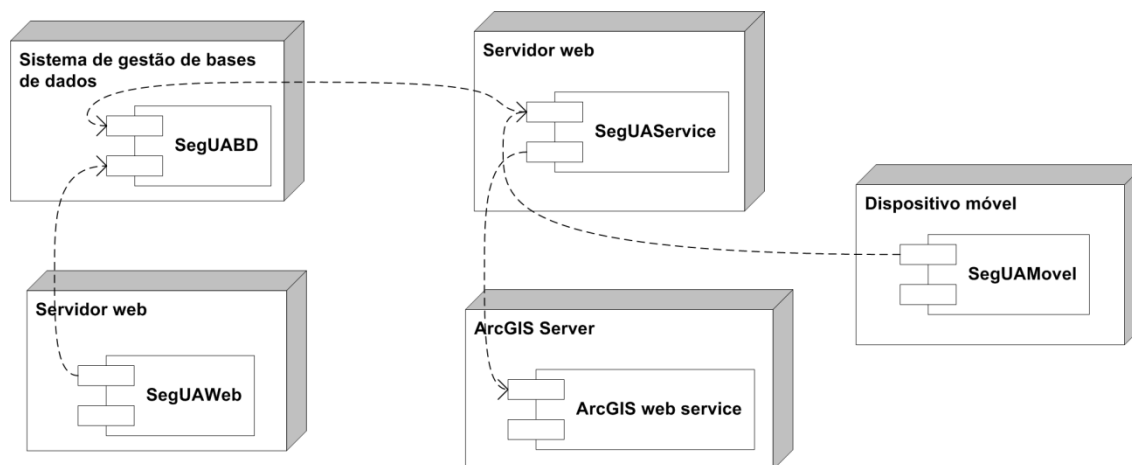


Figura 55 – Arquitectura de instalação do sistema SegUA

5.3 Interação com o utilizador

A interação com o utilizador pode ter tanta importância como têm as funcionalidades da aplicação pois, se o utilizador não conseguir realizar as suas tarefas de uma forma confortável e correcta, a aplicação pode perder a sua utilidade sendo, assim, importante que a interação seja algo mais que imprimir informação e receber ordens do utilizador. Uma má interação com o utilizador tem, assim, impacto emocional, onde o utilizador se sente confuso e frustrado, produzindo um mau desempenho no seu trabalho; e um impacto económico, onde o mau desempenho do utilizador leva a que o trabalho não seja realizado e, portanto, problemas de tempo e de custo [58].

No que diz respeito à interface gráfica, os botões³⁸ no fundo do ecrã têm posição fixa e são facilmente acessíveis, por isso, são usados para criar um fluxo de navegação suave entre formulários. Para criar uma ideia de previsibilidade, o botão esquerdo é utilizado para recuar na acção, e o botão direito é utilizado para avançar. A Figura 56 ilustra um exemplo deste fluxo de navegação, onde o utilizador pretende adicionar um incidente. Após observar que não existe um incidente no seu local, o utilizador avança para a escolha da família de incidentes, e depois avança para as situações da família escolhida. Para alterar a família, o utilizador irá recuar.

³⁸ Soft-buttons



Figura 56 – Fluxo de navegação da interface

5.3.1 Desafios e soluções para aumentar a facilidade de utilização

Devido ao tamanho reduzido dos ecrãs destes dispositivos, as interfaces gráficas em dispositivos móveis exigem mais trabalho em relação às interfaces normais, pois é necessário encontrar formas de apresentar a mesma informação e controlos num espaço muito mais reduzido e sem que isso interfira com a usabilidade.

Os dispositivos móveis apresentam grande diversidade quanto às suas características físicas, nomeadamente, diferentes tamanhos e resoluções de ecrã, teclados físicos ou virtuais, presença de sensores que detectam a orientação do dispositivo, entre outros. Desta forma, um grande obstáculo no desenho de interfaces para aplicações móveis é o de criar uma interface que funcione em qualquer dispositivo, independentemente das suas características e da orientação.

Geralmente, os dispositivos móveis possuem teclas de navegação, por exemplo, as de 4 sentidos que permitem ao utilizador interagir com a aplicação através do teclado. No entanto, este tipo de interacção é pouco interessante pois pode ser necessário percorrer todos os controlos para chegar ao controlo pretendido.

A sobrecarga de informação e de controlos num formulário pode gerar confusão no utilizador e aumenta o tempo necessário para realizar uma tarefa e o risco de erro. Assim, para simplificar a interface, a informação foi distribuída em vários formulários, permitindo que o utilizador foque a sua atenção em cada passo e, assim, reduzir os erros. Além disso, o utilizador apenas tem de introduzir texto quando é estritamente necessário, tendo-se apostado numa estratégia de “selecção ao invés de introdução”[59].

Os dispositivos móveis com ecrã resistivo (sensíveis à pressão) incluem um apontador físico, a *stylus*, que facilita a interacção física do utilizador com o dispositivo, nomeadamente em situações onde é necessária precisão. No entanto, o uso do dedo como apontador é mais natural, sendo desejável que o utilizador utilize a *stylus* apenas quando necessário.

Analisando a Figura 57, observa-se que, utilizando *Combobox*, o utilizador poderá ter problemas em seleccionar o item pretendido pois, devido ao tamanho do dedo representado pelo quadrado vermelho, facilmente poderá pressionar um elemento indesejado em vez da barra lateral de *scroll* tendo que repetir o processo de selecção para seleccionar o item correcto. Utilizando o *Listbox*, se o utilizador pressionar um elemento indesejado, não haverá consequências pois o utilizador poderá facilmente corrigir a selecção bastando corrigir a posição do dedo. Assim, a escolha dos controlos da interface influencia, também, na usabilidade.



Figura 57 – Comparação entre os controlos com utilizado do dedo

A introdução de texto pode ser complicada, especialmente se o dispositivo não possuir um teclado físico, pois as teclas ou botões são, geralmente, condensadas ou de tamanho reduzido. Uma solução pode ser o reconhecimento de escrita à mão, onde o utilizador escreve desenhando no próprio ecrã com o dedo ou *stylus*, e a aplicação analisa o desenho e reconhece o texto introduzido. No entanto, este reconhecimento não está disponível na NETCF e existem apenas soluções comerciais, portanto, esta solução não pôde ser explorada neste projecto. Este problema pode ser minimizado simplificando o método de introdução e os dados esperados. Por exemplo na autenticação do utilizador, substituiu-se a exigência de palavra-chave alfanumérica por uma palavra-chave numérica, como ilustra a Figura 58.



Figura 58 – Simplificação da introdução de dados

5.4 Testes e verificação

O componente GPS do dispositivo móvel tem um papel importante na aplicação móvel, tanto da perspectiva do utilizador, para saber a sua localização e permitir que a aplicação gira a sua ronda, como do ponto de vista administrativo, para observar se o percurso efectuado pelos Vigilantes é o correcto. Sabendo que existem vários factores que influenciam as localizações obtidas, é necessário analisar a qualidade destas localizações. Foi efectuado um percurso pelo campus da Universidade sob condições atmosféricas favoráveis e sempre com linha de vista para o céu, tendo sido registado com um HTC Cruise³⁹. Comparando o registo com o percurso real efectuado, ilustrados pela Figura 59, observa-se que são muito semelhantes, o que valida a precisão do componente GPS.

³⁹ Windows Mobile 6, com o software Catfood Tracker 1.31.0002

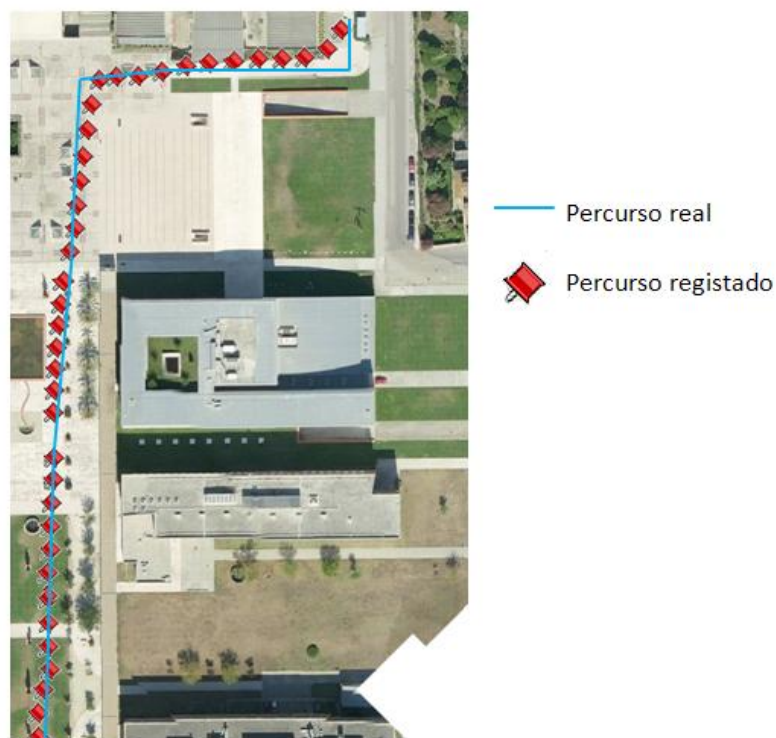


Figura 59 – Percurso registado vs percurso real

Efectuando um teste de precisão num ponto fixo, sem movimento do dispositivo, observou-se uma variação⁴⁰ de cerca de 0.2 metros nas localizações fornecidas pelo componente GPS, assumindo-se este valor como a imprecisão do componente. Na possibilidade de ser utilizada uma ferramenta online para definir pontos de controlo ou áreas de georeferenciação, inseriu-se uma localização no Bing Maps⁴¹ com vista aérea, e obteve-se uma posição a cerca de 2 metros da posição real, como ilustra a Figura 60. Face a estes resultados, conclui-se que existem condições satisfatórias para o uso do componente GPS e de ferramentas externas, apesar de ser necessário ceder uma pequena margem de erro.



Figura 60 – Posição registada vs posição real

⁴⁰ Registaram-se as localizações, e calculou-se a distância entre elas.

⁴¹ Foi também testado o Google Maps, porém com resultados péssimos.

Observou-se que o componente GPS fornece uma taxa de actualização de 1 localização por segundo, ou seja, assumindo que o utilizador se desloca a uma velocidade média de 1 metro por segundo, significa que no espaço de 1 segundo se terá deslocado 1 metro, distância que poderá ser tida em conta na definição das áreas de georeferenciação de edifícios, assim como a imprecisão do componente GPS observada anteriormente. Além disso, foram efectuadas medições do tempo necessário para a aquisição de sinal para se obter uma localização, apresentadas na Tabela 3.

Tentativa	Tempo de espera
1	02m29s
2	01m44s
3	02min44s
4	03min52s

Tabela 3 – Medição de tempos de aquisição de sinal

Analisando os resultados, observa-se que o componente GPS demora um tempo algo significativo para adquirir um sinal válido para cálculo da localização, o que significa que o fluxo de trabalho de trabalho da SegUAMóvel não pode depender apenas da existência de uma localização, pelo menos, inicialmente, devendo oferecer formas alternativas, por exemplo, permitindo que o utilizador defina manualmente a sua localização. A utilização da tecnologia Assisted GPS (AGPS) permite reduzir os tempos de aquisição de sinal, no entanto, suporte para esta tecnologia não está presente em todos os dispositivos nem em todos os operadores móveis[60]. Porém, é possível emular o AGPS transferindo, ao nível aplicacional, os dados necessários de servidores para o efeito e fornecê-los ao dispositivo, reduzindo, assim o tempo necessário para aquisição.

A georeferenciação por áreas rectangulares revelou-se difícil de aplicar, devido à sua pouca flexibilidade com edifícios que se encontram orientados de uma forma oblíqua ao referencial geográfico. Estes edifícios requerem um grande número de áreas comparando com edifícios dispostos horizontalmente ou verticalmente, além de criarem problemas de intercepção com outros edifícios, como ilustra a Figura 61.



Figura 61 – Comparação entre a direcção na georeferenciação

No sentido de se determinar o quão a área de georeferenciação deve ser maior que o edifício que representa, o desempenho da georeferenciação com áreas foi testado no terreno,

através da definição de áreas adjacentes a um edifício, ilustradas na Figura 62, e observação do método em cada uma destas áreas. A área A tem 16 metros entre o seu limite e o contorno edifício (em branco), a área B tem 7 metros, e a área C tem 3 metros.

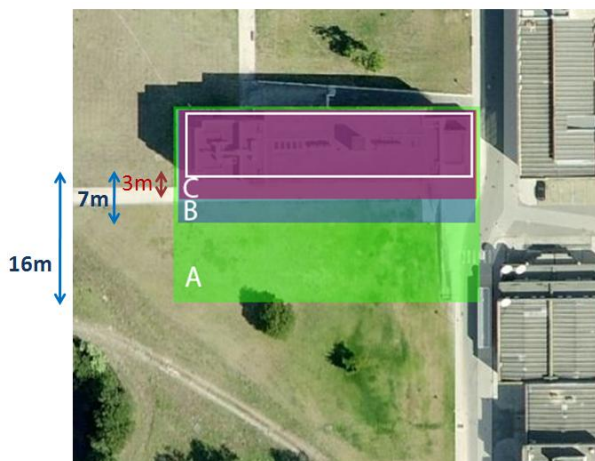


Figura 62 – Teste da georeferenciação

O método detectou sem problemas a sua presença na área A, revelando grande precisão na transição entre a sua área e a área circundante (não georeferenciada). A transição para a área B revelou elevada taxa de sucesso, mas revelou também, alguma instabilidade, demorando, por vezes, um período de tempo significativo a determinar que existiu transição, mas não o suficiente para impedir que o método detecte o edifício antes de o utilizador entrar neste. Quanto à área C, observou-se grande instabilidade, chegando ao ponto de não existir qualquer determinação na maior parte dos testes. A grande diferença na taxa de sucesso entre as várias áreas pode ser explicada pela redução da linha de vista para o céu com a aproximação ao edifício (e consequente aumento da sua altura), e pode-se inferir que a dimensão da área de georeferenciação depende, em grande parte, do grau de bloqueio da linha de vista que se observa no local, sendo necessário efectuar testes locais para se determinar a dimensão necessária.

As equações de tradução entre coordenadas pixel e coordenadas geográficas utilizadas no desenvolvimento no controlo de visualização de mapas da interface gráfica revelaram-se suficientemente correctas e precisas. Utilizando a mesma localização anterior, observa-se que a localização real e a localização obtida pelas equações coincidem, como ilustra a Figura 63.



Figura 63 – Localização obtida das equações de tradução

A fiabilidade dos resultados dos testes de localização depende, também, em grande parte do serviço de mapas utilizado para a obtenção do mapa do exterior do campus. A definição das áreas e a análise da precisão do GPS (localização fixa como percurso) foi feita através do serviço *Bing Maps*, que foi o serviço⁴² que se revelou mais preciso na comparação entre as localizações obtidas no terreno e as localizações obtidas pela visualização dos mapas. A apresentação do mapa no dispositivo é feita através de uma imagem estática, o que levanta problemas de memória devido ao limite de 32MB de memória virtual por processo que o Windows Mobile impõe. Uma imagem de alta resolução permite apresentar um mapa detalhado e reduzir a imprecisão das equações de tradução de coordenadas, no entanto, uma imagem desta natureza requer grandes quantidades de memória, podendo ultrapassar facilmente o limite de memória virtual. Por exemplo, uma imagem com resolução 1000x1000 pixels com 24 bits por píxel pesará $1000 \times 1000 \times 24 = 22.8\text{MB}$. Desta forma, é necessário encontrar um equilíbrio entre resolução e detalhe do mapa, ou então implementar um método de carregamento parcial da imagem na memória.

Para determinar a eficiência da sincronização de dados de trabalhos entre as bases de dados SegUAMóvelBD e SegUA, foram efectuadas medições de tempo (na SegUAMóvel), sob condições óptimas⁴³, para diferentes cenários de actualizações, apresentadas na Tabela 4.

Cenário	Duração (em segundos)	Tráfego gerado	Descrição
Actualização inicial	609	300KB	Edifícios, pisos, salas, e algumas rondas, sem mapas
Actualização subsequente à inicial	6	1KB	Nenhuma alteração
Actualização de alguns dados	55	8KB	1 utilizador, 10 pontos de controlo, 1 situação, 5 causas

Tabela 4 – Medição de tempos de sincronização

⁴² De entre Bing Maps e Google Maps

⁴³ Força de sinal de rede sem fios máximo, SegUAService e SegUABD instalados na mesma máquina.

Como seria de esperar, a actualização inicial representa o cenário mais pesado e mais longo sendo, esta operação, uma operação que, normalmente, será executada aquando da instalação da SegUAMóvel, antes de se fornecer os dispositivos aos utilizadores. Neste cenário, onde as plantas dos pisos não foram incluídos no pacote de actualizações, foram sincronizados cerca de 30 edifícios, e mais de 1000 pisos e salas, entre outros dados, o que gerou uma grande quantidade de ciclos na aplicação, o que justifica a duração observada. Os restantes cenários são cenários prováveis entre várias sessões de utilizações normais, onde uma sincronização sem alterações demora apenas 6 segundos, enquanto que a actualização com algumas alterações demora 55 segundos a ser completada. Face a este tempo e dada a baixa quantidade de informação, tanto alterada como transmitida, pode-se inferir que o método de sincronização de dados de trabalho é muito ineficiente, e que este método não é escalável com o crescimento de informação na SegUABD. É de referir que o encadeamento entre a SegUAMóvel, SegUAService e SegUABD (e desempenho das máquinas destes elementos), assim como os protocolos de transmissão usados, interpretação do XML e operações internas de conversão também têm o seu peso no processo. Desta forma, verifica-se que é necessário otimizar o método, ou que pode ser necessário investir em soluções proprietárias de sincronizações de bases de dados, como a *Merge Replication* para sistemas Microsoft. Uma solução mais drástica seria, antes de iniciar uma ronda, inserir o cartão de memória do dispositivo num computador de secretária mais potente, e executar a sincronização neste computador.

No sentido de verificar o impacto do desempenho reduzido do dispositivo, foram efectuadas medições de tempo na execução de algumas tarefas que podem ser delegadas à aplicação após ordem de envio, com e sem *multithreading*, estando os resultados apresentados na Tabela 5.

Tarefa	Duração sem multithreading (em segundos)	Duração com multithreading (em segundos)	Descrição
Envio de incidente	10	0	Com nota de texto, de voz, e 2 fotografias
Indicação manual de ponto de controlo visitado	3	0	

Tabela 5 – Comparação single threading vs multithreading

Apesar de estas medições terem sido efectuadas sob condições óptimas⁴⁴, observa-se que, sem *multithreading*, a duração de execução destas tarefas pode ter impacto negativo e significativo na experiência do utilizador, enquanto que a execução das tarefas com *multithreading* é instantânea, do ponto de vista do utilizador. Pode-se, assim, inferir que a utilização de modelos de processamento paralelo são especialmente importantes para a experiência do utilizador quando a capacidade é reduzida.

⁴⁴ Força de sinal de rede sem fios máximo, SegUAService e SegUABD instalados na mesma máquina

Um dos pontos-chave do sistema SegUA é o de comunicar a informação em tempo-real e, para tal, é necessário que exista uma boa cobertura da rede sem fios nos campus da Universidade, para que seja sempre possível enviar a informação, incluindo no exterior. Foram efectuados testes de conectividade com a rede sem fios no exterior através da realização de um percurso pelo campus, registando o estado e mapeando com o GPS, estando os resultados apresentados na Figura 64. Analisando estes resultados, verifica-se que existe cobertura mesmo a grandes distâncias dos edifícios, permitindo, assim, que as rondas com pontos de controlo exteriores sejam realizadas sempre com comunicação com o SegUAService.

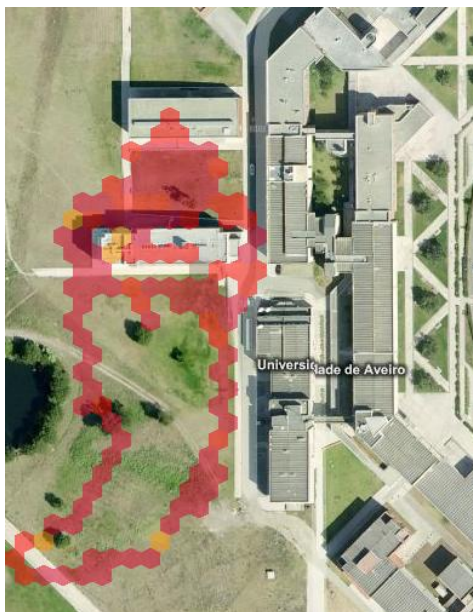


Figura 64 – Mapa de calor da potência do sinal

Devido a restrições de tempo, não foram efectuados testes quanto ao consumo de energia da bateria e efeito das políticas de gestão de energia implementadas.

Do ponto de vista prático, verifica-se que a SegUAMóvel atinge todos os seus objectivos, possibilitando a realização dos objectivos propostos nesta dissertação.

No que diz respeito aos outros elementos do sistema SegUA, é apenas notável referir que a integração dos dados do WebSIG para a SegUABD demora cerca de 8 minutos⁴⁵, para cerca de 30 edifícios, mais de 1000 pisos (e respectivas plantas) e salas, e que, na realidade, nem todas as salas estão registadas no WebSIG.

⁴⁵ Teste efectuado numa máquina algo antiga, com ligação à rede de 20Mb.

6 Resultados

No processo actual, o Vigilante demora, normalmente, cerca de 30 minutos a redigir o seu relatório da ronda, e o Supervisor demora cerca de 1 hora a redigir o seu relatório diário. O Técnico analisa os relatórios diários dos Supervisores e encaminha os incidentes para os Pivots através de email ou fax, demorando entre 30 minutos a 1 hora por dia nesta tarefa. No processo proposto, estas tarefas são realizadas automaticamente pelo sistema, portanto, o tempo dispendido será apenas de operação do sistema e visualização de informação. No novo processo, o Vigilante apenas tem de confirmar a correcção das informações da sua ronda recolhidas pelo sistema, estimando-se 5 minutos dispendidos nesta tarefa; o Supervisor tem de verificar o cumprimento da realização de rondas, estimando-se 10 minutos para esta tarefa; e o Técnico tem de verificar se todos os incidentes foram encaminhados para o Pivot respectivo, estimando-se 15 minutos para esta tarefa. Através destas estimativas de tempo dispendido nas tarefas, é possível estimar o tempo perdido ao longo prazo, e comparar os dois processos, estando apresentados os resultados na Tabela 6.

Papel	Tempo gasto por dia (horas)		Tempo gasto por mês ⁴⁶ (horas)	
	Processo actual	Processo proposto	Processo actual	Processo proposto
Vigilante	0.5	0.083	11	1.83
Supervisor	0.5-1	0.166	11-22	3.65
Técnico	0.5-1	0.25	11-22	5.5

Tabela 6 – Estimativas de tempo dispendido nas tarefas

Observa-se que a acumulação do tempo dispendido diariamente nestas tarefas é significativa. No processo actual, um Pivot recebe um incidente entre meio dia e um dia após a sua detecção, enquanto que, com o processo proposto, o Pivot recebe um incidente meros segundos após a sua detecção, podendo mesmo o Pivot tomar acção quase imediatamente após o registo do incidente. A Figura 65 ilustra a comparação entre o tempo decorrido até que o Pivot possa visualizar um incidente.

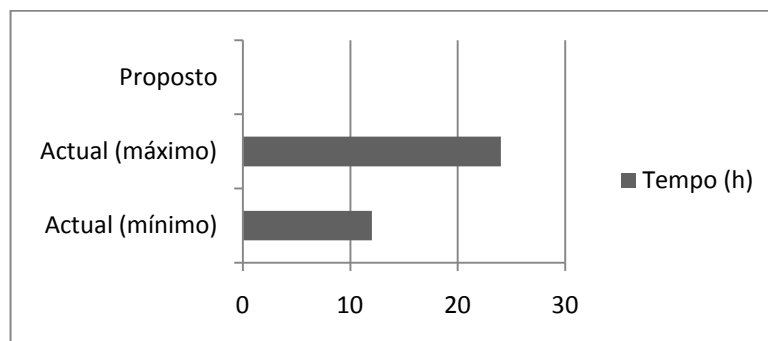


Figura 65 – Comparação entre o tempo de espera na visualização de incidentes nos processos

⁴⁶ Considerando 7 horas diárias em 22 dias.

Como se pode observar, o tempo necessário para o processo proposto é muito reduzido comparado com os tempos necessários do processo actual. Assim, através destas análises, conclui-se que o processo proposto é um processo mais eficiente que o processo actual.

Devido a restrições de tempo, não foi possível realizar testes com utilizadores.

7 Conclusões

A introdução da tecnologia permitiu eliminar passos intermédios no processo de trabalho actual, e assim, reduzir o tempo de resposta e libertar os funcionários da realização de certas tarefas. Devido à capacidade de monitorização das rondas em curso e de emissão de alertas, tanto pelo Supervisor no módulo central, como pelo próprio módulo móvel, foi possível tornar o trabalho do Vigilante mais seguro do ponto de vista da sua integridade física.

Foi possível reduzir a burocracia envolvida no processo actual, nomeadamente na redacção de relatórios, pois estes relatórios já não são necessários no novo processo de trabalho, existindo apenas para efeitos de visualização. As soluções apresentadas permitem uma melhor realização das tarefas por parte de cada papel envolvido no processo, e permitem uma organização central da informação, organização que pode ser utilizada para melhorias na qualidade de serviço do Serviço de Vigilância através de *data mining*, como, por exemplo, detecção de zonas com pouca Vigilância ou zonas com grande número de incidentes e que, por isso, exijam rondas mais frequentes.

A aplicação de conceitos de Computação Móvel permitiu usufruir-se das vantagens da tecnologia, que normalmente apenas se obtêm dentro do espaço de um escritório, em pleno ar livre, como acesso constante à informação.

O comportamento adaptativo permite dotar as aplicações de uma flexibilidade para que estas possam responder à imprevisibilidade e condições do ambiente e, assim, fornecer uma melhor qualidade de serviço aos seus utilizadores. A integração de sistemas possibilita a apresentação de um maior conjunto de informação aos utilizadores, permitindo que estes realizem o seu trabalho com maior qualidade. Desta forma, estes conceitos acrescentam grande valor ao sistema, tanto do ponto de vista do utilizador como do ponto de vista de engenharia, e são áreas que possuem grande potencial para inovação e desenvolvimento.

As plataformas de desenvolvimento de aplicações móveis⁴⁷ são muito limitativas e possuem integração incompleta, o que dificulta a implementação de aplicações, nomeadamente quando estas possuem requisitos avançados de integração com os dispositivos, podendo mesmo estes requisitos serem contraditórios com os próprios conceitos dos sistemas operativos móveis, especialmente no que diz respeito às políticas de gestão de energia destes, como foi o caso neste projecto. O desenvolvimento de interfaces gráficas é difícil devido à imprevisibilidade das acções dos utilizadores e à área reduzida de visualização dos dispositivos móveis, o que significa que o desenvolvimento de aplicações móveis exige uma maior atenção na interacção com o utilizador.

Assim, conclui-se que a introdução de tecnologia permite melhorar os processos de trabalhos existentes nas organizações, tanto ao nível temporal como ao nível da qualidade, e poupar os seus recursos, e que estas melhorias podem permitir que o investimento inicial na tecnologia pode seja rapidamente recuperado.

⁴⁷ Especificamente a NETCF.

7.1 Trabalho futuro

A georeferenciação por áreas necessita de ser melhorada, nomeadamente com utilização de polígonos irregulares para permitir uma georeferenciação mais fácil, do ponto de vista do utilizador, e uma delimitação das áreas mais precisa e suave. Dado que o componente GPS apenas recebe sinal no exterior, pode ser interessante estudar formas para localização no interior de edifícios, por exemplo, através da infraestrutura da rede sem fios[61].

A sincronização de dados móveis também necessita de ser melhorada para que seja mais escalável com grandes quantidades de dados, e que seja efectuada em apenas alguns segundos.

É necessário estudar uma integração mais completa dos dados do WebSIG, especialmente com os objectos existentes neste sistema, para que os utilizadores da SegUAMóvel possam visualizar vários tipos de plantas e indicar quais os objectos em questão.

A interface gráfica precisa de ser melhorada, especialmente para que a interacção com a aplicação possa ser feita com o dedo. É necessário estudar uma estratégia para contornar o limite de memória virtual que o Windows Mobile impõe às aplicações, de forma a ser possível apresentar um mapa com grande resolução.

A auto-actualização da SegUAMóvel, com assistência do SegUAService, por exemplo, é uma ideia interessante, pois permite a correcção de erros ou implementação de novas funcionalidades sem que seja necessário instalar manualmente a SegUAMóvel em cada dispositivo.

A facilidade de acesso que os *web services*, neste caso, o SegUAService, oferecem às aplicações é, também, indesejada pelo facto de que um indivíduo mal intencionado pode facilmente aceder aos serviços e causar problemas, sendo, desta forma, necessário implementar um esquema de segurança no acesso ao SegUAService.

O sistema SegUA representa uma base para a sua expansão com novas funcionalidades e, também, com novas tecnologias devido à interoperabilidade entre os vários elementos do sistema. As funcionalidades do SegUA restringem-se à realização de rondas e registo de incidentes, portanto, existem, possivelmente, outros processos dos STecUA que podem ser optimizados e integrados no sistema.

8 Bibliografia

- [1] Nurvitadhi, Eriko., *Trends in Mobile Computing: A Study of Mobile Phone Usage in the United States and Japan*. s.l. : Oregon State University, 2002.
- [2] Newton MessagePad (Original) Specs. *EveryMac*. [Online] 7 de Julho de 2009. [Citação: 7 de Outubro de 2009.]
http://www.everymac.com/systems/apple/messagepad/stats/newton_mp_omp.html.
- [3] Apple iPhone 3GS 16, 32 GB Specs. *EveryiPod*. [Online] 15 de Setembro de 2009. [Citação: 7 de Outubro de 2009.] <http://www.everyipod.com/iphone-specs/apple-iphone-3g-s-specs.html>.
- [4] Canalys., Smart mobile device shipments hit 118 million in 2007, up 53% on 2006. [Online] 5 de Fevereiro de 2008. [Citação: 25 de Outubro de 2009.]
<http://www.canalys.com/pr/2008/r2008021.htm>.
- [5] Gartner., Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 3.7 Per Cent Increase in Fourth Quarter of 2008. [Online] Gartner, 11 de Março de 2009. [Citação: 23 de Setembro de 2009.] <http://www.gartner.com/it/page.jsp?id=910112>.
- [6] —. Gartner Says Worldwide Smartphone Sales Grew 16 Per Cent in Second Quarter of 2008. [Online] 8 de Setembro de 2008. [Citação: 25 de Outubro de 2009.]
<http://www.gartner.com/it/page.jsp?id=754112>.
- [7] —. Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 11.5 Per Cent Increase in Third Quarter of 2008. [Online] 4 de Dezembro de 2008. [Citação: 25 de Outubro de 2009.] <http://www.gartner.com/it/page.jsp?id=827912>.
- [8] —. Gartner Says Global Economic Downturn Sparked Three-Way Battle for Third Position in Mobile Phone Market in Third Quarter 2008. [Online] 25 de Novembro de 2008. [Citação: 25 de Outubro de 2009.] <http://www.gartner.com/it/page.jsp?id=813812>.
- [9] —. Gartner Says Worldwide Mobile Phone Sales Grew 6 Per Cent in 2008, But Sales Declined 5 Per Cent in the Fourth Quarter. [Online] 2 de Março de 2009. [Citação: 25 de Outubro de 2009.]
<http://www.gartner.com/it/page.jsp?id=904729>.
- [10] —. Gartner Says Worldwide Mobile Phone Sales on Pace to Grow 11 Per Cent in 2008. [Online] 5 de Agosto de 2008. [Citação: 25 de Outubro de 2009.]
<http://www.gartner.com/it/page.jsp?id=736913>.
- [11] —. Gartner Says Worldwide Mobile Phone Sales Increased 14 Per Cent in First Quarter of 2008. [Online] 28 de Maio de 2008. [Citação: 25 de Outubro de 2009.]
<http://www.gartner.com/it/page.jsp?id=680207>.

- [12] —. Gartner Says Worldwide Mobile Phone Sales Increased 16 Per Cent in 2007. [Online] 2008 de Fevereiro de 2008. [Citação: 25 de Outubro de 2009.] <http://www.gartner.com/it/page.jsp?id=612207>.
- [13] —. Gartner Says Worldwide Mobile Phone Sales Increased 12 Percent in Second Quarter of 2008. [Online] 27 de Agosto de 2008. [Citação: 25 de Outubro de 2009.] <http://www.gartner.com/it/page.jsp?id=747414>.
- [14] —. Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 3.7 Per Cent Increase in Fourth Quarter of 2008. [Online] 11 de 03 de 2009. [Citação: 15 de 05 de 2009.] <http://www.gartner.com/it/page.jsp?id=910112>.
- [15] Yao, Paul e Durant, David., *.NET Compact Framework Programming with C#*. s.l. : Prentice Hall, 2004. 0-321-17403-8.
- [16] Babin, Steve., *Developing Software for Symbian OS*. s.l. : Symbian Press, 2007. 978-0-470-72570-2.
- [17] White, James P. e Hemphill, David A., *Java 2 Micro Edition*. s.l. : Manning, 2002. 1930110332.
- [18] Brannan, James A., *Phone SDK Programming: A Beginner's Guide*. s.l. : McGraw-Hill, 2009. 978-0-07-162650-7.
- [19] Meier, Reto., *Professional Android Application Development*. s.l. : Wiley, 2008. 978-0-470-34471-2.
- [20] Canalys., Smart phones defy slowdown . *Canalys*. [Online] 17 de Agosto de 2009. [Citação: 23 de Setembro de 2009.] <http://www.canalys.com/pr/2009/r2009081.htm>.
- [21] Quintã, André Figueiredo., *Integração de sistemas de produção*. s.l. : Universidade de Aveiro, 2008.
- [22] Santos, Vânia Eduarda Albuquerque Domingues., *Motor de integração de dados clínicos distribuídos e sua monitorização*. s.l. : Universidade de Aveiro, 2008.
- [23] Singh, A, et al., *An abandoned object detection system based on dual background segmentation*. s.l. : IIT Delhi, NICTA, University of Queensland, 2009.
- [24] Ivanov, Ivan, et al., *Towards generic detection of unusual events in video surveillance*. s.l. : École Polytechnique Fédérale de Lausanne, 2009.
- [25] Sagrebin, Maria e Pauli, Josef., *Real-time moving object detection for video surveillance*. s.l. : Universitat Duisburg-Essen, 2009.
- [26] Bixio, Luca, et al., *Distributed cognitive sensor network approach for surveillance applications*. s.l. : University of Genoa, 2009.

- [27] Li, Sheng-Tun Li, et al., *PDA Watch for Mobile Surveillance Services*. s.l. : National Kaohsiung First University of Science Technology, 2002.
- [28] Matussek, F., Pujolle, G e R., Reda., *VSLC: Video Surveillance Network Control, Mobile Video Surveillance Local Control, Engineering and Applications*.
- [29] Gualdi, Giovanni, Prati, Andrea e Cucchiara, Rita., *Video streaming for mobile video surveillance*. s.l. : IEEE, 2008.
- [30] Imai, Yoshiro, Hori, Yukio e Masuda, Shin'ichi., *Development and a brief evaluation of a web-based surveillance system for cellular phones and other mobile computing clients*. s.l. : Kagawa University, 2008.
- [31] Wu, Bing-Fei, et al., *An encrypted mobile embedded surveillance system*. s.l. : National Chiao Tung University, 2005.
- [32] Ferro, Duco e Salden, Alfons., *Self-organizing mobile surveillance security networks*. s.l. : Almende, 2007.
- [33] Schilit, William Noah., *A system architecture for context-aware mobile computing*. s.l. : Columbia University, 1995.
- [34] Mridula, Parihar., *ASP .NET Bible*. s.l. : Hungry Minds, 2002. 0764548166.
- [35] Christensen, Erik, et al., *Web Services Description Language (WSDL) 1.1*. [Online] World Wide Web Consortium, 2001. [Citação: 24 de Outubro de 2009.] <http://www.w3.org/TR/wsdl>.
<http://www.w3.org/TR/wsdl>.
- [36] Bray, Tim, et al., *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. [Online] World Wide Web Consortium, 2008 de Novembro de 2008. [Citação: 24 de Outubro de 2009.]
<http://www.w3.org/TR/xml/>.
- [37] Gudgin, Martin, et al., *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*. [Online] World Wide Web Consortium, 27 de Abril de 2007. [Citação: 24 de Outubro de 2009.]
<http://www.w3.org/TR/soap12-part1/>.
- [38] Pereira, Nuno Filipe dos Santos Beça., *Serviços Web e arquiteturas de serviços de broker para empresas ágeis e virtuais*. s.l. : Universidade de Aveiro, 2007.
- [39] Welling, Girish Sharad., *Designing adaptive environment-aware applications for mobile computing*. New Jersey : New Brunswick, 1999.
- [40] Junior, Carlos Fernando Crispim e Fernandes, Anita Maria da Rocha., *Análise das tendências tecnológicas para Computação Móvel aplicada à área da Saúde*. 2006.
- [41] , "Mobile Application Architecture Pocket Guide." [Online] 12 de Janeiro de 2009. [Citação: 7 de Outubro de 2009.]
<http://www.codeplex.com/AppArch/Release/ProjectReleases.aspx?ReleaseId=19798>.

- [42] Veiga, Ivo Gonçalves da., *HOPE+: Dispositivos móveis na avaliação de doentes em enfermagem*. Aveiro : Universidade de Aveiro, 2008.
- [43] Fowler, Marti, et al., *Patterns of Enterprise Application Architecture*. s.l. : Addison Wesley, 2002. 0-321-12742-0.
- [44] Bishop, Judith., *C# 3.0 Design Patterns*. s.l. : O'Reilly Media, 2008. 0-596-52773-X.
- [45] , "[MS-TDS]: Tabular Data Stream Protocol Specification." [Online] 24 de Agosto de 2009. [Citação: 7 de Outubro de 2009.] [http://msdn.microsoft.com/en-us/library/dd304523\(PROT.13\).aspx](http://msdn.microsoft.com/en-us/library/dd304523(PROT.13).aspx).
- [46] Darabant, Adrian Sergiu., *Implementing Efficient Data Synchronization for Mobile Wireless Medical Users*. s.l. : Babes-Bolyai University, 2008.
- [47] Huang, Yongqiang e GarciaMolina, Hector., *Publish/Subscribe in a Mobile Environment*. s.l. : Stanford University, 2001.
- [48] Cugola, Gianpaolo, Murphy, Amy L. e Picco, Gian Pietro., *Content-based Publish-Subscribe in a Mobile Environment*. s.l. : Politecnico di Milano, University of Lugano, 2005.
- [49] , *WGS 84 Implementation Manual*. Brussels : European Organization for the Safety of Air Navigation, Institute of Geodesy and Navigation, 1998.
- [50] DeMers, Michael N., *GIS For Dummies*. s.l. : Wiley Publishing, 2009. 978-0-470-23682-6.
- [51] , Power to the People. [Online] Microsoft, 1 de Agosto de 2005. [Citação: 7 de Outubro de 2009.] <http://blogs.msdn.com/windowsmobile/archive/2005/08/01/446240.aspx>.
- [52] Johnson, Joel Ivory., Windows Mobile Power Management. *CodeProject*. [Online] 21 de Março de 2009. [Citação: 7 de Outubro de 2009.] <http://www.codeproject.com/KB/mobile/WiMoPower1.aspx>.
- [53] Wigley, Andy, Mothand, Daniel e Foot, Peter., *Mobile Development Handbook*. s.l. : Microsoft Press, 2007. 9780735623583.
- [54] Reilly, Douglas J., *Designing Microsoft ASP.NET Applications*. s.l. : Microsoft Press, 2002.
- [55] Foggon, Damien, et al., *Programming Microsoft .NET XML Web Services*. s.l. : Microsoft Press, 2004. 0735619123.
- [56] Malheiro, Miguel dos Santos., *Guia turístico pessoal baseado em contexto através de PDA (servidor)*. s.l. : Universidade de Aveiro, 2008.
- [57] Steiner, Andreas., *A Generalisation Approach to Temporal Data Models and their Implementations*. s.l. : ETH Zurich, 1997.
- [58] Jones, Matt e Marsden, Gary., *Mobile Interaction Design*. s.l. : John Wiley & Sons, 2006. 0-470-09089-8.

- [59] Weiss, Scott., *Handheld Usability*. s.l. : John Wiley & Sons, 2002. 0 470 84446 9.
- [60] Kaplan, Elliott D. e Hegarty, Christopher J., *Understanding GPS - Principles and Applications*. s.l. : Artech House, 2006. 1-58053-894-0.
- [61] Nunes, Bruno Astuto Arouche., *Um sistema de localização para redes wi-fi baseado em níveis de sinal e modelo referenciado de propagação*. s.l. : Universidade Federal do Rio de Janeiro, 2007.
- [62] Microsoft., "Messaging in the .NET Compact Framework." *MSDN*. [Online] [Citação: 25 de Maio de 2009.] <http://msdn.microsoft.com/en-us/library/bb397842.aspx>.
- [63] Fernandes, Nelson Manuel Loureiro., *Gestão de tráfego urbano: integração GPS/SIG/PDA*. s.l. : Universidade de Aveiro, 2008.
- [64] Basílio, João Roberto Ascenção., *Sistemas web acessíveis por qualquer dispositivo*. s.l. : Universidade de Aveiro, 2008.
- [65] Koch, Peter-Paul., Mobile compatibility tables. *quirksmode*. [Online] [Citação: 6 de Outubro de 2009.] <http://www.quirksmode.org/m/table.html>.
- [66] Yang, Baijian, Zhengand, Pei e Ni, Lionel., *Professional Microsoft Smartphone Programming*. s.l. : Wrox Press, 2007. 9780471762935.
- [67] , Accessibility and Ergonomic Guidelines. [Online] Microsoft. [Citação: 7 de Outubro de 2009.] <http://msdn.microsoft.com/en-us/library/bb158589.aspx>.
- [68] Gartner., Gartner Says Worldwide Smartphone Sales Grew 29 Percent in First Quarter of 2008. [Online] 6 de Junho de 2008. [Citação: 25 de Outubro de 2009.] <http://www.gartner.com/it/page.jsp?id=688116>.

9 Anexos

9.1 Modelo de datos

