

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Aplicação de Registo de Ocorrências para Dispositivos Móveis

Hugo Manuel Zenha de Pinho

Relatório de Projecto

Mestrado Integrado em Engenharia Informática e Computação

Orientador: João Manuel Paiva Cardoso

Doutor em Engenharia Electrotécnica e Computadores

28 Julho 2009

Aplicação de Registo de Ocorrências para Dispositivos Móveis

Hugo Manuel Zenha de Pinho

Relatório de Projecto

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo júri:

Presidente: António Fernando Vasconcelos Cunha Castro Coelho (Professor Auxiliar)

Arguente: João Miguel Lobo Fernandes (Professor Associado)

Vogal: João Manuel Paiva Cardoso (Professor Associado)

16 de Julho de 2009

Resumo

A mobilidade no mundo do trabalho é cada vez mais uma constante. Desde que foi criado o primeiro telemóvel, passando pela criação de plataformas que possibilitam trabalhar em qualquer ponto do planeta, como é o caso de dispositivos do tipo *Personal Digital Assistant*, a preocupação de muitas entidades é cada vez mais a de disponibilizar tais ferramentas aos seus colaboradores.

Seguindo esta linha de orientação, a **Câmara Municipal do Porto** sentiu a necessidade não só de integrar os seus serviços de fiscalização, mas também de disponibilizar ao corpo de elementos encarregue de tais serviços ferramentas que aumentem a sua eficácia e eficiência, alicerçadas numa cultura de responsabilidade e que potenciem o prazer pelo trabalho realizado. Uma das ferramentas possíveis, e que será o tema objecto do presente trabalho, utiliza os dispositivos móveis permitindo aos *Agentes Fiscalizadores* realizar o trabalho de registo de ocorrências no terreno, verdadeiramente o seu local de trabalho, e não atrás de uma secretária.

Assim, neste relatório é apresentado todo o trabalho realizado para a criação de uma solução que satisfaça as necessidades da Câmara Municipal do Porto e as dos seus colaboradores. À medida que o problema foi sendo abordado com maior detalhe, percebeu-se que o grande objectivo do projecto era o da criação de um sistema que englobasse todo o processo de reporte de ocorrências no terreno e também o processo de gestão do trabalho dos elementos responsáveis por tal reporte. Por conseguinte, o sistema é composto por dois sub-sistemas, em que um se ocupa da parametrização e da gestão do sistema, enquanto o outro é o responsável por providenciar mecanismos que transportem o trabalho de fiscalização para o terreno.

Através de análises tecnológicas e de uma comunicação intensa com a Câmara Municipal do Porto foi possível a criação de uma solução, apresentada ao longo deste relatório, que procura aproveitar todas as vantagens inerentes à utilização de dispositivos móveis. No final são apresentados os resultados obtidos da utilização da solução idealizada na construção de um protótipo. Este protótipo valida e demonstra a viabilidade da construção do sistema. Importa frisar que, devido ao nível de abstracção conseguido, esta solução apresenta potencial para lidar com o serviço de fiscalização a nível de um município, assim como poderá servir a qualquer outra entidade que necessite de um sistema para reporte de ocorrências, ou seja, o sistema criado através da solução apresentada não está limitado pelo tipo de informação que nele flui.

A informação deste relatório representa o primeiro passo dado em direcção à criação de um sistema que poderá vir a afirmar-se como uma referência em sistemas de reporte de ocorrências.

Abstract

In today's World, mobility at work is becoming a constant. Since the first cell phone was created, passing through the emerge of platforms which give the possibility of working at any point around the globe, for instance, the usage of mobile devices like the *Personal Digital Assistant*, the concern of many entities about giving this tools to their collaborators is rising.

Aware of this, **Câmara Municipal do Porto** felt the need to, not only integrate all their social supervision systems, but also making available to their users the tools which will help them to increase their accuracy and efficiency, improving their responsibility and pleasure at work. One of the tools that can be used, and which is this report's subject of study, is mobile devices. This tool will give the possibility to place the Supervision Agents on the field, their true work site, and not behind a desk.

This report presents a solution that satisfies not only the needs of *Câmara Municipal do Porto*, but also the ones of its collaborators. Thought the process of clarifying the details of the existing problem, the goal of this project was revealed. The goal is to create a system that will gather the whole occurrence reporting process, as well as the one of managing the personal responsible for such reporting. Following this line of thought, the system is divided into two sub-systems; one charged with the configuration and management of the system and the other being responsible for providing the mechanisms to place the supervision work on the field.

Through technology analysis and an intense communication with *Câmara Municipal do Porto*, it was possible to create a solution that tries to use all the advantages taken from the usage of mobile devices. At the end of this report, the results obtained by the application of this solution in the creation of a prototype are shown. This prototype validates and proves the viability of the system development. Another important fact to be mentioned is that, because of the abstraction level reached, this solution has the potential of dealing with supervision services as well as with any other need of any entity, in terms of occurrences reporting. This means the system is not blocked by its information flow type.

The information contained in this report represents the first step towards the creation of what can become a reference as an occurrences report system.

Agradecimentos

A realização deste projecto teve a valiosa contribuição de muitas pessoas a quem publicamente afirmo o meu sincero reconhecimento.

O meu orientador, Professor Doutor João Cardoso, foi inextinguível. A disponibilidade, a dedicação, o encorajamento e os conselhos recebidos foram fundamentais para que o objectivo em vista fosse possível. Se o produto final não tem a qualidade desejada não lhe podem ser atribuídas quaisquer responsabilidades. Para o grupo do Projecto de Sistemas de Informação, coordenado com mestria pelo Engenheiro Manuel Machado, o meu apreço pelas opiniões e conselhos. Também ao Dr. Helder Maia, representante da Câmara Municipal do Porto, quero agradecer toda a colaboração e empenho que tornou realizável este projecto.

Agradeço reconhecidamente a vários grupos de pessoas. De um modo geral a todos os professores da Faculdade de Engenharia da Universidade do Porto e com especial carinho para os ilustres Professores Augusto Sousa, José Almeida e Raul Vidal. Um destaque especial para o grupo constituído pelo Fernando da Silva, Francisco Correia e João Silva pelo companheirismo e entre-ajuda recebida ao longo de todo o nosso percurso académico comum, que remonta aos tempos de secundário, o meu obrigado. Quero ainda mencionar o grupo que me acompanhou mais de perto no decorrer deste projecto, grupo constituído pela Vera Francisco, pelo Filipe Silva e pelo João Marques, para eles vai igualmente o meu reconhecimento.

Dum modo geral agradeço a toda a minha família pelo apoio e estímulo recebidos. Aos meus pais, Maria Adelina e Manuel, uma especial gratidão, pois, eles que estiveram na base de tudo, sempre tiveram a capacidade, a disponibilidade e a paciência necessárias. Também ao meu irmão, Rui Manuel, quero agradecer o apoio e companheirismo de irmão mais velho.

À Joana fico grato por ser a garante do equilíbrio indispensável ao longo deste meu percurso e que, de certeza, para sempre a ele ficará ligada.

Hugo Zenha
Julho, 2009.

“Wireless is freedom”

Martin Cooper
Father of Cell Phone

Conteúdo

1	Introdução	1
1.1	Contexto/Enquadramento	1
1.2	Entidades Envolvidas	2
1.2.1	Projecto de Sistemas de Informação (Faculdade de Engenharia da Universidade do Porto)	2
1.2.2	Câmara Municipal do Porto	2
1.3	Objectivos	2
1.4	Estrutura do Relatório	3
2	O Projecto	5
2.1	Sistema de Gestão Integrada de Fiscalização Municipal	5
2.2	Abordagem	6
2.3	Planeamento	7
3	Análise do Problema	9
3.1	Especificação de Requisitos	9
3.1.1	Descrição Geral	9
3.1.2	Casos de Uso	13
3.1.3	Requisitos Específicos	19
3.2	Revisão Tecnológica	29
3.2.1	SuS - Supervision System	29
3.2.2	MSuS - Mobile Supervision System	29
3.3	Sumário	43
4	Solução Apresentada	44
4.1	Arquitectura	44
4.1.1	Visão Geral	44
4.1.2	Visão Conceptual	45
4.1.3	Visão Estática	47
4.1.4	Visão Dinâmica	50
4.1.5	Visão de Dados	53
4.1.6	Testabilidade da Arquitectura	58
4.2	Planeamento do Desenvolvimento	59
5	Desenvolvimento e Validação	61
5.1	Arquitectura	62
5.1.1	Implementação para Dispositivos Móveis	62

CONTEÚDO

5.1.2	Base de Dados	66
5.2	Protótipo	68
5.2.1	Reporte de Ocorrências	69
5.2.2	Gestão do Trabalho	72
5.2.3	Pesquisa de Informação	73
6	Conclusão	77
6.1	Considerações sobre o Projecto	77
6.1.1	Acordo com o Planeamento	78
6.1.2	Trabalho Futuro	78
6.2	Reflexão	78
	Referências	83
A	Teste de Desempenho - Servidores de Aplicações Web	84
A.1	Hardware	84
A.2	Software	84
A.2.1	Servidor	84
A.2.2	Cliente	85
A.3	Implementação	85
A.3.1	Código dos Web Services	85
A.4	Resultados	87
A.5	Resumo ou Conclusão	88
B	Diagrama de Classes	89
C	Relatório do Projecto	90

Lista de Figuras

2.1	Planeamento do Projecto.	8
3.1	Visão Física do Sistema.	10
3.2	Diagrama de casos de uso do <i>SuS</i> - Administrador de Sistema.	14
3.3	Diagrama de casos de uso do <i>SuS</i> - Pivot de Serviços de Fiscalização.	16
3.4	Diagrama de casos de uso do <i>SuS</i> - Agente Fiscalizador.	17
3.5	Diagrama de casos de uso do <i>MSuS</i>	18
3.6	Principais Características dos Web services Java	35
3.7	Principais Características dos Web services .NET	36
3.8	Teste TPS (Primeira Implementação).	39
3.9	Teste AVG (Primeira Implementação).	39
3.10	Teste CNT (Primeira Implementação).	40
3.11	Teste Mbytes (Primeira Implementação).	40
3.12	Teste TPS (Segunda Implementação).	41
3.13	Teste AVG (Segunda Implementação).	41
3.14	Teste CNT (Segunda Implementação).	41
3.15	Teste Mbytes (Segunda Implementação).	42
3.16	Escolhas Tecnológicas.	43
4.1	Visão Física do Sistema.	45
4.2	Camadas de Software.	46
4.3	Visão Geral das Classes do <i>SuS</i>	48
4.4	Visão Geral das Classes do <i>MSuS</i>	48
4.5	Visão das Classes que constituem a definição de Comando.	49
4.6	Visão das Classes que constituem a definição de Pergunta/Resposta a Web services e de Dados.	50
4.7	Exemplo de inserção de dados.	51
4.8	Exemplo de leitura de dados.	51
4.9	Exemplo de interacção entre sistemas.	52
4.10	Visão de Dados - Membros.	54
4.11	Visão de Dados - Ocorrências.	55
4.12	Visão de Dados - Alocação de Trabalho.	56
4.13	Visão de Dados - Localização.	57
4.14	Exemplo de Teste ao <i>IDataStorage</i>	59
4.15	Planeamento do Desenvolvimento.	60
5.1	Organização da Arquitectura no Microsoft Visual Studio.	63
5.2	Aspecto do Oracle Designer.	66

LISTA DE FIGURAS

5.3	Aspecto do Oracle Mobile Database Workbench.	67
5.4	Sincronização dos dispositivos móveis.	67
5.5	Protótipo: ecrã inicial.	68
5.6	Protótipo: Menu principal.	69
5.7	Protótipo: Reporte - Ambiente de escolha do tipo de ocorrência.	69
5.8	Protótipo: Reporte - Escolha do tipo de ocorrência.	70
5.9	Protótipo: Reporte - Introdução de informação geral.	71
5.10	Protótipo: Reporte - Introdução de informação específica.	71
5.11	Protótipo: Reporte - Introdução de informação específica.	72
5.12	Protótipo: Trabalho - Menu.	72
5.13	Protótipo: Trabalho - Informação sobre equipas.	73
5.14	Protótipo: Trabalho - Informação sobre actividades pessoais.	73
5.15	Protótipo: Pesquisa - Menu.	74
5.16	Protótipo: Pesquisa - Tipos de Ocorrência.	74
5.17	Protótipo: Pesquisa - Tipos de Ocorrência.	75
5.18	Protótipo: Pesquisa - Tipos de Ocorrência.	75
6.1	Comparação Planeado/Realizado.	78
B.1	Diagrama de Classes Completo.	89

Lista de Tabelas

3.1	Casos de Uso - Administrador do Sistema sobre o SuS.	15
3.2	Casos de Uso - Pivot de Serviços de Fiscalização sobre o SuS.	17
3.3	Casos de Uso - Agente Fiscalizador sobre o SuS.	18
3.4	Casos de Uso - Agente Fiscalizador sobre o MSuS.	18
3.5	Casos de Uso - Comum sobre o SuS e MSuS.	19
3.6	Requisitos Funcionais - Tratar Área, Característica e Categoria.	21
3.7	Requisitos Funcionais - Tratar Lista, Elemento, Membro, Perfil, Permissão e Tipo de Ocorrência.	22
3.8	Requisitos Funcionais - Tratar Tipo de Tarefa, Topônimo, Número de Polícia, Zona e Actividade.	23
3.9	Requisitos Funcionais - Tratar Tarefa, Alocação, Conhecimento e Equipa.	24
3.10	Requisitos Funcionais - Validar Ocorrência Reportada e Tratar Ocorrência.	25
3.11	Requisitos Funcionais - Tratar Ocorrência.	25
3.12	Requisitos Funcionais - Pesquisa de Informação Interna e Externa.	26
3.13	Requisitos Não-Funcionais	27
3.14	Requisitos de Documentação	28
A.1	Teste de Desempenho: Apache Tomcat 6.0.18 com AXIS2	87
A.2	Teste de Desempenho: IBM WebSphere Application Server CE Version 2.1	87
A.3	Teste de Desempenho: Oracle Weblogic 10gR3	87
A.4	Teste de Desempenho: Sun Application Server 2.1	88

Abreviaturas e Símbolos

API	Application Programming Interface
AVG	Average
ARM	Advanced RISC Machine
CICA	Centro de Informática Professor Correia de Araújo
CMP	Câmara Municipal do Porto
CNT	Connections
CRUD	Create, Read, Update and Delete
DMSI	Direcção Municipal de Sistemas Informáticos
ESB	Enterprise Service Bus
FEUP	Faculdade de Engenharia da Universidade
GIC	Gestão Integrada de Contra-Ordenações
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communications
HSDPA	High-Speed Downlink Packet Access
IDE	Integrated Development Environment
MSuS	Mobile Supervision System
PDA	Personal Digital Assistant
PL/SQL	Procedural Language/Structured Query Language
PSI	Projecto de Sistemas de Informação - FEUP
SIGARRA	Sistema de Informação para a Gestão Agregada de Recursos e Registos Académicos
SDK	Software Development Kit
SGIFM	Sistema de Gestão Integrada de Fiscalização Municipal
SGS	Sistema de Gestão de Sócios
SQL	Structured Query Language
SOAP	Simple Object Access Protocol
SuS	Supervision System
TLCS	Tecnologias da Informação e Comunicação
TPS	Transactions Per Second
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
WS-Security	Web Services Security
W3C	World Wide Web Consortium

Capítulo 1

Introdução

A **Mobilidade** é uma das preocupações em crescendo no mundo actual, não só no que diz respeito a pessoas e bens, mas também em relação ao trabalho.

Cada vez mais os profissionais procuram soluções que permitam realizar o seu trabalho sem que para tal seja necessária a sua deslocação. Os dispositivos móveis aparecem, então, como uma das soluções para este problema. Por exemplo, para vendedores, empresários e até agentes fiscalizadores, a possibilidade de consultar e criar informação através de um dispositivo móvel é uma mais valia, tornando-os mais eficazes e eficientes.

1.1 Contexto/Enquadramento

Neste contexto, a **Câmara Municipal do Porto (CMP)** inicia um esforço para providenciar aos seus agentes de fiscalização municipal mecanismos que lhes permitam a mobilidade do trabalho a realizar.

É neste âmbito que o projecto sobre o qual se debruça este relatório ganha forma. Em colaboração com a Faculdade de Engenharia da Universidade do Porto (*FEUP*), mais propriamente com a unidade **Projecto de Sistemas de Informação (PSI)**, a *CMP* deu início ao processo de criação de tal tecnologia móvel que permitirá que os seus colaboradores se libertem do trabalho de secretária. Sendo assim, o *PSI* encontra-se a desenvolver o **Sistema de Gestão Integrada de Fiscalização Municipal (SGIFM)**.

O *PSI*, em sequência do início deste projecto em parceria com a *CMP*, propôs a realização do projecto final do *Mestrado Integrado em Engenharia Informática e Computação* cujo resultado final é o assunto deste relatório.

1.2 Entidades Envolvidas

1.2.1 Projecto de Sistemas de Informação (Faculdade de Engenharia da Universidade do Porto)

O *PSI* é constituído por colaboradores da *FEUP* que pertencem maioritariamente aos quadros do **Centro de Informática Professor Correia de Araújo (CICA)**, e depende funcionalmente da Direcção da *FEUP*, tendo por objectivo o desenvolvimento de produtos e serviços na área dos Sistemas de Informação.

Os principais projectos em que actualmente se encontra envolvido são o desenvolvimento do *Sistema de Informação para a Gestão Agregada de Recursos e Registos Académicos - SIGARRA* [eGD01], a *Gestão Integrada de Contra-Ordenações (GIC)* [dSdI06] para a *CMP* e o *Sistema de Gestão de Sócios (SGS)* [Str09] para o *Sindicato de Trabalhadores da Função Pública do Norte*.

1.2.2 Câmara Municipal do Porto

A Câmara Municipal do Porto tem vindo a desenvolver um conjunto de programas tendentes a melhorar a sua máquina administrativa e a oferecer a todos os colaboradores instrumentos necessários para melhorar a sua eficiência profissional. Todos os departamentos municipais se envolveram nesta "batalha", nem sempre fácil, destacando-se neste trabalho, entre outras unidades, a **Direcção Municipal de Sistemas Informáticos (DMSI)** em conjunto com a qual este projecto está a ser realizado. Fruto deste esforço colectivo interno, a autarquia tem tido um reconhecimento a nível nacional e, internacionalmente, é já vista como um dos exemplos em Portugal de boas práticas no sector público [dCeIdCMdP08]. Espera-se que com este projecto, a autarquia cimente ainda mais a sua posição exemplar dos últimos anos no âmbito da utilização de *Tecnologias da Informação e Comunicação (TLCS)*.

1.3 Objectivos

Inicialmente o projecto aqui apresentado tinha como objectivo, e título, "*Aplicação de registo de autos para dispositivos móveis*", mas devido ao aumento dos requisitos e da complexidade, decidiu-se alterar para "*Aplicação de registo de ocorrências para dispositivos móveis*", em consonância com os novos objectivos.

Era pedido que fosse seguido o caderno de encargos já criado e se procedesse a uma análise tecnológica e à criação de uma solução, implementando o maior número de funcionalidades descritas em tal caderno.

Com a evolução, o pedido foi reestruturado. O projecto evoluiu, assim como o trabalho pedido no âmbito deste projecto final e foi então necessário desenvolver o projecto

partindo de uma etapa mais embrionária. Assim sendo, o pretendido para o projecto em questão foi o seguinte:

- Especificação de Requisitos;
- Análise Tecnológica;
- Apresentação de uma Solução;
- Desenvolvimento de um protótipo para dispositivos móveis, mais concretamente para *Personal Digital Assistants (PDAs)*.

Para além dos pontos até aqui referidos, é do interesse do *PSI* que a solução criada apresente o nível de abstracção necessária para que esta não seja apenas aplicada no contexto de municípios, mas também em outras áreas em que as entidades a elas associadas necessitem de sistemas de reporte de ocorrências.

1.4 Estrutura do Relatório

Após a introdução ao projecto e às entidades intervenientes, incluídas nesta primeira secção, o presente relatório encontra-se estruturado nas secções seguintes:

- **2. O Projecto**

Uma abordagem geral ao projecto é apresentada neste capítulo. O problema é definido, assim como a abordagem e planeamento para a sua concretização.

- **3. Análise do Problema**

Neste capítulo são apresentados elementos mais técnicos. Referenciam-se os requisitos que a solução apresentada deverá satisfazer, assim como a análise tecnológica realizada com o objectivo de apurar quais as melhores escolhas para a criação de tal solução.

- **4. Solução Apresentada**

A solução idealizada é apresentada neste capítulo. Aqui se referencia a arquitectura da solução a seguir e o planeamento para o desenvolvimento do projecto.

- **5. Desenvolvimento e Validação**

Neste capítulo explica-se como foi concretizada a implementação da arquitectura, a nível do dispositivo móvel, e o desenvolvimento do protótipo para a validação da solução apresentada. Os problemas encontrados constituem, também, assunto abordado neste capítulo.

- **6. Conclusão**

Por fim, são tecidas considerações sobre o decorrer do projecto. É também resumido o trabalho que poderá ser realizado no futuro, tendo por base este projecto.

No final são apresentados, como anexos, o teste de desempenho de servidores de aplicações web, o diagrama de classes completo e os relatórios elaborados ao longo deste projecto.

Este documento foi elaborado tendo por base os relatórios realizados no âmbito do projecto (*ver referências [dP09b], [dP09c] e [dP09d]*). Para informação mais detalhada é recomendada a leitura dos mesmos.

Capítulo 2

O Projecto

Na presente secção é introduzida, mais detalhadamente, a problemática que levou ao início do projecto da *CMP*, em parceria com o *PSI*. É também apresentada a abordagem seguida para a criação da solução, assim como o planeamento realizado para a sua concretização.

2.1 Sistema de Gestão Integrada de Fiscalização Municipal

O *Sistema de Gestão Integrada de Fiscalização Municipal*, conhecido simplesmente por *SGIFM*, é a solução a implementar para resolver o problema apresentado pela *CMP*.

Análise do Problema

Actualmente, a organização dos recursos e talentos é fundamental em qualquer sector de actividade. A actividade da fiscalização não é excepção. O enfoque na produtividade, na satisfação dos Clientes/Cidadãos e no valor produzido com o serviço de fiscalização, torna essencial apostar na gestão eficaz de todo o seu leque de actividades. As competências principais dos vários serviços fiscalizadores de uma autarquia passam principalmente por fazer cumprir as posturas e regulamentos municipais, por cooperar com outros serviços municipais e entidades externas, e por assegurar o licenciamento e fiscalização de actividades económicas. É aqui que os Sistemas de Informação podem ser um aliado precioso desta actividade tão nobre e essencial ao bom funcionamento das organizações de administração pública local.

Desta forma, desenvolveu-se uma solução integrada de fiscalização que vai permitir melhorar a organização dos processos de fiscalização e otimizar a execução das actividades inerentes às suas competências. Um dos problemas vigentes em grande parte das autarquias, está relacionado com o facto dos agentes fiscalizadores passarem muito tempo

em gabinete, a reportar as ocorrências por eles verificadas no trabalho de exterior, quando é desejável que seja mais longa a sua permanência no terreno.

Por outro lado, existem vários serviços de fiscalização com competências de fiscalização sobrepostas, fazendo com que a ameaça da duplicação de procedimentos esteja sempre presente.

Finalmente, devido ao aumento dos tecidos empresariais e núcleos de actividades económicas nos territórios concelhios, torna-se imperativo que as rotinas de fiscalização sejam alvo de uma programação prévia, de forma a que o resultado da sua actividade seja mais eficaz.

Em resposta a todos estes constrangimentos e condicionantes, é necessária a construção de uma solução baseada em duas grandes componentes aplicacionais. A primeira, é o Sistema de Supervisão Central, que irá ser ao longo deste relatório referido como *Supervision System (SuS)*, que permite assegurar toda a parametrização, controlo e integração com outros sistemas de informação. E a segunda, é o Sistema de Supervisão Mobile, que será referido como *Mobile Supervision System (MSuS)*, que pode ser instalado em todos os equipamentos móveis (neste caso *PDAs*) de fiscalização, e vai permitir a rentabilização e optimização de todo o trabalho de exterior realizado pelas equipas de fiscalização. Esta solução assume-se como uma ferramenta estruturante da actividade de fiscalização, permitindo a normalização do registo de ocorrências, e contribuindo para a optimização dos seus processos. Esta solução vai sustentar a actividade de centenas de utilizadores (cerca de 400 utilizadores) entre agentes fiscalizadores, gestores de processos de fiscalização, chefias e responsáveis de serviços de fiscalização e pivots de ligação com outros serviços da autarquia. A implementação da solução vai ser progressiva, uma vez que numa fase inicial vai albergar um conjunto de serviços piloto, e depois, numa fase posterior, vai-se estender ao resto da estrutura da autarquia.

2.2 Abordagem

Para que o projecto seja concluído com sucesso é necessária a definição da abordagem ao problema para que a solução seja obtida eficaz e eficientemente.

Devido aos requisitos de natureza variada (ver *Secção 3.1 Especificação de Requisitos*) foram tomados vários passos para a elaboração de uma solução.

A abordagem definida para o projecto em causa inclui os seguintes passos:

- Análise Tecnológica;
- Especificação de Requisitos;
- Definição da Arquitectura;

- Implementação da Solução;
- Testes.

Tendo em conta a abordagem seguida para a realização do projecto e os recursos disponíveis, é apresentado de seguida o planeamento do desenvolvimento da solução proposto.

2.3 Planeamento

Para que fosse possível a conclusão do projecto com êxito foi necessário planear o seu desenrolar, tendo sempre atenção aos prazos estipulados.

O planeamento do projecto, no âmbito do projecto, é o seguinte:

- **Dias 26 e 27 de Fevereiro e Semanas de 02 e de 09 de Março**

Análise Tecnológica - Levantamento de informação sobre tecnologias que sejam uma mais valia para o projecto. Comparação e escolha das tecnologias a utilizar. Elaboração do Relatório de Análise Tecnológica;

- **Semana de 16 de Março**

Definição da Arquitectura - Desenho da arquitectura a ser utilizada no projecto. Elaboração do Relatório de Definição de Arquitectura;

- **Semanas de 23 e 30 de Março**

Especificação de Requisitos - Levantamento dos requisitos para a solução a criar para o cliente. Elaboração do Relatório de Especificação de Requisitos;

- **Semana de 06 de Abril**

Mecanismos de Sincronização - Estudo sobre os mecanismos de sincronização escolhidos na fase de análise tecnológica;

- **Semana de 13 de Abril**

Mecanismos de Segurança - Estudo sobre os mecanismos de segurança a utilizar;

- **Semanas de 20 e 27 de Abril, mês de Maio e Semanas de 01 e 08 de Junho**

Desenvolvimento do Protótipo - Implementação da Arquitectura para os dispositivos móveis e desenvolvimento de um protótipo com vista a validar a solução apresentada;

- **Semanas de 15 e 22 de Junho**

Revisão da Tese - Leitura atenta da Tese realizada e correcção desta.

Este planeamento foi realizado em conjunto com o proponente e aprovado pelo cliente para que todas as partes estivessem cientes de como foi realizado o trabalho.

A informação apresentada até ao momento está sintetizada na *Figura 2.1*.

O Projecto

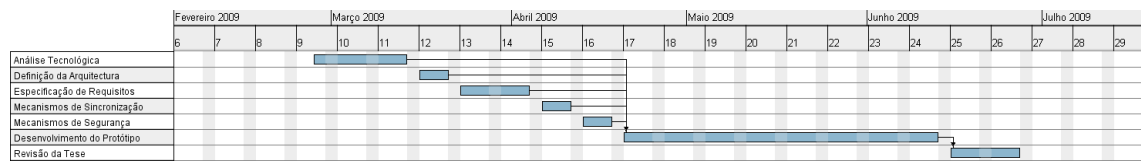


Figura 2.1: Planeamento do Projecto.

Seguidamente serão apresentados os pormenores do trabalho realizado no âmbito do projecto.

Capítulo 3

Análise do Problema

A presente secção apresenta os requisitos do *SGIFM* e a revisão tecnológica realizada com o objectivo da escolha das tecnologias a utilizar no desenvolvimento de uma solução que satisfaça tais requisitos.

3.1 Especificação de Requisitos

A especificação de requisitos aparece com o objectivo de explicitar com o maior detalhe possível todas as funcionalidades, casos de uso e requisitos do sistema a desenvolver no âmbito do projecto de parceria da *CMP* com o *PSI*. Tal é necessário para que quaisquer dúvidas que possam subsistir sobre o pretendido sejam convenientemente esclarecidas e dissipadas.

3.1.1 Descrição Geral

3.1.1.1 Contexto

A solução pretendida insere-se no contexto da gestão de ocorrências e tarefas inerentes à actividade de fiscalização em autarquias (contra-ordenações, ocorrências na via pública, gestão urbanística, licenciamentos, *etc*). Advém da necessidade da *Câmara Municipal do Porto* em fornecer aos seus *Agentes Fiscalizadores* uma ferramenta simples, eficaz, eficiente e fiável, que lhes permita melhorar a qualidade das acções de fiscalização levadas a cabo no município.

Arquitectura Global

A arquitectura global baseia-se numa estrutura cliente-servidor em que os mecanismos de comunicação são disponibilizados numa camada intermédia, através da utilização

de *Web services*. A comunicação realizada entre os dois sistemas a desenvolver e outros que poderão, posteriormente, interagir também com estes, será realizada através da disponibilização e procura de *Web services*.

O *SGIFM* é constituído por dois sub-sistemas, o *SuS* e o *MSuS*, que comunicarão através de mecanismos disponibilizados por um servidor de aplicações. Este servidor permitirá não só comunicações internas ao sistema, mas também a comunicação deste com aplicações externas. De notar que mesmo o caso especial da *Gestão Integrada de Contra-Ordenações* (mais sucintamente *GIC*) [dSdI06], sistema já em utilização na *CMP*, é vista como mais uma aplicação externa, mesmo que no futuro se espere que a informação contida no *SGIFM* seja utilizada por esta. A *Figura 3.1* apresenta, de uma forma elucidativa, a visão física do sistema.

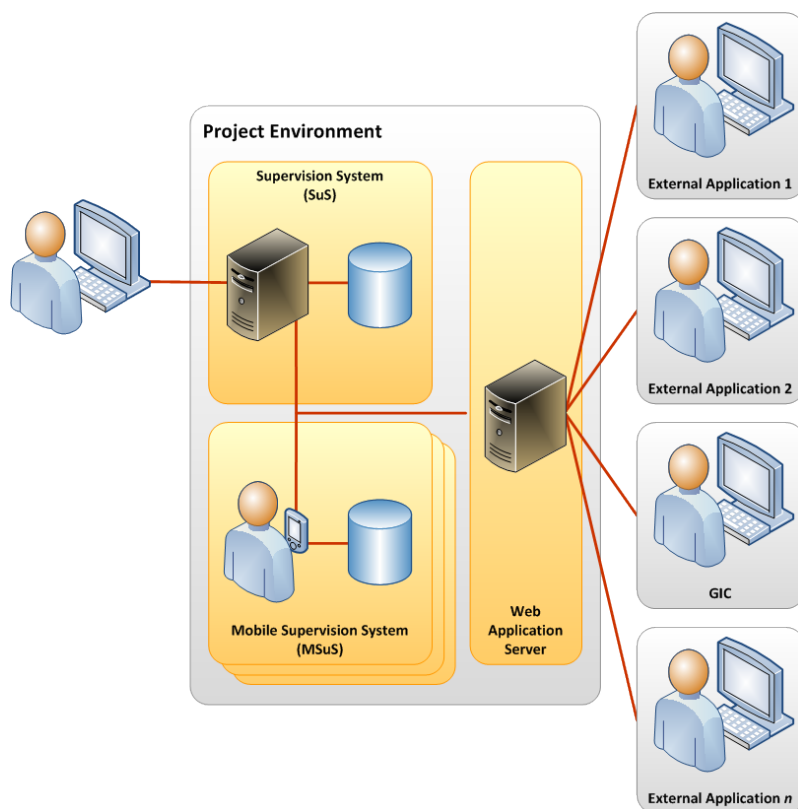


Figura 3.1: Visão Física do Sistema.

Ambiente de Operação

Do lado do Sistema de Fiscalização (designado por *Supervision System - SuS*), o ambiente em servidor Windows será o ambiente de eleição. Este disponibiliza todas as especificidades necessárias para a utilização do Sistema de Fiscalização.

O Sistema de Fiscalização Móvel (designado por *Mobile Supervision System - MSuS*) enquadrar-se-á num contexto de utilização de âmbito generalizado, em plataformas móveis

(vulgo *PDA* ou *SmartPhones*) com o sistema operativo Windows Mobile 6 ou superior, podendo ser adaptado, posteriormente, às especificidade de cada um.

De notar que o sistema deve possibilitar a ligação simultânea de múltiplos clientes a um servidor. Deve ser também possível a gestão e configuração simultânea da aplicação a nível do servidor, por múltiplos utilizadores.

3.1.1.2 Funções do Sistema

O sistema em questão, o *SGIFM*, divide-se em dois sub-sistemas, o sistema de *Back-Office*, o *SuS*, e o sistema para executar no dispositivo móvel, o *MSuS*. Assim sendo, as funções dependem do sub-sistema em questão.

SuS - Supervision System

O *SuS*, *Back-Office* do sistema de fiscalização, deverá disponibilizar as seguintes funções:

- **Consultas** - deverão ser disponibilizadas consultas às ocorrências reportadas e mecanismos de consulta a aplicações externas;
- **Controlo/Validação das Ocorrências** - deverão ser disponibilizadas funcionalidades com vista ao controlo e validação das ocorrências reportadas pelos agentes fiscalizadores;
- **Parametrizações do Sistema** - deverá permitir ao Administrador do Sistema configurar este sistema de forma a satisfazer as necessidades do ambiente de funcionamento;
- **Planeamento de Tarefas** - o sistema deverá disponibilizar mecanismos para planear o trabalho dos agentes fiscalizadores, sendo que estes apenas têm de consultar as suas tarefas no dispositivo móvel.

MSuS - Mobile Supervision System

O sistema a executar nos dispositivos móveis dos agentes fiscalizadores, o *MSuS*, deverá disponibilizar as seguintes funcionalidades:

- **Consultas** - deverão ser disponibilizadas consultas às ocorrências reportadas e mecanismos de consulta a aplicações externas;
- **Registo de Ocorrências** - sendo este o objectivo principal de todo o produto, o *MSuS* deverá disponibilizar o mecanismo de registo de ocorrências por parte do agente fiscalizador.

3.1.1.3 Características dos Utilizadores

A aplicação a desenvolver deverá envolver três tipos distintos de utilizadores:

- **Agentes Fiscalizadores**, que serão os principais actores deste sistema;
- **Pivots dos Serviços Fiscalizadores**, que terão a competência de gerir as ocorrências do Serviço Fiscalizador a que pertencem, assim como gerir o trabalho dos agentes;
- **Administradores do Sistema**, responsáveis pela parametrização e gestão das funcionalidades a que os Agentes Fiscalizadores e *Pivots* têm acesso a partir dos dispositivos móveis (caso *MSuS*), ou da interface web (caso *SuS*).

3.1.1.4 Restrições

As únicas restrições a apontar são os factos do *MSuS* ser implementado para correr em dispositivos móveis com sistema operativo *Windows* e do *SuS* ser implementado em tecnologia *Oracle*, especificamente *PL/SQL* [McL08] sobre *Oracle Database* [Lon09].

No âmbito do projecto, a primeira restrição não é vista como problema, pois os dispositivos existentes/a comprar terão o *Windows* como sistema operativo, no entanto, se no futuro se mudar de plataforma terá de haver um esforço adicional na criação da nova versão do *MSuS*.

3.1.1.5 Pressupostos e Dependências

Pressupõe-se que:

- o dispositivo móvel está ligado para assegurar o funcionamento do *MSuS*;
- o sistema operativo do dispositivo móvel sobre o qual o *MSuS* está a executar possui a capacidade para garantir o funcionamento deste último sem falhas;
- os *Agentes Fiscalizadores*, *Pivots dos Serviços Fiscalizadores* e *Administradores do Sistema* possuem os conhecimentos necessários para utilizar um browser Web;
- os *Agentes Fiscalizadores* possuem os conhecimentos necessários para utilizar uma aplicação num dispositivo móvel;
- o *SuS* será acedido por um dos seguintes *browsers web*:
 - *Internet Explorer 8* [Cor09d];
 - *Mozilla Firefox 3* [eMF09];
 - qualquer *browser web* compatível com os standards, tal como definidos pela entidade *W3C* [W3C09].

- o *SuS* possui capacidade de armazenamento de dados considerada ilimitada;
- a ligação entre *SuS* e *MSuS* estará assegurada.

3.1.2 Casos de Uso

Nesta secção são especificadas as funcionalidades que os dois sistemas, *SuS* e *MSuS*, devem disponibilizar aos seus utilizadores. Estas funcionalidades encontram-se representadas recorrendo a diagramas de caso de uso para uma maior facilidade na leitura do documento. De notar que ambos os sub-sistemas aqui referidos contêm funcionalidades específicas, embora exista, no entanto, um ramo comum aos dois.

As tabelas que aparecem durante esta secção descrevem brevemente cada caso de uso introduzido, tendo em consideração qual o actor que lhe tem acesso e a prioridade de desenvolvimento. A prioridade pode ser *Essencial*, *Elevada*, *Média* ou *Baixa*. O sistema de reporte de ocorrências funciona se os de prioridade *Essencial* estiverem realizados. Contudo, para estar pronto para ser utilizado no terreno é necessário que os de prioridade *Elevada* também o estejam.

3.1.2.1 SuS - Supervision System

No sistema *SuS* existem três actores: o *Administrador de Sistema*, o *Pivot de Serviços de Fiscalização* e o *Agente Fiscalizador*. Estes actores irão utilizar as funcionalidades disponibilizadas pelo *SuS*. O *Administrador de Sistema* é o responsável pela parametrização dos valores do sistema e dos serviços disponibilizados por este. Cabe ao *Pivot de Serviços de Fiscalização* gerir e validar o trabalho do *Agente Fiscalizador*, sendo que este último pode pesquisar informação contida no sistema e reportar ocorrências.

Administrador de Sistema

Como já referido anteriormente, o *Administrador de Sistema* é responsável pela parametrização do sistema. Para tal, o leque de casos de uso a que tem acesso é bastante extenso, desde o tratamento de informação que flui no sistema à edição de permissões dos seus utilizadores. A *Figura 3.2* apresenta os casos de uso que representam as funcionalidades do *SuS* a que este tem acesso. De seguida é explicados cada um destes casos de uso.

Análise do Problema

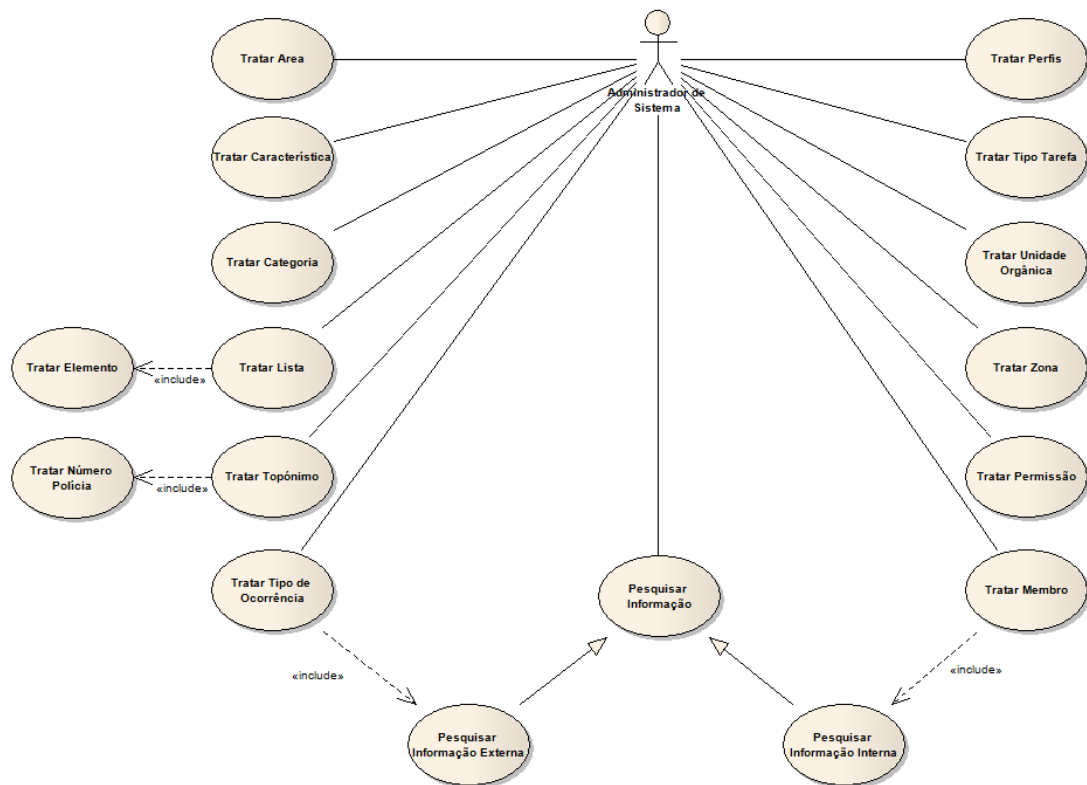


Figura 3.2: Diagrama de casos de uso do SuS - Administrador de Sistema.

A Tabela 3.1 apresenta cada um dos casos de uso referenciados na Figura 3.2. De referir que, no SuS, os casos de uso destinam-se, mais concretamente, à gestão do sistema e do trabalho a ser realizado pelos *Agentes Fiscalizadores*. Os casos de uso aqui apresentados inserem-se no primeiro grupo, a que os *Administradores de Sistema* têm acesso.

Destacam-se neste grupo de casos de uso, os que se destinam à criação dos Tipos de Ocorrência a serem reportados pelos *Agentes Fiscalizadores* (essencial para este projecto cumprir o seu objectivo principal - Reporte de Ocorrências) e à criação de Membros. Os restantes casos de uso servem, por um lado, para a gestão de permissões, caso da criação de perfis, e por outro lado, para facilitarem o serviço de reporte, como é o caso da criação de topónimos.

Os casos de uso referentes à pesquisa de informação fazem parte dos casos de uso comuns a todos os utilizadores dos sub-sistemas. Assim, estes estão explicados mais à frente na Secção 3.1.2.3 Comum.

Análise do Problema

Tabela 3.1: Casos de Uso - Administrador do Sistema sobre o SuS.

Identificador	Nome	Descrição	Prioridade
uc_sus01	<i>Tratar Área</i>	Permite adicionar, editar e remover Áreas (conjunto de Tipo de Ocorrência).	Elevada
uc_sus02	<i>Tratar Característica</i>	Providencia as acções de adição, edição e remoção de Características (conjunto de dados específicos a um ou vários Tipos de Ocorrência).	Elevada
uc_sus03	<i>Tratar Categoria</i>	Permite adicionar, editar e remover Categorias profissionais a que os Membros podem pertencer.	Elevada
uc_sus04	<i>Tratar Lista</i>	Permite adicionar, editar e remover Listas para selecção de valores para as Características.	Elevada
uc_sus05	<i>Tratar Elemento</i>	As acções de adição, edição e remoção de elementos que constituem as Listas.	Elevada
uc_sus06	<i>Tratar Membro</i>	Permite adicionar, editar e remover registo de Membros, ou seja, de possíveis utilizadores do sistema a construir.	Elevada
uc_sus07	<i>Tratar Perfil</i>	Este caso de uso refere-se à adição, edição e remoção de Perfis a que um Membro pode estar associado.	Elevada
uc_sus08	<i>Tratar Permissão</i>	Permite adicionar, editar e remover os tipos de Permissão que podem estar associados a um Perfil.	Elevada
uc_sus09	<i>Tratar Tipo de Ocorrência</i>	Permite adicionar, editar e remover Tipos de Ocorrência, ou seja, quais ocorrências podem ser reportadas através do sistema.	Elevada
uc_sus10	<i>Tratar Tipo de Tarefa</i>	Permite a adição, edição e remoção de Tipo de Tarefas que podem ser criadas.	Elevada
uc_sus11	<i>Tratar Topónimo</i>	Permite adicionar, editar e remover Topónimos, ou seja, nomes de lugares, sítios ou localidades.	Elevada
uc_sus12	<i>Tratar Número de Polícia</i>	Providencia acções de adição, edição e remoção de Números de Polícia que estarão associados a um Topónimo. É uma forma de criar uma maior eficiência nos serviços.	Elevada
uc_sus13	<i>Tratar Unidade Orgânica</i>	Permite adicionar, editar e remover Unidades Orgânicas do Município a que um Membro pode pertencer.	Elevada
uc_sus14	<i>Tratar Zona</i>	Permite a adição, edição e remoção de Zonas (conjunto de Topónimos do Município).	Elevada

Pivot de Serviços de Fiscalização

Os *Pivots de Serviços de Fiscalização* são responsáveis pela gestão das ocorrências reportadas pelos *Agentes Fiscalizadores*. Compete aos *Pivots* zelar pelo bom trabalho desses *Agentes*.

A *Figura 3.3* apresenta as funcionalidades, através de casos de uso, a que o *Pivot* terá acesso no sistema de *Back-Office*.

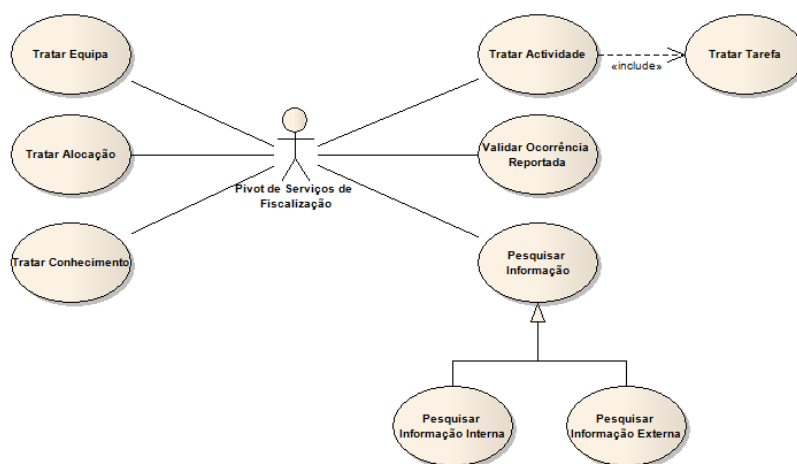


Figura 3.3: Diagrama de casos de uso do SuS - Pivot de Serviços de Fiscalização.

A *Tabela 3.2* apresenta cada um dos casos de uso referenciados na *Figura 3.3*. Como referido anteriormente, no SuS, os casos de uso destinam-se, mais concretamente, à gestão do sistema e do trabalho a ser realizado pelos *Agentes Fiscalizadores*. Os casos de uso apresentados na *Tabela 3.2* e na *Figura 3.3* inserem-se no segundo grupo, a que os *Pivots de Serviços de Fiscalização* têm acesso. Destaca-se neste grupo de casos de uso, o que se destina à validação das Ocorrências Reportadas pelos *Agentes Fiscalizadores*. Os restantes casos de uso destinam-se à gestão do trabalho destes agentes.

Os casos de uso referentes à pesquisa de informação fazem parte dos casos de uso comuns a todos os utilizadores dos sub-sistemas. Assim, estes estão explicados mais à frente na *Secção 3.1.2.3 Comum*.

Tabela 3.2: Casos de Uso - Pivot de Serviços de Fiscalização sobre o SuS.

Identificador	Nome	Descrição	Prioridade
uc_sus15	<i>Tratar Atividade</i>	Permite adicionar, editar e remover Atividades (conjunto de Tarefas que um <i>Agente Fiscalizador</i> tem de realizar).	Média
uc_sus16	<i>Tratar Tarefa</i>	Permite a adição, edição e remoção de Tarefas que um <i>Agente Fiscalizador</i> terá de realizar.	Média
uc_sus17	<i>Tratar Alocação</i>	Permite a adição, edição e remoção de Alocações de Equipas a Zonas do Município.	Média
uc_sus18	<i>Tratar Conhecimento</i>	Permite adicionar, editar e remover linhas de orientação de como proceder ao reporte de uma Ocorrência de um certo Tipo.	Média
uc_sus19	<i>Tratar Equipa</i>	Permite adicionar, editar e remover Equipas constituídas por Membros.	Média
uc_sus20	<i>Validar Ocorrência Reportada</i>	Possibilita a validação, devolução para edição, eliminação e arquivamento de ocorrências reportadas.	Média

Agente Fiscalizador

O trabalho do *Agente Fiscalizador* é reportar ocorrências que testemunha. Assim sendo, os casos de uso a que tem acesso restringem-se ao reporte e à pesquisa de informação que o auxiliará no seu trabalho. A *Figura 3.4* apresenta os casos de uso em relação ao *Agente Fiscalizador*.

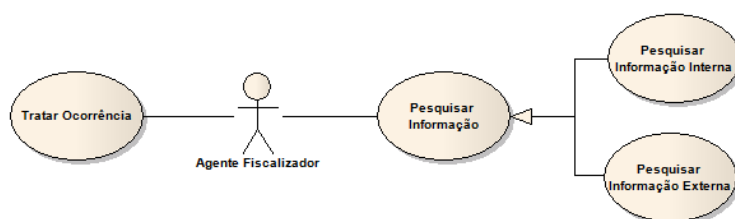


Figura 3.4: Diagrama de casos de uso do SuS - Agente Fiscalizador.

A *Tabela 3.3* apresenta cada um dos casos de uso referenciados na *Figura 3.4*. O SuS permite aos *Agentes Fiscalizadores* reportarem Ocorrências sem recorrerem a dispositivos móveis.

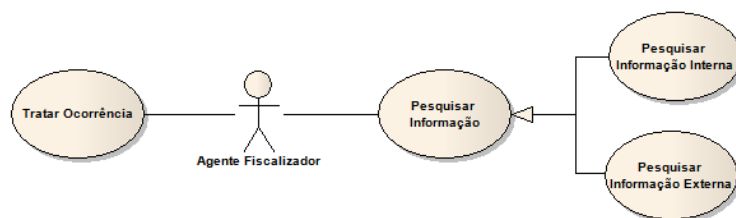
Os casos de uso referentes à pesquisa de informação fazem parte dos casos de uso comuns a todos os utilizadores dos sub-sistemas. Assim, estes estão explicados mais à frente na *Secção 3.1.2.3 Comum*.

Tabela 3.3: Casos de Uso - Agente Fiscalizador sobre o SuS.

Identificador	Nome	Descrição	Prioridade
uc_sus21	<i>Tratar Ocorrência</i>	Permite adicionar, editar e remover ocorrências, assim como imprimir informações referentes a uma ocorrência.	Baixa

3.1.2.2 MSuS - Mobile Supervision System

A *Figura 3.5* apresenta os casos de uso referentes ao *MSuS*. Neste sistema existe apenas um actor, o *Agente Fiscalizador*. O *Agente Fiscalizador* utiliza este sistema para tarefas de pesquisa de informação importante para a realização do seu trabalho, reportar ocorrências.

Figura 3.5: Diagrama de casos de uso do *MSuS*.

A *Tabela 3.4* apresenta cada um dos casos de uso referenciados na *Figura 3.5*. O *MSuS* permite aos *Agentes Fiscalizadores* reportarem Ocorrências, com recurso ao dispositivo móvel.

Os casos de uso referentes à pesquisa de informação fazem parte dos casos de uso comuns a todos os utilizadores dos sub-sistemas. Assim, estes estão explicados mais à frente na *Secção 3.1.2.3 Comum*.

Tabela 3.4: Casos de Uso - Agente Fiscalizador sobre o MSuS.

Identificador	Nome	Descrição	Prioridade
uc_msus01	<i>Tratar Ocorrência</i>	Permite adicionar, editar, remover ocorrências, assim como imprimir informações referentes a uma ocorrência e associar a esta uma posição geo-referenciada.	Essencial

3.1.2.3 Comum

O *SuS* e o *MSuS* têm em comum os casos de uso referentes à pesquisa de informação. Estes casos de uso apenas se dividem em duas possíveis situações. A primeira situação diz respeito a pesquisa de informação criada única e exclusivamente pela utilização do sistema enquanto a segunda tem a ver com a informação proveniente de actividades externas à utilização do sistema, como é o caso da criação de legislações que dão origem a tipos de ocorrência utilizados neste sistema. A *Tabela 3.5* apresenta cada um destes casos de uso.

Tabela 3.5: Casos de Uso - Comum sobre o *SuS* e *MSuS*.

Identificador	Nome	Descrição	Prioridade
uc_com01	<i>Pesquisar Informação Interna</i>	Permite pesquisar informação criada pela utilização do sistema.	Elevada
uc_com02	<i>Pesquisar Informação Externa</i>	Permite pesquisar informação criada externamente à utilização do sistema.	Elevada

3.1.3 Requisitos Específicos

Nesta secção são detalhados todos os requisitos do sistema. Estes dividem-se em requisitos externos, requisitos funcionais, requisitos não-funcionais e requisitos de documentação, demonstrando a política a seguir no desenvolvimento do sistema.

3.1.3.1 Requisitos Externos

Interface com o Utilizador

O desenvolvimento deste sistema prevê interface com o utilizador em duas situações, uma para o sistema *SuS*, outra para o *MSuS*. É esperado que ambas as interfaces prezem a simplicidade e usabilidade, com especial atenção no caso do *MSuS*, pois em dispositivos móveis tal requisito é mais sensível. No caso do *SuS* é esperado uma interface Web.

Interfaces de Hardware

Em relação ao dispositivo móvel, será necessário que esteja equipado com receptor *GPS*, assim como mecanismos de ligação à rede (quer com fios, quer sem fios). De notar que o dispositivo móvel, para poder imprimir informação relativa às ocorrências reportadas, terá de possibilitar a comunicação com uma impressora portátil.

É necessário que o servidor onde o *SuS* estará a correr tenha ligação à Internet.

Interfaces de Software

É necessário que o dispositivo móvel tenha o sistema operativo *Windows Mobile 6* ou superior, assim como o *Oracle Database Lite* (ver [dP09a]).

A nível do servidor será necessária a interligação com um sistema de base de dados, neste caso *Oracle Database*. Todo o mecanismo de sincronização com os dispositivos móveis será feito utilizando o *Oracle Mobile Server* (parte integrante do *Oracle Database Lite*) e *Web services* (ver [dP09a]).

Interfaces de Comunicação

O processo de comunicação dos dispositivos móveis com o servidor e com aplicações externas irá efectuar-se através de um canal de comunicação. O meio de comunicação poderá passar por uma ligação por rede móvel (*GSM*, *GPRS*, *UMTS*, *HSDPA* ou qualquer outra), por rede sem fios *Wi-Fi* ou com um cabo para ligar dispositivos via *USB*.

No caso de o servidor e a base de dados serem executados em máquinas diferentes é necessária a ligação por rede (com ou sem fios); em alternativa será utilizada uma ligação entre módulos no caso de serem executados na mesma máquina.

3.1.3.2 Requisitos Funcionais

As tabelas ao longo desta secção apresentam os requisitos funcionais ligados a cada caso de uso apresentado anteriormente na *Secção 3.1.2 Casos de Uso*. Cada tabela apresenta os requisitos de cada caso de uso, seus identificadores e uma breve descrição do pretendido.

SuS - Supervision System

Aqui são apresentados os vários requisitos funcionais referentes a cada um dos casos de uso apresentados na *Secção 3.1.2.1 SuS - Supervision System*.

Tratar Área, Característica e Categoria

Na *Tabela 3.6* são apresentados os requisitos funcionais referentes aos casos de uso *Tratar Área*, *Tratar Característica* e *Tratar Categoria*.

Tabela 3.6: Requisitos Funcionais - Tratar Área, Característica e Categoria.

Caso de uso	Identificador	Descrição
<i>Tratar Área</i>	<i>uc_sus01_r01</i>	Criação do mecanismo de adição de Áreas.
	<i>uc_sus01_r02</i>	Criação do mecanismo de edição de Áreas.
	<i>uc_sus01_r03</i>	Criação do mecanismo de remoção de Áreas.
<i>Tratar Característica</i>	<i>uc_sus02_r01</i>	Criação do mecanismo de adição de Características.
	<i>uc_sus02_r02</i>	Criação do mecanismo de edição de Características.
	<i>uc_sus02_r03</i>	Criação do mecanismo de remoção de Características.
<i>Tratar Categoria</i>	<i>uc_sus03_r01</i>	Criação do mecanismo de adição de Categorias profissionais.
	<i>uc_sus03_r02</i>	Criação do mecanismo de edição de Categorias profissionais.
	<i>uc_sus03_r03</i>	Criação do mecanismo de remoção de Categorias profissionais.

Tratar Lista, Elemento, Membro, Perfil, Permissão e Tipo de Ocorrência

Na *Tabela 3.7* são apresentados os requisitos funcionais referentes aos casos de uso *Tratar Lista*, *Tratar Elemento*, *Tratar Membro*, *Tratar Perfil*, *Tratar Permissão* e *Tratar Tipo de Ocorrência*.

Tabela 3.7: Requisitos Funcionais - Tratar Lista, Elemento, Membro, Perfil, Permissão e Tipo de Ocorrência.

Caso de uso	Identificador	Descrição
<i>Tratar Lista</i>	<i>uc_sus04_r01</i>	Criação do mecanismo de adição de Listas de selecção.
	<i>uc_sus04_r02</i>	Criação do mecanismo de edição de Listas de selecção.
	<i>uc_sus04_r03</i>	Criação do mecanismo de remoção de Listas de selecção.
<i>Tratar Elemento</i>	<i>uc_sus05_r01</i>	Criação do mecanismo de adição de Elementos de selecção.
	<i>uc_sus05_r02</i>	Criação do mecanismo de edição de Elementos de selecção.
	<i>uc_sus05_r03</i>	Criação do mecanismo de remoção de Elementos de selecção.
<i>Tratar Membro</i>	<i>uc_sus06_r01</i>	Criação do mecanismo de adição de Membros.
	<i>uc_sus06_r02</i>	Criação do mecanismo de edição de Membros.
	<i>uc_sus06_r03</i>	Criação do mecanismo de remoção de Membros.
<i>Tratar Perfil</i>	<i>uc_sus07_r01</i>	Criação do mecanismo de adição de Perfis de utilização.
	<i>uc_sus07_r02</i>	Criação do mecanismo de edição de Perfis de utilização.
	<i>uc_sus07_r03</i>	Criação do mecanismo de remoção de Perfis de utilização.
<i>Tratar Permissão</i>	<i>uc_sus08_r01</i>	Criação do mecanismo de adição de Permissões de utilização.
	<i>uc_sus08_r02</i>	Criação do mecanismo de edição de Permissões de utilização.
	<i>uc_sus08_r03</i>	Criação do mecanismo de remoção de Permissões de utilização.
<i>Tratar Tipo de Ocorrência</i>	<i>uc_sus09_r01</i>	Criação do mecanismo de adição de Tipos de Ocorrência a ser reportada.
	<i>uc_sus09_r02</i>	Criação do mecanismo de simulação do reporte de Tipos de Ocorrência no dispositivo móvel.
	<i>uc_sus09_r03</i>	Criação do mecanismo de edição de Tipos de Ocorrência a ser reportada.
	<i>uc_sus09_r04</i>	Criação do mecanismo de remoção de Tipos de Ocorrência a ser reportada.

Tratar Tipo de Tarefa, Topônimo, Número de Polícia, Unidade Orgânica, Zona e Actividade

Na *Tabela 3.8* são apresentados os requisitos funcionais referentes aos casos de uso *Tratar Tipo de Tarefa*, *Tratar Topônimo*, *Tratar Número de Polícia*, *Tratar Unidade Orgânica*, *Tratar Zona* e *Tratar Actividade*.

Tabela 3.8: Requisitos Funcionais - Tratar Tipo de Tarefa, Topônimo, Número de Polícia, Zona e Actividade.

Caso de uso	Identificador	Descrição
<i>Tratar Tipo de Tarefa</i>	<i>uc_sus10_r01</i>	Criação do mecanismo de adição de Tipos de Tarefa a ser realizada por <i>Agentes Fiscalizadores</i> .
	<i>uc_sus10_r02</i>	Criação do mecanismo de edição de Tipos de Tarefa a ser realizada por <i>Agentes Fiscalizadores</i> .
	<i>uc_sus10_r03</i>	Criação do mecanismo de remoção de Tipos de Tarefa a ser realizada por <i>Agentes Fiscalizadores</i> .
<i>Tratar Topônimo</i>	<i>uc_sus11_r01</i>	Criação do mecanismo de adição de Topônimos.
	<i>uc_sus11_r02</i>	Criação do mecanismo de edição de Topônimos.
	<i>uc_sus11_r03</i>	Criação do mecanismo de remoção de Topônimos.
<i>Tratar Número de Polícia</i>	<i>uc_sus12_r01</i>	Criação do mecanismo de adição de Números de Polícia.
	<i>uc_sus12_r02</i>	Criação do mecanismo de edição de Números de Polícia.
	<i>uc_sus12_r03</i>	Criação do mecanismo de remoção de Números de Polícia.
<i>Tratar Unidade Orgânica</i>	<i>uc_sus13_r01</i>	Criação do mecanismo de adição de Unidades Orgânicas do Município.
	<i>uc_sus13_r02</i>	Criação do mecanismo de edição de Unidades Orgânicas do Município.
	<i>uc_sus13_r03</i>	Criação do mecanismo de remoção de Unidades Orgânicas do Município.
<i>Tratar Zona</i>	<i>uc_sus14_r01</i>	Criação do mecanismo de adição de Zonas do Município.
	<i>uc_sus14_r02</i>	Criação do mecanismo de edição de Zonas do Município.
	<i>uc_sus14_r03</i>	Criação do mecanismo de remoção de Zonas do Município.
	<i>uc_sus14_r04</i>	Criação do mecanismo de adição de localizações georeferenciadas a Zonas do Município.
<i>Tratar Actividade</i>	<i>uc_sus15_r01</i>	Criação do mecanismo de adição de Actividades a serem realizadas por <i>Agentes Fiscalizadores</i> .
	<i>uc_sus15_r02</i>	Criação do mecanismo de edição de Actividades a serem realizadas por <i>Agentes Fiscalizadores</i> .
	<i>uc_sus15_r03</i>	Criação do mecanismo de remoção de Actividades a serem realizadas por <i>Agentes Fiscalizadores</i> .
	<i>uc_sus15_r04</i>	Criação do mecanismo de cancelamento de Actividades a serem realizadas por <i>Agentes Fiscalizadores</i> .

Tratar Tarefa, Alocação, Conhecimento e Equipa

Na *Tabela 3.9* são apresentados os requisitos funcionais referentes aos casos de uso *Tratar Tarefa*, *Tratar Alocação*, *Tratar Conhecimento*, *Tratar Equipa*.

Tabela 3.9: Requisitos Funcionais - Tratar Tarefa, Alocação, Conhecimento e Equipa.

Caso de uso	Identificador	Descrição
<i>Tratar Tarefa</i>	<i>uc_sus16_r01</i>	Criação do mecanismo de adição de Tarefas associadas a Actividades a serem realizadas por <i>Agentes Fiscalizadores</i> .
	<i>uc_sus16_r02</i>	Criação do mecanismo de edição de Tarefas associadas a Actividades a serem realizadas por <i>Agentes Fiscalizadores</i> .
	<i>uc_sus16_r03</i>	Criação do mecanismo de remoção de Tarefas associadas a Actividades a serem realizadas por <i>Agentes Fiscalizadores</i> .
	<i>uc_sus16_r04</i>	Criação do mecanismo de adição de localizações geo-referenciadas a Tarefas associadas a Actividades a serem realizadas por <i>Agentes Fiscalizadores</i> .
<i>Tratar Alocação</i>	<i>uc_sus17_r01</i>	Criação do mecanismo de adição de Alocações de Equipas a Zonas do Município.
	<i>uc_sus17_r02</i>	Criação do mecanismo de edição de Alocações de Equipas a Zonas do Município.
	<i>uc_sus17_r03</i>	Criação do mecanismo de remoção de Alocações de Equipas a Zonas do Município.
<i>Tratar Conhecimento</i>	<i>uc_sus18_r01</i>	Criação do mecanismo de adição de Linhas de Orientação para o reporte de Ocorrência dos vários tipos.
	<i>uc_sus18_r02</i>	Criação do mecanismo de edição de Linhas de Orientação para o reporte de Ocorrência dos vários tipos.
	<i>uc_sus18_r03</i>	Criação do mecanismo de remoção de Linhas de Orientação para o reporte de Ocorrência dos vários tipos.
<i>Tratar Equipa</i>	<i>uc_sus19_r01</i>	Criação do mecanismo de adição de Equipas de <i>Agentes Fiscalizadores</i> .
	<i>uc_sus19_r02</i>	Criação do mecanismo de edição de Equipas de <i>Agentes Fiscalizadores</i> .
	<i>uc_sus19_r03</i>	Criação do mecanismo de remoção de Equipas de <i>Agentes Fiscalizadores</i> .

Validar Ocorrência Reportada e Tratar Ocorrência

Na *Tabela 3.10* são apresentados os requisitos funcionais referentes aos casos de uso *Validar Ocorrência Reportada e Tratar Ocorrência*.

Tabela 3.10: Requisitos Funcionais - Validar Ocorrência Reportada e Tratar Ocorrência.

Caso de uso	Identificador	Descrição
<i>Validar Ocorrência Reportada</i>	<i>uc_sus20_r01</i>	Criação do mecanismo de validação de Ocorrências Reportadas.
	<i>uc_sus20_r02</i>	Criação do mecanismo de devolução para edição de Ocorrências Reportadas.
	<i>uc_sus20_r03</i>	Criação do mecanismo de eliminação de Ocorrências Reportadas.
	<i>uc_sus20_r04</i>	Criação do mecanismo de arquivamento de Ocorrências Reportadas.
<i>Tratar Ocorrência</i>	<i>uc_sus21_r01</i>	Criação do mecanismo de adição de Ocorrências.
	<i>uc_sus21_r02</i>	Criação do mecanismo de edição de Ocorrências.
	<i>uc_sus21_r03</i>	Criação do mecanismo de remoção de Ocorrências.
	<i>uc_sus21_r04</i>	Criação do mecanismo de impressão da informação referente a Ocorrências Reportadas.

MSuS - Mobile Supervision System

Aqui são apresentados os vários requisitos funcionais referentes a cada um dos casos de uso apresentados na *Secção 3.1.2.2 MSuS - Mobile Supervision System*.

Tratar Ocorrência

Na *Tabela 3.11* são apresentados os requisitos funcionais do caso de uso *Tratar Ocorrência*.

Tabela 3.11: Requisitos Funcionais - Tratar Ocorrência.

Caso de uso	Identificador	Descrição
<i>Tratar Ocorrência</i>	<i>uc_msus01_r01</i>	Criação do mecanismo de adição de Ocorrências.
	<i>uc_msus01_r02</i>	Criação do mecanismo de edição de Ocorrências.
	<i>uc_msus01_r03</i>	Criação do mecanismo de remoção de Ocorrências.
	<i>uc_msus01_r04</i>	Criação do mecanismo de associação de localização geo-referenciada a Ocorrências Reportadas.
	<i>uc_msus01_r05</i>	Criação do mecanismo de impressão da informação referente a Ocorrências Reportadas.

Comum

Aqui são apresentados os requisitos funcionais referentes a cada um dos casos de uso apresentados na *Secção 3.1.2.3 Comum*.

Pesquisar Informação Interna e Externa

Na *Tabela 3.12* são apresentados os requisitos funcionais referentes aos casos de uso *Pesquisar Informação Interna e Pesquisar Informação Externa*.

Tabela 3.12: Requisitos Funcionais - Pesquisa de Informação Interna e Externa.

Caso de uso	Identificador	Descrição
<i>Pesquisar Informação Interna</i>	<i>uc_com01_r01</i>	Criação do mecanismo de pesquisa de informação criada através da utilização do sistema.
<i>Pesquisar Informação Externa</i>	<i>uc_com02_r01</i>	Criação do mecanismo de pesquisa de informação criada externamente à utilização do sistema.

3.1.3.3 Requisitos Não-Funcionais

Nesta secção descrevem-se um pouco os requisitos do foro não funcional exigidos pelo cliente. Os requisitos suplementares ou requisitos não-funcionais são requisitos transversais aos vários casos de utilização do software. São requisitos de qualidade do produto e definem as propriedades do sistema, apresentando-se como um complemento aos requisitos funcionais. Compreendem tipicamente características como tempos de resposta, segurança, fiabilidade, normas, etc. É importante adequar este tipo de requisitos aos utilizadores finais, reforçando a implementação num ou noutro aspecto, conforme as necessidades. Os requisitos não-funcionais podem assumir um carácter mais crítico do que os requisitos funcionais uma vez que os podem inviabilizar, tornando o sistema obsoleto, enquanto os requisitos funcionais apenas nos descrevem a informação acerca da real utilidade do sistema. Torna-se crucial seguir a regra dos 3 *Us* (*Usability*, *Used* e *Utility*) na implementação de uma interface de qualidade, e ter sempre em consideração que na relação custo/rentabilidade destes requisitos muitas vezes o ganho ao atingir os mesmos, a nível de qualidade e de grau de certeza, não compensa o trabalho despendido para os atingir. Em relação à criticidade, os requisitos apresentados são classificados em **Essencial**, **Elevada** e **Média**.

Na *Tabela 3.13* são descritos os requisitos não-funcionais associados ao software pretendido, apresentando, para cada um deles, uma pequena descrição e a sua prioridade, de acordo com a classificação apresentada [dSdI09].

Tabela 3.13: Requisitos Não-Funcionais

Requisito	Descrição	Prioridade
<i>Usabilidade</i>	O sistema pretendido deverá ser simples de utilizar, de modo a minimizar a formação a efectuar aos utilizadores finais. Para esse efeito é essencial tornar a interface intuitiva (ou pelo menos intuitiva para utilizadores regulares de computadores). O percurso do utilizador deverá ser acompanhado com tópicos de ajuda e mensagens de erro claras e esclarecedoras. A facilidade de configuração do sistema deverá ser também uma aposta tendo em vista esta adaptação ao utilizador.	Essencial
<i>Eficiência</i>	A garantia da eficiência da aplicação passa por factores tais como as tecnologias usadas (o uso de tecnologias de ponta e reconhecidas como soluções para este tipo de software é importante para garantir a sua longevidade) e a proximidade e qualidade dos servidores.	Elevada
<i>Segurança</i>	As funcionalidades devem depender do nível de acesso ao sistema que o utilizador tem, limitando assim o acesso dos utilizadores a funcionalidades para as quais não possuem permissões. Existem dados privados e protegidos que devem ser salvaguardados. O software deve possuir mecanismos para impedir a visualização ou modificação destes dados por pessoal não autorizado.	Essencial
<i>Hardware</i>	Quando a solução apresentada for posta no terreno é necessário que esta tenha em consideração as especificidades dos dispositivos móveis que irão executar o <i>MSuS</i> . Em causa está a utilização de <i>PDA</i> s industriais acompanhados de impressora portátil. Um estudo sobre as melhor opções para a <i>CMP</i> está a ser realizado.	Essencial
<i>Software</i>	A base de dados a utilizar para o desenvolvimento do <i>SuS</i> será o <i>Oracle Database 11g</i> e a linguagem de programação <i>PL/SQL</i> . Em relação ao <i>MSuS</i> , este deverá executar sobre o sistema operativo da <i>Microsoft</i> , o <i>Windows Mobile 6.5</i>	Essencial
<i>Fiabilidade</i>	A fiabilidade representa o nível de confiança que é depositado pelo utilizador no sistema. Assim, pretende-se um produto consistente e robusto (que efectue as operações 100% correctas e responda sempre de forma previsível ao utilizador), essencial para garantir a satisfação do cliente, assim como a boa utilização do produto.	Elevada

Requisito	Descrição	Prioridade
<i>Manutenção</i>	Pretende-se um sistema fácil de manter, de forma a minimizar o tempo perdido com as operações de manutenção. É vital que o sistema não fique sem funcionar, salvo, porventura, alguma excepção, que pode levar a avultados prejuízos.	Média
<i>Portabilidade</i>	O software (sistema <i>SuS</i>) deve ser compatível com os browsers mais conhecidos (Internet Explorer, Mozilla Firefox, Opera), abrangendo assim uma gama de utilizadores suficientemente vasta para se considerar que a cobertura do universo de utilizadores é virtualmente total.	Média

3.1.3.4 Requisitos de Documentação

A este nível é habitual o cliente exigir alguns documentos, de forma a minimizar a necessidade de formação e/ou contacto com os fornecedores. Assim sendo, julga-se ser interessante que seja produzida a seguinte documentação de forma a minimizar tempos de aprendizagem e de formação. A *Tabela 3.14* apresenta a documentação a criar [dSdI09].

Tabela 3.14: Requisitos de Documentação

Manual de Instalação	Pretende documentar todos os passos necessários para que o produto seja instalado nas máquinas do cliente. Sendo assim, deve conter um conjunto de secções para que o utilizador não precise de contactar a empresa fornecedora.
Manual do Utilizador	Tem como finalidade assistir os utilizadores do software, de maneira a tomarem conhecimento de como lidar com ele. Deve conter um guião escrito e imagens associadas, sobre como o sistema deve ser apresentado e responder aos comandos introduzidos pelo utilizador. É importante que contenha uma secção de problemas conhecidos e respectivas soluções, evitando a perda de tempo à procura de soluções. Sendo um documento a ser distribuído pelos vários utilizadores finais, este deve ter em conta as características do público-alvo, evitando demasiados termos técnicos e descrição de implementação. O manual não tem de ser exclusivamente distribuído em papel, pode também ser utilizada a ajuda <i>online</i> .

3.2 Revisão Tecnológica

Em relação ao estudo das tecnologias a utilizar na construção da solução, e como se pode ver na *Figura 3.1* é necessário escolher quais serão as melhores opções em relação:

- à aplicação de Back-Office (*SuS*);
- à aplicação a correr nos dispositivos móveis, mais concretamente nos *PDA*s (*MSuS*);
- à base de dados a utilizar nestes *PDA*s;
- aos *Web services* a criar;
- e ao Servidor de Aplicações Web para a disponibilização de *Web services*.

O estudo efectuado é apresentado de seguida.

3.2.1 SuS - Supervision System

Em relação à aplicação de *Back-Office* do projecto existem algumas restrições impostas pela instituição de acolhimento, o *PSI*. De acordo com esta, a aplicação terá de ser desenvolvida recorrendo à base de dados *Oracle* e à linguagem *PL/SQL* para criação do ambiente de interacção.

3.2.2 MSuS - Mobile Supervision System

Tendo a aplicação em causa o intuito de correr num *Personal Digital Assistant (PDA)* é de todo conveniente estudar quais as tecnologias que podem ser utilizadas para a criação de todos os módulos em causa, suas vantagens e desvantagens, para que, então, se proceda à decisão sobre quais serão utilizadas no âmbito deste projecto. Foi dada liberdade ao autor deste relatório para a escolha das tecnologias.

3.2.2.1 Aplicação

Em relação à aplicação a criar, primeiro é necessário optar pelo desenvolvimento em ambiente web ou de uma aplicação mobile. Para além disso, a tecnologia a utilizar dentro da opção anterior, é também uma escolha a realizar.

Aquando da escolha da tecnologia a utilizar é necessário ter em consideração os seguintes aspectos:

- Acesso às características do dispositivo Móvel;
- Custo;

- Desempenho;
- Experiência com a tecnologia;
- Ferramentas de desenvolvimento e *IDEs* (*Integrated Development Environment*);
- Ferramentas para teste.

Desenvolvimento Orientado à Web

Existem várias opções no campo de desenvolvimento web. Devido às especificidades do projecto em causa, como acesso a funcionalidades do dispositivo móvel, utilização de geo-posicionamento, segurança, entre outras, esta opção foi descartada.

Contudo, o *Microsoft Silverlight* [Cor09g] vocacionado para dispositivos móveis, uma tecnologia que está em fase de desenvolvimento, poderá, num futuro próximo, vir a ser uma opção a ponderar para próximos desenvolvimentos na área e/ou em outros projectos semelhantes.

Desenvolvimento não Orientado à Web

No desenvolvimento de aplicações não orientadas à Web existem várias opções à escolha. Assim sendo, é necessário analisar as que parecem ser as mais viáveis no contexto do projecto.

Brew

O *Brew* [Qua09], *Binary Runtime Environment for Wireless*, é uma plataforma de desenvolvimento de aplicações criada pela *Qualcomm* para telemóveis. Existe compatibilidade com alguns *PDA*s.

No entanto, para que seja possível obter o *SDK* (*Software Development Kit*) e as ferramentas base para iniciar o desenvolvimento, é necessário o registo na *Qualcomm*. Para a criação de aplicações comerciais, é necessário seguir uma série de passos que implicam um custo associado dependendo do número de aplicações a desenvolver, que nunca poderá ser menor que 100. Após as aplicações estarem desenvolvidas e testadas - de notar que apenas podem ser testadas em emuladores obtidos com o *SDK* e em hardware que permita correr aplicações *Brew*, que por si só já restringe muito o número de dispositivos móveis compatíveis -, tais aplicações serão testadas pela *Qualcomm*. Só assim a aplicação pode, então, ser comercializada via mecanismos da própria *Qualcomm*.

Existem muitas críticas a esta forma de desenvolvimento, desde logo pelo facto da obrigação ao registo na *Qualcomm* e submissão das aplicações para testes, acarretando um esforço monetário adicional. Estes custos provocam uma menor adesão, assim como quase nenhum apelo ao desenvolvimento por *hobbie*. Para além de todo este processo, ainda existe a possibilidade da aplicação ser rejeitada devido a insuficiente rentabilidade ou mesmo devido a competição directa com produtos da própria *Qualcomm* [Qua09].

Sendo baseado nas linguagens de programação *C/C++*, os *IDEs* que podem ser utilizados são diversos, desde *open source* (*Eclipse*, por exemplo) até versões comerciais (*Microsoft Visual Studio*, por exemplo).

Em relação a desempenho, o *Brew* aparece como sendo uma das melhores opções, muito devido ao facto de se basear numa linguagem de programação de mais baixo nível que os seus concorrentes [Whi08].

Java

O Java é uma tecnologia da *Sun Microsystems* que aparece na criação de aplicações móveis sob a sua versão *Java ME* [Net09a] [Ris08]. Esta plataforma é uma solução aberta, ao contrário de muitas outras que se encontram no mercado. Foi desenhada com o intuito de providenciar portabilidade de aplicações [Net09b].

Sendo uma tecnologia sob a licença *GNU General Public License*, os custos associados à utilização desta tecnologia apenas se focam na experiência e capacidades dos desenvolvedores.

Em relação a ferramentas para o desenvolvimento de aplicações, a *Sun Microsystems* providencia o *Java ME SDK* e ferramentas base. No entanto, e como é conhecido, devido aos múltiplos ambientes e *APIs* (*Application Programming Interface*), existem diversos *SDKs* e *toolkits* para satisfazer necessidades específicas. Para que a experiência do utilizador seja melhor, existem no mercado diversos *IDEs*, quer *open source*, quer comerciais. No mundo do *open source* os mais populares são o *Eclipse* e o *NetBeans*, este último da própria *Sun*. No lado comercial, soluções como o *Rational Application Developer* da *IBM* estão à disposição dos desenvolvedores. Muitas ferramentas sofrem da pouca oferta de soluções de desenvolvimento, no mundo do Java acontece precisamente o contrário, existem demasiadas soluções. A complexidade da arquitectura do *Java ME* permite uma boa adaptação a um vasto leque de plataformas mas, devido ao problema referido anteriormente, a escolha do ambiente de desenvolvimento mais adequado pode ser um desafio.

Para que seja possível proceder a testes à aplicação desenvolvida, e sendo a aplicação a desenvolver para *PDA*s, a necessidade de utilização de emuladores torna-se bastante crítica. Neste campo o *Java ME* oferece emuladores genéricos para alguns dispositivos, através dos *SDKs* e *IDEs*. Adicionalmente, alguns dos criadores de dispositivos, como a *Nokia* ou a *Motorola*, providenciam *SDKs* que oferecem emuladores mais precisos para certos dispositivos, sendo que alguns podem ainda ser integrados com *IDEs*. É também necessário referir que os emuladores genéricos não testam as *APIs* específicas dos criadores do dispositivo em causa, podendo não reflectir problemas específicos deste dispositivo. Assim sendo, é recomendado testar as aplicações criadas no dispositivo alvo.

O acesso às capacidades do dispositivo móvel, como é o caso da câmara, normalmente requer a inclusão manual destes acessos nas configurações, perfil ou mesmo na *API* para esse dispositivo.

Em relação ao desempenho de uma aplicação desenvolvida utilizando *Java ME*, esta sofre mais pelo poder de processamento limitado dos dispositivos móveis, uma vez que a aplicação é executada numa máquina virtual [Whi08].

SuperWaba

Depois da tecnologia *Java ME* ter sido abordada anteriormente, o *SuperWaba* [Sup09], mais concretamente o *TotalCross*, aparece como uma alternativa viável, na medida em que este é uma máquina virtual baseada na do *Java* tendo como alvo os dispositivos móveis.

Todas as vantagens e desvantagens ligadas à utilização de *Java* são encontradas nesta tecnologia, no entanto existem algumas diferenças. Mesmo sendo parecido com a especificação do *Java 2 Standard Edition*, o *SuperWaba* aparece bastante vocacionado para as restrições ligadas às limitações dos dispositivos móveis. Sendo assim, o seu desempenho, devido à optimização criada, consegue ser até duas vezes mais rápido que o concorrente *Java ME* [Eve06].

Mesmo sendo uma tecnologia *open source*, é necessário, se utilizada para criação de aplicações comerciais, o licenciamento por cada dispositivo móvel em que tal aplicação será instalada [Sup09].

Microsoft .NET

Dentro da leque de soluções oferecidas pela tecnologia *Microsoft .NET* encontra-se o *.NET Compact Framework* [Cen09]. Este é uma componente dos dispositivos *Windows Mobile* e *Windows Embedded CE* que permite o desenvolvimento e utilização de aplicações e *Web services*.

Para desenvolvimento, o *IDE* de eleição é o *Microsoft Visual Studio*, existindo, contudo, outras possibilidades. Claro que tal ferramenta tem custos monetários associados à sua utilização, no entanto, esta simplifica muito o desenvolvimento para o *.NET Compact Framework* e tem uma curva de aprendizagem pequena para desenvolvedores já familiarizados com o *.NET* e desenvolvimento em *C#* e/ou *VB*.

Em relação a ferramentas para teste das aplicações, o *Windows Mobile 6 SDK* providencia o chamado *Device Emulator* que é considerado uma das vantagens em desenvolver aplicações móveis através do *.NET*. Este emulador providencia *pseudo-GPS* (*Global Positioning System*) e todas as possibilidades de interacção com o dispositivo. Este emulador é um verdadeiro *Advanced RISC Machine (ARM) emulator*, o que quer dizer que corre o mesmo código que dispositivos móveis baseados em arquitecturas com processadores *ARM*. Segundo o fabricante, 90% dos dispositivos móveis estão equipados com processadores *ARM*, conjunto onde se encontram os *PDA*s [Ltd09].

Como o *.NET Compact Framework* foi criado para responder às necessidades de desenvolvimento no sistema operativo *Windows Mobile*, tendo acesso a todas as capacidades do dispositivo móvel suportadas pelo sistema operativo.

Em relação ao desempenho, este é considerado razoável em todas as vertentes [Whi08].

Escolha

Após uma análise cuidada de várias tecnologias existentes no mercado (e certas implementações) foi escolhido optar pela tecnologia *.NET*. Isto deve-se, não só à existência de licenciamento em tecnologia Microsoft mas, também, devido aos dispositivos alvo do projecto serem *PDA*s com *Windows Mobile*. Desta forma, a integração com todas as especificidades do dispositivo é facilitada, pois a tecnologia do sistema operativo e da aplicação a desenvolver é a mesma.

No entanto, não seria de todo uma má escolha a utilização de um tecnologia baseada em *Java*, como por exemplo o *SuperWaba* ou mesmo o *Java ME*. O caso do *Brew* estará sempre fora de questão devido à necessidade de licenciamento e à forma como todo o processo de desenvolvimento e entrega ao cliente é realizado, assim como a compatibilidade deste com os dispositivos móveis ser restrita.

3.2.2.2 Base de Dados

A escolha da tecnologia de bases de dados para o *PDA* depende muito dos mecanismos existentes de sincronização com a base de dados utilizada no sistema com o qual a aplicação desenvolvida vai comunicar - o *SuS*. A base de dados em causa é proveniente da tecnologia *Oracle*. Logo, a opção que tornará mais fácil esta integração será a escolha mais acertada. No entanto, ao desenvolver a aplicação em causa, a mudança de tecnologia a utilizar deverá ser facilmente realizada para responder a necessidades da troca de aplicação com a qual vai sincronizar.

Borland JDataStore

Existe uma tecnologia de base de dados, denominada de *JDataStore* [Cor09c], que parece ser um opção interessante para utilizar em dispositivos móveis. No entanto, no próprio site da *Borland* não existe a indicação de como obter este software. Sendo assim, como parece ser um produto descontinuado, esta opção acaba por deixar de fazer parte do lote de tecnologias a considerar.

Microsoft SQL Server Compact

A *Microsoft* entra no leque de opções devido ao produto *SQL Server Compact 3.5* [Cor09e].

Esta base de dados é desenhada para desenvolvedores que tenham como alvo quer dispositivos móveis baseados em *Windows Mobile*, quer *Desktops*. Providencia sincronização com *SQL Server*, assim como *API* para desenvolvimento. O desenvolvimento pode ser integrado através do *Microsoft Visual Studio* que providencia todos os mecanismos de

teste e desenvolvimento inerentes.

Através da componente *SQL Server Integration Services* são providenciados mecanismos de troca de informação com bases de dados *Oracle*, *IBM*, entre outras [Cor09f] [Cor08a].

Oracle Database Lite

A *Oracle* providencia aos seus clientes a possibilidade de adquirir o produto *Oracle Database Lite* [Cor08b]. Este produto providencia a possibilidade de criar bases de dados que têm por alvo dispositivos móveis, e, sendo uma tecnologia *Java-based*, pode ser utilizada sobre várias plataformas.

Esta base de dados vem acompanhada de mecanismos de sincronização para bases de dados *Oracle*, sendo uma mais valia no caso de a aplicação a desenvolver necessitar de sincronizar com esta tecnologia.

A solução da *Oracle* permite a integração com vários *IDEs*, entre os quais o *Microsoft Visual Studio*, *JDeveloper*, *Eclipse*, entre outros, tendo para isso mecanismos **ODBC**, **JDBC** e **ADO.NET**. [Cor09i] [Cor08b]

Escolha

Como foi dito no início da análise destas opções, a opção para a base de dados no *PDA* cairá sobre aquela que fornecer mais facilidade de sincronização com a base de dados *Oracle* a utilizar no sistema de *Back-Office*. A escolha fica, então, dividida entre o *Oracle Database Lite* e o *Microsoft SQL Serve Compact*, uma vez que ambas as soluções apresentam mecanismos de sincronização com bases de dados *Oracle*.

A possibilidade de teste em ambiente *desktop* ser apenas possível numa das soluções poderia limitar a escolha. No entanto, recorrendo ao *Microsoft Device Emulator* quer uma, quer outra podem ser instaladas e testadas no ambiente *Windows Mobile*.

Sendo assim, a escolha passa por perceber qual a solução que apresenta maior flexibilidade numa possível manutenção e/ou alteração da tecnologia da aplicação. Seguindo este critério, a solução da *Oracle* aparece como sendo a melhor opção, pois ao utilizar o *Microsoft SQL Server* fica-se mais dependente das tecnologias da *Microsoft*.

3.2.2.3 Web Services

Para além da utilização de Web services também foi equacionada a utilização de um Enterprise Service Bus (mais conhecido apenas por *ESB*) [Ric06]. Esta tecnologia permite a integração de vários sistemas de uma forma mais transparente, servindo como intermediário na transmissão de informação entre os vários intervenientes.

Esta tecnologia não será utilizada, uma vez que o esforço necessário para que esta seja utilizada não compensa as vantagens que daí advêm, uma vez que, a solução não prevê uma grande interacção entre os intervenientes, existindo apenas ligação pontual entre um dispositivo móvel e o sistema de *Back-Office* [Cha04].

A discussão em relação à implementação dos *Web services*, que vão ser a ligação entre a aplicação no dispositivo móvel e os sistemas de registo de autos e sistemas de pesquisa, é sobre se será utilizada tecnologia *Java* ou *Microsoft .NET*.

Java

Em relação à solução *Java*, é disponibilizado um serviço, através do projecto *Glassfish - Metro* [Med09], de criação de *Web services* de fácil utilização, com bom desempenho e extensibilidade. As principais características estão representadas na *Figura 3.6* ¹. Estes serviços apresentam uma vasta panóplia de soluções de transporte e de segurança, em que se pode referir, com especial importância, a utilização de *SOAP* (*Simple Object Access Protocol*) e do *WS-Security* (*Web Services Security*).

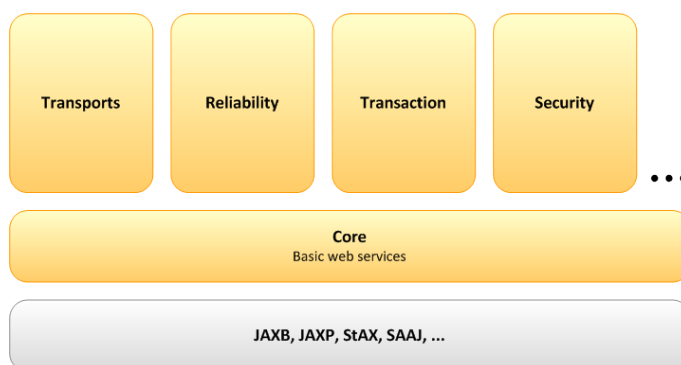


Figura 3.6: Principais Características dos Web services Java

Microsoft .NET

No caso da tecnologia *.NET*, é disponibilizado um serviço de criação de *Web services* totalmente integrado. As principais características estão ilustradas na *Figura 3.7* ².

Tal como a solução *Java*, o *.NET* disponibiliza transporte *SOAP*, segurança *WS-Security*, entre outras soluções [Cor09f].

¹baseado no diagrama existente em [Med09] sobre Web services

²baseado no diagrama existente em [Cor09f] sobre Web services

Análise do Problema

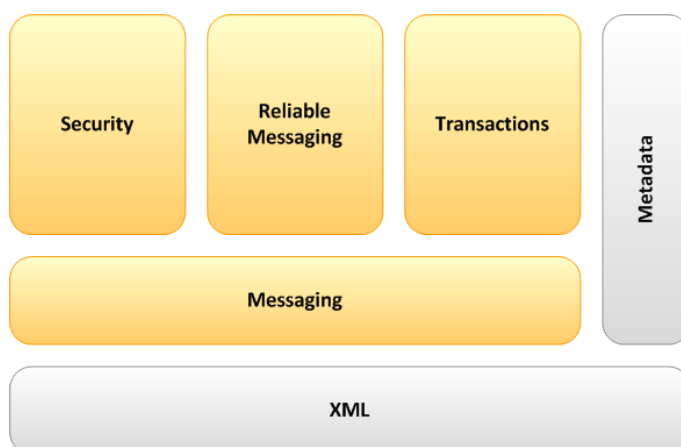


Figura 3.7: Principais Características dos Web services .NET

Escolha

A diferença entre as duas tecnologias não é significativa. Em ambas, o acesso por parte de outras plataformas de desenvolvimento é garantido (objectivo dos *Web services*). A escolha recai sobre a tecnologia *Java*, na medida em que a base de dados que irá suportar o projecto a desenvolver é *Oracle* (restrição de desenvolvimento), que é uma solução baseada em *Java*. Assim, a compatibilidade e integração será facilitada pela criação de *Web services* em *Java*.

Esta escolha não influencia a aplicação a executar no *PDA*, pois os *Web services* permitem acesso a qualquer aplicação em qualquer plataforma.

3.2.2.4 Servidor de Aplicações Web

Existem várias tecnologias no mercado que providenciam *deploy* para os *Web services* a desenvolver no âmbito deste projecto.

Para além de procura de informação sobre as várias soluções, foram realizados testes de comparação de desempenho para uma melhor fundamentação na escolha.

Apache Tomcat

A *Apache Software Foundation* providencia suporte à comunidade de projectos de software *open source*. O servidor de aplicações fornecido é o *Apache Tomcat* [Fou09a], e evidencia-se pelo papel crucial que representou no crescimento inicial da *World Wide Web*.

Este servidor de aplicações está disponível para um grande leque de sistemas operativos (como *Windows*, mas principalmente para os baseados em *Unix*), fornecendo de uma forma gratuita ferramentas para disponibilização de serviços online.

Dentro do projecto *Apache*, o *Axis* [Fou09c] aparece como uma das soluções para *deployment* dos *Web services*.

Esta tecnologia suporta as especificidades da escolha de *Web services* discutida anteriormente (como *SOAP* e *WS-Security*), tendo como vantagem, sobre alguns dos seus competidores, o facto de ser uma tecnologia licenciada segundo *GNU General Public License*, o que tem levado a que seja uma opção muito tentadora [Fou09b].

IBM WebSphere

A *International Business Machines Corporation (IBM)* fornece uma linha de produtos na área de servidores de aplicações.

Websphere [Cor09b] é uma linha de produtos de software da *IBM*, tendo como produto essencial o chamado *IBM WebSphere Application Server*. Este foi desenhado para criar, operar e integrar negócio electrónico através de múltiplas plataformas de computação. Esta tecnologia é baseada nas tecnologias web de *Java*. Inclui quer as componentes de *run-time*, quer as ferramentas para desenvolver aplicações que irão correr no *IBM WebSphere Application Server*.

Esta solução, tal como as discutidas até ao momento, permitem utilização das especificidades dos *Web services* a desenvolver. Em relação à solução da *Apache*, o *Websphere* tem a desvantagem de ser necessária a sua compra, existindo, no entanto, uma versão não paga, o *WebSphere Application Server Community Edition* [Cor09a].

Sun Application Server

Em relação a servidores de aplicações, a *Sun* aparece com o produto proveniente do projecto *Glassfish*, *Sun Application Server* [SM09]. Sendo um servidor de aplicações *open source*, o *Sun Application Server* aparece como uma alternativa viável para o *deployment* dos *Web services*.

O *Metro*, parte integrante do projecto *Glassfish*, suporta todas as tecnologias necessárias ao desenvolvimento deste projecto.

Oracle

A *Oracle* aparece com uma linha de produtos que vão de encontro às necessidades do projecto. Especificamente, o software para servidores de aplicações web que a *Oracle* disponibiliza são o *Oracle Application Server* [Cor09j] e o *Oracle Weblogic Server* [Cor09i].

Oracle Application Server

Esta tecnologia, desenvolvida pela *Oracle*, irá ser descontinuada. Esta situação acontecerá devido à compra da *BEA* por parte da *Oracle*. A *Oracle* irá continuar a desenvolver

o servidor anteriormente da *BEA*, uma vez que esta é uma das líderes em soluções de infraestruturas para aplicações empresariais [Cor09i].

Sendo assim, esta opção aparece como uma das que são descartadas à partida devido à sua descontinuação, facto que pode provocar um problema no futuro aquando da migração, obrigada, para outra solução.

Oracle WebLogic Server

Como dito anteriormente, devido à compra da *BEA* por parte da *Oracle*, esta aparece como uma das soluções para *deployment* de aplicações web considerada com uma das líderes do mercado. Tal como as soluções analisadas até ao momento, este servidor suporta todas as tecnologias necessárias para a conclusão do projecto.

Comparação de Desempenhos

Para proceder à escolha de qual o servidor de aplicações web a utilizar no âmbito do projecto foram realizados testes para comparação de desempenhos.

Os testes realizados tiveram por objectivo comparar desempenhos de *Web services* nas várias opções apresentadas anteriormente. Foram utilizadas duas implementações de *Web services* (dois testes com objectivos diferentes):

1. uma implementação avalia apenas a rapidez em responder a um pedido simples, sem processamento no lado do servidor;
2. a outra avalia métricas no caso de existir processamento do lado do servidor.

As métricas utilizadas para a comparação foram as seguintes:

- Transacções por segundo (**Teste TPS**);
- Média de tempo de resposta em milisegundos (**Teste AVG**);
- Número de conexões, em milhares, estabelecidas em cinco minutos (**Teste CNT**);
- Mbytes trocados em cinco minutos (**Teste Mbytes**).

Os testes foram realizados utilizando para o servidor e cliente a mesma máquina, tendo o servidor sido executado numa máquina virtual.

No *Anexo A* são apresentados mais pormenores sobre o teste de desempenho realizado.

Primeira Implementação

Analisando as transacções existentes, informação ilustrada na *Figura 3.8*, é possível concluir que neste teste de resposta a solução apresentada pela *Oracle* destaca-se pela positiva, com *659 tps*. No entanto, o servidor *Apache* não está muito longe desse valor. O servidor *WebSphere*, da *IBM*, aparece como sendo o que tem menor capacidade de resposta rápida a pedidos enviados, com *522 tps*.

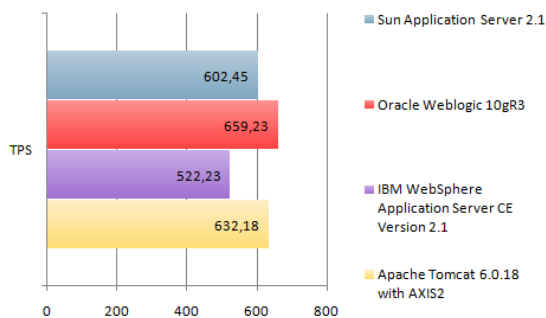


Figura 3.8: Teste TPS (Primeira Implementação).

Em relação ao tempo médio de resposta, e como pode ser observado na *Figura 3.9*, mais uma vez o servidor *Weblogic* da *Oracle* aparece em lugar de destaque, sendo o mais rápido a responder a pedidos (*13,78 milisegundos*). A solução da *IBM* continua a ter o pior resultado, com resposta média em *17,59 milisegundos*.

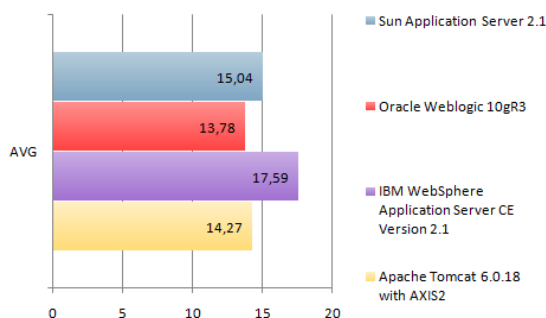


Figura 3.9: Teste AVG (Primeira Implementação).

Análise do Problema

A *Figura 3.10* apresenta o número de conexões estabelecidas em cinco minutos com cada um dos servidores. Este número é maior utilizando o servidor *Weblogic* da *Oracle*, aproximadamente, 198 mil conexões estabelecidas. Neste teste, mais uma vez, o servidor da *IBM* aparece como sendo o mais limitado, ficando-se pelas 157 mil conexões.

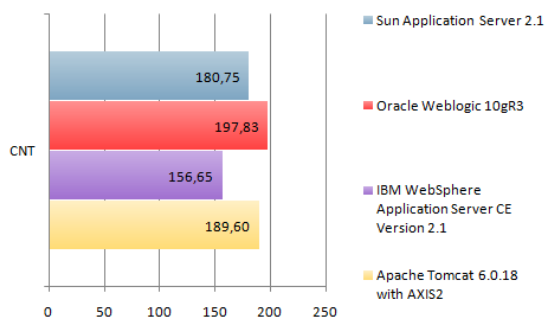


Figura 3.10: Teste CNT (Primeira Implementação).

Durante as conexões estabelecidas, o número de Mbytes trocados entre o servidor e cliente varia de solução para solução (ver *Figura 3.11*). O servidor da *IBM* destaca-se novamente pela negativa, na medida em que, sendo o servidor que estabeleceu menos conexões, aparece como uma das soluções que necessita de uma maior troca de informação entre os intervenientes, com um total de 50,34 Mbytes trocados. O servidor da *Sun* aparece como sendo o que necessita de menor troca de informação (40,15 Mbytes). No entanto, o *Weblogic* da *Oracle*, tendo por base a relação Mbytes/conexões, não fica muito atrás do concorrente anterior.

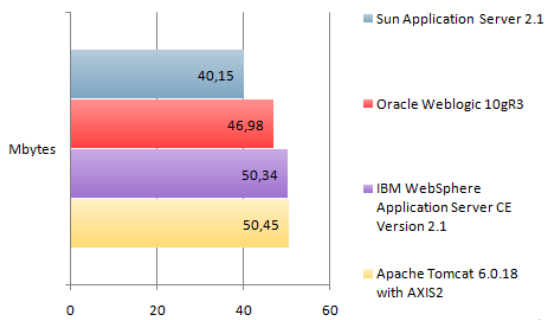


Figura 3.11: Teste Mbytes (Primeira Implementação).

Para além da informação contida nos gráficos apresentados, deve notar-se que o único servidor que originou erros de resposta foi o *Sun Application Server*, não tendo respondido a pedidos por 75 vezes num total de 180751 conexões.

Segunda Implementação

Os valores para as métricas definidas estão representados nas *Figuras 3.12, 3.13, 3.14 e 3.15*. Os resultados obtidos na segunda implementação são semelhantes e por isso as conclusões a tirar são as mesmas que na primeira implementação. No entanto, é de evidenciar que as diferenças entre as soluções já não são tão acentuadas.

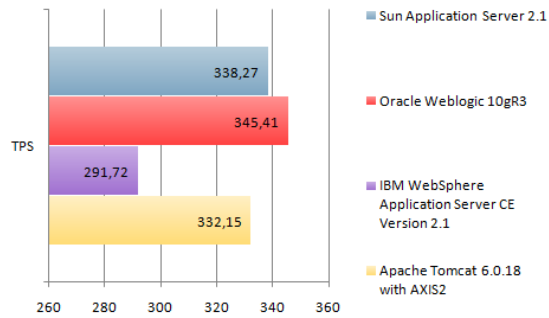


Figura 3.12: Teste TPS (Segunda Implementação).

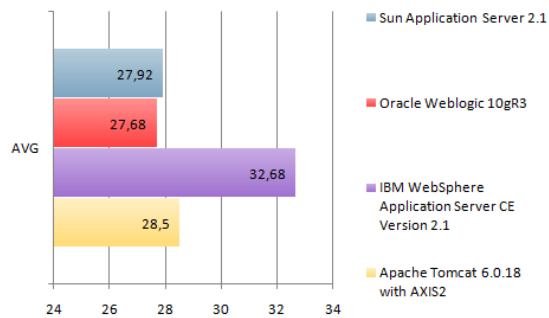


Figura 3.13: Teste AVG (Segunda Implementação).

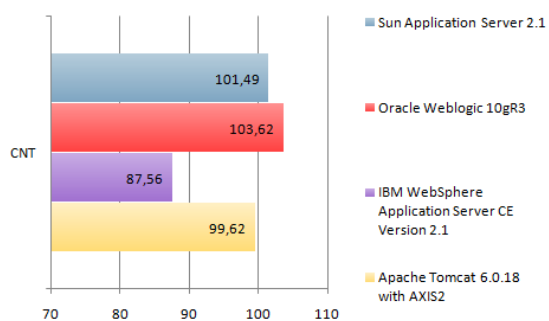


Figura 3.14: Teste CNT (Segunda Implementação).

Análise do Problema

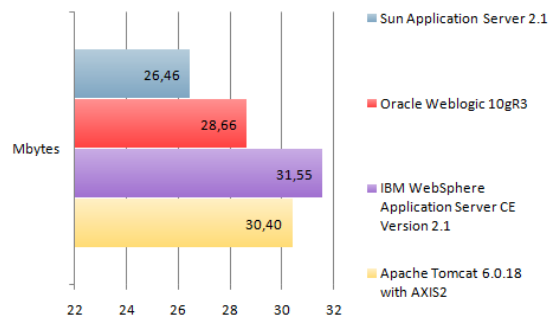


Figura 3.15: Teste Mbytes (Segunda Implementação).

De notar, que a solução da *Oracle* continua a ser a melhor do teste, e que a solução da *Sun* continua a originar erros, não respondendo a 225 pedidos num total de 101487 conexões.

Escolha

Tendo por base a análise dos resultados dos testes realizados e descritos anteriormente, a escolha recai sobre a solução da *Oracle*, o *Weblogic Server*.

Se, porventura, não fosse possível enveredar por um solução paga, o servidor *Apache Tomcat com Axis2* seria a melhor opção. A última opção é ocupada pela solução da *Sun*, com o *Sun Application Server*, na medida em que a possibilidade de não resposta por este servidor é maior do que em relação aos seus competidores, sendo efectivamente o único que nos testes realizados originou erros.

3.3 Sumário

A especificação de requisitos é um dos passos mais importantes para a construção de uma solução. Sem a identificação das necessidades não é possível criar algo que as satisfaça. No decurso deste capítulo, foram apresentados os requisitos funcionais e não funcionais mais importantes.

Após a especificação dos requisitos procedeu-se a análises tecnológicas no âmbito deste projecto. Sendo esta uma área tecnológica em grande expansão no momento, existem muitas opções que podem servir para o desenvolvimento do projecto em causa. Nesta análise foram consideradas as opções mais representativas existentes no mercado.

A *Figura 3.16* sintetiza as escolhas tecnológicas realizadas ao longo da análise tecnológica efectuada.

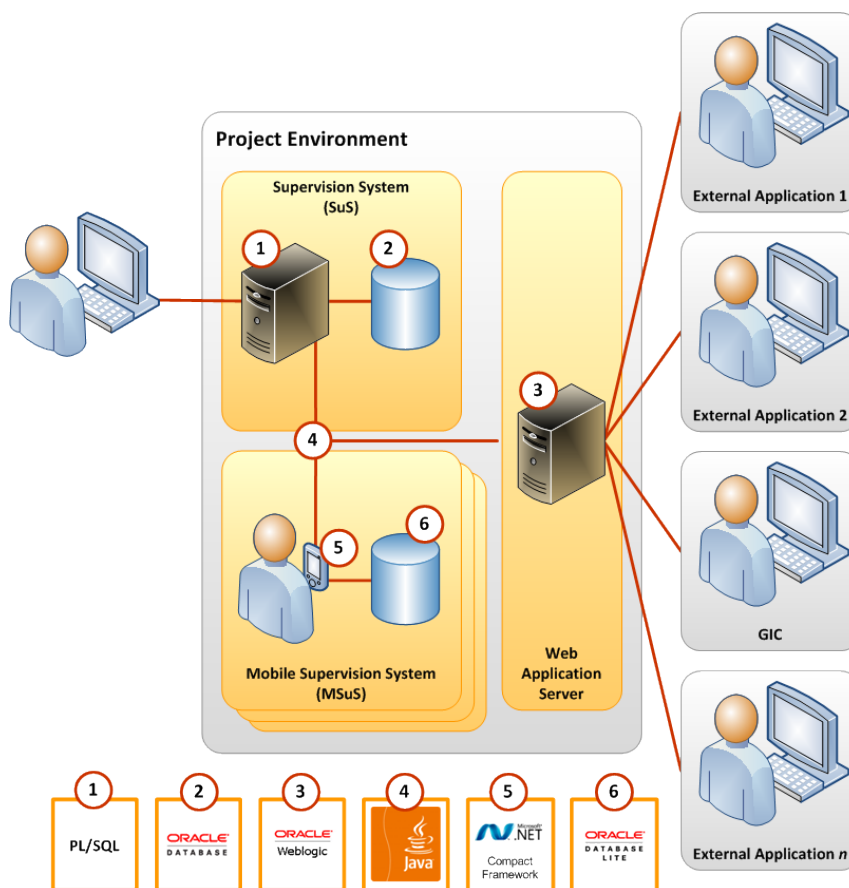


Figura 3.16: Escolhas Tecnológicas.

Capítulo 4

Solução Apresentada

Neste capítulo são apresentados a solução proposta para satisfazer as necessidades da *CMP* e o planeamento para o desenvolvimento desta.

4.1 Arquitectura

São aqui apresentados os pormenores em relação à arquitectura desenvolvida para a solução a criar. Desta forma, serão inicialmente apresentadas as visões mais globais e simples da arquitectura (Visão geral e Visão Conceptual). De seguida será apresentado um conjunto de informações mais detalhado de toda a arquitectura (Visão Estática, Visão Dinâmica e Visão de Dados). No final, serão analisadas as soluções arquitecturais criadas para possibilitar e facilitar os testes do sistema a desenvolver.

4.1.1 Visão Geral

Como referido anteriormente, o projecto consiste na criação de um Sistema de Supervisão Móvel (*MSuS*) que irá correr numa plataforma móvel, neste caso num *PDA*. Para que este sistema seja possível de administrar, existirá um Sistema de *Back-Office*, Sistema de Supervisão (*SuS*).

A *Figura 4.1* apresenta a visão física do sistema. O *MSuS* sincroniza com o *SuS* através de *Web services*, que este último disponibiliza. Aparte desta sincronização, o *SuS* permite também interacção com aplicações externas, que pode ser através da disponibilização de *Web services* ou consulta destes.

Entre os sistemas externos, pode ser evidenciada a aplicação *GIC - Gestão Integrada de Contra-Ordenações* (para mais informações sobre esta aplicação consultar [dSdI06]), desenvolvida no *PSI*. É também objectivo possibilitar a sincronização com esta aplicação,

no entanto, o *SuS/MSuS* não deverá ficar preso a essa sincronização. O *GIC* é apenas mais uma aplicação externa com que estes sistemas podem trocar informação.

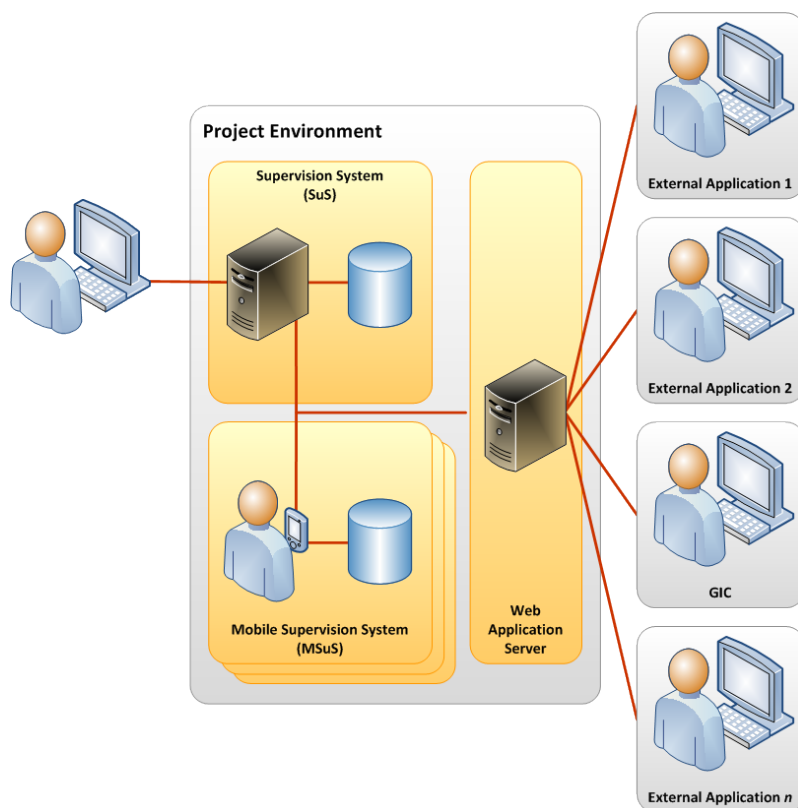


Figura 4.1: Visão Física do Sistema.

4.1.2 Visão Conceptual

A *Figura 4.2* representa a visão conceptual pensada para os Sistemas de Fiscalização e para a forma como estes interagem.

Primeiro que tudo, é necessário ter em atenção que a forma como a Aplicação Externa (*External Application*) é organizada não é do âmbito do projecto, no entanto foi incluída para que exista um visão global das interacções possíveis.

A troca de informação entre os vários sistemas é realizada através de *Web services* (representados com conexões azuis na *Figura 4.2*). Os módulos denominados por *Responder* disponibilizam serviços, e os denominados por *Querist* consultam esses serviços disponibilizados. Estes módulos são incorporados na aplicação para que seja possível a interacção entre o *SuS/MSuS* e aplicações externas. Os desenvolvedores das aplicações externas têm de se preocupar em implementar os mecanismos de consulta de serviços do *SuS*.

Solução Apresentada

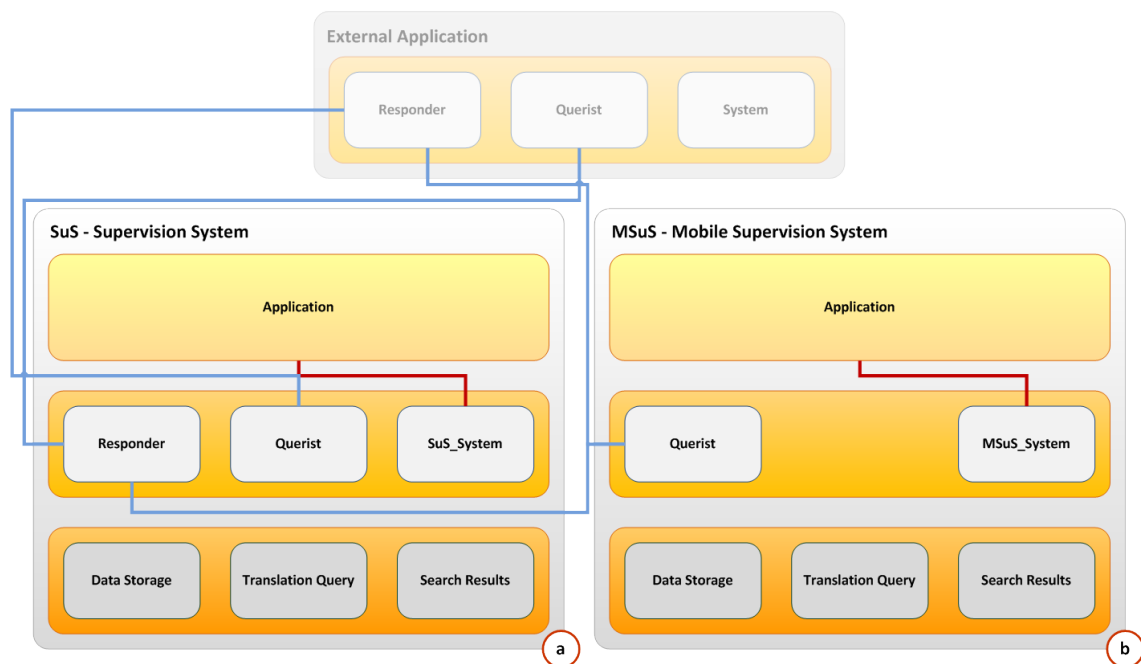


Figura 4.2: Camadas de Software.

De seguida é explicado cada um dos dois sistemas apresentados na *Figura 4.2*.

4.1.2.1 SuS - Supervision System

Em relação ao *SuS* (representado na *Figura 4.2 (a)*) foram definidas três camadas de software:

- *Camada da Aplicação* (camada superior) - Interface de interacção com o utilizador, como é o caso do Administrador do Sistema de Fiscalização, através do qual são disponibilizados os serviços do *SuS*;
- *Camada de Regras ou da Lógica de Negócio* (segunda camada) - inclui todos os mecanismos de interacção com aplicações externas e com o *MSuS* (*Responder/Querist*), e também o sistema responsável pela gestão dos serviços disponibilizados (denominado por *SuS_System*);
- *Camada de Acesso a Dados* (camada inferior) - inclui todos os mecanismos de *CRUD* (*Create, Read, Update and Delete*) do armazenamento de dados (são acedidos através do *SuS_System*). Nesta camada existem o módulo de armazenamento de dados, *Data Storage*, o módulo responsável pelas tradução dos pedidos do sistema para a linguagem do módulo anterior, *Translation Query*, e por fim o módulo responsável pelo tratamento dos resultados, *Search Results*.

De notar que a *Camada de Aplicação* apenas tem acesso aos serviços disponibilizados pelo *SuS* (representado na *Figura 4.2* pela conexão vermelha).

4.1.2.2 MSuS - Mobile Supervision System

Em relação à estrutura em camadas do *MSuS* a ideia é a mesma que no *SuS*. No entanto, existem diferenças no que diz respeito aos *Web services*. Enquanto que o *SuS* disponibiliza e utiliza *Web services* disponibilizados por aplicações externas, o *MSuS* apenas os utiliza, como ilustra a *Figura 4.2 (b)*.

4.1.3 Visão Estática

4.1.3.1 SuS - Supervision System

No *SuS*, *SuS_System* é a classe principal. Esta tem a responsabilidade de receber e interpretar pedidos provenientes da camada de aplicação (como apresentado na *Figura 4.2 (a)*). Para além de tal responsabilidade, também é através desta classe que são disponibilizados *Web services*, assim como mecanismos de consulta de *Web services* disponibilizados por aplicações externas.

A interface *IDataStorage* representa os dados guardados do sistema. Sendo assim, a sua funcionalidade é providenciar os mecanismos de *CRUD* do armazenamento de dados, através do método *readData*, responsável pela leitura de dados, e do método *cudData*, responsável pelas outras acções, ou seja, pela manipulação de dados.

As classes do tipo *Responder* disponibilizam *Web services* para que aplicações exteriores possam interagir com o *SuS*. A disponibilização de serviços segue uma estrutura definida através do método *execute*, recebendo *IQuestion* (explicado na *Secção 4.1.3.4 Perguntas/Respostas*).

Por fim, a classe *Querist* possibilita o acesso a *Web services* disponibilizados por aplicações externas. A forma como tal interacção é realizada terá de ser implementada através do método *execute*, que deverá responder às necessidades específicas de cada *Web service*. As classes definidas para o *SuS* são apresentadas na *Figura 4.3*.

Solução Apresentada

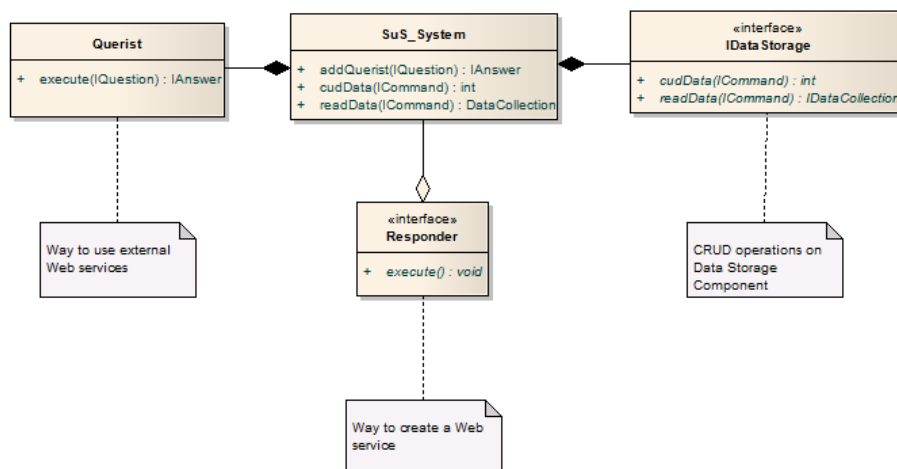


Figura 4.3: Visão Geral das Classes do *SuS*.

4.1.3.2 MSuS - Mobile Supervision System

O *MSuS* é idêntico ao *SuS* apresentado na secção anterior. A única excepção é a ausência dos mecanismos de disponibilização de *Web services*. Tal alteração pode ser constatada na *Figura 4.4*.

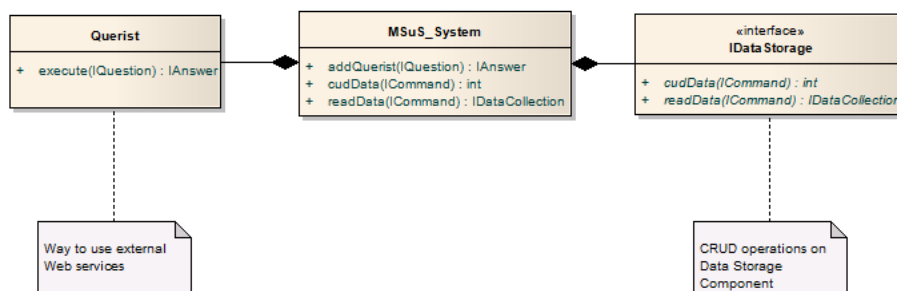


Figura 4.4: Visão Geral das Classes do *MSuS*.

4.1.3.3 Comandos

A abstracção do conceito de Comandos - interface *ICommand* - aparece devido à necessidade de generalizar a forma de criação de acções sobre o armazenamento de dados, tirando esse esforço da interface *IStorage* (ver *Figura 4.2*). Esta interface disponibiliza o método *GetQueryString()* que permite a conversão de pedidos do sistema para a linguagem do módulo de armazenamento de dados. Esta informação está representada na *Figura 4.5* .

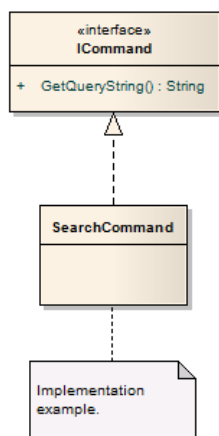


Figura 4.5: Visão das Classes que constituem a definição de Comando.

À partida são criados mecanismos que possibilitam operações *CRUD* que são, basicamente, especializações da interface *ICommand* .

4.1.3.4 Perguntas/Respostas

Tal como no conceito de Comandos, as Perguntas, Respostas e Dados retirados do armazenamento de dados aparecem como interfaces - *IQuestion* , *IAnswer* e *IDataCollection* - para que exista uma generalização deste conceito, tirando o esforço, quer da camada de aplicação, quer das classes *Querist* e *IDataCollection* (ver *Figura 4.2*). A *Figura 4.6* ilustra estes mecanismos. No caso do *IQuestion* é disponibilizado o método *execute* responsável por realizar o pedido.

Solução Apresentada

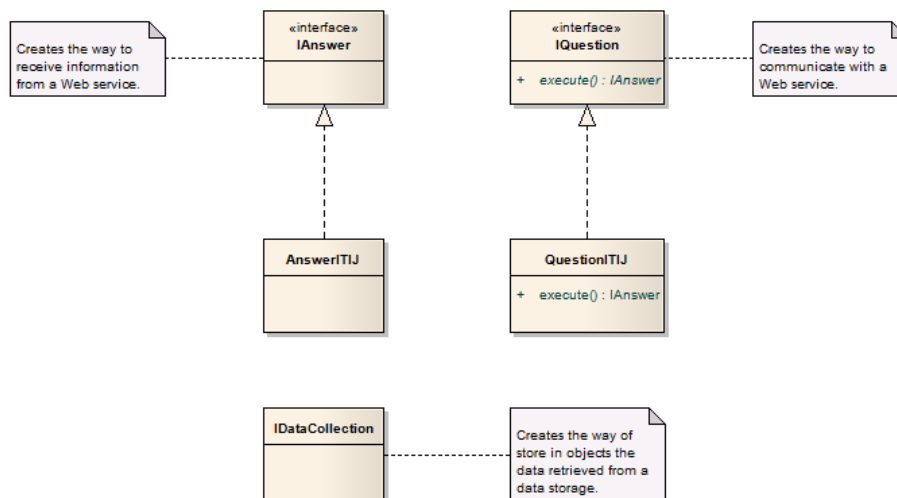


Figura 4.6: Visão das Classes que constituem a definição de Pergunta/Resposta a Web services e de Dados.

4.1.4 Visão Dinâmica

O comportamento dinâmico dos sistemas abordados pelo projecto é apresentado nesta secção, recorrendo-se a diagramas de sequência. Estes diagramas espelham exemplos de utilização do sistema.

4.1.4.1 Interação com Armazenamento de Dados

Quando se fala da interacção com o armazenamento de dados, a questão recai sobre quais os mecanismos disponibilizados para responder aos métodos de *CRUD*. Na arquitectura proposta, existem duas formas de interacção com este armazenamento:

- *Manipulação de Dados* - através do método `cudData(ICommand):int` da interface *IDataStorage* (ver Figura 4.3 e 4.4);
- *Consulta de Dados* - através do método `rData(ICommand):IDataCollection` da mesma interface.

Nas Figuras 4.7 e 4.8 está representado o *SuS_System*, no entanto, as operações aí apresentadas, que serão explicadas de seguida, são também válidas para o *MSuS_System*.

O exemplo ilustrado na Figura 4.7 representa a inserção de um novo terminal no armazenamento de dados. Para tal, a camada de aplicação terá de criar uma instância de *ICommand* que reflecta esse pedido para depois pedir ao *SuS_System* para interagir com a implementação do *IDataStorage* através do método `cudData(ICommand):int`. Este

método pode por alguma razão falhar. E por isso é transmitida se a interação foi bem sucedida ou não.

No caso do segundo exemplo, ilustrado na *Figura 4.8*, o processo desenrola-se da mesma forma. No entanto, para pedir informações sobre os terminais (caso retratado neste exemplo) o *ICommand* é diferente do necessário no exemplo anterior, assim como o pedido realizado ao *SuS_System* pela *Camada de Aplicação*. O método chamado *rData(ICommand):IDataCollection* retorna a coleção de dados pedida. No entanto, em caso de erro, a classe retornada contém mecanismos de comunicação que informam dessa ocorrência à camada de aplicação.

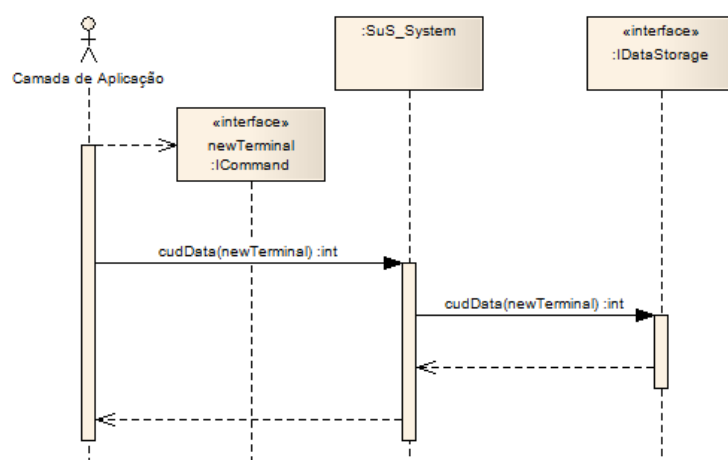


Figura 4.7: Exemplo de inserção de dados.

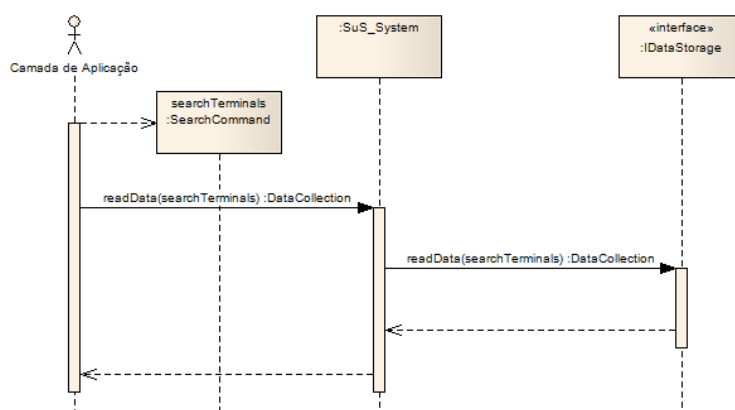


Figura 4.8: Exemplo de leitura de dados.

4.1.4.2 Interação entre Sistemas

Um pedido de informação por parte do *MSuS* ao *SuS* através de um Web service é ilustrado na *Figura 4.9*. Tal pedido provém da camada de aplicação. Para tal, uma instância da interface *IQuestion* terá de ser criada. Após tal procedimento, a camada de aplicação pede ao *MSuS_System* para realizar a execução do pedido. Para que isto seja possível, o sistema adiciona um *Querist* com a *IQuestion* criada (*infringementQuestion*, que tem por objectivo pedir ao *SuS* dados sobre uma infracção). *Querist* ao executar tal pedido acede a Web services disponibilizados recebendo a resposta esperada e criando a resposta que será devolvida à camada da aplicação (contém mecanismos para comunicar ocorrências de erro). Esta primeira sequência é válida para pedidos a quaisquer Web services, quer sejam disponibilizados pelo *SuS*, quer por aplicações externas, sendo também válidas no caso da camada de aplicação ser parte integrante do *SuS* e não do *MSuS* (neste caso o *MSuS_System* é trocado pelo *SuS_System*).

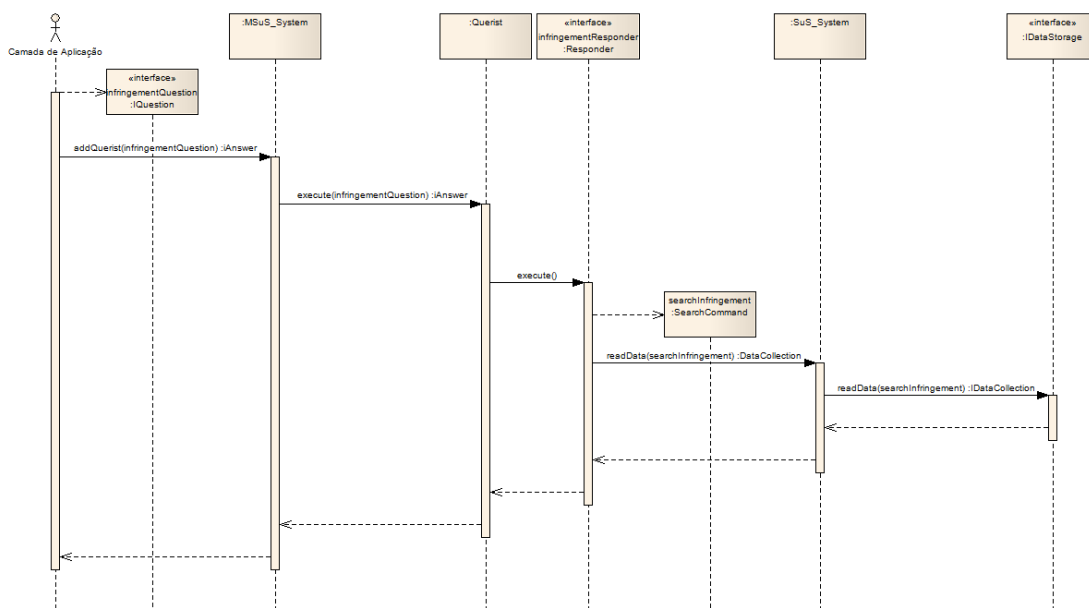


Figura 4.9: Exemplo de interacção entre sistemas.

Do lado do *SuS*, o *Responder* recebe os pedidos provenientes do exterior e executa os procedimentos necessários para responder. Estes mecanismos estão explicados na *Secção 4.1.4.1 Interação com Armazenamento de Dados*.

4.1.5 Visão de Dados

Nesta secção é apresentada a forma como será guardada a informação dos sistemas. Para tal, serão utilizados diagramas de classes para que os dados necessários sejam mais facilmente entendidos e, também, a forma como estes se relacionam.

Para uma melhor compreensão, o diagrama de classes foi dividido em partes que são apresentadas seguidamente.

4.1.5.1 Membros

Sendo a solução a desenvolver centrada na mobilidade do trabalho, primeiro que tudo, é necessário criar mecanismos de armazenamento dos dados dos membros - qualquer utilizador da solução. Para tal, a informação geral sobre estes, é guardada sob a forma da classe *Membro*, na medida em que não é necessário guardar dados como o número do bilhete de identidade, entre outros (afinal é uma solução com vista a agilizar a forma de trabalho e não para gerir recursos humanos). Cada membro terá associado a si um *Perfil* - também visto como uma classe - que lhe associará as permissões que terá sobre o sistema. As diversas permissões (escrita, leitura, entre outras) dão origem à classe *Permissao*. Para além da informação associada ao *Membro* até agora descrita, também serão associadas a sua *Categoria* profissional e a *Unidade_Organica* a que pertence.

Cada *Membro* pode pertencer a uma ou várias equipas (classe *Equipa*), cujo objectivo da sua existência será explicado na *Secção 4.1.5.3 Alocação de Trabalho*.

Foi criado o conceito de *Login* para que os dados do Membro sejam separados dos mecanismos de autenticação no sistema.

Toda esta informação está representada na *Figura 4.10*, onde se pode visualizar com maior detalhe quais os atributos associados a cada classe.

4.1.5.2 Ocorrências

Para a solução a desenvolver, a parte mais crucial é a forma como serão armazenadas as ocorrências reportadas pelos *Agentes Fiscalizadores*. Assim sendo, a classe *Ocorrenci*a aparece naturalmente no modelo de dados proposto, sendo caracterizada pela data e local da sua detecção por parte do *Agente Fiscalizador*, e por determinadas características que dependem do tipo da ocorrência em questão. Logo, a classe *Tipo_Ocorrenci*a guarda a informação necessária para ser possível identificar quais as características (representadas pela classe *Caracteristica*) específicas a preencher aquando da detecção de tal ocorrência.

As características podem ser de variados tipos, com texto e números; mas no caso de serem ficheiros uma nova classe foi criada para tratar de tal possibilidade - classe *Ficheiro*.

Solução Apresentada

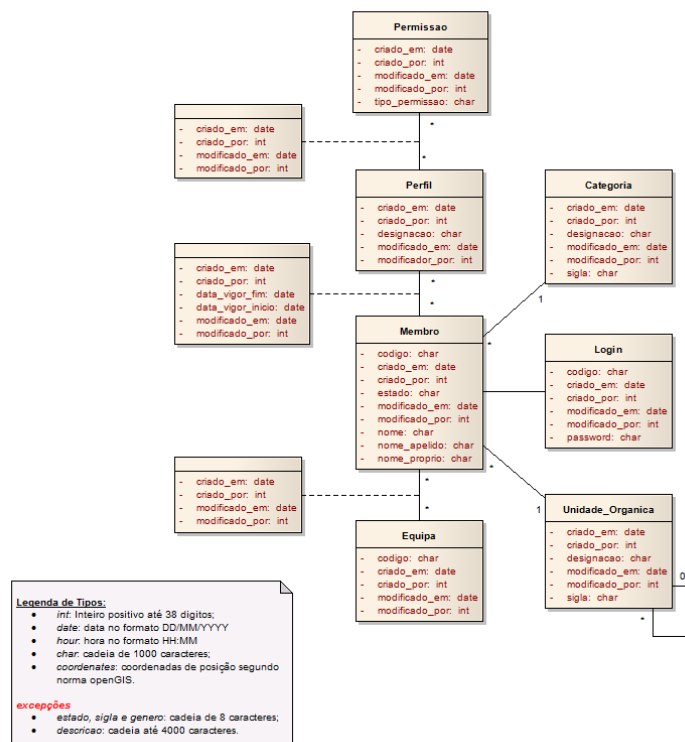


Figura 4.10: Visão de Dados - Membros.

Também foi criada a classe *Lista* para possibilitar a criação de listas de selecção para o valor de uma característica.

Foi criada a classe *Area* para facilitar a procura de tipos de ocorrência. Esta divide os tipos de ocorrência em conjuntos. A classe *Conhecimento* cria uma forma de disponibilização de informação sobre o método de reporte de um *Tipo_Ocorrência* ou um conjunto destas, uma *Area*. Por fim, a ligação destas áreas ao perfil do membro aparece para criar as permissões sobre estas.

Toda esta informação está representada na *Figura 4.11*, onde se pode visualizar com maior detalhe quais os atributos associados a cada classe.

Solução Apresentada

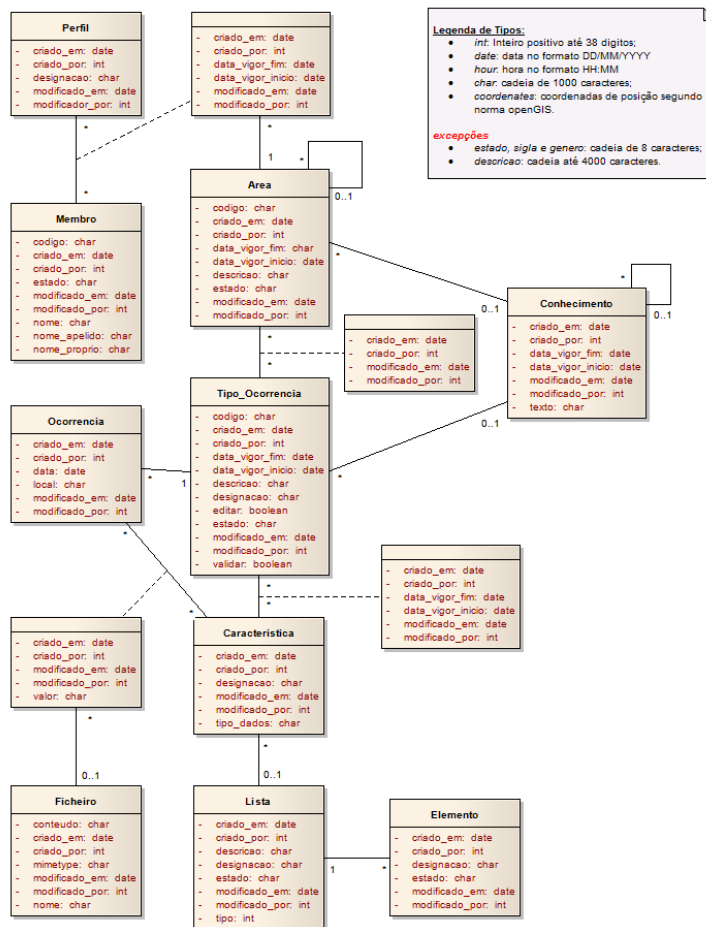


Figura 4.11: Visão de Dados - Ocorrências.

4.1.5.3 Alocação de Trabalho

A existência de equipas justifica-se pela necessidade da alocação de um ou mais membros a zonas do município. Uma equipa pode ser alocada (através da classe *Alocacao*) a uma zona do município e um membro a uma actividade (classe *Actividade*).

Uma actividade é composta por um conjunto de tarefas (classe *Tarefa*), que dirão se a primeira foi concluída com sucesso ou não. Uma tarefa, para um melhor aproveitamento de informação, é dividida em tipos de tarefas (classe *Tipo_Tarefa*).

Uma actividade é concluída tendo em consideração o seu atributo *genero*. Este pode assumir uma de três situações:

- *Todos*

Todos os membros associados à actividade têm de a ter terminado. O atributo *situacao* na classe associação do membro à actividade transmite este dado;

- *Um*

Quando esta actividade foi de facto completada com sucesso por um dos membros associados;

- *Cancelada*

Quando a actividade for cancelada por um membro superior, neste caso um *Pivot*.

Toda esta informação está representada na *Figura 4.12*, onde se pode visualizar com maior detalhe quais os atributos associados a cada classe.

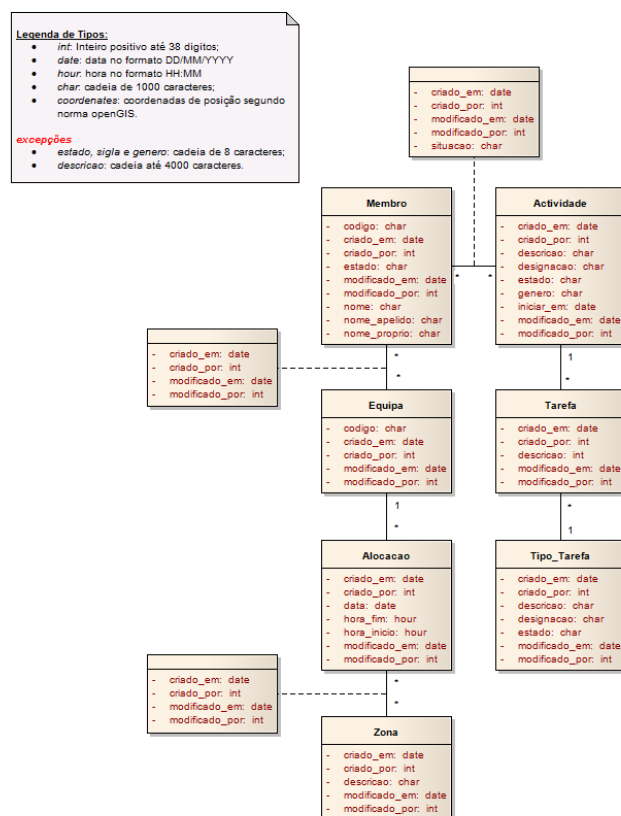


Figura 4.12: Visão de Dados - Alocação de Trabalho.

4.1.5.4 Localização

Toda a informação contida na solução pretende-se que seja localizada no município. Para tal foram criadas as classes de topónimos e números de polícia organizados em zonas do município. A localização não fica por aqui. É pedida a integração do geo-posicionamento. Assim sendo, uma ocorrência para além de estar associada a um topónimo e número de polícia, também estará a uma localização geo-referenciada, dada pela classe *Localização*. Também as tarefas estarão associadas a uma localização geo-referenciada,

Solução Apresentada

enquanto que uma zona será definida por um polígono geo-referenciado. A convenção OpenGIS [OGC09] permite organizar esta informação num só atributo, neste caso é o atributo *coordenadas* na classe *Localização*.

Toda esta informação está representada na *Figura 4.13*, onde se encontra detalhado em mais pormenor os atributos associados a cada classe.

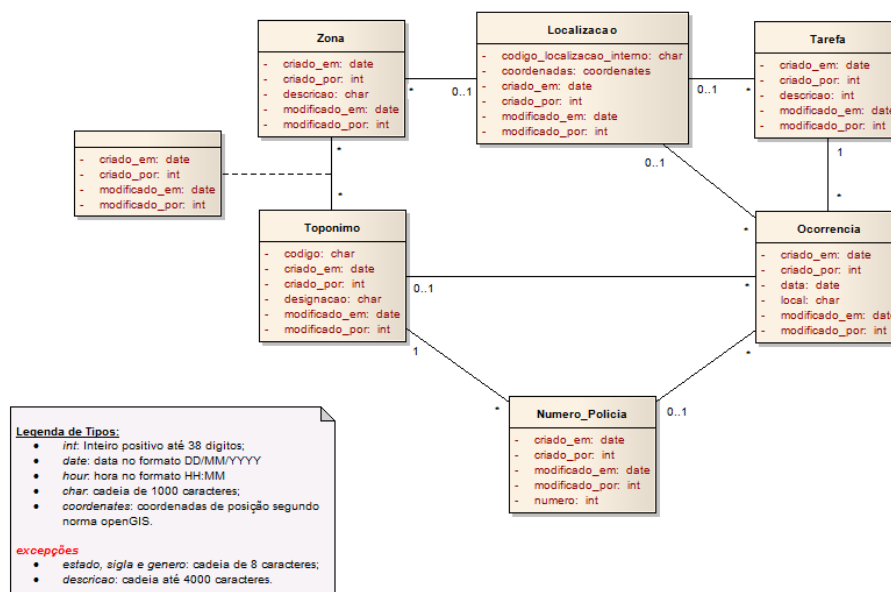


Figura 4.13: Visão de Dados - Localização.

4.1.6 Testabilidade da Arquitectura

Como qualquer boa solução, a aqui apresentada necessita ser testável, ou seja, a arquitectura deverá permitir um teste facilitado ao bom funcionamento das várias partes que a compõem, bem como às aplicações que a utilizam.

Tendo em conta as especificidades do projecto alguns problemas no teste da solução poderão aparecer. Ao proceder à fase de testes dever-se-á ter em conta as seguintes limitações:

- **Dispositivos de Teste Limitados**

Estão disponíveis para ser utilizados em testes, um *PDA* e uma impressora para ligação a este, tendo também em consideração a possibilidade de utilização do emulador de dispositivos móveis da *Microsoft*. Os testes a grande escala só poderão ser realizados na altura de testes no terreno;

- **Tempo Necessário para um Teste em Ambiente Real**

Devido a ser uma aplicação com a vertente móvel, os testes no terreno poderão demorar mais tempo do que o normal e ser mais complicado proceder às correcções necessárias. Será complicado recriar as condições de erro em alguns casos (principalmente aquando da integração com o geo-posicionamento);

- **Complexidade de Dados**

A complexidade dos dados em questão torna complicada a introdução manual de dados para a criação de testes.

Tendo em consideração as figuras na *Secção 4.1.3 Visão Estática* torna-se óbvio que os testes da arquitectura apresentada centrar-se-ão na escrita e leitura de dados e para tal as *Interfaces* podem servir de plataformas de criação dos testes necessários para validar esta solução.

Um exemplo para um possível teste à arquitectura é a criação de uma implementação do *IDataStorage* para ligação a uma base de dados *Oracle*, como é demonstrado na *Figura 4.14*.

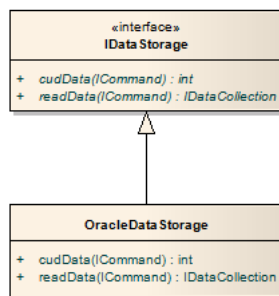


Figura 4.14: Exemplo de Teste ao IDataStorage.

4.2 Planeamento do Desenvolvimento

O desenvolvimento da solução foi dividido nas seguintes fases:

- **Fase F01**
Desenvolvimento da Arquitectura da Solução;
- **Fase F02**
Desenvolvimento do Protótipo do Reporte de Ocorrências, a nível do dispositivo móvel;
- **Fase F03**
Desenvolvimento dos Requisitos de Prioridade Essencial e Elevada;
- **Fase F04**
Desenvolvimento dos Requisitos de Prioridade Média;
- **Fase F05**
Desenvolvimento dos Requisitos de Prioridade Baixa;
- **Fase F06**
Fase de Testes Finais e Testes no Terreno.

Todas estas fases são seguidas de testes, com a excepção da última fase. No âmbito deste projecto apenas as fases F01 e F02 serão abordadas, com o seguinte planeamento:

- **Semanas de 20 e 27 de Abril - Fase F01;**
- **Semanas de 27 de Abril, de 04, 11, 18 e 25 de Maio e de 01 e 08 de Junho - Fase F02.**

Solução Apresentada

A informação apresentada até ao momento está sintetizada na *Figura 4.15*.

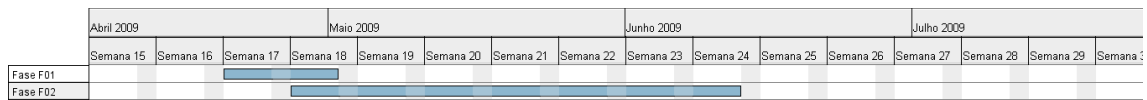


Figura 4.15: Planeamento do Desenvolvimento.

Capítulo 5

Desenvolvimento e Validação

Na presente secção é abordada a implementação da arquitectura, a nível do dispositivo (foco central no âmbito do projecto de final do curso), assim como o desenvolvimento de um protótipo. O objectivo do protótipo, como já referido anteriormente, é validar a solução proposta pelo aluno, mostrando algumas das funcionalidades pedidas numa possível interface de utilização.

Em relação à arquitectura da solução, toda ela foi implementada com sucesso, com a excepção do Sistema de *Back-Office*, o *SuS*, em que foi planeada a sua criação numa fase posterior, não fazendo parte do âmbito do projecto do aluno. Durante a presente secção são, então, apresentados os seguintes elementos constituintes da arquitectura:

- **Implementação da Arquitectura para o dispositivo móvel;**

Utilização da ferramenta *Microsoft Visual Studio 2008* para a criação da arquitectura na linguagem de programação *C#*. Para mais pormenores ver *Secção 4.1.3.2 MSuS - Mobile Supervision System*;

- **Criação do Repositório de Base de Dados**

Criação da base de dados na tecnologia *Oracle Database* que irá servir de repositório de toda a informação que flui no sistema. Este servirá de ponte de informação para os dispositivos móveis. Para mais pormenores ver *Secção 4.1.5 Visão de Dados*.

- **Sincronização com o dispositivo móvel**

A sincronização com o dispositivo móvel é efectuada recorrendo às ferramentas englobadas no software da *Oracle*, o *Oracle Database Lite*. De notar que os mecanismos de Web services para possibilitar a utilização de outras tecnologias também foi criado, como é explicado na *Secção 4.1.3 Visão Estática*. No entanto, a sincronização no *Oracle Database Lite* já é possibilitada através dos seus próprios mecanismos.

Em relação ao desenvolvimento do protótipo, este foi desenvolvido sobre a arquitectura implementada. O protótipo consiste na criação de uma camada de aplicação para a utilização das outras apresentadas na *Secção 4.1.2 Visão Conceptual*, neste caso referente ao *MSuS*.

5.1 Arquitectura

5.1.1 Implementação para Dispositivos Móveis

Foi concluída com sucesso a implementação da visão estática da arquitectura na versão para o *MSuS* (ver *Secção 4.1.3.2 MSuS - Mobile Supervision System*). Como esperado, a arquitectura foi de fácil implementação e, decompondo-a nas camadas referidas na *Secção 4.1.2 Visão Conceptual*, a sua utilização é simples. A *Figura 5.1* mostra qual a estrutura final. A arquitectura foi dividida nas camadas esperadas, a *Camada de Acesso a Dados*, a *de Lógica de Negócio* e a *de Aplicação* (que será o foco do protótipo apresentado mais à frente). De notar que a arquitectura já foi extendida no que se refere à ligação à base de dados local (neste caso o *Oracle Database Lite*) e no que se refere aos comandos de interacção com esta (possibilita instruções *CRUD* sobre qualquer armazenamento de dados que possibilite a utilização de instruções *SQL*).

Toda a arquitectura foi pensada para que qualquer que fosse a informação que flui no sistema este funcione, logo, nesta arquitectura não existem elementos que fechem a sua utilização ao reporte de ocorrências a nível de um Município, podendo ser aplicada a qualquer outro ramo que necessite de reporte de ocorrências (por exemplo no ramo da vinicultura).

5.1.1.1 Exemplos de Implementação

Nesta secção são apresentadas algumas das partes implementadas consideradas mais relevantes.

MSuS.System

Toda a fluência de informação no *MSuS* passa pela classe *MSuS.System* que é a responsável por gerir todos os pedidos que chegam da *Camada de Aplicação*. Esta classe possibilita a interacção com o exterior, através da criação de um *Querist* (mecanismo de acesso a Web services disponibilizados por aplicações externas), e também a interacção com o armazenamento de dados, através da criação de instruções *CRUD* à *Camada de Acesso a Dados*.

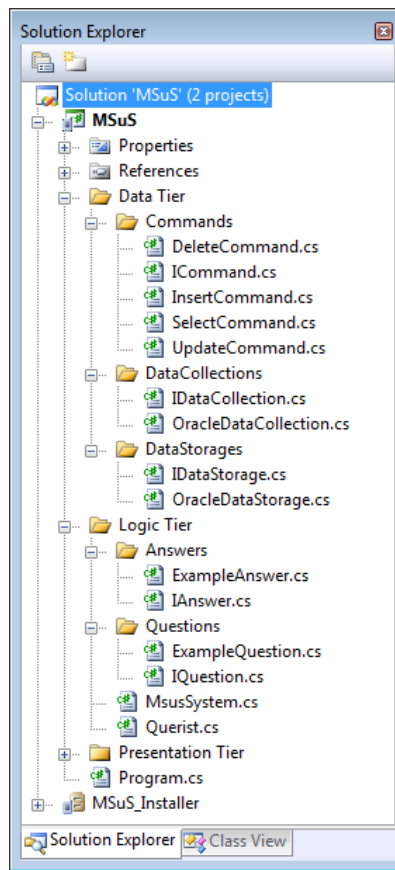


Figura 5.1: Organização da Arquitectura no Microsoft Visual Studio.

O código referente a esta classe é o seguinte:

```

1 namespace MSuS.Logic_Tier
2 {
3     /// ...
4     class MsusSystem
5     {
6         private IDataStorage _dataStorage;
7         public MsusSystem(String username, String password)
8         {
9             _dataStorage = (IDataStorage)new OracleDataStorage
10                             (username, password);
11         }
12         /// ...
13         public IAnswer AddQuerist(IQuestion question)
14         {
15             Querist newQuerist = new Querist(question);
16             IAnswer newAnswer = newQuerist.Execute();
17             return newAnswer;

```

```

18     }
19     /// ...
20     public int CudData(ICommand command)
21     {
22         return _dataStorage.CudData(command);
23     }
24     /// ...
25     public ICollection ReadData(ICommand command)
26     {
27         return _dataStorage.ReadData(command);
28     }
29 }
30 }

```

Para a criação de um *Querist* o método *AddQuerist* (linha 13) é executado, enquanto que para a leitura e actualização de dados do armazenamento de dados associado são executados os métodos *CudData* (linha 20) e *ReadData* (linha 25), respectivamente.

IDataStorage

A ligação à base de dados é realizada através da criação de mecanismos que possibilitem acções *CRUD*. Para tal, a interface *IDataStorage* foi criada possibilitando a extensibilidade para a utilização de diferentes tipos de armazenamento de dados com o *MSuS*. Neste caso, foi criada uma classe, *OracleDataStorage*, que implementa esta interface para a comunicação com uma base de dados da tecnologia *Oracle*.

O código da interface é o seguinte:

```

1 namespace MSuS.Data_Tier.DataStorages
2 {
3     /// ...
4     interface IDataStorage
5     {
6         /// ...
7         int CudData(ICommand command);
8         /// ...
9         ICollection ReadData(ICommand command);
10    }
11 }

```

Para a leitura e actualização dos dados de um armazenamento de dados são disponibilizados os métodos *CudData* (linha 20) e *ReadData* (linha 25) respectivamente.

O código a seguir apresentado, representa a implementação para a base de dados *Oracle*:

```

1  namespace MSuS.Data_Tier.DataStorages
2  {
3      class OracleDataStorage : IDataStorage
4      {
5          private OracleConnection _connection;
6          private String _username;
7          private String _password;
8
9          public OracleDataStorage(String username, String password)
10         {
11             _username = username;
12             _password = password;
13             String connectionString = "dsn=mobilesus;uid=system;pwd="
14                                     + _password;
15             _connection = new OracleConnection(connectionString);
16             _connection.Open();
17         }
18
19         public int CudData(ICommand command)
20         {
21             OracleCommand oc = new OracleCommand
22                                     (command.GetQueryString(), _connection);
23             try
24             {
25                 oc.ExecuteNonQuery();
26             }
27             catch (Exception)
28             {
29                 return 0;
30             }
31             return 1;
32         }
33
34         public IDataCollection ReadData(ICommand command)
35         {
36             OracleDataCollection odc = new OracleDataCollection();
37             OracleCommand oc = new OracleCommand
38                                     (command.GetQueryString(), _connection);
39             odc.DataReader = (OracleDataReader)oc.ExecuteReader();
40             return (IDataCollection)odc;
41         }
42     }
43 }

```

5.1.2 Base de Dados

A base de dados que servirá de repositório da solução apresentada foi criada com sucesso. Para tal, alguns passos tiveram de ser tomados para a criação de uma solução que irá sincronizar com dispositivos móveis. Como apresentado nas tecnologias a utilizar (ver *Secção 3.2 Revisão Tecnológica*), o software escolhido para o armazenamento de dados é o da *Oracle*, o *Oracle Database*.

Recorrendo à ferramenta *Oracle Designer* [Cor09k], que providencia, entre outras funcionalidades, a possibilidade de desenhar a base de dados para posterior geração, foi criada a base de dados pretendida. A *Figura 5.2* apresenta o ambiente desta ferramenta.

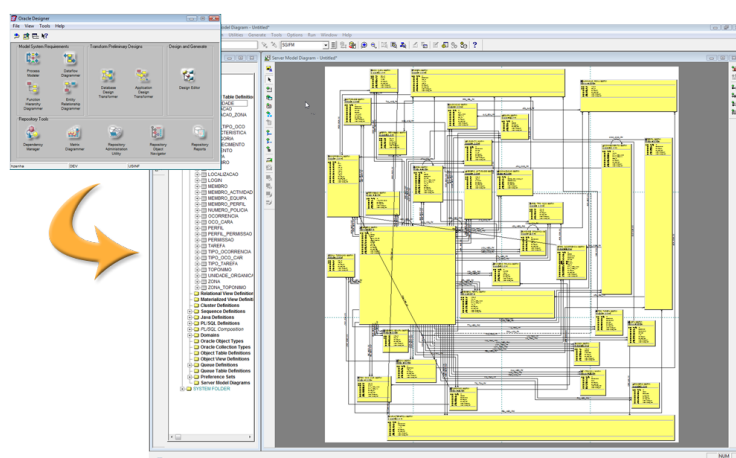


Figura 5.2: Aspecto do Oracle Designer.

5.1.2.1 Sincronização com Dispositivos Móveis

Após a geração da base de dados que serve de suporte ao protótipo e, no futuro, à solução a implementar, é necessária a criação dos mecanismos de sincronização com os dispositivos móveis a utilizarem tais aplicações. Para tal, a tecnologia da *Oracle*, o *Oracle Database Lite*, disponibiliza todo um leque de aplicações que permitem uma rápida configuração da informação a disponibilizar e a sincronizar com estes dispositivos. É neste contexto que aparece a utilização do *Mobile Database Workbench*. Com esta ferramenta, parte integrante do *Oracle Database Lite*, é possível criar uma base de dados baseada no repositório onde foi criada a apresentada anteriormente, podendo dar privilégios de actualização ou não aos dispositivos que a utilizarão.

A *Figura 5.3* apresenta a utilização deste ferramenta.

É possível reparar na *Figura 5.3* (tabela do lado esquerdo, campo *Updatability*) que alguns dos itens publicados podem ser actualizados nos dispositivos móveis, enquanto outros não. Estas diferenças fazem sentido se se pensar que um *Agente Fiscalizador*

Desenvolvimento e Validação

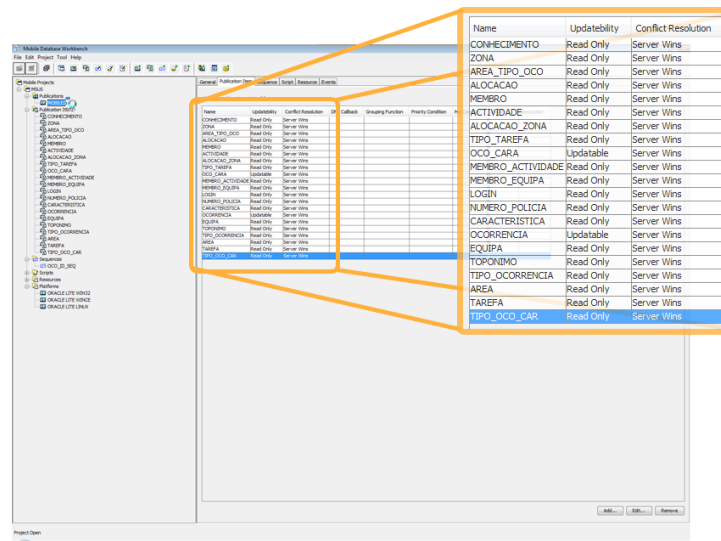


Figura 5.3: Aspecto do Oracle Mobile Database Workbench.

apenas reporta ocorrências, logo não faz parte do seu trabalho criar, por exemplo, novos tipos de ocorrência através da utilização do seu dispositivo móvel.

Após a criação desta base de dados a sincronizar com os dispositivos móveis, é necessário, precisamente, sincronizar esta nova informação. Tal é possível, após a instalação nos vários dispositivos móveis do *Oracle Database Lite* na sua versão cliente, recorrendo ao mecanismo de sincronização da *Oracle*, o *mSync* (futuramente será integrado no *MSuS*). Este mecanismo é de simples utilização; apenas se tem de inserir os dados referentes à máquina onde está a ser executado o servidor para os dispositivos móveis, o *Mobile Server* do *Oracle Database Lite*, e iniciar a sincronização. No final pode-se verificar que a base de dados se encontra no dispositivo móvel (recorrendo ao *MSQL*, também englobado na instalação do *Oracle Database Lite*). A Figura 5.4 mostra esta sequência de passos.

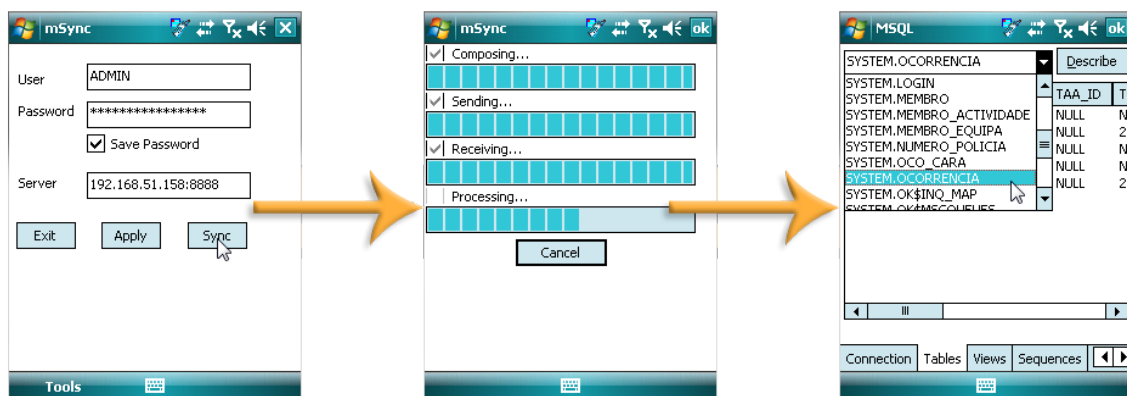


Figura 5.4: Sincronização dos dispositivos móveis.

Tendo a base de dados criada e toda a arquitectura que irá sustentar a aplicação nos dispositivos móveis, foi desenvolvido um protótipo que demonstra como será a utilização do *MSuS* por parte dos *Agentes Fiscalizadores*.

5.2 Protótipo

Como referido anteriormente, um dos objectivos deste projecto era a criação de um protótipo do sistema que corresse nos dispositivos móveis, com principal enfoque na funcionalidade principal, o reporte de ocorrências.

O protótipo criado aborda os seguintes conjuntos de funcionalidades:

- **Reporte de Ocorrências**

Permite ao utilizador reportar novas ocorrências;

- **Gestão do Trabalho**

Permite ao utilizador saber a que equipas pertence e em que zonas do município estas têm de exercer o serviço de fiscalização, assim como, que conjunto de tarefas, actividades, têm de executar enquanto entidade individual;

- **Pesquisa de Informação**

Permite aos seus utilizadores pesquisar informação sobre os tipos de ocorrência que podem reportar e quais as ocorrências já reportadas por estes e que ainda se encontram na base de dados do dispositivo móvel.

A *Figura 5.5* apresenta o ecrã inicial da utilização do *MSuS* no dispositivo móvel e a *Figura 5.6* o menu que, após autenticação, é apresentado ao utilizador.



Figura 5.5: Protótipo: ecrã inicial.



Figura 5.6: Protótipo: Menu principal.

De seguida, cada um destes conjuntos de funcionalidades é explicado com maior pormenor, recorrendo a exemplos de utilização do protótipo.

5.2.1 Reporte de Ocorrências

Quando o utilizador escolhe a *Opção 1* do menu principal (ver *Figura 5.6*) pode reportar uma nova ocorrência. É apresentado, então, ao utilizador a possibilidade de introdução da informação necessária para tal reporte, em quatro passos: Escolha do tipo de ocorrência, Introdução de informação geral, Introdução de informação específica e Confirmação do reporte.

Aquando do primeiro passo, o formulário apresentado é o que é representado na *Figura 5.7*.



Figura 5.7: Protótipo: Reporte - Ambiente de escolha do tipo de ocorrência.

Ao escolher uma *Área* irão aparecendo ao utilizador os *Tipos de Ocorrência* que pode escolher, assim como *Sub-Áreas*. A *Figura 5.8* apresenta duas situações. Na primeira, o utilizador escolhe a área *Via Pública* e tipo de ocorrência *Buraco na Estrada*, enquanto no segundo, ao escolher a área *Urbanismo* possibilitou ao utilizador escolher uma sub-área,

que neste caso foi *Elevadores*, escolhendo então *Elevadores: Falta de Inspeção* como o tipo de ocorrência. No resto da apresentação do *Reporte de Ocorrências*, as figuras que aparecem têm do lado esquerdo o referente à primeira situação e do lado direito o referente à segunda.



Figura 5.8: Protótipo: Reporte - Escolha do tipo de ocorrência.

Após a escolha do tipo de ocorrência, o próximo passo é a introdução das informações gerais da ocorrência a reportar. Estas são a data e o conjunto topónimo/número de polícia, podendo este último par ser substituído pela introdução manual do local. Qualquer que seja o tipo de ocorrência escolhida, a forma de introdução desta informação é a mesma. Na primeira situação foi inserido o local manualmente, enquanto na segunda tal não se verificou, como apresenta a *Figura 5.9*. De notar que, ao inserir manualmente o local, as escolhas sobre o par topónimo/número de polícia são ignoradas.

Após ter preenchido estes valores, o utilizador passa ao próximo passo. Neste terá de inserir informações específicas de cada um dos tipos de ocorrência. Na primeira situação apenas tem de inserir as observações que considera relevantes, enquanto que na segunda situação, para além destas observações, é necessária a identificação da hora da ocorrência, do denunciante e de uma testemunha. A criação deste formulário é dinâmica (podendo resultar em vários formulários, como se verifica na segunda situação), recorrendo para isso às características que estão associadas a um tipo de ocorrência a nível da base de dados. A *Figura 5.10* apresenta a situação aqui descrita.

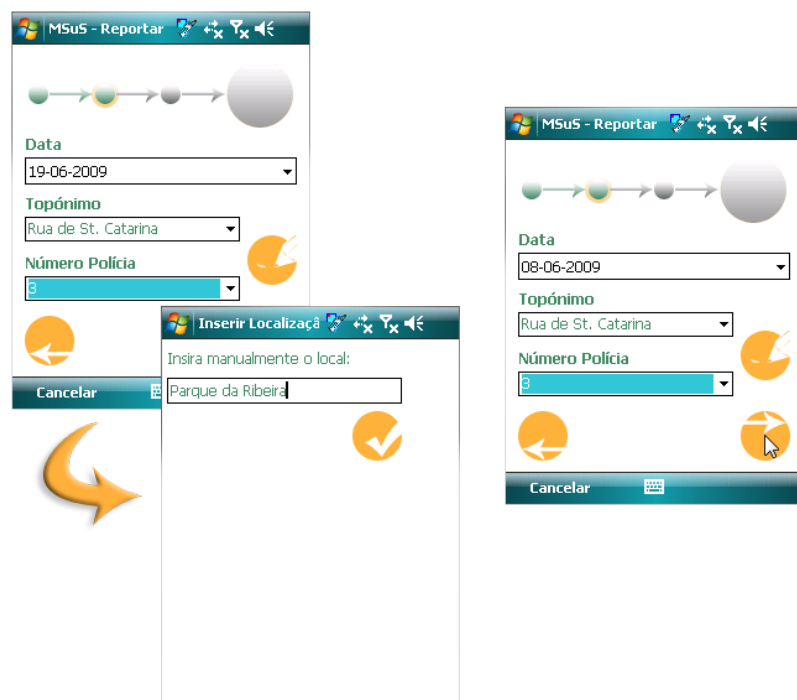


Figura 5.9: Protótipo: Reporte - Introdução de informação geral.

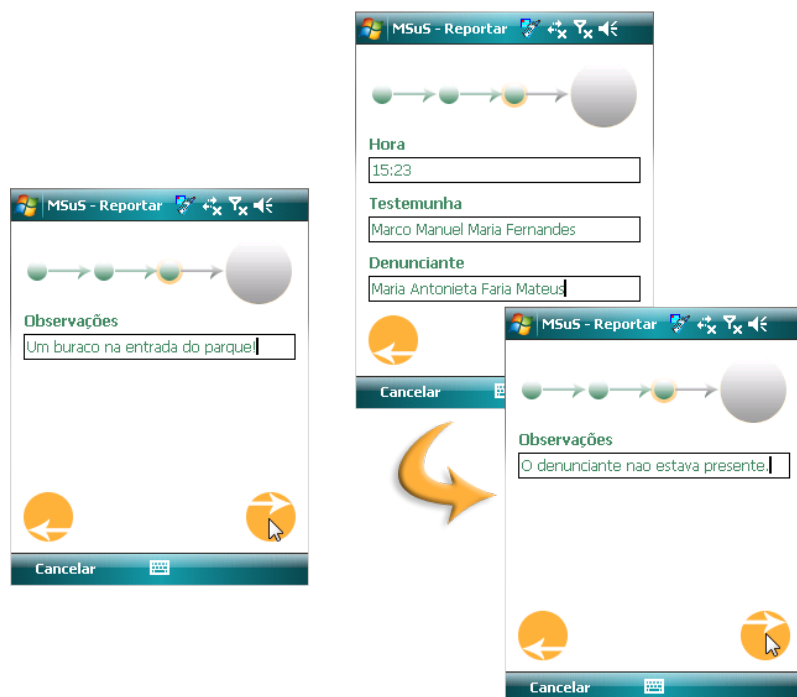


Figura 5.10: Protótipo: Reporte - Introdução de informação específica.

Após a introdução dos dados necessários é possível, no último passo, confirmar o reporte da ocorrência em questão. Esta confirmação está representada na *Figura 5.11*

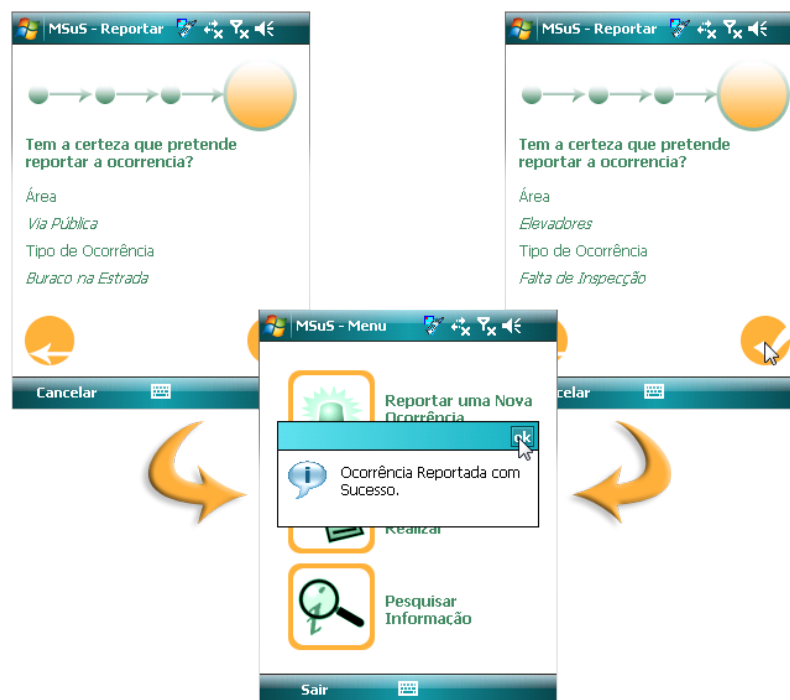


Figura 5.11: Protótipo: Reporte - Introdução de informação específica.

5.2.2 Gestão do Trabalho

Para além do reporte de ocorrências, o utilizador pode escolher a *Opção 2* no menu principal (ver na *Figura 5.6*) que lhe permite ter acesso ao trabalho a realizar durante o dia. A sua actividade divide-se entre as equipas a que pertence e o conjunto de tarefas (actividades) que tem de realizar individualmente. Para aceder a esta informação terá de escolher entre as opções que lhe aparecem no menu seguinte, representado na *Figura 5.12*.



Figura 5.12: Protótipo: Trabalho - Menu.

Ao escolher a *Opção a*, o utilizador tem acesso às equipas a que pertence e quais são

os membros que a constituem. O acesso a esta informação é apresentado pela *Figura 5.13*.



Figura 5.13: Protótipo: Trabalho - Informação sobre equipas.

Ao optar por escolher a *Opção b*, o utilizador tem acesso às várias actividades a ele associadas e respectivas tarefas. Tal informação aparece organizada como mostra a *Figura 5.14*.

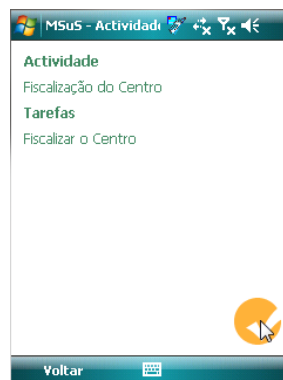


Figura 5.14: Protótipo: Trabalho - Informação sobre actividades pessoais.

5.2.3 Pesquisa de Informação

Por fim, é dada ao utilizador a possibilidade de pesquisar informação, optando este por escolher a Opção 3 do menu principal (ver *Figura 5.6*). Será então apresentado ao utilizador um novo menu, onde este poderá escolher entre ver informação referente aos

tipos de ocorrência que pode reportar e a referente às ocorrências que já reportou. O menu está representado na *Figura 5.15*.



Figura 5.15: Protótipo: Pesquisa - Menu.

O utilizador ao escolher a *Opção a* poderá pesquisar informação referente aos tipos de ocorrência que pode reportar através do seu dispositivo móvel. Para tal, encontra uma forma de pesquisa idêntica aquela com que se depara no reporte de ocorrências já abordado anteriormente na *Secção 5.2.1 Reporte de Ocorrências*. A *Figura 5.16* apresenta essa forma de escolha.



Figura 5.16: Protótipo: Pesquisa - Tipos de Ocorrência.

Após a escolha do tipo de ocorrência, o utilizador tem acesso às informações referentes a este, tendo ainda a possibilidade de aceder a uma série de linhas de orientação para o seu reporte. A *Figura 5.16* demonstra a informação a que o utilizador tem acesso.

Se o utilizador pretender pesquisar informação referente às ocorrências que reportou, e que ainda se encontram no dispositivo móvel, opta pela *Opção b*. Quando esta é acedida o utilizador pode ver a informação geral das ocorrências e, se pretender, a informação específica do tipo de ocorrência a que pertence. Tendo como referências as duas ocorrências

Desenvolvimento e Validação



Figura 5.17: Protótipo: Pesquisa - Tipos de Ocorrência.

registadas como exemplo na *Secção 5.2.1 Reporte de Ocorrências*, a *Figura 5.18* apresenta a informação que foi inserida, sendo que a primeira situação aparece à esquerda e a segunda à direita.

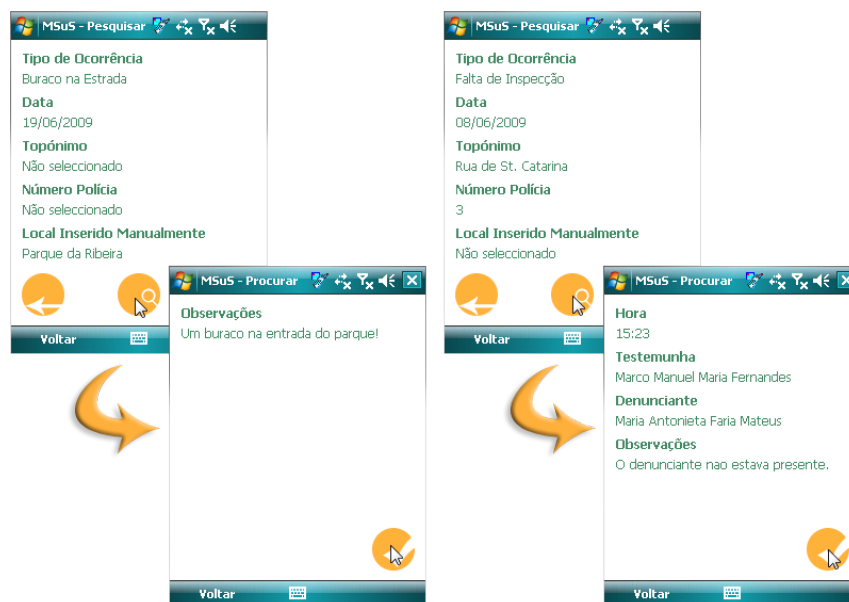


Figura 5.18: Protótipo: Pesquisa - Tipos de Ocorrência.

Ao longo deste capítulo foi apresentada a forma de lidar com as ferramentas necessárias para a criação da solução. Com enfoque na solução a nível dos dispositivos

móveis, foi criado um protótipo que possibilita quer o reporte de ocorrências, quer a pesquisa de informações.

O protótipo revelou ser um bom instrumento de validação da solução proposta. No decorrer do seu desenvolvimento, foram realizados pequenos ajustes à solução apresentada, melhorando-a.

Capítulo 6

Conclusão

No presente capítulo são tecidas conclusões sobre o projecto, como foi o trajecto deste em relação ao planeamento e discutidas quais as perspectivas de desenvolvimento futuro.

6.1 Considerações sobre o Projecto

No âmbito do projecto, foram cumpridos os objectivos estipulados em relação ao projecto em questão. Como referido inicialmente, foi pedido a especificação de requisitos, uma análise tecnológica, a apresentação de uma solução e o desenvolvimento de um protótipo que valide a solução apresentada.

Estes pontos foram sendo apresentados ao longo do presente relatório, culminando no sucesso do projecto *”Aplicação de registo de ocorrências para dispositivos móveis”*. Não esquecer que a solução apresentada será utilizada para a continuação e conclusão do desenvolvimento do *Sistema de Gestão Integrada de Fiscalização Municipal*.

À medida que a solução foi sendo criada, a necessidade que esta pudesse ser de fácil adaptação a um contexto diferente do de um município foi demonstrada por parte do *PSI*. Tal foi concretizado, e o nível de abstracção conseguido permite a utilização desta pelas mais diversas entidades que necessitem de um sistema de reporte de ocorrência. Para tal, apenas a informação que flui no sistema e que é armazenada necessita de ser alterada, algo que será da inteira responsabilidade dos utilizadores do sistema, podendo, porventura, necessitar de adaptação das camadas de aplicação, ponto de interacção com estes utilizadores.

6.1.1 Acordo com o Planeamento

Como referido no início deste relatório, a complexidade do projecto em causa aumentou consideravelmente ao longo do mesmo. Assim sendo, todo o trabalho de especificação de requisitos foi realizado novamente, atrasando todo o processo de criação da solução. Por conseguinte, o planeamento foi sendo reestruturado para que no final do projecto os resultados esperados fossem cumpridos.

Pode evidenciar-se o facto de que a especificação de requisitos e a definição de arquitectura ocuparam grande parte do trabalho realizado. No entanto, à medida que o estudo das tecnologias foi sendo realizado, estas foram sendo aplicadas para que no final não só fosse possível a apresentação de uma solução para o problema da *CMP*, mas também existisse a possibilidade de transmitir segurança a esta entidade e de garantir que esta solução funciona e vai de encontro ao pretendido. A *Figura 6.1* apresenta as diferenças entre o planeamento e a realidade.

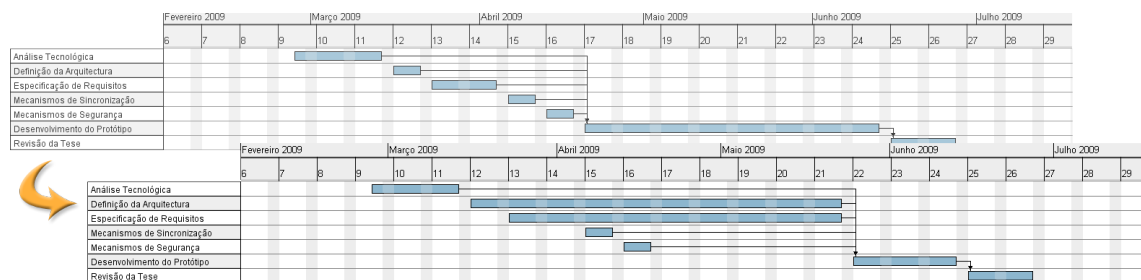


Figura 6.1: Comparação Planeado/Realizado.

6.1.2 Trabalho Futuro

Primeiro que tudo é necessário referir que este projecto vai continuar e o trabalho aqui apresentado é visto como o ponto de partida para um produto que se pensa vir a ter grande sucesso. Estão a ser criadas as condições para que este projecto seja finalizado, apontando o final de 2009 como meta.

O sistema que resultará deste projecto é já alvo de interesse por parte de outras entidades para além da *CMP*, podendo tornar-se uma solução de referência num futuro próximo, na medida que preza pela abstracção da informação que nele fluirá.

6.2 Reflexão

É inegável que cada vez mais se preza a eficiência dos processos no mundo empresarial. Nesta linha de orientação, a *CMP* não podia deixar de procurar aumentar a eficiência dos seus serviços, neste caso a nível dos serviços de fiscalização. Possibilitando o reporte

Conclusão

de ocorrências do mais variado tipo, desde infracções até a um simples reporte de um buraco na estrada, a utilização do sistema apresentado ao longo deste relatório aumentará a qualidade dos serviços prestados por este organismo. Tal aumento traduzir-se-á numa melhoria da rentabilidade económica e também da qualidade de vida dos cidadãos.

Como se está perante um sistema com potencial de comercialização, não se pode deixar de tentar perceber como poderá ser aproveitado para utilização noutros contextos. Seguindo o pedido de abstracção formulado pelo *PSI*, a solução apresentada neste relatório preza tal característica. Tendo em consideração tal facto, a solução poderá ser utilizada no reporte de ocorrências das mais variadas entidades. Assim sendo, esta solução possibilitará a criação de uma linha de produtos que poderá colocar o *PSI* em lugar de destaque no que diz respeito a sistemas de reporte de ocorrências.

Em suma, o primeiro grande passo foi dado na criação de uma solução que satisfaz não só o problema apresentado pela **Câmara Municipal do Porto**, mas também os requisitos do **Projecto de Sistemas de Informação**.

Referências

- [Cen09] .NET Framework Developer Center. .net compact framework, Maio 2009. Microsoft Corporation, disponível em <http://msdn.microsoft.com/en-us/netframework/aa497273.aspx>.
- [Cha04] David A. Chappell. *Enterprise Service Bus*. O'Reilly Media, Inc., 2004.
- [Cor08a] Microsoft Corporation. Sql server integration services with oracle database 10g. In Microsoft Corporation, editor, *SQL Server Technical Article*. Microsoft Corporation, May 2008.
- [Cor08b] Oracle Corporation. Oracle database lite 10gr3 technical white paper. In Oracle Corporation, editor, *An Oracle White Paper*. Oracle Corporation, August 2008.
- [Cor09a] International Business Machines Corp. Ibm - websphere application server community edition, Maio 2009. International Business Machines Corp., disponível em [WebSphereApplicationServerCommunityEdition](http://www.ibm.com/software/websphere/application/ServerCommunityEdition/).
- [Cor09b] International Business Machines Corp. Ibm software - websphere, Maio 2009. International Business Machines Corp., disponível em <http://www-01.ibm.com/software/websphere/>.
- [Cor09c] Borland Software Corporation. Jdatastore developer support, Maio 2009. Borland Software Corporation, disponível em <http://info.borland.com/devsupport/jdatastore/>.
- [Cor09d] Microsoft Corporation. Internet explorer 8, Maio 2009. Microsoft Corporation, disponível em <http://www.microsoft.com/windows/internet-explorer/default.aspx>.
- [Cor09e] Microsoft Corporation. Microsoft sql server compact 3.5, Maio 2009. Microsoft Corporation, disponível em <http://www.microsoft.com/Sqlserver/2005/en/us/compact.aspx>.
- [Cor09f] Microsoft Corporation. Msdn: Microsoft development network, Fevereiro 2009. Microsoft Corporation, disponível em <http://msdn.microsoft.com/en-us/default.aspx>.
- [Cor09g] Microsoft Corporation. The official microsoft silverlight site, Maio 2009. Microsoft Corporation, disponível em <http://silverlight.net/>.

REFERÊNCIAS

- [Cor09h] Microsoft Corporation. Windows virtual pc, Maio 2009. Microsoft Corporation, disponível em <http://www.microsoft.com/windows/virtual-pc/default.aspx>.
- [Cor09i] Oracle Corporation. Oracle and bea systems, Fevereiro 2009. Oracle Corporation, disponível em <http://www.oracle.com/bea/index.html>.
- [Cor09j] Oracle Corporation. Oracle application server 10g standard edition, Fevereiro 2009. Oracle Corporation, disponível em http://www.oracle.com/appserver/standard_edition.html.
- [Cor09k] Oracle Corporation. Oracle designer, Junho 2009. Oracle Corporation, disponível em <http://www.oracle.com/technology/products/designer/index.html>.
- [Cor09l] Oracle Corporation. Oracle weblogic server, Fevereiro 2009. Oracle Corporation, disponível em <http://www.oracle.com/technology/products/weblogic/index.html>.
- [dCeIdCMdP08] Gabinete de Comunicação e Imagem da Câmara Municipal do Porto. *Discurso Directo*. Câmara Municipal do Porto, Dezembro 2008. Disponível em <http://cm-porto.pt/discursodirecto-n1.pdf>.
- [dP09a] Hugo Manuel Zenha de Pinho. Análise tecnológica. In *Aplicação de registo de autos para dispositivos móveis - Relatório de Análise Tecnológica*. Projecto de Sistemas de Software - Faculdade de Engenharia da Universidade do Porto, Março 2009.
- [dP09b] Hugo Manuel Zenha de Pinho. *Relatório de Análise Tecnológica*. Projecto de Sistemas de Software - Faculdade de Engenharia da Universidade do Porto, Maio 2009. Disponível através de requerimento em <http://www.hzenha.pt.vu/>.
- [dP09c] Hugo Manuel Zenha de Pinho. *Relatório de Definição de Arquitectura*. Projecto de Sistemas de Software - Faculdade de Engenharia da Universidade do Porto, Maio 2009. Disponível através de requerimento em <http://www.hzenha.pt.vu/>.
- [dP09d] Hugo Manuel Zenha de Pinho. *Relatório de Especificação de Requisitos*. Projecto de Sistemas de Software - Faculdade de Engenharia da Universidade do Porto, Maio 2009. Disponível através de requerimento em <http://www.hzenha.pt.vu/>.
- [dSdI06] Direcção Municipal de Sistemas de Informação. *GIC - Especificação de Requisitos*. Câmara Municipal do Porto, Outubro 2006.
- [dSdI09] Direcção Municipal de Sistemas de Informação. Outros requisitos não-funcionais. In *SGIFM - Especificação de Requisitos*. Câmara Municipal do Porto, Março 2009.

REFERÊNCIAS

- [eGD01] Lúcia M. Ribeiro e Gabriel David. *SiFEUP - Relatório Técnico*. Faculdade de Engenharia da Universidade do Porto, 2001. Disponível em http://www.fe.up.pt/si/publs_web.show_publ_file?p_id=736.
- [eMF09] Mozilla Europe e Mozilla Foundation. Firefox o navegador web, Maio 2009. The Mozilla Foundation, disponível em <http://www.mozilla-europe.org/pt/firefox/>.
- [Eve06] Software Everywhere. Interview with superwaba developer guilherme c. hazan, Fevereiro 2006. David Beers on Mobile Computing, disponível em <http://www.superwaba.com.br/pt/default.asp>.
- [Fou09a] The Apache Software Foundation. The apache software foundation - meritocracy in action, Fevereiro 2009. The Apache Software Foundation, disponível em <http://tomcat.apache.org/>.
- [Fou09b] The Apache Software Foundation. Apache tomcat, Fevereiro 2009. The Apache Software Foundation, disponível em <http://www.apache.org/>.
- [Fou09c] The Apache Software Foundation. Webservices - axis, Fevereiro 2009. The Apache Software Foundation, disponível em <http://tomcat.apache.org/>.
- [Lon09] Kevin Loney. *Oracle Database 11g The Complete Reference*. The McGraw-Hill Companies, Inc., 2009.
- [Ltd09] ARM Ltd. Arm - the architecture for the digital world, Maio 2009. ARM Ltd, disponível em <http://www.arm.com/>.
- [McL08] Michael McLaughlin. *Oracle Database 11g PL/SQL Programming*. The McGraw-Hill Companies, Inc., 2008.
- [Med09] O'Reilly Media. metro: Home, Fevereiro 2009. Sun Microsystems, Inc., disponível em <https://metro.dev.java.net/>.
- [Net09a] Sun Developer Network. Java me - the most ubiquitous application platform for mobile devices, Maio 2009. Sun Microsystems, Inc., disponível em <http://java.sun.com/javame/index.jsp>.
- [Net09b] Sun Developer Network. Why java me platform?, Fevereiro 2009. Sun Microsystems, Inc., disponível em <http://java.sun.com/javame/overview/why.jsp>.
- [OGC09] Inc. Open Geospatial Consortium. Opengis® standards and specifications, Maio 2009. Wikimedia, disponível em <http://www.opengeospatial.org/standards>.
- [Qua09] Qualcomm. Qualcomm brew, Fevereiro 2009. QUALCOMM Incorporated, disponível em <http://brew.qualcomm.com/brew/en/>.

REFERÊNCIAS

- [Ric06] Mark Richards. Infoq: The role of the enterprise service bus, Outubro 2006. C4Media Inc., disponível em <http://www.infoq.com/presentations/Enterprise-Service-Bus>.
- [Ris08] Ray Rischpater. *Beginning Java ME Platform*. Springer-Verlag, Inc., 2008.
- [SM09] Inc. Sun Microsystems. Sun glassfish enterprise server, Maio 2009. Sun Microsystems, Inc., disponível em <http://www.sun.com/software/products/appsrvr/>.
- [Str09] Pedro Strecht. *Sistema de Gestão de Sócios* - apresentação. Projecto de Sistemas de Software - Faculdade de Engenharia da Universidade do Porto, Maio 2009.
- [Sup09] SuperWaba. Superwaba: The real power of mobile computing, Fevereiro 2009. SuperWaba, disponível em <http://www.superwaba.com.br/pt/default.asp>.
- [W3C09] W3C. World wide web consortium, Maio 2009. The World Wide Web Consortium, disponível em <http://www.w3.org/>.
- [Whi08] Jim White. Start your engines: Mobile application development, Abril 2008. Jupitermedia Corporation, disponível em <http://www.devx.com/SpecialReports/Article/37693/0/page/1>.

Anexo A

Teste de Desempenho - Servidores de Aplicações Web

A.1 Hardware

O hardware utilizado para proceder aos testes dos servidores de aplicações web foi o mesmo para correr quer o cliente, quer o servidor.

As características são as seguintes:

- **Modelo** Dell OptiPlex 755
- **Processador** Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz
- **Memória RAM** 3325 MB

De notar que o servidor correu numa máquina virtual, para a qual foram alocados 1024 MB de RAM.

A.2 Software

O software utilizado para executar os testes no cliente é diferente do utilizado no servidor. Todo o software foi instalado com os valores definidos por omissão sem qualquer tipo de melhoramentos por parte do testador.

A.2.1 Servidor

O servidor foi instalado numa máquina virtual numa máquina com o hardware especificado anteriormente. O software para criar esta máquina virtual foi o *Microsoft Virtual PC 6.0.192.0* [Cor09h].

O software instalado para a execução do teste foi:

- **Sistema Operativo** Windows Server 2003 Enterprise Edition com Service Pack 1;
- **Servidores de Aplicações;**
 - Apache Tomcat 6.0.18 com AXIS2;

- IBM WebSphere Application Server CE Version 2.1;
- Oracle Weblogic 10gR3;
- Sun Application Server 2.1.

- **Java JDK 1.6.0_12.**

A.2.2 Cliente

O software instalado no cliente para a execução do teste foi:

- **Sistema Operativo** Windows Vista Business N;
- **Aplicação para execução de testes** eviware soapUI 2.5.1;
- **Java JDK 1.6.0_12.**

A.3 Implementação

A implementação dos testes foi realizada recorrendo à ferramenta de desenvolvimento *Oracle JDeveloper 11.1.1.0.1* para criação dos ficheiros necessários para *deployment* nos servidores IBM WebSphere, Oracle Weblogic e Sun Application Server, enquanto que para o caso do Apache Tomcat, que pede tipos de ficheiro específicos para a sua implementação, foi necessário recorrer à ferramenta NetBeans IDE 6.5.

A.3.1 Código dos Web Services

Foram criados Web services para testar os servidores de aplicações. O código destes Web services é o seguinte:

```
1 import java.io.BufferedReader;
2 import java.io.BufferedWriter;
3 import java.io.FileInputStream;
4 import java.io.FileNotFoundException;
5 import java.io.FileOutputStream;
6 import java.io.IOException;
7 import java.io.InputStreamReader;
8 import java.io.OutputStreamWriter;
9
10 import javax.jws.WebMethod;
11 import javax.jws.WebService;
12
13 @WebService
14 public class ServerTester {
15     public ServerTester () {
16     }
17
18     @WebMethod
19     public String answeringTester () {
20         return "answering!";
21     }
22
23     @WebMethod
```

```

24  public String processingTester(){
25
26      FileOutputStream fos;
27      try {
28          fos = new FileOutputStream("/testingFile");
29      } catch (FileNotFoundException e) {
30          return "error in opening file for writing!";
31      }
32      BufferedWriter out;
33      out = new BufferedWriter(new OutputStreamWriter(fos));
34      try {
35          out.write("This was written to the file!");
36      } catch (IOException e) {
37          return "error in writing to file!";
38      }
39      try {
40          out.close();
41      } catch (IOException e) {
42          return "error in closing file after writing!";
43      }
44      FileInputStream fis;
45      try {
46          fis = new FileInputStream("/testingFile");
47      } catch (FileNotFoundException e) {
48          return "error in opening file for reading!";
49      }
50      BufferedReader in;
51      in = new BufferedReader(new InputStreamReader(fis));
52      String answer;
53      try {
54          answer = in.readLine();
55      } catch (IOException e) {
56          return "error in reading file!";
57      }
58      try {
59          in.close();
60      } catch (IOException e) {
61          return "error in closing file after reading!";
62      }
63      return answer + " and it is answering!";
64  }
65  }
66  }

```

O método *answeringTester()* corresponde ao teste referente à **Primeira Implementação** e o *processingTester()* corresponde à **Segunda Implementação**.

Em relação à configuração do soapUI para correr os testes de carregamento, os valores utilizados foram os seguintes:

- **Limit** 300 seconds;
- **Threads** 10;
- **Test Delay** 0;
- **Random** 0.5;

A.4 Resultados

Os resultados referentes a cada um dos servidores estão ilustrados nas *Tabelas A.1, A.2, A.3 e A.4*. Cada tabela é referente a um dos servidores testados nas duas implementações de *Web services* realizadas. Cada tabela apresenta os tempos mínimo e máximo de resposta e sua média, qual o tempo da última conexão, número de conexões realizadas, número de *threads* por segundo (tps), *MBytes* trocados, *bits* por segundo e número de erros em respostas do servidor.

Tabela A.1: Teste de Desempenho: Apache Tomcat 6.0.18 com AXIS2

Implementação	min	max	avg	last	cnt	tps	Mbytes	bps	err
1	3	444	14,27	18	189602	632,18	50,44838	176379	0
2	3	5156	28,5	21	99618	332,15	30,40086	106289	0

Tabela A.2: Teste de Desempenho: IBM WebSphere Application Server CE Version 2.1

Implementação	min	max	avg	last	cnt	tps	Mbytes	bps	err
1	2	673	17,59	48	156645	522,23	50,34386	175991	0
2	3	3137	32,68	197	87564	291,72	31,55205	110222	0

Tabela A.3: Teste de Desempenho: Oracle Weblogic 10gR3

Implementação	min	max	avg	last	cnt	tps	Mbytes	bps	err
1	3	564	13,78	165	197830	659,23	46,97768	164149	0
2	6	4516	27,68	23	103620	345,41	28,65765	100169	0

Tabela A.4: Teste de Desempenho: Sun Application Server 2.1

Implementação	min	max	avg	last	cnt	tps	Mbytes	bps	err
1	3	640	15,04	10	180751	602,45	40,14731	140391	75
2	4	465	27,92	26	101487	338,27	26,46045	92481	225

A.5 Resumo ou Conclusão

Os resultados obtidos mostram que o servidor *Oracle Weblogic 10gR3* foi o **Melhor do Teste**, sendo que numa vista mais económica, o servidor *Apache Tomcat 6.0.18 com AXIS2* apresenta a melhor relação qualidade/preço.

Diagrama de Classes

O diagrama apresentado pela *Figura B.1* representa a visão de dados do *SuS*, sendo que a mesma é espelhada para o *MSuS*. De notar que o *MSuS* só poderá actualizar os dados referentes às *Ocorrências* e *Características* a elas associadas.

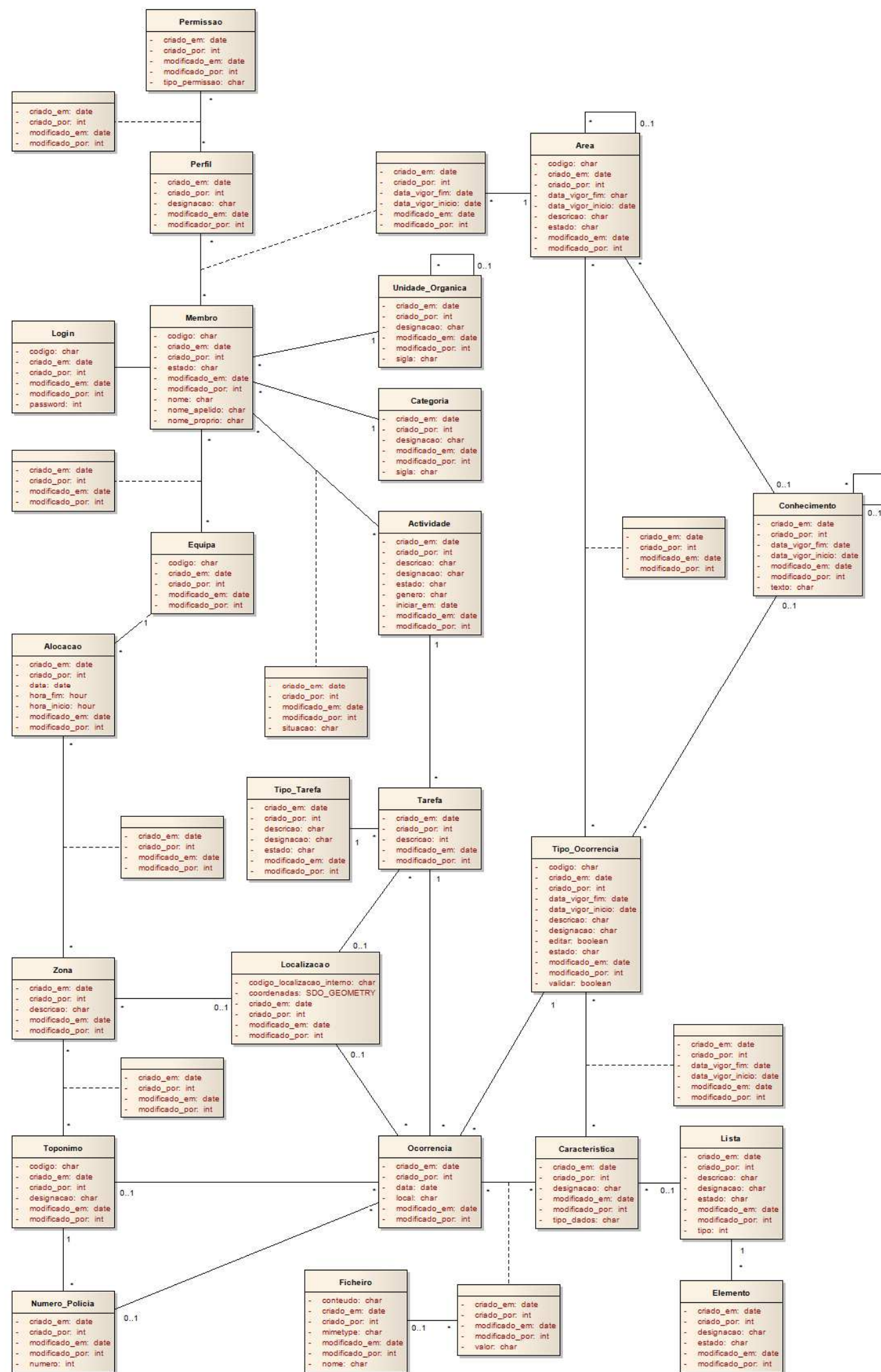


Figura B.1 - Diagrama de Classes Completo

Anexo C

Relatório do Projecto

Os seguintes relatórios estão contidos num CD, parte integrante do presente relatório:

- Relatório de Análise Tecnológica;
- Relatório de Especificação de Requisitos;
- Relatório de Definição da Arquitectura.

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Aplicação de Registo de Ocorrências para Dispositivos Móveis

Hugo Manuel Zenha de Pinho

Relatório de Análise Tecnológica

Mestrado Integrado em Engenharia Informática e Computação

Orientador: João Manuel Paiva Cardoso

Doutor em Engenharia Electrotécnica e Computadores

29 Junho 2009

Conteúdo

1	Introdução	1
1.1	Contexto/Enquadramento	1
2	Análise Tecnológica	3
2.1	SuS - Supervision System	3
2.2	MSuS - Mobile Supervision System	3
2.2.1	Aplicação	4
2.2.2	Base de Dados	7
2.2.3	Web Services	9
2.2.3.1	Servidor de Aplicações Web	10
3	Conclusões	17
	Referências	19
A	Teste de Desempenho - Servidores de Aplicações Web	20
A.1	Hardware	20
A.2	Software	20
A.2.1	Servidor	20
A.2.2	Cliente	21
A.3	Implementação	21
A.3.1	Código dos Web Services	21
A.4	Resultados	23
A.5	Resumo ou Conclusão	24

Lista de Figuras

1.1	Aspecto Geral da Área de Actuação do Projecto.	1
2.1	Principais Características dos Web services Java	9
2.2	Principais Características dos Web services .NET	10
2.3	Teste TPS (Primeira Implementação).	13
2.4	Teste AVG (Primeira Implementação).	13
2.5	Teste CNT (Primeira Implementação).	14
2.6	Teste Mbytes (Primeira Implementação).	14
2.7	Teste TPS (Segunda Implementação).	15
2.8	Teste AVG (Segunda Implementação).	15
2.9	Teste CNT (Segunda Implementação).	15
2.10	Teste Mbytes (Segunda Implementação).	16
3.1	Escolhas Tecnológicas.	17

Lista de Tabelas

A.1	Teste de Desempenho: Apache Tomcat 6.0.18 com AXIS2	23
A.2	Teste de Desempenho: IBM WebSphere Application Server CE Version 2.1 . . .	23
A.3	Teste de Desempenho: Oracle Weblogic 10gR3	23
A.4	Teste de Desempenho: Sun Application Server 2.1	24

Abreviaturas e Símbolos

API	Application Programming Interface
AVG	Average
ARM	Advanced RISC Machine
CNT	Connections
ESB	Enterprise Service Bus
GIC	Gestão Integrada de Contra-Ordenações
GPS	Global Positioning System
IDE	Integrated Development Environment
PDA	Personal Digital Assistant
PL/SQL	Procedural Language/Structured Query Language
SDK	Software Development Kit
SQL	Structured Query Language
SOAP	Simple Object Access Protocol
TPS	Transactions Per Second
WS-Security	Web Services Security

Capítulo 1

Introdução

1.1 Contexto/Enquadramento

Este documento apresenta a análise tecnológica efectuada para escolher as tecnologias a utilizar para o desenvolvimento do projecto em causa. O aspecto geral da área de actuação do projecto é ilustrado na *Figura 1.1*.

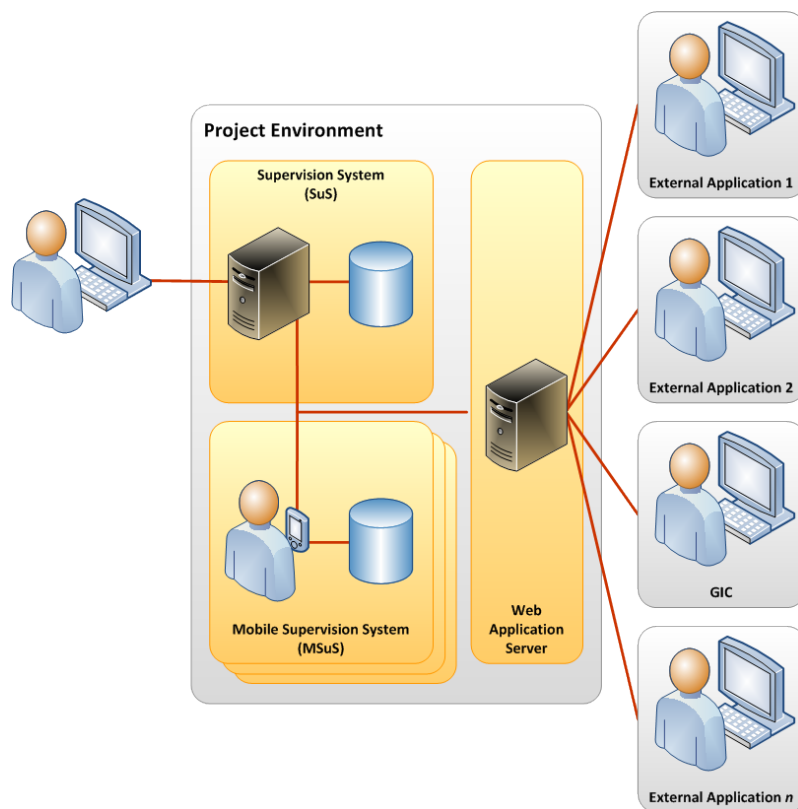


Figura 1.1: Aspecto Geral da Área de Actuação do Projecto.

Introdução

De acordo com a *Figura 1.1*, a análise terá de ser efectuada em relação às tecnologias a utilizar em:

- Aplicação *Back-Office* do Sistema de Fiscalização (*Supervision System*);
- Aplicação no *PDA* (*Personal Digital Assistant*) do Sistema de Fiscalização Móvel (*Mobile Supervision System*);
- Base de Dados no *PDA*;
- Web Services;
- Servidor de Aplicações Web.

Capítulo 2

Análise Tecnológica

Em relação ao estudo das tecnologias a utilizar na construção da solução, e como se pode ver na *Figura 1.1* é necessário escolher quais serão as melhores opções em relação:

- à aplicação de Back-Office (*SuS*);
- à aplicação a correr nos dispositivos móveis, mais concretamente nos *PDA*s (*MSuS*);
- à base de dados a utilizar nestes *PDA*s;
- aos *Web services* a criar;
- e ao Servidor de Aplicações Web para a disponibilização de *Web services*.

O estudo efectuado é apresentado de seguida.

2.1 SuS - Supervision System

Em relação à aplicação de *Back-Office* do projecto existem algumas restrições impostas pela instituição de acolhimento, o *PSI*. De acordo com esta, a aplicação terá de ser desenvolvida recorrendo à base de dados *Oracle* e à linguagem *PL/SQL* para criação do ambiente de interacção.

2.2 MSuS - Mobile Supervision System

Tendo a aplicação em causa o intuito de correr num *Personal Digital Assistant (PDA)* é de todo conveniente estudar quais as tecnologias que podem ser utilizadas para a criação de todos os módulos em causa, suas vantagens e desvantagens, para que, então, se proceda à decisão sobre quais serão utilizadas no âmbito deste projecto. Foi dada liberdade ao autor deste relatório para a escolha das tecnologias.

2.2.1 Aplicação

Em relação à aplicação a criar, primeiro é necessário optar pelo desenvolvimento em ambiente web ou de uma aplicação mobile. Para além disso, a tecnologia a utilizar dentro da opção anterior, é também uma escolha a realizar.

Aquando da escolha da tecnologia a utilizar é necessário ter em consideração os seguintes aspectos:

- Acesso às características do dispositivo Móvel;
- Custo;
- Desempenho;
- Experiência com a tecnologia;
- Ferramentas de desenvolvimento e *IDEs (Integrated Development Environment)*;
- Ferramentas para teste.

Desenvolvimento Orientado à Web

Existem várias opções no campo de desenvolvimento web. Devido às especificidades do projecto em causa, como acesso a funcionalidades do dispositivo móvel, utilização de geo-posicionamento, segurança, entre outras, esta opção foi descartada.

Contudo, o *Microsoft Silverlight* [Cor09f] vocacionado para dispositivos móveis, uma tecnologia que está em fase de desenvolvimento, poderá, num futuro próximo, vir a ser uma opção a ponderar para próximos desenvolvimentos na área e/ou em outros projectos semelhantes.

Desenvolvimento não Orientado à Web

No desenvolvimento de aplicações não orientadas à Web existem várias opções à escolha. Assim sendo, é necessário analisar as que parecem ser as mais viáveis no contexto do projecto.

Brew

O *Brew* [Qua09], *Binary Runtime Environment for Wireless*, é uma plataforma de desenvolvimento de aplicações criada pela *Qualcomm* para telemóveis. Existe compatibilidade com alguns *PDA*s.

No entanto, para que seja possível obter o *SDK (Software Development Kit)* e as ferramentas base para iniciar o desenvolvimento, é necessário o registo na *Qualcomm*. Para a criação de aplicações comerciais, é necessário seguir uma série de passos que implicam um custo associado dependendo do número de aplicações a desenvolver, que nunca poderá ser menor que 100. Após as aplicações estarem desenvolvidas e testadas - de notar que apenas podem ser testadas em emuladores obtidos com o *SDK* e em hardware que permita correr aplicações *Brew*, que por si só já restringe muito o número de dispositivos móveis compatíveis -, tais aplicações serão testadas

pela *Qualcomm*. Só assim a aplicação pode, então, ser comercializada via mecanismos da própria *Qualcomm*.

Existem muitas críticas a esta forma de desenvolvimento, desde logo pelo facto da obrigação ao registo na *Qualcomm* e submissão das aplicações para testes, acarretando um esforço monetário adicional. Estes custos provocam uma menor adesão, assim como quase nenhum apelo ao desenvolvimento por *hobbie*. Para além de todo este processo, ainda existe a possibilidade da aplicação ser rejeitada devido a insuficiente rentabilidade ou mesmo devido a competição directa com produtos da própria *Qualcomm* [Qua09].

Sendo baseado nas linguagens de programação C/C++, os *IDEs* que podem ser utilizados são diversos, desde *open source* (*Eclipse*, por exemplo) até versões comerciais (*Microsoft Visual Studio*, por exemplo).

Em relação a desempenho, o *Brew* aparece como sendo uma das melhores opções, muito devido ao facto de se basear numa linguagem de programação de mais baixo nível que os seus concorrentes [Whi08].

Java

O Java é uma tecnologia da *Sun Microsystems* que aparece na criação de aplicações móveis sob a sua versão *Java ME* [Net09a] [Ris08]. Esta plataforma é uma solução aberta, ao contrário de muitas outras que se encontram no mercado. Foi desenhada com o intuito de providenciar portabilidade de aplicações [Net09b].

Sendo uma tecnologia sob a licença *GNU General Public License*, os custos associados à utilização desta tecnologia apenas se focam na experiência e capacidades dos desenvolvedores.

Em relação a ferramentas para o desenvolvimento de aplicações, a *Sun Microsystems* providencia o *Java ME SDK* e ferramentas base. No entanto, e como é conhecido, devido aos múltiplos ambientes e *APIs* (*Application Programming Interface*), existem diversos *SDKs* e *toolkits* para satisfazer necessidades específicas. Para que a experiência do utilizador seja melhor, existem no mercado diversos *IDEs*, quer *open source*, quer comerciais. No mundo do *open source* os mais populares são o *Eclipse* e o *NetBeans*, este último da própria *Sun*. No lado comercial, soluções como o *Rational Application Developer* da *IBM* estão à disposição dos desenvolvedores. Muitas ferramentas sofrem da pouca oferta de soluções de desenvolvimento, no mundo do Java acontece precisamente o contrário, existem demasiadas soluções. A complexidade da arquitectura do *Java ME* permite uma boa adaptação a um vasto leque de plataformas mas, devido ao problema referido anteriormente, a escolha do ambiente de desenvolvimento mais adequado pode ser um desafio.

Para que seja possível proceder a testes à aplicação desenvolvida, e sendo a aplicação a desenvolver para *PDAs*, a necessidade de utilização de emuladores torna-se bastante crítica. Neste campo o *Java ME* oferece emuladores genéricos para alguns dispositivos, através dos *SDKs* e *IDEs*. Adicionalmente, alguns dos criadores de dispositivos, como a *Nokia* ou a *Motorola*, providenciam *SDKs* que oferecem emuladores mais precisos para certos dispositivos, sendo que alguns podem ainda ser integrados com *IDEs*. É também necessário referir que os emuladores genéricos

não testam as *APIs* específicas dos criadores do dispositivo em causa, podendo não reflectir problemas específicos deste dispositivo. Assim sendo, é recomendado testar as aplicações criadas no dispositivo alvo.

O acesso às capacidades do dispositivo móvel, como é o caso da câmara, normalmente requer a inclusão manual destes acessos nas configurações, perfil ou mesmo na *API* para esse dispositivo.

Em relação ao desempenho de uma aplicação desenvolvida utilizando *Java ME*, esta sofre mais pelo poder de processamento limitado dos dispositivos móveis, uma vez que a aplicação é executada numa máquina virtual [Whi08].

SuperWaba

Depois da tecnologia *Java ME* ter sido abordada anteriormente, o *SuperWaba* [Sup09], mais concretamente o *TotalCross*, aparece como uma alternativa viável, na medida em que este é uma máquina virtual baseada na do *Java* tendo como alvo os dispositivos móveis.

Todas as vantagens e desvantagens ligadas à utilização de *Java* são encontradas nesta tecnologia, no entanto existem algumas diferenças. Mesmo sendo parecido com a especificação do *Java 2 Standard Edition*, o *SuperWaba* aparece bastante vocacionado para as restrições ligadas às limitações dos dispositivos móveis. Sendo assim, o seu desempenho, devido à optimização criada, consegue ser até duas vezes mais rápido que o concorrente *Java ME* [Eve06].

Mesmo sendo uma tecnologia *open source*, é necessário, se utilizada para criação de aplicações comerciais, o licenciamento por cada dispositivo móvel em que tal aplicação será instalada [Sup09].

Microsoft .NET

Dentro da leque de soluções oferecidas pela tecnologia *Microsoft .NET* encontra-se o *.NET Compact Framework* [Cen09]. Este é uma componente dos dispositivos *Windows Mobile* e *Windows Embedded CS* que permite o desenvolvimento e utilização de aplicações e *Web services*.

Para desenvolvimento, o *IDE* de eleição é o *Microsoft Visual Studio*, existindo, contudo, outras possibilidades. Claro que tal ferramenta tem custos monetários associados à sua utilização, no entanto, esta simplifica muito o desenvolvimento para o *.NET Compact Framework* e tem uma curva de aprendizagem pequena para desenvolvedores já familiarizados com o *.NET* e desenvolvimento em *C#* e/ou *VB*.

Em relação a ferramentas para teste das aplicações, o *Windows Mobile 6 SDK* providencia o chamado *Device Emulator* que é considerado uma das vantagens em desenvolver aplicações móveis através do *.NET*. Este emulador providencia *pseudo-GPS* (*Global Positioning System*) e todas as possibilidades de interacção com o dispositivo. Este emulador é um verdadeiro *Advanced RISC Machine (ARM) emulator*, o que quer dizer que corre o mesmo código que dispositivos móveis baseados em arquitecturas com processadores *ARM*. Segundo o fabricante, 90% dos dispositivos móveis estão equipados com processadores *ARM*, conjunto onde se encontram os *PDA's* [Ltd09].

Como o *.NET Compact Framework* foi criado para responder às necessidades de desenvolvimento no sistema operativo *Windows Mobile*, tendo acesso a todas as capacidades do dispositivo móvel suportadas pelo sistema operativo.

Em relação ao desempenho, este é considerado razoável em todas as vertentes [Whi08].

Escolha

Após uma análise cuidada de várias tecnologias existentes no mercado (e certas implementações) foi escolhido optar pela tecnologia *.NET*. Isto deve-se, não só à existência de licenciamento em tecnologia Microsoft mas, também, devido aos dispositivos alvo do projecto serem *PDA*s com *Windows Mobile*. Desta forma, a integração com todas as especificidades do dispositivo é facilitada, pois a tecnologia do sistema operativo e da aplicação a desenvolver é a mesma.

No entanto, não seria de todo uma má escolha a utilização de uma tecnologia baseada em *Java*, como por exemplo o *SuperWaba* ou mesmo o *Java ME*. O caso do *Brew* estará sempre fora de questão devido à necessidade de licenciamento e à forma como todo o processo de desenvolvimento e entrega ao cliente é realizado, assim como a compatibilidade deste com os dispositivos móveis ser restrita.

2.2.2 Base de Dados

A escolha da tecnologia de bases de dados para o *PDA* depende muito dos mecanismos existentes de sincronização com a base de dados utilizada no sistema com o qual a aplicação desenvolvida vai comunicar - o *SuS*. A base de dados em causa é proveniente da tecnologia *Oracle*. Logo, a opção que tornará mais fácil esta integração será a escolha mais acertada. No entanto, ao desenvolver a aplicação em causa, a mudança de tecnologia a utilizar deverá ser facilmente realizada para responder a necessidades da troca de aplicação com a qual vai sincronizar.

Borland JDataStore

Existe uma tecnologia de base de dados, denominada de *JDataStore* [Cor09c], que parece ser um opção interessante para utilizar em dispositivos móveis. No entanto, no próprio site da *Borland* não existe a indicação de como obter este software. Sendo assim, como parece ser um produto descontinuado, esta opção acaba por deixar de fazer parte do lote de tecnologias a considerar.

Microsoft SQL Server Compact

A *Microsoft* entra no leque de opções devido ao produto *SQL Server Compact 3.5* [Cor09d].

Esta base de dados é desenhada para desenvolvedores que tenham como alvo quer dispositivos móveis baseados em *Windows Mobile*, quer *Desktops*. Providencia sincronização com *SQL Server*, assim como *API* para desenvolvimento. O desenvolvimento pode ser integrado através do *Microsoft Visual Studio* que providencia todos os mecanismos de teste e desenvolvimento inerentes.

Através da componente *SQL Server Integration Services* são providenciados mecanismos de troca de informação com bases de dados *Oracle*, *IBM*, entre outras [Cor09e] [Cor08a].

Oracle Database Lite

A *Oracle* providencia aos seus clientes a possibilidade de adquirir o produto *Oracle Database Lite* [Cor08b]. Este produto providencia a possibilidade de criar bases de dados que têm por alvo dispositivos móveis, e, sendo uma tecnologia *Java-based*, pode ser utilizada sobre várias plataformas.

Esta base de dados vem acompanhada de mecanismos de sincronização para bases de dados *Oracle*, sendo uma mais valia no caso de a aplicação a desenvolver necessitar de sincronizar com esta tecnologia.

A solução da *Oracle* permite a integração com vários *IDEs*, entre os quais o *Microsoft Visual Studio*, *JDeveloper*, *Eclipse*, entre outros, tendo para isso mecanismos **ODBC**, **JDBC** e **ADO.NET**. [Cor09h] [Cor08b]

Escolha

Como foi dito no início da análise destas opções, a opção para a base de dados no *PDA* cairá sobre aquela que fornecer mais facilidade de sincronização com a base de dados *Oracle* a utilizar no sistema de *Back-Office*. A escolha fica, então, dividida entre o *Oracle Database Lite* e o *Microsoft SQL Serve Compact*, uma vez que ambas as soluções apresentam mecanismos de sincronização com bases de dados *Oracle*.

A possibilidade de teste em ambiente *desktop* ser apenas possível numa das soluções poderia limitar a escolha. No entanto, recorrendo ao *Microsoft Device Emulator* quer uma, quer outra podem ser instaladas e testadas no ambiente *Windows Mobile*.

Sendo assim, a escolha passa por perceber qual a solução que apresenta maior flexibilidade numa possível manutenção e/ou alteração da tecnologia da aplicação. Seguindo este critério, a solução da *Oracle* aparece como sendo a melhor opção, pois ao utilizar o *Microsoft SQL Server* fica-se mais dependente das tecnologias da *Microsoft*.

2.2.3 Web Services

Para além da utilização de Web services também foi equacionada a utilização de um Enterprise Service Bus (mais conhecido apenas por *ESB*) [Ric06]. Esta tecnologia permite a integração de vários sistemas de uma forma mais transparente, servindo como intermediário na transmissão de informação entre os vários intervenientes.

Esta tecnologia não será utilizada, uma vez que o esforço necessário para que esta seja utilizada não compensa as vantagens que daí advêm, uma vez que, a solução não prevê uma grande interacção entre os intervenientes, existindo apenas ligação pontual entre um dispositivo móvel e o sistema de *Back-Office* [Cha04].

A discussão em relação à implementação dos *Web services*, que vão ser a ligação entre a aplicação no dispositivo móvel e os sistemas de registo de autos e sistemas de pesquisa, é sobre se será utilizada tecnologia *Java* ou *Microsoft .NET*.

Java

Em relação à solução *Java*, é disponibilizado um serviço, através do projecto *Glassfish - Metro* [Med09], de criação de *Web services* de fácil utilização, com bom desempenho e extensibilidade. As principais características estão representadas na Figura 2.1¹. Estes serviços apresentam uma vasta panóplia de soluções de transporte e de segurança, em que se pode referir, com especial importância, a utilização de *SOAP* (*Simple Object Access Protocol*) e do *WS-Security* (*Web Services Security*).

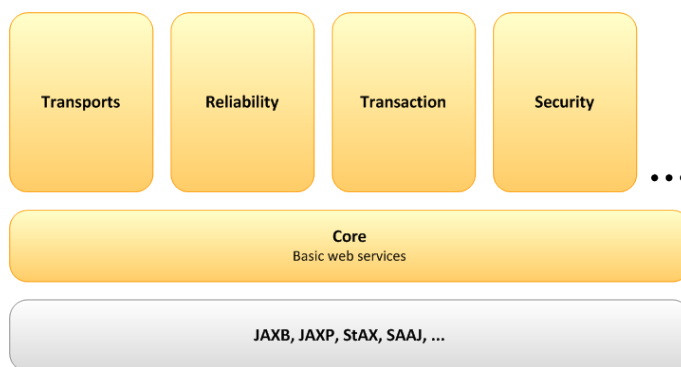


Figura 2.1: Principais Características dos Web services Java

Microsoft .NET

No caso da tecnologia *.NET*, é disponibilizado um serviço de criação de *Web services* totalmente integrado. As principais características estão ilustradas na Figura 2.2².

Tal como a solução *Java*, o *.NET* disponibiliza transporte *SOAP*, segurança *WS-Security*, entre outras soluções [Cor09e].

¹baseado no diagrama existente em [Med09] sobre Web services

²baseado no diagrama existente em [Cor09e] sobre Web services

Análise Tecnológica

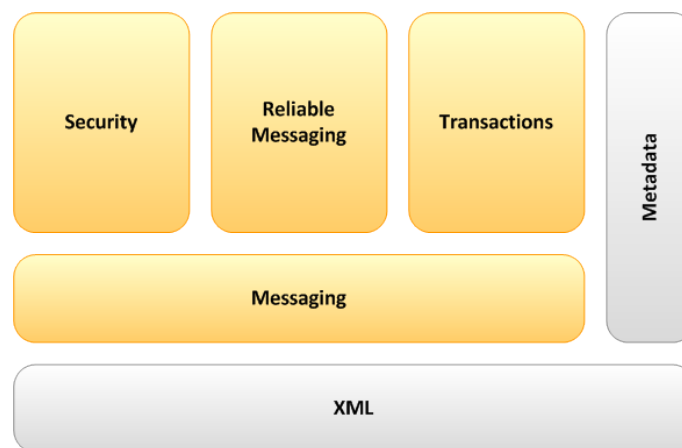


Figura 2.2: Principais Características dos Web services .NET

Escolha

A diferença entre as duas tecnologias não é significativa. Em ambas, o acesso por parte de outras plataformas de desenvolvimento é garantido (objectivo dos *Web services*). A escolha recai sobre a tecnologia *Java*, na medida em que a base de dados que irá suportar o projecto a desenvolver é *Oracle* (restrição de desenvolvimento), que é uma solução baseada em *Java*. Assim, a compatibilidade e integração será facilitada pela criação de *Web services* em *Java*.

Esta escolha não influencia a aplicação a executar no *PDA*, pois os *Web services* permitem acesso a qualquer aplicação em qualquer plataforma.

2.2.3.1 Servidor de Aplicações Web

Existem várias tecnologias no mercado que providenciam *deploy* para os *Web services* a desenvolver no âmbito deste projecto.

Para além de procura de informação sobre as várias soluções, foram realizados testes de comparação de desempenho para uma melhor fundamentação na escolha.

Apache Tomcat

A *Apache Software Foundation* providencia suporte à comunidade de projectos de software *open source*. O servidor de aplicações fornecido é o *Apache Tomcat* [Fou09a], e evidencia-se pelo papel crucial que representou no crescimento inicial da *World Wide Web*.

Este servidor de aplicações está disponível para um grande leque de sistemas operativos (como *Windows*, mas principalmente para os baseados em *Unix*), fornecendo de uma forma gratuita ferramentas para disponibilização de serviços online.

Dentro do projecto *Apache*, o *Axis* [Fou09c] aparece como uma das soluções para *deployment* dos *Web services*.

Esta tecnologia suporta as especificidades da escolha de *Web services* discutida anteriormente (como *SOAP* e *WS-Security*), tendo como vantagem, sobre alguns dos seus competidores, o facto

de ser uma tecnologia licenciada segundo *GNU General Public License*, o que tem levado a que seja uma opção muito tentadora [Fou09b].

IBM WebSphere

A *International Business Machines Corporation (IBM)* fornece uma linha de produtos na área de servidores de aplicações.

Websphere [Cor09b] é uma linha de produtos de software da *IBM*, tendo como produto essencial o chamado *IBM WebSphere Application Server*. Este foi desenhado para criar, operar e integrar negócio electrónico através de múltiplas plataformas de computação. Esta tecnologia é baseada nas tecnologias web de *Java*. Inclui quer as componentes de *run-time*, quer as ferramentas para desenvolver aplicações que irão correr no *IBM WebSphere Application Server*.

Esta solução, tal como as discutidas até ao momento, permitem utilização das especificidades dos *Web services* a desenvolver. Em relação à solução da *Apache*, o *Websphere* tem a desvantagem de ser necessária a sua compra, existindo, no entanto, uma versão não paga, o *WebSphere Application Server Community Edition* [Cor09a].

Sun Application Server

Em relação a servidores de aplicações, a *Sun* aparece com o produto proveniente do projecto *Glassfish*, *Sun Application Server* [SM09]. Sendo um servidor de aplicações *open source*, o *Sun Application Server* aparece como uma alternativa viável para o *deployment* dos *Web services*.

O *Metro*, parte integrante do projecto *Glassfish*, suporta todas as tecnologias necessárias ao desenvolvimento deste projecto.

Oracle

A *Oracle* aparece com uma linha de produtos que vão de encontro às necessidades do projecto. Especificamente, o software para servidores de aplicações web que a *Oracle* disponibiliza são o *Oracle Application Server* [Cor09i] e o *Oracle Weblogic Server* [Cor09j].

Oracle Application Server

Esta tecnologia, desenvolvida pela *Oracle*, irá ser descontinuada. Esta situação acontecerá devido à compra da *BEA* por parte da *Oracle*. A *Oracle* irá continuar a desenvolver o servidor anteriormente da *BEA*, uma vez que esta é uma das líderes em soluções de infraestruturas para aplicações empresariais [Cor09h].

Sendo assim, esta opção aparece como uma das que são descartadas à partida devido à sua descontinuação, facto que pode provocar um problema no futuro aquando da migração, obrigada, para outra solução.

Oracle WebLogic Server

Como dito anteriormente, devido à compra da *BEA* por parte da *Oracle*, esta aparece como

uma das soluções para *deployment* de aplicações web considerada com uma das líderes do mercado. Tal como as soluções analisadas até ao momento, este servidor suporta todas as tecnologias necessárias para a conclusão do projecto.

Comparação de Desempenhos

Para proceder à escolha de qual o servidor de aplicações web a utilizar no âmbito do projecto foram realizados testes para comparação de desempenhos.

Os testes realizados tiveram por objectivo comparar desempenhos de *Web services* nas várias opções apresentadas anteriormente. Foram utilizadas duas implementações de *Web services* (dois testes com objectivos diferentes):

1. uma implementação avalia apenas a rapidez em responder a um pedido simples, sem processamento no lado do servidor;
2. a outra avalia métricas no caso de existir processamento do lado do servidor.

As métricas utilizadas para a comparação foram as seguintes:

- Transacções por segundo (**Teste TPS**);
- Média de tempo de resposta em milisegundos (**Teste AVG**);
- Número de conexões, em milhares, estabelecidas em cinco minutos (**Teste CNT**);
- Mbytes trocados em cinco minutos (**Teste Mbytes**).

Os testes foram realizados utilizando para o servidor e cliente a mesma máquina, tendo o servidor sido executado numa máquina virtual.

No *Anexo A* são apresentados mais pormenores sobre o teste de desempenho realizado.

Primeira Implementação

Analisando as transacções existentes, informação ilustrada na *Figura 2.3*, é possível concluir que neste teste de resposta a solução apresentada pela *Oracle* destaca-se pela positiva, com 659 *tps*. No entanto, o servidor *Apache* não está muito longe desse valor. O servidor *WebSphere*, da *IBM*, aparece como sendo o que tem menor capacidade de resposta rápida a pedidos enviados, com 522 *tps*.

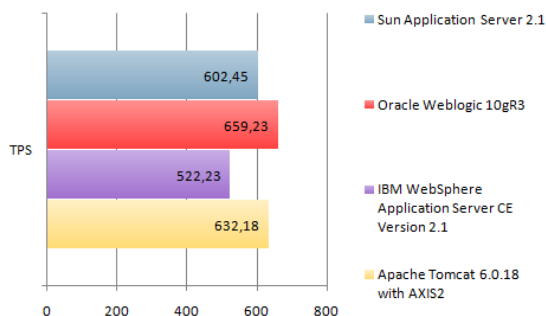


Figura 2.3: Teste TPS (Primeira Implementação).

Em relação ao tempo médio de resposta, e como pode ser observado na *Figura 2.4*, mais uma vez o servidor *Weblogic* da *Oracle* aparece em lugar de destaque, sendo o mais rápido a responder a pedidos (13,78 *milisegundos*). A solução da *IBM* continua a ter o pior resultado, com resposta média em 17,59 *milisegundos*.

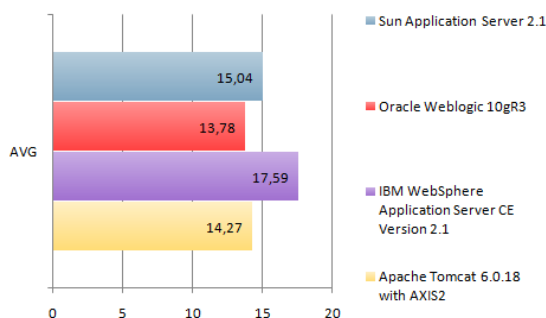


Figura 2.4: Teste AVG (Primeira Implementação).

Análise Tecnológica

A *Figura 2.5* apresenta o número de conexões estabelecidas em cinco minutos com cada um dos servidores. Este número é maior utilizando o servidor *Weblogic* da *Oracle*, aproximadamente, 198 mil conexões estabelecidas. Neste teste, mais uma vez, o servidor da *IBM* aparece como sendo o mais limitado, ficando-se pelas 157 mil conexões.

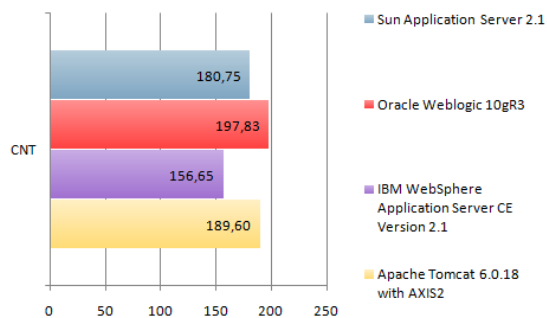


Figura 2.5: Teste CNT (Primeira Implementação).

Durante as conexões estabelecidas, o número de Mbytes trocados entre o servidor e cliente varia de solução para solução (ver *Figura 2.6*). O servidor da *IBM* destaca-se novamente pela negativa, na medida em que, sendo o servidor que estabeleceu menos conexões, aparece como uma das soluções que necessita de uma maior troca de informação entre os intervenientes, com um total de 50,34 Mbytes trocados. O servidor da *Sun* aparece como sendo o que necessita de menor troca de informação (40,15 Mbytes). No entanto, o *Weblogic* da *Oracle*, tendo por base a relação Mbytes/conexões, não fica muito atrás do concorrente anterior.

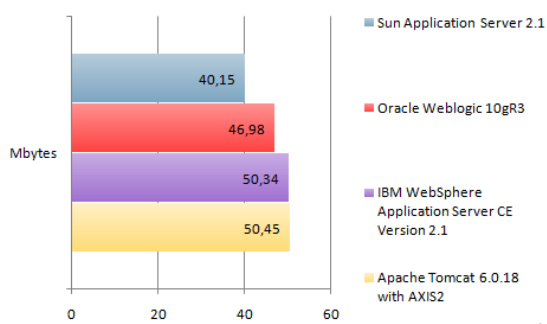


Figura 2.6: Teste Mbytes (Primeira Implementação).

Para além da informação contida nos gráficos apresentados, deve notar-se que o único servidor que originou erros de resposta foi o *Sun Application Server*, não tendo respondido a pedidos por 75 vezes num total de 180751 conexões.

Segunda Implementação

Os valores para as métricas definidas estão representados nas *Figuras 2.7, 2.8, 2.9 e 2.10*. Os resultados obtidos na segunda implementação são semelhantes e por isso as conclusões a tirar são as mesmas que na primeira implementação. No entanto, é de evidenciar que as diferenças entre as soluções já não são tão acentuadas.

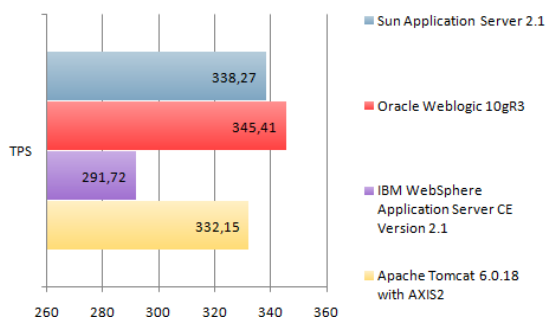


Figura 2.7: Teste TPS (Segunda Implementação).

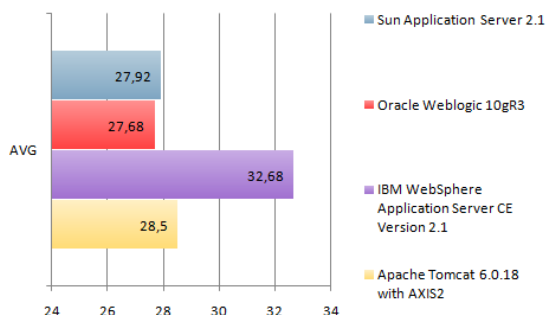


Figura 2.8: Teste AVG (Segunda Implementação).

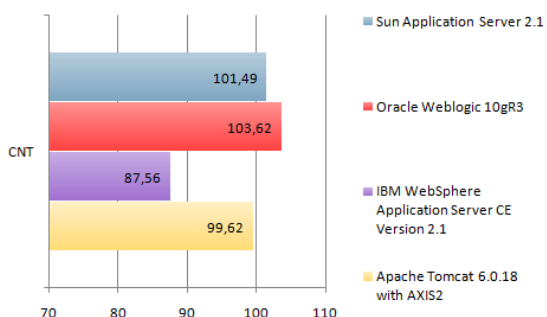


Figura 2.9: Teste CNT (Segunda Implementação).

Análise Tecnológica

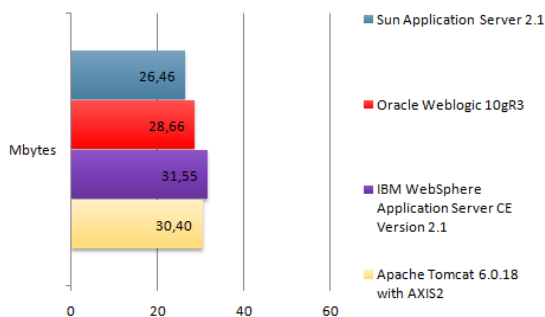


Figura 2.10: Teste Mbytes (Segunda Implementação).

De notar, que a solução da *Oracle* continua a ser a melhor do teste, e que a solução da *Sun* continua a originar erros, não respondendo a 225 pedidos num total de 101487 conexões.

Escolha

Tendo por base a análise dos resultados dos testes realizados e descritos anteriormente, a escolha recai sobre a solução da *Oracle*, o *Weblogic Server*.

Se, porventura, não fosse possível enveredar por um solução paga, o servidor *Apache Tomcat com Axis2* seria a melhor opção. A última opção é ocupada pela solução da *Sun*, com o *Sun Application Server*, na medida em que a possibilidade de não resposta por este servidor é maior do que em relação aos seus competidores, sendo efectivamente o único que nos testes realizados originou erros.

Capítulo 3

Conclusões

Sendo esta uma área tecnológica em grande expansão no momento, existem muitas opções que podem servir para o desenvolvimento do projecto em causa. Nesta análise foram consideradas as opções mais representativas existentes no mercado.

A *Figura 3.1* sintetiza as escolhas tecnológicas realizadas ao longo da análise tecnológica efectuada.

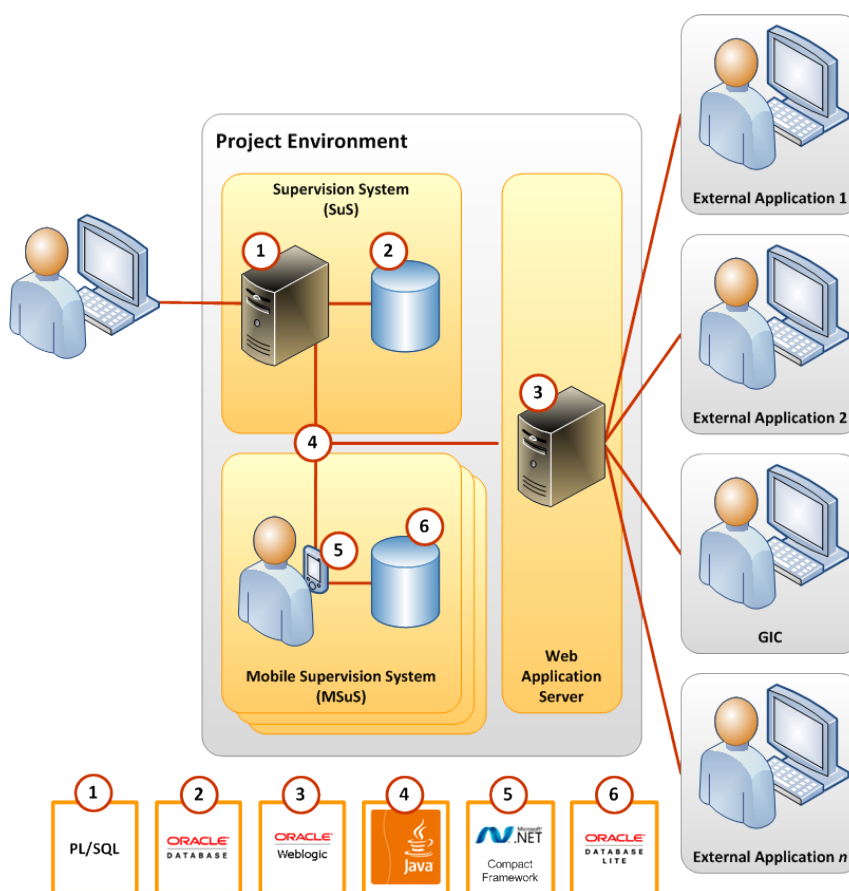


Figura 3.1: Escolhas Tecnológicas.

Referências

- [Cen09] .NET Framework Developer Center. .net compact framework, Maio 2009. Microsoft Corporation, disponível em <http://msdn.microsoft.com/en-us/netframework/aa497273.aspx>.
- [Cha04] David A. Chappell. *Enterprise Service Bus*. O'Reilly Media, Inc., 2004.
- [Cor08a] Microsoft Corporation. Sql server integration services with oracle database 10g. In Microsoft Corporation, editor, *SQL Server Technical Article*. Microsoft Corporation, May 2008.
- [Cor08b] Oracle Corporation. Oracle database lite 10gr3 technical white paper. In Oracle Corporation, editor, *An Oracle White Paper*. Oracle Corporation, August 2008.
- [Cor09a] International Business Machines Corp. Ibm - websphere application server community edition, Maio 2009. International Business Machines Corp. , disponível em [WebSphereApplicationServerCommunityEdition](http://www.ibm.com/websphere/applicationservercommunityedition/).
- [Cor09b] International Business Machines Corp. Ibm software - websphere, Maio 2009. International Business Machines Corp., disponível em <http://www-01.ibm.com/software/websphere/>.
- [Cor09c] Borland Software Corporation. Jdatastore developer support, Maio 2009. Borland Software Corporation, disponível em <http://info.borland.com/devsupport/jdatastore/>.
- [Cor09d] Microsoft Corporation. Microsoft sql server compact 3.5, Maio 2009. Microsoft Corporation, disponível em <http://www.microsoft.com/Sqlserver/2005/en/us/compact.aspx>.
- [Cor09e] Microsoft Corporation. Msdn: Microsoft development network, Fevereiro 2009. Microsoft Corporation, disponível em <http://msdn.microsoft.com/en-us/default.aspx>.
- [Cor09f] Microsoft Corporation. The official microsoft silverlight site, Maio 2009. Microsoft Corporation, disponível em <http://silverlight.net/>.
- [Cor09g] Microsoft Corporation. Windows virtual pc, Maio 2009. Microsoft Corporation, disponível em <http://www.microsoft.com/windows/virtual-pc/default.aspx>.
- [Cor09h] Oracle Corporation. Oracle and bea systems, Fevereiro 2009. Oracle Corporation, disponível em <http://www.oracle.com/bea/index.html>.

REFERÊNCIAS

- [Cor09i] Oracle Corporation. Oracle application server 10g standard edition, Fevereiro 2009. Oracle Corporation, disponível em http://www.oracle.com/appserver/standard_edition.html.
- [Cor09j] Oracle Corporation. Oracle weblogic server, Fevereiro 2009. Oracle Corporation, disponível em <http://www.oracle.com/technology/products/weblogic/index.html>.
- [Eve06] Software Everywhere. Interview with superwaba developer guilherme c. hazan, Fevereiro 2006. David Beers on Mobile Computing, disponível em <http://www.superwaba.com.br/pt/default.asp>.
- [Fou09a] The Apache Software Foundation. The apache software foundation - meritocracy in action, Fevereiro 2009. The Apache Software Foundation, disponível em <http://tomcat.apache.org/>.
- [Fou09b] The Apache Software Foundation. Apache tomcat, Fevereiro 2009. The Apache Software Foundation, disponível em <http://www.apache.org/>.
- [Fou09c] The Apache Software Foundation. Webservices - axis, Fevereiro 2009. The Apache Software Foundation, disponível em <http://tomcat.apache.org/>.
- [Ltd09] ARM Ltd. Arm - the architecture for the digital world, Maio 2009. ARM Ltd, disponível em <http://www.arm.com/>.
- [Med09] O'Reilly Media. metro: Home, Fevereiro 2009. Sun Microsystems, Inc., disponível em <https://metro.dev.java.net/>.
- [Net09a] Sun Developer Network. Java me - the most ubiquitous application platform for mobile devices, Maio 2009. Sun Microsystems, Inc., disponível em <http://java.sun.com/javame/index.jsp>.
- [Net09b] Sun Developer Network. Why java me platform?, Fevereiro 2009. Sun Microsystems, Inc., disponível em <http://java.sun.com/javame/overview/why.jsp>.
- [Qua09] Qualcomm. Qualcomm brew, Fevereiro 2009. QUALCOMM Incorporated, disponível em <http://brew.qualcomm.com/brew/en/>.
- [Ric06] Mark Richards. Infoq: The role of the enterprise service bus, Outubro 2006. C4Media Inc., disponível em <http://www.infoq.com/presentations/Enterprise-Service-Bus>.
- [Ris08] Ray Rischpater. *Beginning Java ME Platform*. Springer-Verlag, Inc., 2008.
- [SM09] Inc. Sun Microsystems. Sun glassfish enterprise server, Maio 2009. Sun Microsystems, Inc., disponível em <http://www.sun.com/software/products/appsrvr/>.
- [Sup09] SuperWaba. Superwaba: The real power of mobile computing, Fevereiro 2009. SuperWaba, disponível em <http://www.superwaba.com.br/pt/default.asp>.
- [Whi08] Jim White. Start your engines: Mobile application development, Abril 2008. Jupitermedia Corporation, disponível em <http://www.devx.com/SpecialReports/Article/37693/0/page/1>.

Anexo A

Teste de Desempenho - Servidores de Aplicações Web

A.1 Hardware

O hardware utilizado para proceder aos testes dos servidores de aplicações web foi o mesmo para correr quer o cliente, quer o servidor.

As características são as seguintes:

- **Modelo** Dell OptiPlex 755
- **Processador** Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz
- **Memória RAM** 3325 MB

De notar que o servidor correu numa máquina virtual, para a qual foram alocados 1024 MB de RAM.

A.2 Software

O software utilizado para executar os testes no cliente é diferente do utilizado no servidor. Todo o software foi instalado com os valores definidos por omissão sem qualquer tipo de melhoramentos por parte do testador.

A.2.1 Servidor

O servidor foi instalado numa máquina virtual numa máquina com o hardware especificado anteriormente. O software para criar esta máquina virtual foi o *Microsoft Virtual PC 6.0.192.0* [Cor09g].

O software instalado para a execução do teste foi:

- **Sistema Operativo** Windows Server 2003 Enterprise Edition com Service Pack 1;
- **Servidores de Aplicações**;
 - Apache Tomcat 6.0.18 com AXIS2;

- IBM WebSphere Application Server CE Version 2.1;
- Oracle Weblogic 10gR3;
- Sun Application Server 2.1.

- **Java JDK 1.6.0_12.**

A.2.2 Cliente

O software instalado no cliente para a execução do teste foi:

- **Sistema Operativo** Windows Vista Business N;
- **Aplicação para execução de testes** eviware soapUI 2.5.1;
- **Java JDK 1.6.0_12.**

A.3 Implementação

A implementação dos testes foi realizada recorrendo à ferramenta de desenvolvimento *Oracle JDeveloper 11.1.1.0.1* para criação dos ficheiros necessários para *deployment* nos servidores IBM WebSphere, Oracle Weblogic e Sun Application Server, enquanto que para o caso do Apache Tomcat, que pede tipos de ficheiro específicos para a sua implementação, foi necessário recorrer à ferramenta NetBeans IDE 6.5.

A.3.1 Código dos Web Services

Foram criados Web services para testar os servidores de aplicações. O código destes Web services é o seguinte:

```
1 import java.io.BufferedReader;
2 import java.io.BufferedWriter;
3 import java.io.FileInputStream;
4 import java.io.FileNotFoundException;
5 import java.io.FileOutputStream;
6 import java.io.IOException;
7 import java.io.InputStreamReader;
8 import java.io.OutputStreamWriter;
9
10 import javax.jws.WebMethod;
11 import javax.jws.WebService;
12
13 @WebService
14 public class ServerTester {
15     public ServerTester () {
16     }
17
18     @WebMethod
19     public String answeringTester () {
20         return "answering!";
21     }
22
23     @WebMethod
24     public String processingTester () {
```

```

25
26     FileOutputStream fos;
27     try {
28         fos = new FileOutputStream("/testingFile");
29     } catch (FileNotFoundException e) {
30         return "error in opening file for writing!";
31     }
32     BufferedWriter out;
33     out = new BufferedWriter(new OutputStreamWriter(fos));
34     try {
35         out.write("This was written to the file!");
36     } catch (IOException e) {
37         return "error in writing to file!";
38     }
39     try {
40         out.close();
41     } catch (IOException e) {
42         return "error in closing file after writing!";
43     }
44     FileInputStream fis;
45     try {
46         fis = new FileInputStream("/testingFile");
47     } catch (FileNotFoundException e) {
48         return "error in opening file for reading!";
49     }
50     BufferedReader in;
51     in = new BufferedReader(new InputStreamReader(fis));
52     String answer;
53     try {
54         answer = in.readLine();
55     } catch (IOException e) {
56         return "error in reading file!";
57     }
58     try {
59         in.close();
60     } catch (IOException e) {
61         return "error in closing file after reading!";
62     }
63     return answer + " and it is answering!";
64 }
65 }
66 }

```

O método *answeringTester()* corresponde ao teste referente à **Primeira Implementação** e o *processingTester()* corresponde à **Segunda Implementação**.

Em relação à configuração do soapUI para correr os testes de carregamento, os valores utilizados foram os seguintes:

- **Limit** 300 seconds;
- **Threads** 10;
- **Test Delay** 0;
- **Random** 0.5;

A.4 Resultados

Os resultados referentes a cada um dos servidores estão ilustrados nas *Tabelas A.1, A.2, A.3 e A.4*. Cada tabela é referente a um dos servidores testados nas duas implementações de *Web services* realizadas. Cada tabela apresenta os tempos mínimo e máximo de resposta e sua média, qual o tempo da última conexão, número de conexões realizadas, número de *threads* por segundo (tps), *MBytes* trocados, *bits* por segundo e número de erros em respostas do servidor.

Tabela A.1: Teste de Desempenho: Apache Tomcat 6.0.18 com AXIS2

Implementação	min	max	avg	last	cnt	tps	Mbytes	bps	err
1	3	444	14,27	18	189602	632,18	50,44838	176379	0
2	3	5156	28,5	21	99618	332,15	30,40086	106289	0

Tabela A.2: Teste de Desempenho: IBM WebSphere Application Server CE Version 2.1

Implementação	min	max	avg	last	cnt	tps	Mbytes	bps	err
1	2	673	17,59	48	156645	522,23	50,34386	175991	0
2	3	3137	32,68	197	87564	291,72	31,55205	110222	0

Tabela A.3: Teste de Desempenho: Oracle Weblogic 10gR3

Implementação	min	max	avg	last	cnt	tps	Mbytes	bps	err
1	3	564	13,78	165	197830	659,23	46,97768	164149	0
2	6	4516	27,68	23	103620	345,41	28,65765	100169	0

Tabela A.4: Teste de Desempenho: Sun Application Server 2.1

Implementação	min	max	avg	last	cnt	tps	Mbytes	bps	err
1	3	640	15,04	10	180751	602,45	40,14731	140391	75
2	4	465	27,92	26	101487	338,27	26,46045	92481	225

A.5 Resumo ou Conclusão

Os resultados obtidos mostram que o servidor *Oracle Weblogic 10gR3* foi o **Melhor do Teste**, sendo que numa vista mais económica, o servidor *Apache Tomcat 6.0.18 com AXIS2* apresenta a melhor relação qualidade/preço.

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Aplicação de Registo de Ocorrências para Dispositivos Móveis

Hugo Manuel Zenha de Pinho

Relatório de Especificação de Requisitos

Mestrado Integrado em Engenharia Informática e Computação

Orientador: João Manuel Paiva Cardoso

Doutor em Engenharia Electrotécnica e Computadores

29 Junho 2009

Conteúdo

1	Introdução	1
1.1	Objectivo	1
1.2	Âmbito	1
1.3	Destinatários do Documento	2
1.4	Estrutura do Documento	2
2	Descrição Geral	3
2.1	Contexto do Produto	3
2.1.1	Arquitectura Global	3
2.1.2	Ambiente de Operação	3
2.2	Funções do Sistema	4
2.2.1	SuS - Supervision System	4
2.2.2	MSuS - Mobile Supervision System	5
2.3	Características dos Utilizadores	5
2.4	Restrições	5
2.5	Pressupostos e Dependências	6
3	Casos de Uso	7
3.1	SuS - Supervision System	7
3.1.1	Administrador de Sistema	7
3.1.2	Pivot de Serviços de Fiscalização	16
3.1.3	Agente Fiscalizador	20
3.2	MSuS - Mobile Supervision System	21
3.3	Comum	22
4	Requisitos Específicos	24
4.1	Requisitos Externos	24
4.1.1	Interface com o Utilizador	24
4.1.2	Interfaces de Hardware	24
4.1.3	Interfaces de Software	24
4.1.4	Interfaces de Comunicação	25
4.2	Requisitos Funcionais	26
4.2.1	SuS - Supervision System	26
4.2.2	MSuS - Mobile Supervision System	50
4.2.3	Comum	52
4.3	Requisitos Não-Funcionais	53
4.4	Requisitos de Documentação	55

CONTEÚDO

Referências

56

Lista de Figuras

2.1	Visão Física do Sistema.	4
3.1	Diagrama de casos de uso do <i>SuS</i> - Administrador de Sistema.	8
3.2	Diagrama de casos de uso do <i>SuS</i> - Pivot de Serviços de Fiscalização.	16
3.3	Diagrama de casos de uso do <i>SuS</i> - Agente Fiscalizador.	20
3.4	Diagrama de casos de uso do <i>MSuS</i>	21

Lista de Tabelas

3.1	Tratar Área	9
3.2	Tratar Característica	9
3.3	Tratar Categoria	10
3.4	Tratar Lista	10
3.5	Tratar Elemento	11
3.6	Tratar Membro	11
3.7	Tratar Perfil	12
3.8	Tratar Permissão	12
3.9	Tratar Tipo de Ocorrência	13
3.10	Tratar Tipo de Tarefa	13
3.11	Tratar Topônimo	14
3.12	Tratar Número de Polícia	14
3.13	Tratar Unidade Orgânica	15
3.14	Tratar Zona	15
3.15	Tratar Actividade	17
3.16	Tratar Tarefa	17
3.17	Tratar Alocação	18
3.18	Tratar Conhecimento	18
3.19	Tratar Equipa	19
3.20	Validar Ocorrência Reportada	19
3.21	Tratar Ocorrência	20
3.22	Tratar Ocorrência	21
3.23	Pesquisar Informação Interna	22
3.24	Pesquisar Informação Externa	23
4.1	Requisito - Adicionar Área	26
4.2	Requisito - Editar Área	26
4.3	Requisito - Remover Área	27
4.4	Requisito - Adicionar Característica	27
4.5	Requisito - Editar Característica	27
4.6	Requisito - Remover Característica	28
4.7	Requisito - Adicionar Categoria	28
4.8	Requisito - Editar Categoria	28
4.9	Requisito - Remover Categoria	29
4.10	Requisito - Adicionar Lista	29
4.11	Requisito - Editar Lista	29
4.12	Requisito - Remover Lista	30
4.13	Requisito - Adicionar Elemento	30

LISTA DE TABELAS

4.14	Requisito - Editar Elemento	30
4.15	Requisito - Remover Elemento	31
4.16	Requisito - Adicionar Membro	31
4.17	Requisito - Editar Membro	31
4.18	Requisito - Remover Membro	32
4.19	Requisito - Adicionar Perfil	32
4.20	Requisito - Editar Perfil	32
4.21	Requisito - Remover Perfil	33
4.22	Requisito - Adicionar Permissão	33
4.23	Requisito - Editar Permissão	33
4.24	Requisito - Remover Permissão	34
4.25	Requisito - Adicionar Tipo de Ocorrência	34
4.26	Requisito - Simular Tipo de Ocorrência	34
4.27	Requisito - Editar Tipo de Ocorrência	35
4.28	Requisito - Remover Tipo de Ocorrência	35
4.29	Requisito - Adicionar Tipo de Tarefa	36
4.30	Requisito - Editar Tipo de Tarefa	36
4.31	Requisito - Remover Tipo de Tarefa	36
4.32	Requisito - Adicionar Topônimo	37
4.33	Requisito - Editar Topônimo	37
4.34	Requisito - Remover Topônimo	37
4.35	Requisito - Adicionar Número de Polícia	38
4.36	Requisito - Editar Número de Polícia	38
4.37	Requisito - Remover Número de Polícia	38
4.38	Requisito - Adicionar Unidade Orgânica	39
4.39	Requisito - Editar Unidade Orgânica	39
4.40	Requisito - Remover Unidade Orgânica	39
4.41	Requisito - Adicionar Zona	40
4.42	Requisito - Editar Zona	40
4.43	Requisito - Remover Zona	40
4.44	Requisito - Adicionar Localização GPS à Zona	41
4.45	Requisito - Adicionar Actividade	41
4.46	Requisito - Editar Actividade	41
4.47	Requisito - Remover Actividade	42
4.48	Requisito - Cancelar Actividade	42
4.49	Requisito - Adicionar Tarefa	43
4.50	Requisito - Editar Tarefa	43
4.51	Requisito - Remover Tarefa	43
4.52	Requisito - Adicionar Localização GPS à Tarefa	44
4.53	Requisito - Adicionar Alocação	44
4.54	Requisito - Editar Alocação	44
4.55	Requisito - Remover Alocação	45
4.56	Requisito - Adicionar Conhecimento	45
4.57	Requisito - Editar Conhecimento	45
4.58	Requisito - Remover Conhecimento	46
4.59	Requisito - Adicionar Equipa	46
4.60	Requisito - Editar Equipa	46
4.61	Requisito - Remover Equipa	47

LISTA DE TABELAS

4.62	Requisito - Validar Ocorrência Reportada	47
4.63	Requisito - Devolver Ocorrência Reportada	47
4.64	Requisito - Eliminar Ocorrência Reportada	48
4.65	Requisito - Arquivar Ocorrência Reportada	48
4.66	Requisito - Adicionar Ocorrência	49
4.67	Requisito - Editar Ocorrência	49
4.68	Requisito - Remover Ocorrência	49
4.69	Requisito - Imprimir Ocorrência	50
4.70	Requisito - Adicionar Ocorrência	50
4.71	Requisito - Editar Ocorrência	51
4.72	Requisito - Remover Ocorrência	51
4.73	Requisito - Adicionar Localização GPS à Ocorrência	51
4.74	Requisito - Imprimir Ocorrência	51
4.75	Requisito Instável - Pesquisar Informação Interna	52
4.76	Requisito Instável - Pesquisar Informação Externa	52
4.77	Requisitos Não-Funcionais	53
4.78	Requisitos de Documentação	55

Abreviaturas e Símbolos

GIC	Gestão Integrada de Contra-Ordenações
MSuS	Mobile Supervision System
PDA	Personal Digital Assistant
PSI	Projecto de Sistemas de Informação - FEUP
SuS	Supervision System
W3C	World Wide Web Consortium

Capítulo 1

Introdução

1.1 Objectivo

Este documento tem como finalidade explicitar com o maior detalhe possível todas as funcionalidades, casos de uso e requisitos do sistema a desenvolver no âmbito do projecto final de curso do Hugo Manuel Zenha de Pinho. Tal é necessário para que quaisquer dúvidas que possam subsistir sejam convenientemente esclarecidas e dissipadas.

O presente documento é parte integrante do projecto a desenvolver para o *Projecto de Sistemas de Informação - FEUP* (conhecido também por *PSI*), que tem por objectivo a criação de um Sistema de Gestão Centralizada de Fiscalização Municipal, aplicado em dispositivos móveis.

1.2 Âmbito

Este documento pretende descrever os requisitos, tanto funcionais como não-funcionais, para a implementação de um Sistema de Gestão de Centralizada de Fiscalização Municipal, aplicado em dispositivos móveis.

O sistema descrito neste documento e as suas funcionalidades decorrem da necessidade, por parte do mercado, da existência de um sistema que permita aos *Agentes Fiscalizadores Municipais* poderem utilizar as potencialidades dos dispositivos móveis para tornar o seu trabalho mais eficaz e eficiente.

O sistema consiste em dois sub-sistemas:

- Sistema de Administração da Fiscalização (que será denominado por *Supervision System - SuS*), é o *Back-Office*, permitirá todas as funcionalidades de administração, como adicionar um novo dispositivo ao sistema;

- Sistema de Fiscalização Móvel (que será denominado por *Mobile Supervision System - MSuS*), que será o sistema a que os *Agentes Fiscalizadores* terão acesso para proceder ao seu trabalho.

1.3 Destinatários do Documento

Pretende-se que o documento seja um elo de ligação entre o cliente, o estagiário e outros possíveis futuros desenvolvedores. Devido a tal pretensão, a linguagem utilizada ao longo deste documento será uma perfeitamente ao alcance de uma pessoa dentro do contexto da fiscalização municipal, tanto ao nível do reporte de ocorrências como ao nível da gestão.

É importante que o documento seja o mais claro possível e que permita aos desenvolvedores perceberem com relativa facilidade as necessidades do cliente. Atingindo este objectivo, os desenvolvedores conseguem construir um produto que vai de encontro às necessidades reais dos utilizadores, num tempo mais curto de implementação.

1.4 Estrutura do Documento

Este documento encontra-se estruturado tendo em vista uma compreensão rápida de todas as especificidades do sistema a desenvolver. Para tal, encontra-se dividido nas seguintes secções:

- **2. Descrição Geral** - nesta secção apresenta-se uma contextualização do produto, sendo resumida a sua arquitectura e o seu ambiente de funcionamento, seguindo-se a apresentação das funções do sistema. A descrição prossegue com a caracterização dos utilizadores, as restrições e os pressupostos e dependências do produto;
- **3. Casos de Uso** - são apresentados os casos de uso do sistema a desenvolver nesta secção, i.e., quais as funcionalidades que os utilizadores esperam que este disponibilize;
- **4. Requisitos Específicos** - aqui aparecem detalhados todos os requisitos do sistema. Divide-se em requisitos externos, onde aparecem as interfaces, requisitos funcionais, requisitos não-funcionais, requisitos de documentação e cumprimento de standards, demonstrando a política a seguir no desenvolvimento do sistema.

Capítulo 2

Descrição Geral

2.1 Contexto do Produto

A solução pretendida insere-se no contexto da gestão de ocorrências e tarefas inerentes à actividade de fiscalização em autarquias (contra-ordenações, ocorrências na via pública, gestão urbanística, licenciamentos, *etc*). Advém da necessidade da *Câmara Municipal do Porto* em fornecer aos seus *Agentes Fiscalizadores* uma ferramenta simples, eficaz, eficiente e fiável, que lhes permita melhorar a qualidade das acções de fiscalização levadas a cabo no município.

2.1.1 Arquitectura Global

A arquitectura global baseia-se numa estrutura cliente-servidor em que os mecanismos de comunicação são disponibilizados numa camada intermédia, através da utilização de *Web services*. A comunicação realizada entre os dois sistemas a desenvolver e outros que poderão, posteriormente, interagir também com estes, será realizada através da disponibilização e procura de *Web services*.

O *SGIFM* é constituído por dois sub-sistemas, o *SuS* e o *MSuS*, que comunicarão através de mecanismos disponibilizados por um servidor de aplicações. Este servidor permitirá não só comunicações internas ao sistema, mas também a comunicação deste com aplicações externas. De notar que mesmo o caso especial da *Gestão Integrada de Contra-Ordenações* (mais sucintamente *GIC*) [dSdI06], sistema já em utilização na *CMP*, é vista como mais uma aplicação externa, mesmo que no futuro se espere que a informação contida no *SGIFM* seja utilizada por esta. A *Figura 2.1* apresenta, de uma forma elucidativa, a visão física do sistema.

2.1.2 Ambiente de Operação

Do lado do Sistema de Fiscalização (designado por *Supervision System - SuS*), o ambiente em servidor Windows será o ambiente de eleição. Este disponibiliza todas as especificidades

Descrição Geral

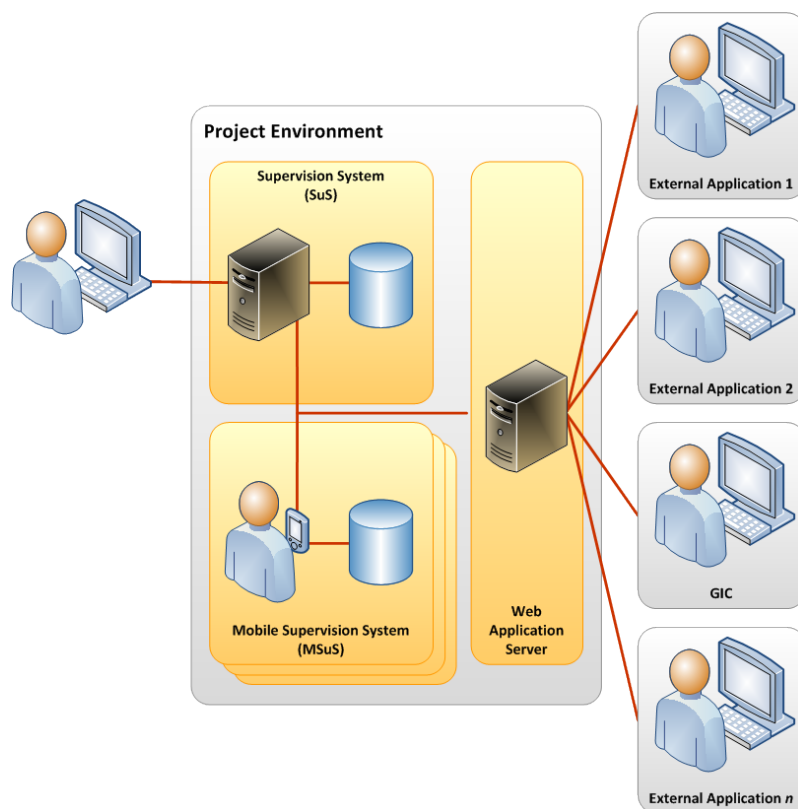


Figura 2.1: Visão Física do Sistema.

necessárias para a utilização do Sistema de Fiscalização.

O Sistema de Fiscalização Móvel (designado por *Mobile Supervision System - MSuS*) enquadrar-se-á num contexto de utilização de âmbito generalizado, em plataformas móveis (vulgo *PDA* ou *SmartPhones*) com o sistema operativo Windows Mobile 6 ou superior, podendo ser adaptado, posteriormente, às especificidade de cada um.

De notar que o sistema deve possibilitar a ligação simultânea de múltiplos clientes a um servidor. Deve ser também possível a gestão e configuração simultânea da aplicação a nível do servidor, por múltiplos utilizadores.

2.2 Funções do Sistema

O sistema em questão, o *SGIFM*, divide-se em dois sub-sistemas, o sistema de *Back-Office*, o *SuS*, e o sistema para executar no dispositivo móvel, o *MSuS*. Assim sendo, as funções dependem do sub-sistema em questão.

2.2.1 SuS - Supervision System

O *SuS*, *Back-Office* do sistema de fiscalização, deverá disponibilizar as seguintes funções:

- **Consultas** - deverão ser disponibilizadas consultas às ocorrências reportadas e mecanismos de consulta a aplicações externas;
- **Controlo/Validação das Ocorrências** - deverão ser disponibilizadas funcionalidades com vista ao controlo e validação das ocorrências reportadas pelos agentes fiscalizadores;
- **Parametrizações do Sistema** - deverá permitir ao Administrador do Sistema configurar este sistema de forma a satisfazer as necessidades do ambiente de funcionamento;
- **Planeamento de Tarefas** - o sistema deverá disponibilizar mecanismos para planear o trabalho dos agentes fiscalizadores, sendo que estes apenas têm de consultar as suas tarefas no dispositivo móvel.

2.2.2 MSuS - Mobile Supervision System

O sistema a executar nos dispositivos móveis dos agentes fiscalizadores, o *MSuS*, deverá disponibilizar as seguintes funcionalidades:

- **Consultas** - deverão ser disponibilizadas consultas às ocorrências reportadas e mecanismos de consulta a aplicações externas;
- **Registo de Ocorrências** - sendo este o objectivo principal de todo o produto, o *MSuS* deverá disponibilizar o mecanismo de registo de ocorrências por parte do agente fiscalizador.

2.3 Características dos Utilizadores

A aplicação a desenvolver deverá envolver três tipos distintos de utilizadores:

- **Agentes Fiscalizadores**, que serão os principais actores deste sistema;
- **Pivots dos Serviços Fiscalizadores**, que terão a competência de gerir as ocorrências do Serviço Fiscalizador a que pertencem, assim como gerir o trabalho dos agentes;
- **Administradores do Sistema**, responsáveis pela parametrização e gestão das funcionalidades a que os Agentes Fiscalizadores e *Pivots* têm acesso a partir dos dispositivos móveis (caso *MSuS*), ou da interface web (caso *SuS*).

2.4 Restrições

As únicas restrições a apontar são os factos do *MSuS* ser implementado para correr em dispositivos móveis com sistema operativo *Windows* e do *SuS* ser implementado em tecnologia *Oracle*, especificamente *PL/SQL* [McL08] sobre *Oracle Database* [Lon09].

No âmbito do projecto, a primeira restrição não é vista como problema, pois os dispositivos existentes/a comprar terão o *Windows* como sistema operativo, no entanto, se no futuro se mudar de plataforma terá de haver um esforço adicional na criação da nova versão do *MSuS*.

2.5 Pressupostos e Dependências

Pressupõe-se que:

- o dispositivo móvel está ligado para assegurar o funcionamento do *MSuS*;
- o sistema operativo do dispositivo móvel sobre o qual o *MSuS* está a executar possui a capacidade para garantir o funcionamento deste último sem falhas;
- os *Agentes Fiscalizadores*, *Pivots dos Serviços Fiscalizadores* e *Administradores do Sistema* possuem os conhecimentos necessários para utilizar um browser Web;
- os *Agentes Fiscalizadores* possuem os conhecimentos necessários para utilizar uma aplicação num dispositivo móvel;
- o *SuS* será acedido por um dos seguintes *browsers web*:
 - *Internet Explorer 8* [Cor09];
 - *Mozilla Firefox 3* [eMF09];
 - qualquer *browser web* compatível com os standards, tal como definidos pela entidade W3C [W3C09].
- o *SuS* possui capacidade de armazenamento de dados considerada ilimitada;
- a ligação entre *SuS* e *MSuS* estará assegurada.

Capítulo 3

Casos de Uso

Nesta secção são especificadas as funcionalidades que os dois sistemas, *SuS* e *MSuS*, devem disponibilizar aos seus utilizadores. Estas funcionalidades encontram-se representadas recorrendo a diagramas de caso de uso para uma maior facilidade na leitura do documento. De notar que ambos os sub-sistemas aqui referidos contêm funcionalidades específicas, embora exista, no entanto, um ramo comum aos dois.

As tabelas que aparecem durante esta secção descrevem cada caso de uso introduzido, tendo em consideração qual o actor que a ele tem acesso, prioridade de desenvolvimento, assim com pressupostos, pré e pós-condições para as funcionalidades que do caso em questão advêm. A prioridade pode ser *Essencial*, *Elevada*, *Média* ou *Baixa*. O sistema de reporte de ocorrências funciona se os de prioridade *Essencial* estiverem realizados. Contudo, para estar pronto para ser utilizado no terreno é necessário que os de prioridade *Elevada* também o estejam.

3.1 SuS - Supervision System

No sistema *SuS* existem três actores: o *Administrador de Sistema*, o *Pivot de Serviços de Fiscalização* e o *Agente Fiscalizador*. Estes actores irão utilizar as funcionalidades disponibilizadas pelo *SuS*. O *Administrador de Sistema* é o responsável pela parametrização dos valores do sistema e dos serviços disponibilizados por este. Cabe ao *Pivot de Serviços de Fiscalização* gerir e validar o trabalho do *Agente Fiscalizador*, sendo que este último pode pesquisar informação contida no sistema e reportar ocorrências.

3.1.1 Administrador de Sistema

Como já referido anteriormente, o *Administrador de Sistema* é responsável pela parametrização do sistema. Para tal, o leque de casos de uso a que tem acesso é bastante extenso, desde o tratamento de informação que flui no sistema à edição de permissões dos seus utilizadores. A *Figura*

Casos de Uso

3.1 apresenta os casos de uso que representam as funcionalidades do *SuS* a que este tem acesso. De seguida é explicado cada um destes casos de uso.

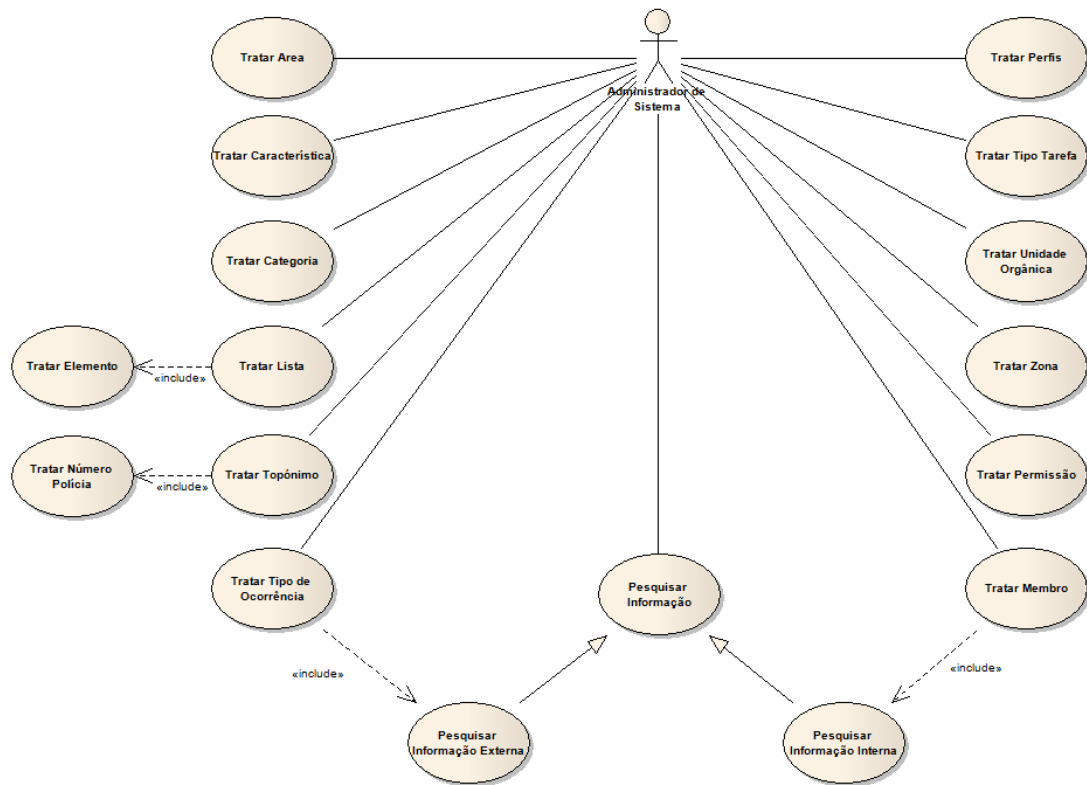


Figura 3.1: Diagrama de casos de uso do *SuS* - Administrador de Sistema.

Caso de Uso Tratar Área

Este caso de uso permite adicionar, editar e remover áreas do sistema. Uma área é um conjunto de outras áreas e/ou de conjunto de tipos de ocorrência que podem ser reportados através desta solução.

A *Tabela 3.1* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.1: Tratar Área

Identificador	uc_sus01
Descrição	Permite Adicionar, Editar e Remover áreas.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Característica

A adição, edição e remoção de características na solução é providenciada através deste caso de uso. Uma característica é um dado sobre um tipo de ocorrência em particular, não sendo por isso geral a todos os tipos de ocorrência existentes na solução. Por exemplo, todos os tipos de ocorrência têm um código associado, mas nem todos necessitam um registo de uma matrícula de um veículo.

A *Tabela 3.2* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.2: Tratar Característica

Identificador	uc_sus02
Descrição	Providencia as acções de adição, edição e remoção de características.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Categoria

Cada membro - qualquer utilizador do sistema - pertence a uma categoria profissional. Assim sendo, este caso de uso permite a adição, edição e remoção de tais categorias do sistema.

A *Tabela 3.3* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.3: Tratar Categoria

Identificador	uc_sus03
Descrição	Permite adicionar, editar e remover categorias profissionais.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Lista

Cada característica de um tipo de ocorrência pode ser de vários tipos, pode ser texto, número, entre outros. No entanto, é necessário providenciar ao *Agente Fiscalizador* listas de selecção de dados onde este escolhe a opção que irá ser o valor de tal característica. A lista aparece como sendo o mecanismo para que tal seja possível. Este caso de uso ira permitir adicionar, editar e remover listas do sistema.

A *Tabela 3.4* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.4: Tratar Lista

Identificador	uc_sus04
Descrição	Permite ao utilizador adicionar, editar e remover listas.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso *Tratar Elemento*

No seguimento do caso de uso anterior, a lista pode ser composta por vários elementos que dão origem a este caso de uso. Adição, edição e remoção são as acções providenciadas por este caso de uso.

A *Tabela 3.5* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.5: Tratar Elemento

Identificador	uc_sus05
Descrição	Acções de adição, edição e remoção de elementos do sistema.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso *Tratar Membro*

Um membro é visto no sistema como um utilizador desta. A gestão da informação relacionada com estes é realizada através deste caso de uso, providenciando as acções de adição, edição e remoção de membros.

A *Tabela 3.6* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.6: Tratar Membro

Identificador	uc_sus06
Descrição	Permite adicionar, editar e remover membros do sistema.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Perfil

Na solução as permissões são atribuídas a um perfil. Este caso de uso permite adicionar, editar e remover perfis.

A *Tabela 3.7* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.7: Tratar Perfil

Identificador	uc_sus07
Descrição	Adição, edição e remoção são providenciadas por este caso de uso.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Permissão

Cada perfil pode ter a si associado vários tipos de permissão sobre as funcionalidades disponibilizadas nesta solução. Este caso de uso permite a gestão destas permissões providenciando as acções de adição, edição e remoção.

A *Tabela 3.8* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.8: Tratar Permissão

Identificador	uc_sus08
Descrição	Permite adicionar, editar e remover permissões.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Tipo de Ocorrência

Uma das partes mais cruciais do sistema de *Back-Office*, os tipos de ocorrência transmitem o que pode ser registado como ocorrência por parte do *Agente Fiscalizador*. Este caso de uso permite adicionar, editar e remover tal informação.

A Tabela 3.9 apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.9: Tratar Tipo de Ocorrência

Identificador	uc_sus09
Descrição	Permite adicionar, editar e remover tipos de ocorrência.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Tipo de Tarefa

Uma actividade pode ser adicionada a um membro. Esta actividade é um conjunto de tarefas que podem derivar de vários tipos. Este caso de uso permite a adição, edição e remoção de tais tipos de tarefa.

A Tabela 3.10 apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.10: Tratar Tipo de Tarefa

Identificador	uc_sus10
Descrição	Permite a adição, edição e remoção de tipos de tarefa.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Topónimo

Um topónimo é um nome de um lugar, sítio ou localidade, que neste caso será utilizado para guardar informação sobre uma morada existente no município. Este caso de uso permite a gestão destes topónimos.

A Tabela 3.11 apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.11: Tratar Topónimo

Identificador	uc_sus11
Descrição	Permite adicionar, editar e remover topónimos do sistema.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Número de Polícia

Cada topónimo pode ter a ele associado vários números de polícia. Este número é utilizado para uma maior eficiência dos serviços. A gestão destes números é providenciada por este caso de uso.

A Tabela 3.12 apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.12: Tratar Número de Polícia

Identificador	uc_sus12
Descrição	Providencia as acções de adição, edição e remoção de número de polícia.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Unidade Orgânica

Cada membro faz parte de uma unidade orgânica dentro da organização do município. Para tal este caso de uso permite a adição, edição e remoção de tais unidades.

A *Tabela 3.13* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.13: Tratar Unidade Orgânica

Identificador	uc_sus13
Descrição	Permite adicionar, editar e remover unidades orgânicas.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Zona

O município está dividido por zonas de fiscalização. Este caso de uso permite a adição, edição e remoção de zonas.

A *Tabela 3.14* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.14: Tratar Zona

Identificador	uc_sus14
Descrição	Permite adicionar, editar e remover zonas.
Actor	<i>Administrador do Sistema.</i>
Prioridade	Elevada.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

O caso de uso Pesquisar Informação é explicado na *Secção 3.3*.

3.1.2 Pivot de Serviços de Fiscalização

Os *Pivots de Serviços de Fiscalização* são responsáveis pela gestão das ocorrências reportadas pelos *Agentes Fiscalizadores*. Compete aos *Pivots* zelar pelo bom trabalho desses *Agentes*.

A *Figura 3.2* apresenta as funcionalidades, através de casos de uso, a que o *Pivot* terá acesso no sistema de *Back-Office*.

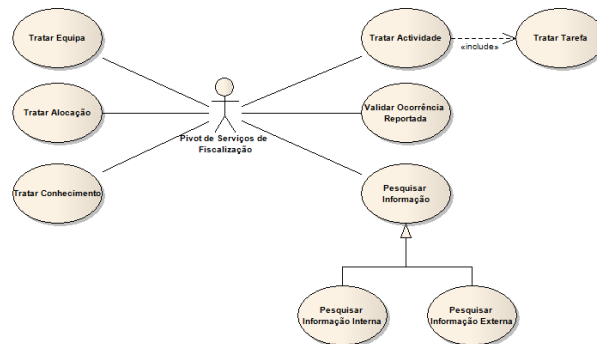


Figura 3.2: Diagrama de casos de uso do SuS - Pivot de Serviços de Fiscalização.

De seguida são explicados cada um destes casos.

Caso de Uso Tratar Actividade

Cada membro associado ao sistema pode ser incumbido da realização de uma actividade. Sendo o Pivot o responsável pela gestão do trabalho dos Agentes Fiscalizadores, este pode, através deste caso de uso, adicionar, editar, remover e cancelar actividades.

A Tabela 3.15 apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.15: Tratar Actividade

Identificador	uc_sus15
Descrição	Permite adicionar, editar, remover e cancelar actividades.
Actor	<i>Pivot dos Serviços Fiscalizadores.</i>
Prioridade	Média.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Tarefa

Cada actividade criada é composta por tarefas. Sendo assim, o pivot pode criar, editar e remover tais tarefas.

A Tabela 3.16 apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.16: Tratar Tarefa

Identificador	uc_sus16
Descrição	Permite adição, edição e remoção de tarefas.
Actor	<i>Pivot dos Serviços Fiscalizadores.</i>
Prioridade	Média.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Alocação

Cada membro poderá fazer parte de uma equipa. Esta equipa pode ser alocada a uma zona do município para exercer o serviço de fiscalização. Um conjunto de alocações transmite o horário de trabalho dos agentes fiscalizadores no terreno. Este caso de uso expõe a possibilidade de adição, edição e remoção de tal alocação.

A *Tabela 3.17* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.17: Tratar Alocação

Identificador	uc_sus17
Descrição	A adição, edição e remoção são possibilitadas por este caso de uso.
Actor	<i>Pivot dos Serviços Fiscalizadores.</i>
Prioridade	Média.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Conhecimento

Para cada tipo de ocorrência ou conjunto destas (*Área*) podem existir linhas de orientação para o trabalho de fiscalização. Estas aparecem sobre a forma de um banco de conhecimento. Este caso de uso debruça-se sobre a possibilidade de adição, edição e remoção de tais linhas de orientação.

A *Tabela 3.18* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.18: Tratar Conhecimento

Identificador	uc_sus18
Descrição	Possibilidade de adição, edição e remoção de linhas de orientação (banco de conhecimento)
Actor	<i>Pivot dos Serviços Fiscalizadores.</i>
Prioridade	Média.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Tratar Equipa

Um membro, que tenha perfil de *Agente Fiscalizador*, pode ser associado a uma equipa para proceder ao trabalho de fiscalização numa zona do município.

A *Tabela 3.19* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.19: Tratar Equipa

Identificador	uc_sus19
Descrição	Permite adicionar, editar e remover equipas do sistema.
Actor	<i>Pivot dos Serviços Fiscalizadores.</i>
Prioridade	Média.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

Caso de Uso Validar Ocorrência Reportada

Um agente quando reporta uma ocorrência pode acontecer uma de várias situações: pode ser submetida no sistema imediatamente, pode necessitar de validação por parte do pivot ou pode ser necessário editá-la, pelo *Agente*, quando mais informação sobre esta estiver disponível. No caso de necessitar de validação, o *Pivot* pode submeter a ocorrência, validando-a, pode devolver para edição, por falta de informações necessárias, pode eliminá-la, por não ser relevante, ou pode arquivá-la. Este caso de uso transmite todas estas possibilidades.

A *Tabela 3.20* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.20: Validar Ocorrência Reportada

Identificador	uc_sus20
Descrição	Possibilita a validação, devolução para edição, eliminação ou arquivamento da ocorrência reportada.
Actor	<i>Pivot dos Serviços Fiscalizadores.</i>
Prioridade	Média.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso. Não existência de conflitos com dependências.

O caso de uso Pesquisar Informação é explicado na *Secção 3.3*.

3.1.3 Agente Fiscalizador

O trabalho do *Agente Fiscalizador* é reportar ocorrências que testemunha. Assim sendo, os casos de uso a que tem acesso restringem-se ao reporte e à pesquisa de informação que o auxiliará no seu trabalho. A *Figura 3.3* apresenta os casos de uso em relação ao *Agente Fiscalizador*.

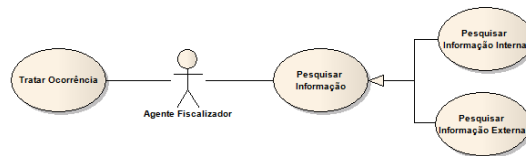


Figura 3.3: Diagrama de casos de uso do *SuS* - Agente Fiscalizador.

Caso de Uso *Tratar Ocorrência*

O *Agente Fiscalizador* pode adicionar, editar e remover ocorrências do sistema, assim como imprimir a informação referente a ocorrências reportadas.

A *Tabela 3.21* apresenta o resto da informação necessária sobre este caso de uso.

Tabela 3.21: Tratar Ocorrência

Identificador	uc_sus21
Descrição	Permite adicionar, editar, remover e imprimir ocorrências.
Actor	<i>Agente Fiscalizador</i> .
Prioridade	Baixa.
Pressupostos	Acesso ao servidor é possível. O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pré-condições	O utilizador é o criador da ocorrência.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso.

O caso de uso Pesquisar Informação é explicado na *Secção 3.3*.

3.2 MSuS - Mobile Supervision System

A [Figura 3.4](#) apresenta os casos de uso referentes ao *MSuS*. Neste sistema existe apenas um actor, o *Agente Fiscalizador*. O *Agente Fiscalizador* utiliza este sistema para tarefas de pesquisa de informação importante para a realização do seu trabalho, reportar ocorrências.

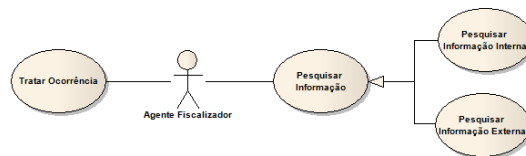


Figura 3.4: Diagrama de casos de uso do *MSuS*.

Caso de Uso *Tratar Ocorrência*

Ambos os sistemas que irão ser desenvolvidos centram-se na possibilidade de um *Agente Fiscalizador* reportar *Ocorrências*. Em analogia ao caso de uso idêntico no *SuS*, a [Tabela 3.22](#) explica este caso de uso.

Tabela 3.22: Tratar Ocorrência

Identificador	uc_msus01
Descrição	Permite adicionar, editar, remover e imprimir ocorrências.
Actor	<i>Agente Fiscalizador</i> .
Prioridade	Essencial.
Pressupostos	O dispositivo móvel encontra-se ligado. Acesso à base de dados do dispositivo móvel é possível. O Utilizador está devidamente autenticado. O dispositivo encontra-se ligada à impressora portátil e esta está ligada.
Pré-condições	O utilizador tem as permissões necessárias.
Pré-condições	O utilizador é o criador da ocorrência.
Pós-condições	As alterações ao armazenamento de dados são realizadas com sucesso.

A informação referente à pesquisa de informação é apresentada seguidamente na [Secção 3.3 Comum](#).

3.3 Comum

Caso de Uso *Pesquisar Informação*

O *SuS* e o *MSuS* têm em comum os casos de uso referentes à pesquisa de informação. Estes casos de uso apenas se dividem em duas possíveis situações. A primeira situação diz respeito a pesquisa de informação criada única e exclusivamente pela utilização do sistema enquanto a segunda tem a ver com a informação proveniente de actividades externas à utilização do sistema, como é o caso da criação de legislações que dão origem a tipos de ocorrência utilizados neste sistema.

Pesquisar Informação Interna

Este caso de uso é uma especialização da pesquisa de informação. Esta foi a forma encontrada para transparecer que as pesquisas efectuadas por este caso de uso tem por objectivo obter informações criadas pela a utilização do sistema. Um exemplo é a pesquisa de uma ocorrência reportada.

A [Tabela 3.23](#) descreve este caso de uso.

Tabela 3.23: Pesquisar Informação Interna

Identificador	uc_com01
Descrição	Permite pesquisar informação criada pela utilização do sistema.
Actor	<i>Todos.</i>
Prioridade	Elevada.
Pressupostos	O servidor do sistema encontra-se em funcionamento; Ligação à Internet existe possibilitando troca de informação entre o servidor e o cliente;
Pré-condições	A autenticação do utilizador foi realizada com sucesso.
Pós-condições	-
Pressupostos	Acesso ao servidor é possível. (<i>SuS</i>) O dispositivo móvel encontra-se ligado. (<i>MSuS</i>) Acesso à base de dados do dispositivo móvel é possível. (<i>MSuS</i>) O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	

Pesquisar Informação Externa

Tal como o caso de uso anterior, esta é um a especialização. Esta foi a forma encontrada para transparecer que as pesquisas efectuadas por estes casos de uso tem por objectivo obter informações as quais não são criadas directamente derivado ao contexto de utilização do sistema. Um exemplo é a pesquisa de um tipo de ocorrência.

A *Tabela 3.24* descreve este caso de uso.

Tabela 3.24: Pesquisar Informação Externa

Identificador	uc_com01
Descrição	Permite pesquisar informação criada externamente à utilização do sistema.
Actor	<i>Todos.</i>
Prioridade	Elevada.
Pressupostos	O servidor do sistema encontra-se em funcionamento; Ligação à Internet existe possibilitando troca de informação entre o servidor e o cliente;
Pré-condições	A autenticação do utilizador foi realizada com sucesso.
Pós-condições	-
Pressupostos	Acesso ao servidor é possível. (<i>SuS</i>) O dispositivo móvel encontra-se ligado. (<i>MSuS</i>) Acesso à base de dados do dispositivo móvel é possível. (<i>MSuS</i>) O Utilizador está devidamente autenticado.
Pré-condições	O utilizador tem as permissões necessárias.
Pós-condições	

Capítulo 4

Requisitos Específicos

Nesta secção são detalhados todos os requisitos do sistema. Estes dividem-se em requisitos externos, requisitos funcionais, requisitos não-funcionais e requisitos de documentação, demonstrando a política a seguir no desenvolvimento do sistema.

4.1 Requisitos Externos

4.1.1 Interface com o Utilizador

O desenvolvimento deste sistema prevê interface com o utilizador em duas situações, uma para o sistema *SuS*, outra para o *MSuS*. É esperado que ambas as interfaces prezem a simplicidade e usabilidade, com especial atenção no caso do *MSuS*, pois em dispositivos móveis tal requisito é mais sensível. No caso do *SuS* é esperado uma interface Web.

4.1.2 Interfaces de Hardware

Em relação ao dispositivo móvel, será necessário que esteja equipado com receptor *GPS*, assim como mecanismos de ligação à rede (quer com fios, quer sem fios). De notar que o dispositivo móvel, para poder imprimir informação relativa às ocorrências reportadas, terá de possibilitar a comunicação com uma impressora portátil.

É necessário que o servidor onde o *SuS* estará a correr tenha ligação à Internet.

4.1.3 Interfaces de Software

É necessário que o dispositivo móvel tenha o sistema operativo *Windows Mobile 6* ou superior, assim como o *Oracle Database Lite* (ver [dP09]).

A nível do servidor será necessária a interligação com um sistema de base de dados, neste caso *Oracle Database*. Todo o mecanismo de sincronização com os dispositivos móveis será feito

utilizando o *Oracle Mobile Server* (parte integrante do *Oracle Database Lite*) e *Web services* (ver [dP09]).

4.1.4 Interfaces de Comunicação

O processo de comunicação dos dispositivos móveis com o servidor e com aplicações externas irá efectuar-se através de um canal de comunicação. O meio de comunicação poderá passar por uma ligação por rede móvel (*GSM*, *GPRS*, *UMTS*, *HSDPA* ou qualquer outra), por rede sem fios *Wi-Fi* ou com um cabo para ligar dispositivos via *USB*.

No caso de o servidor e a base de dados serem executados em máquinas diferentes é necessária a ligação por rede (com ou sem fios); em alternativa será utilizada uma ligação entre módulos no caso de serem executados na mesma máquina.

4.2 Requisitos Funcionais

Cada requisito funcional apresentado nesta secção está representado por uma tabela. Nesta são apresentados um identificador, sua descrição e dados manipulados. Poderá ou não ter também preocupações que a sua satisfação possa originar.

4.2.1 SuS - Supervision System

Aqui são apresentados os vários requisitos funcionais referentes a cada um dos casos de uso apresentados na *Secção 3.1 SuS - Supervision System*.

Caso de Uso *Tratar Área*

Como já referido anteriormente, uma área é um conjunto de sub-áreas e/ou tipos de ocorrência. Para mais informações sobre este caso de uso ver *Tabela 3.1*. Este caso de uso compreende os requisitos de adição (ver *Tabela 4.1*), de edição (ver *Tabela 4.2*) e de remoção (ver *Tabela 4.3*).

Tabela 4.1: Requisito - Adicionar Área

Identificador	uc_sus01_r01
Descrição	Criação de mecanismo para adição de áreas. Ou seja, é criado um novo registo de área no armazenamento de dados.
Dados Manipulados	Código interno; Área da qual pode ser sub-área; Descrição; Estado em que área se encontra; Datas de vigor; Dados de auditoria.

Tabela 4.2: Requisito - Editar Área

Identificador	uc_sus01_r02
Descrição	Criação do mecanismo de edição de áreas, isto é, alterar certos valores de um registo existente no armazenamento de dados.
Dados Manipulados	Descrição; Datas de vigor; Dados de auditoria.
Preocupações	Necessária a atenção para eventuais conflitos entre dependências.

Requisitos Específicos

Tabela 4.3: Requisito - Remover Área

Identificador	uc_sus01_r03
Descrição	Criação do mecanismo de remoção de áreas. Para tal não se remove realmente as áreas do armazenamento de dados, apenas se mudam as datas de vigor e estado destas.
Dados Manipulados	Estado em que área se encontra; Datas de vigor; Dados de auditoria.
Preocupações	Necessária a atenção para eventuais conflitos entre dependências.

Caso de Uso *Tratar Característica*

Este caso de uso é composto por requisitos de adição, edição e remoção de características. As Tabelas 4.4, 4.5 e 4.6 apresentam as informações sobre tais operações, respectivamente. Para mais informações sobre o caso de uso em questão ver Tabela 3.2.

Tabela 4.4: Requisito - Adicionar Característica

Identificador	uc_sus02_r01
Descrição	Permitir ao utilizador autorizado a criação de um novo registo no armazenamento de dados referente a uma característica.
Dados Manipulados	Designação; Tipo de dados em questão (ex: texto, imagem, entre outros); Dados de auditoria.

Tabela 4.5: Requisito - Editar Característica

Identificador	uc_sus02_r02
Descrição	Possibilitar a edição de alguns dos valores referentes a uma característica existente no armazenamento de dados
Dados Manipulados	Designação; Tipo de dados em questão (ex: texto, imagem, entre outros); Dados de auditoria.
Preocupações	Necessária a atenção para eventuais conflitos entre dependências.

Requisitos Específicos

Tabela 4.6: Requisito - Remover Característica

Identificador	uc_sus02_r03
Descrição	Retirar ligação aos tipos de ocorrência que utilizam tal característica.
Dados Manipulados	Datas de vigor de ligação entre tipos de ocorrência e a característica em questão.

Caso de Uso *Tratar Categoria*

Sendo que cada membro pode pertencer a um categoria profissional, a adição, edição e remoção destas aparecem com requisitos do sistema. Estas ações sobre o armazenamento de dados estão explicadas nas Tabelas 4.7, 4.8 e 4.9, respectivamente. Para informação mais detalhada sobre o caso de uso em questão ver Tabela 3.3.

Tabela 4.7: Requisito - Adicionar Categoria

Identificador	uc_sus03_r01
Descrição	Permitir adicionar categorias profissionais no armazenamento de dados.
Dados Manipulados	Sigla; Designação; Dados de auditoria.

Tabela 4.8: Requisito - Editar Categoria

Identificador	uc_sus03_r02
Descrição	Permitir alteração dos dados referentes a uma categoria profissional existente no armazenamento de dados.
Dados Manipulados	Sigla; Designação; Dados de auditoria.

Tabela 4.9: Requisito - Remover Categoria

Identificador	uc_sus03_r03
Descrição	Remover do armazenamento de dados o registo da categoria profissional em questão.
Dados Manipulados	Todo o registo de categoria profissional; Retirar membros desta categoria.
Preocupações	Necessária a atenção para eventuais conflitos entre dependências.

Caso de Uso *Tratar Lista*

As listas para utilização como um dos tipos de dados das características são adicionadas, editadas e removidas através dos requisitos contidos neste caso de uso. As *Tabelas 4.10, 4.11 e 4.12* apresentam estas acções sobre os registos de listas, respectivamente. Para mais informações sobre este caso de uso ver *Tabela 3.4*.

Tabela 4.10: Requisito - Adicionar Lista

Identificador	uc_sus04_r01
Descrição	Permite adicionar uma nova lista no armazenamento de dados.
Dados Manipulados	Descrição; Designação; Estado em que se encontra a lista. Forma de apresentação da lista (ex: <i>combo box</i>); Ligação com elementos que compõe esta nova lista; Dados de auditoria.

Tabela 4.11: Requisito - Editar Lista

Identificador	uc_sus04_r02
Descrição	Permitir editar dados da lista.
Dados Manipulados	Descrição; Designação; Forma de apresentação da lista (ex: <i>combo box</i>); Ligação com elementos que compõe esta nova lista; Dados de auditoria.

Requisitos Específicos

Tabela 4.12: Requisito - Remover Lista

Identificador	uc_sus04_r03
Descrição	Permitir alterar o estado da lista para removida.
Dados Manipulados	Estado em que se encontra a lista. Dados de auditoria.

Caso de Uso *Tratar Elemento*

As listas são constituídas por elementos, os quais tem de ser adicionados, editados e removidos. As Tabelas 4.13, 4.14 e 4.15 apresentam estas acções, respectivamente. Para mais informações sobre este caso de uso ver Tabela 3.5.

Tabela 4.13: Requisito - Adicionar Elemento

Identificador	uc_sus05_r01
Descrição	Permitir adicionar um novo registo de elemento constituinte de uma lista.
Dados Manipulados	Designação; Estado em que o elemento se encontra; Dados de auditoria.

Tabela 4.14: Requisito - Editar Elemento

Identificador	uc_sus05_r02
Descrição	Permitir editar dados referentes a um elemento existente no armazenamento de dados.
Dados Manipulados	Designação; Dados de auditoria.

Tabela 4.15: Requisito - Remover Elemento

Identificador	uc_sus05_r03
Descrição	Modificar estado para removido.
Dados Manipulados	Estado em que o elemento se encontra; Dados de auditoria.

Caso de Uso *Tratar Membro*

Um dos principais casos de uso do sistema, este compreende acções de adição, edição e remoção de membros - utilizadores do sistema. As [Tabelas 4.16, 4.17 e 4.18](#) apresentam estas acções sobre membros, respectivamente. Para mais informações sobre este caso de uso ver [Tabela 3.6](#).

Tabela 4.16: Requisito - Adicionar Membro

Identificador	uc_sus06_r01
Descrição	Permitir a um utilizador com permissões a adição de um novo membro.
Dados Manipulados	Código; Nome completo, nome próprio e apelido; A que unidade orgânica e categoria profissional pertence; Estado em que o membro se encontra; Password de entrada no sistema; Ligação a perfis e áreas; Dados de auditoria.

Tabela 4.17: Requisito - Editar Membro

Identificador	uc_sus06_r02
Descrição	Permitir editar alguns dos dados de um membro existente no armazenamento de dados.
Dados Manipulados	Nome completo, nome próprio e apelido; A que unidade orgânica e categoria profissional pertence; Password de entrada no sistema; Ligação a perfis e áreas; Dados de auditoria.
Preocupações	Necessária a atenção para eventuais conflitos entre dependências.

Requisitos Específicos

Tabela 4.18: Requisito - Remover Membro

Identificador	uc_sus06_r03
Descrição	Mudar o estado do membro para um estado que não o permita mais aceder ao sistema. Este pode ter sido despedido ou mesmo ter ido para a reforma.
Dados Manipulados	Estado em que o membro se encontra; Dados de auditoria.

Caso de Uso *Tratar Perfil*

Na solução irão existir perfis que reflectem as permissões que os membros a eles associados têm. Para tal associação, primeiro tem de existir os mecanismos de adição, edição e remoção de tais perfis. As Tabelas 4.19, 4.20 e 4.21 apresentam estas acções, respectivamente. Para mais informações sobre este caso de uso ver Tabela 3.7.

Tabela 4.19: Requisito - Adicionar Perfil

Identificador	uc_sus07_r01
Descrição	Criação de mecanismos de adição de um perfil.
Dados Manipulados	Designação; Ligação às permissões; Dados de auditoria.

Tabela 4.20: Requisito - Editar Perfil

Identificador	uc_sus07_r02
Descrição	Permitir alterar dados relativos ao perfil.
Dados Manipulados	Designação; Ligação às permissões; Dados de auditoria.

Requisitos Específicos

Tabela 4.21: Requisito - Remover Perfil

Identificador	uc_sus07_r03
Descrição	Permitir remover um perfil.
Dados Manipulados	Remoção das ligações dos vários membros ao perfil em causa; Remoção das ligações às permissões; Remoção do registo do perfil.
Preocupações	Necessária a atenção para eventuais conflitos entre dependências.

Caso de Uso *Tratar Permissão*

Esta casos de uso reflecte a necessidade de atribuição de permissões aos perfis dos membros registados. Para tal, aparecem os requisitos de adicionar (ver [Tabela 4.22](#)), editar (ver [Tabela 4.23](#)) e remover (ver [Tabela 4.24](#)) permissões do sistema. Para mais informações sobre este caso de uso ver [Tabela 3.8](#).

Tabela 4.22: Requisito - Adicionar Permissão

Identificador	uc_sus08_r01
Descrição	Permitir adicionar uma nova permissão na solução.
Dados Manipulados	Tipo de permissão; Dados de auditoria.

Tabela 4.23: Requisito - Editar Permissão

Identificador	uc_sus08_r02
Descrição	Permitir editar um permissão existente.
Dados Manipulados	Tipo de permissão; Dados de auditoria.

Requisitos Específicos

Tabela 4.24: Requisito - Remover Permissão

Identificador	uc_sus08_r03
Descrição	Permitir eliminar registo da permissão.
Dados Manipulados	Remoção das ligações dos perfis à permissão em causa. Remoção do registo da permissão.
Preocupações	Necessária a atenção para eventuais conflitos entre dependências.

Caso de Uso *Tratar Tipo de Ocorrência*

Este caso de uso é um dos mais cruciais do sistema, sendo que cria quais os tipos de ocorrência que podem ser reportados. Para tal, é necessário mecanismos de parametrização. Os requisitos em causa são a adição, edição e remoção destes tipos de ocorrência. As *Tabelas 4.25, 4.26 e 4.27* apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver *Tabela 3.9*.

Tabela 4.25: Requisito - Adicionar Tipo de Ocorrência

Identificador	uc_sus09_r01
Descrição	Permitir adicionar um novo tipo de ocorrência possível de reportar.
Dados Manipulados	Código; Descrição; Designação; Estado em que se encontra; Possibilidade de edição posterior por parte do <i>Agente</i> ; Validação necessária pelo <i>Pivot</i> ; Ligações às características adicionais; Ligação à área onde está contido; Datas de vigor; Dados de auditoria.

Tabela 4.26: Requisito - Simular Tipo de Ocorrência

Identificador	uc_sus09_r02
Descrição	Poder simular qual será o interface de reporte do tipo de ocorrência no dispositivo móvel.
Dados Manipulados	-

Requisitos Específicos

Tabela 4.27: Requisito - Editar Tipo de Ocorrência

Identificador	uc_sus09_r03
Descrição	Permitir editar alguns dos dados associados a um tipo de ocorrência existente no armazenamento de dados.
Dados Manipulados	Descrição; Designação; Ligações às características adicionais; Ligação à área onde está contido; Datas de vigor; Dados de auditoria.

Tabela 4.28: Requisito - Remover Tipo de Ocorrência

Identificador	uc_sus09_r04
Descrição	Permitir cancelar a possibilidade de reportar o tipo de ocorrência em questão, modificando o seu estado.
Dados Manipulados	Estado em que se encontra; Datas de vigor; Dados de auditoria.

Caso de Uso *Tratar Tipo de Tarefa*

Um tipo de tarefa é para ser associado a tarefas que serão atribuídas a actividades de membros registados no sistema. Para que estas associações sejam possíveis e necessária a adição, edição e remoção de tais tipos. As *Tabelas 4.29, 4.30 e 4.31* apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver *Tabela 3.10*.

Tabela 4.29: Requisito - Adicionar Tipo de Tarefa

Identificador	uc_sus10_r01
Descrição	Permitir adicionar um novo tipo de tarefa.
Dados Manipulados	Descrição; Designação; Estado em que se encontra; Dados de auditoria.

Tabela 4.30: Requisito - Editar Tipo de Tarefa

Identificador	uc_sus10_r02
Descrição	Permitir editar dados referente a um tipo de tarefa.
Dados Manipulados	Descrição; Designação; Dados de auditoria.

Tabela 4.31: Requisito - Remover Tipo de Tarefa

Identificador	uc_sus10_r03
Descrição	Permitir deixar de ser associado a novas tarefas.
Dados Manipulados	Estado em que se encontra; Dados de auditoria.

Caso de Uso *Tratar Topónimo*

Podem ser adicionados, editados e removidos topónimos do armazenamento de dados. As Tabelas 4.32, 4.33 e 4.34 apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver Tabela 3.10.

Tabela 4.32: Requisito - Adicionar Topónimo

Identificador	uc_sus11_r01
Descrição	Permitir adicionar um novo topónimo.
Dados Manipulados	Código; Designação; Ligação às zonas a que pertence; Dados de auditoria.

Tabela 4.33: Requisito - Editar Topónimo

Identificador	uc_sus11_r02
Descrição	Permitir edição dos dados referentes a um topónimo existente no armazenamento de dados.
Dados Manipulados	Designação; Ligação às zonas a que pertence; Dados de auditoria.

Tabela 4.34: Requisito - Remover Topónimo

Identificador	uc_sus11_r03
Descrição	Permitir remover topónimos que ainda não foram associados a nenhuma ocorrência.
Dados Manipulados	Eliminar registo do topónimo.
Preocupações	Necessária a atenção para eventuais conflitos entre dependências. Não é possível no caso de existirem ocorrências associadas ao topónimo.

Caso de Uso Tratar Número de Polícia

Podem ser adicionados, editados e removidos números de polícia do armazenamento de dados. As Tabelas 4.35, 4.36 e 4.37 apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver Tabela 3.12.

Tabela 4.35: Requisito - Adicionar Número de Polícia

Identificador	uc_sus12_r01
Descrição	Permitir adicionar um novo número de polícia.
Dados Manipulados	Número; Ligação aos topónimos a que pertence; Dados de auditoria.

Tabela 4.36: Requisito - Editar Número de Polícia

Identificador	uc_sus12_r02
Descrição	Permitir edição dos dados referentes a um número de polícia existente no armazenamento de dados.
Dados Manipulados	Número; Ligação aos topónimos a que pertence; Dados de auditoria.

Tabela 4.37: Requisito - Remover Número de Polícia

Identificador	uc_sus12_r03
Descrição	Permitir remover números de polícia que ainda não foram associados a nenhuma ocorrência.
Dados Manipulados	Eliminar registo do número de polícia.
Preocupações	Necessária a atenção para eventuais conflitos entre dependências. Não é possível no caso de existirem ocorrências associadas ao número de polícia.

Caso de Uso *Tratar Unidade Orgânica*

Sendo que cada membro pode pertencer a uma unidade orgânica, a adição, edição e remoção destas aparecem com requisitos do sistema. Estes acções sobre o armazenamento de dados estão explicadas nas *Tabelas 4.38, 4.39 e 4.40*. Para informação mais detalhada sobre o caso de uso em questão ver *Tabela 3.13*.

Tabela 4.38: Requisito - Adicionar Unidade Orgânica

Identificador	uc_sus13_r01
Descrição	Permitir adicionar um nova unidade orgânica no armazenamento de dados.
Dados Manipulados	Sigla; Designação; Dados de auditoria.

Tabela 4.39: Requisito - Editar Unidade Orgânica

Identificador	uc_sus13_r02
Descrição	Permitir alteração dos dados referentes a uma unidade orgânica existente no armazenamento de dados.
Dados Manipulados	Sigla; Designação; Dados de auditoria.

Tabela 4.40: Requisito - Remover Unidade Orgânica

Identificador	uc_sus13_r03
Descrição	Remover do armazenamento de dados o registo da unidade orgânica em questão.
Dados Manipulados	Todo o registo de unidade orgânica; Retirar membros desta unidade orgânica.
Preocupações	Necessária a atenção para eventuais conflitos entre dependências.

Caso de Uso *Tratar Zona*

Podem ser adicionadas, editadas e removidas zonas do município do armazenamento de dados, assim como associar a esta uma localização geo-posicional. As *Tabelas 4.41, 4.42, 4.43 e 4.44* apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver *Tabela 3.14*.

Tabela 4.41: Requisito - Adicionar Zona

Identificador	uc_sus14_r01
Descrição	Permitir a adição de uma nova zona.
Dados Manipulados	Descrição; Dados de auditoria.

Tabela 4.42: Requisito - Editar Zona

Identificador	uc_sus14_r02
Descrição	Permitir editar dados de uma zona existentes no armazenamento de dados.
Dados Manipulados	Descrição; Dados de auditoria.

Tabela 4.43: Requisito - Remover Zona

Identificador	uc_sus14_r03
Descrição	Permitir remover zonas que ainda não foram associada a nenhum outra classe existente na solução.
Dados Manipulados	Todo o registo da zona em questão.
Preocupações	Necessária a atenção para eventuais conflitos entre dependências.

Tabela 4.44: Requisito - Adicionar Localização GPS à Zona

Identificador	uc_sus14_r04
Descrição	Permitir adicionar a uma zona um registo de localização.
Dados Manipulados	Código de localização interno; Coordenadas GPS; Dados de auditoria;
Preocupações	No caso de já existir uma localização associada, elimina-se esta primeiro e depois cria-se um novo registo.

Caso de Uso Tratar Actividade

Um membro, neste caso um *Agente Fiscalizador*, pode ser associado a várias actividades de fiscalização. Para tal, mecanismos de adição, edição, remoção e cancelamento aparecem como requisitos. As Tabelas 4.45, 4.46, 4.47 e 4.48 apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver Tabela 3.15.

Tabela 4.45: Requisito - Adicionar Actividade

Identificador	uc_sus15_r01
Descrição	Permitir a criação de uma nova actividade.
Dados Manipulados	Descrição; Designação; Estado em que se encontra; Data de inicio; Ligações a membros a realizar actividade; Ligações a tarefas que compões a actividade; Dados de auditoria.

Tabela 4.46: Requisito - Editar Actividade

Identificador	uc_sus15_r02
Descrição	Permitir edição da informação relativa a um actividade existente.
Dados Manipulados	Descrição; Designação; Data de inicio; Ligações a membros a realizar actividade; Ligações a tarefas que compões a actividade; Dados de auditoria.

Requisitos Específicos

Tabela 4.47: Requisito - Remover Actividade

Identificador	uc_sus15_r03
Descrição	Permitir remover actividade que ainda não foram associada a nenhum membro.
Dados Manipulados	Remoção das ligações a tarefas; Remoção do registo de actividade em questão;

Tabela 4.48: Requisito - Cancelar Actividade

Identificador	uc_sus15_r04
Descrição	Permitir cancelar a actividade.
Dados Manipulados	Estado em que se encontra; Dados de auditoria.

Caso de Uso *Tratar Tarefa*

Um membro, neste caso um *Agente Fiscalizador*, pode ser associado a várias actividades de fiscalização, e estas, por sua vez, são compostas por tarefas. Assim sendo, mecanismos de adição, edição, remoção de tarefas aparecem como requisitos, não esquecendo a possibilidade de lhes associar uma localização geo-referenciada. As *Tabelas 4.49, 4.50, 4.51 e 4.52* apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver *Tabela 3.16*.

Tabela 4.49: Requisito - Adicionar Tarefa

Identificador	uc_sus16_r01
Descrição	Permitir adicionar uma nova tarefa.
Dados Manipulados	Descrição; Tipo de tarefa; Dados de auditoria.

Tabela 4.50: Requisito - Editar Tarefa

Identificador	uc_sus16_r02
Descrição	Permitir edição de dados da tarefa.
Dados Manipulados	Descrição; Tipo de tarefa; Dados de auditoria.

Tabela 4.51: Requisito - Remover Tarefa

Identificador	uc_sus16_r03
Descrição	Permitir remover registo da tarefa.
Dados Manipulados	Todo o registo de tarefa.

Requisitos Específicos

Tabela 4.52: Requisito - Adicionar Localização GPS à Tarefa

Identificador	uc_sus16_r04
Descrição	Permitir adicionar a uma tarefa um registo de localização.
Dados Manipulados	Código de localização interno; Coordenadas GPS; Dados de auditoria;
Preocupações	No caso de já existir uma localização associada, elimina-se esta primeiro e depois cria-se um novo registo.

Caso de Uso *Tratar Alocação*

Os *Agentes Fiscalizadores* podem estar associados a uma equipa, que por sua vez terá um horário composto por várias alocações a zonas do município para procederem a trabalho de fiscalização geral. Para tal, é necessário cumprir os requisitos de adição, edição e remoção de tais alocações. As *Tabelas 4.53, 4.54 e 4.55* apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver *Tabela 3.17*.

Tabela 4.53: Requisito - Adicionar Alocação

Identificador	uc_sus17_r01
Descrição	Permitir definir uma nova alocação de uma equipa a uma zona do município.
Dados Manipulados	Data de alocação; Horas de início e fim; Zonas do município a fiscalizar; Equipas associadas; Dados de auditoria;

Tabela 4.54: Requisito - Editar Alocação

Identificador	uc_sus17_r02
Descrição	Permitir editar uma alocação já existente.
Dados Manipulados	Data de alocação; Horas de início e fim; Zonas do município a fiscalizar; Equipas associadas; Dados de auditoria;
Preocupações	No caso de já existir de a data de alocação se encontrar no passado não é possível editar.

Requisitos Específicos

Tabela 4.55: Requisito - Remover Alocação

Identificador	uc_sus17_r03
Descrição	Permitir a remoção de alocação de uma equipa a uma zona do município.
Dados Manipulados	Remoção do registo de alocação.
Preocupações	No caso de já existir de a data de alocação se encontrar no passado não é possível remover.

Caso de Uso *Tratar Conhecimento*

Cada área ou tipo de ocorrência pode ter associado a ele um banco de conhecimento. Para tal é necessário permitir adição, edição e remoção de pequenas fracções deste conhecimento. As Tabelas 4.56, 4.57 e 4.58 apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver Tabela 3.18.

Tabela 4.56: Requisito - Adicionar Conhecimento

Identificador	uc_sus18_r01
Descrição	Permitir adicionar uma nova linha de orientação no banco de conhecimento.
Dados Manipulados	Texto; Datas de vigor; Dados de auditoria;

Tabela 4.57: Requisito - Editar Conhecimento

Identificador	uc_sus18_r02
Descrição	Permitir editar uma linha de orientação existente.
Dados Manipulados	Texto; Datas de vigor; Dados de auditoria;

Tabela 4.58: Requisito - Remover Conhecimento

Identificador	uc_sus18_r03
Descrição	Permitir terminar a data de vigor do conhecimento em causa.
Dados Manipulados	Datas de vigor; Dados de auditoria.

Caso de Uso *Tratar Equipa*

Os *Agentes Fiscalizadores* podem fazer parte de equipas. Este caso de uso transmite a necessidade de permitir a adição, edição e remoção de tais equipas. As *Tabelas 4.59, 4.60 e 4.61* apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver *Tabela 3.19*.

Tabela 4.59: Requisito - Adicionar Equipa

Identificador	uc_sus19_r01
Descrição	Permitir adicionar uma nova equipa.
Dados Manipulados	Código; Ligações a membros; Dados de auditoria;

Tabela 4.60: Requisito - Editar Equipa

Identificador	uc_sus19_r02
Descrição	Permitir editar um equipa existente no armazenamento de dados.
Dados Manipulados	Ligações a membros; Dados de auditoria;

Requisitos Específicos

Tabela 4.61: Requisito - Remover Equipa

Identificador	uc_sus19_r03
Descrição	Permitir remover equipas
Dados Manipulados	Remoção de ligações a membros; Remoção do registo da equipa.
Preocupações	No caso de já existir de alocações associadas à equipa não é possível remover.

Caso de Uso *Validar Ocorrência Reportada*

Quando uma ocorrência é reportada pode ser necessária a sua validação por parte do *Pivot*. Uma de várias coisas pode acontecer. Um ocorrência pode ser validada, devolvida para completar, eliminada ou arquivada. As *Tabelas 4.62, 4.63, 4.64 e 4.65* apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver *Tabela 3.20*.

Tabela 4.62: Requisito - Validar Ocorrência Reportada

Identificador	uc_sus20_r01
Descrição	Permitir a validação de uma ocorrência reportada.
Dados Manipulados	Estado em que se encontra; Dados de auditoria;

Tabela 4.63: Requisito - Devolver Ocorrência Reportada

Identificador	uc_sus20_r02
Descrição	Permitir, na ausência de informação necessária para validar, devolver ao <i>Agente</i> que reportou a ocorrência a possibilidade de a editar.
Dados Manipulados	Estado em que se encontra; Dados de auditoria;
Preocupações	No caso da ocorrência já ter sido validada ou arquivada não é possível devolver para edição.

Requisitos Específicos

Tabela 4.64: Requisito - Eliminar Ocorrência Reportada

Identificador	uc_sus20_r03
Descrição	Permitir eliminar o registo da ocorrência.
Dados Manipulados	Remoção do registo da ocorrência e valores de características associadas.
Preocupações	No caso da ocorrência já ter sido validada ou arquivada não é possível remove-la.

Tabela 4.65: Requisito - Arquivar Ocorrência Reportada

Identificador	uc_sus20_r04
Descrição	Permitir arquivar ocorrência sem que para isso tenha sido necessário mais algum tipo de intervenção no terreno.
Dados Manipulados	Estado em que se encontra; Dados de auditoria;
Preocupações	No caso da ocorrência já ter sido validada não é possível arquivá-la.

Caso de Uso Tratar Ocorrência

Para além de ser possível ao Agente Fiscalizador reportar ocorrências no terreno através do dispositivo móvel, também o é através do sistema de *Back-Office*, o *SuS*. Para tal, este deverá possibilitar a adição de uma nova ocorrência, assim como editar e remover já existentes, não esquecendo que deverá possibilitar a impressão da informação referente às ocorrências adicionadas. As Tabelas 4.66, 4.67, 4.68 e 4.69 apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver Tabela 3.21.

Tabela 4.66: Requisito - Adicionar Ocorrência

Identificador	uc_sus21_r01
Descrição	Permitir o reporte de uma nova ocorrência.
Dados Manipulados	Data de observação; Local; Estado em que se encontra; Tipo de ocorrência; Valor das características do tipo de ocorrência; Dados de auditoria.

Tabela 4.67: Requisito - Editar Ocorrência

Identificador	uc_sus21_r02
Descrição	Permitir editar um ocorrência já reportada.
Dados Manipulados	Data de observação; Local; Estado em que se encontra; Valor das características do tipo de ocorrência; Dados de auditoria.
Preocupações	No caso da ocorrência já ter sido validada ou arquivada não é possível editá-la.

Tabela 4.68: Requisito - Remover Ocorrência

Identificador	uc_sus21_r03
Descrição	Permitir remover um ocorrência reportada.
Dados Manipulados	Remoção do registo da ocorrência e valor das características associadas.
Preocupações	No caso da ocorrência já ter sido validada ou arquivada não é possível remove-la.

Tabela 4.69: Requisito - Imprimir Ocorrência

Identificador	uc_sus21_r04
Descrição	Permitir imprimir a informação referente a um ocorrência reportada.
Dados Manipulados	Data de observação; Local; Estado em que se encontra; Tipo de ocorrência; Valor das características do tipo de ocorrência; Dados de auditoria.

4.2.2 MSuS - Mobile Supervision System

Aqui são apresentados os vários requisitos funcionais referentes a cada um dos casos de uso apresentados na *Secção 3.2 MSuS - Mobile Supervision System*.

Caso de Uso *Tratar Ocorrência*

Este é o caso de uso mais crucial de toda a solução, pois é no reporte de ocorrência que todos os outros casos assentam. Para tal, este sistema deverá possibilitar adicionar uma nova ocorrência, assim como editar, remover e associar localizações geo-referenciadas, assim como imprimir, utilizando um impressora portátil, as informações referentes a ocorrências adicionadas. As Tabelas 4.70, 4.71, 4.72, 4.73 e 4.74 apresentam estes requisitos, respectivamente. Para mais informações sobre este caso de uso ver *Tabela 3.22*.

Tabela 4.70: Requisito - Adicionar Ocorrência

Identificador	uc_msus01_r01
Descrição	Permitir o reporte de uma nova ocorrência.
Dados Manipulados	Data de observação; Local; Estado em que se encontra; Tipo de ocorrência; Valor das características do tipo de ocorrência; Dados de auditoria.

Requisitos Específicos

Tabela 4.71: Requisito - Editar Ocorrência

Identificador	uc_msus01_r02
Descrição	Permitir editar um ocorrência já reportada.
Dados Manipulados	Data de observação; Local; Estado em que se encontra; Valor das características do tipo de ocorrência; Dados de auditoria.
Preocupações	No caso da ocorrência já ter sido validada ou arquivada não é possível editá-la.

Tabela 4.72: Requisito - Remover Ocorrência

Identificador	uc_msus01_r03
Descrição	Permitir remover um ocorrência reportada.
Dados Manipulados	Remoção do registo da ocorrência e valor das características associadas.
Preocupações	No caso da ocorrência já ter sido validada ou arquivada não é possível remove-la.

Tabela 4.73: Requisito - Adicionar Localização GPS à Ocorrência

Identificador	uc_msus01_r04
Descrição	Permitir adicionar a uma ocorrência um registo de localização.
Dados Manipulados	Código de localização interno; Coordenadas GPS; Dados de auditoria;
Preocupações	No caso de já existir uma localização associada, elimina-se esta primeiro e depois cria-se um novo registo.

Tabela 4.74: Requisito - Imprimir Ocorrência

Identificador	uc_msus01_r05
Descrição	Permitir imprimir a informação referente a um ocorrência reportada.
Dados Manipulados	Data de observação; Local; Estado em que se encontra; Tipo de ocorrência; Valor das características do tipo de ocorrência; Dados de auditoria.
Preocupações	Como efectuar ligação a um impressora portátil.

4.2.3 Comum

Caso de Uso *Pesquisar Informação Interna*

Este caso de uso tem associada a si um único requisito. Este transmite a necessidade de criação de mecanismos de pesquisa sobre dados criadas com a utilização do sistema. As [Tabela 4.75](#) apresenta este requisito. Para mais informações sobre este caso de uso ver [Tabela 3.23](#).

Tabela 4.75: Requisito Instável - Pesquisar Informação Interna

Identificador	uc_com01_r01
Descrição	Permite pesquisar informação criada internamente com a utilização do sistema.
Dados Manipulados	-

Caso de Uso *Pesquisar Informação Externa*

Este caso de uso tem associada a si um único requisito. Este transmite a necessidade de criação de mecanismos de pesquisa sobre dados externos à solução. As [Tabela 4.76](#) apresenta este requisito. Para mais informações sobre este caso de uso ver [Tabela 3.24](#).

Tabela 4.76: Requisito Instável - Pesquisar Informação Externa

Identificador	uc_com02_r01
Descrição	Permite pesquisar informação criada externamente à utilização do sistema.
Dados Manipulados	-

4.3 Requisitos Não-Funcionais

Nesta secção descrevem-se um pouco os requisitos do foro não funcional exigidos pelo cliente. Os requisitos suplementares ou requisitos não-funcionais são requisitos transversais aos vários casos de utilização do software. São requisitos de qualidade do produto e definem as propriedades do sistema, apresentando-se como um complemento aos requisitos funcionais. Compreendem tipicamente características como tempos de resposta, segurança, fiabilidade, normas, etc. É importante adequar este tipo de requisitos aos utilizadores finais, reforçando a implementação num ou noutro aspecto, conforme as necessidades. Os requisitos não-funcionais podem assumir um carácter mais crítico do que os requisitos funcionais uma vez que os podem inviabilizar, tornando o sistema obsoleto, enquanto os requisitos funcionais apenas nos descrevem a informação acerca da real utilidade do sistema. Torna-se crucial seguir a regra dos 3 *Us* (*Usability*, *Used* e *Utility*) na implementação de uma interface de qualidade, e ter sempre em consideração que na relação custo/rentabilidade destes requisitos muitas vezes o ganho ao atingir os mesmos, a nível de qualidade e de grau de certeza, não compensa o trabalho despendido para os atingir. Em relação à criticidade, os requisitos apresentados são classificados em **Essencial**, **Elevada** e **Média**.

Na *Tabela 4.77* são descritos os requisitos não-funcionais associados ao software pretendido, apresentando, para cada um deles, uma pequena descrição e a sua prioridade, de acordo com a classificação apresentada [dSdI09].

Tabela 4.77: Requisitos Não-Funcionais

Requisito	Descrição	Prioridade
<i>Usabilidade</i>	O sistema pretendido deverá ser simples de utilizar, de modo a minimizar a formação a efectuar aos utilizadores finais. Para esse efeito é essencial tornar a interface intuitiva (ou pelo menos intuitiva para utilizadores regulares de computadores). O percurso do utilizador deverá ser acompanhado com tópicos de ajuda e mensagens de erro claras e esclarecedoras. A facilidade de configuração do sistema deverá ser também uma aposta tendo em vista esta adaptação ao utilizador.	Essencial
<i>Eficiência</i>	A garantia da eficiência da aplicação passa por factores tais como as tecnologias usadas (o uso de tecnologias de ponta e reconhecidas como soluções para este tipo de software é importante para garantir a sua longevidade) e a proximidade e qualidade dos serviços.	Elevada

Requisitos Específicos

Requisito	Descrição	Prioridade
<i>Manutenção</i>	Pretende-se um sistema fácil de manter, de forma a minimizar o tempo perdido com as operações de manutenção. É vital que o sistema não fique sem funcionar, salvo, porventura, alguma excepção, que pode levar a avultados prejuízos.	Média
<i>Portabilidade</i>	O software (sistema <i>SuS</i>) deve ser compatível com os browsers mais conhecidos (Internet Explorer, Mozilla Firefox, Opera), abrangendo assim uma gama de utilizadores suficientemente vasta para se considerar que a cobertura do universo de utilizadores é virtualmente total.	Média
<i>Segurança</i>	As funcionalidades devem depender do nível de acesso ao sistema que o utilizador tem, limitando assim o acesso dos utilizadores a funcionalidades para as quais não possuem permissões. Existem dados privados e protegidos que devem ser salvaguardados. O software deve possuir mecanismos para impedir a visualização ou modificação destes dados por pessoal não autorizado.	Essencial
<i>Hardware</i>	Quando a solução apresentada for posta no terreno é necessário que esta tenha em consideração as especificidades dos dispositivos móveis que irão executar o <i>MSuS</i> . Em causa está a utilização de <i>PDA</i> s industriais acompanhados de impressora portátil. Um estudo sobre as melhor opções para a <i>CMP</i> está a ser realizado.	Essencial
<i>Software</i>	A base de dados a utilizar para o desenvolvimento do <i>SuS</i> será o <i>Oracle Database 11g</i> e a linguagem de programação <i>PL/SQL</i> . Em relação ao <i>MSuS</i> , este deverá executar sobre o sistema operativo da <i>Microsoft</i> , o <i>Windows Mobile 6.5</i>	Essencial
<i>Fiabilidade</i>	A fiabilidade representa o nível de confiança que é depositado pelo utilizador no sistema. Assim, pretende-se um produto consistente e robusto (que efectue as operações 100% correctas e responda sempre de forma previsível ao utilizador), essencial para garantir a satisfação do cliente, assim como a boa utilização do produto.	Elevada

4.4 Requisitos de Documentação

A este nível é habitual o cliente exigir alguns documentos, de forma a minimizar a necessidade de formação e/ou contacto com os fornecedores. Assim sendo, julga-se ser interessante que seja produzida a seguinte documentação de forma a minimizar tempos de aprendizagem e de formação. A *Tabela 4.78* apresenta a documentação a criar [dSdl09].

Tabela 4.78: Requisitos de Documentação

Manual de Instalação	Pretende documentar todos os passos necessários para que o produto seja instalado nas máquinas do cliente. Sendo assim, deve conter um conjunto de secções para que o utilizador não precise de contactar a empresa fornecedora.
Manual do Utilizador	Tem como finalidade assistir os utilizadores do software, de maneira a tomarem conhecimento de como lidar com ele. Deve conter um guião escrito e imagens associadas, sobre como o sistema deve ser apresentado e responder aos comandos introduzidos pelo utilizador. É importante que contenha uma secção de problemas conhecidos e respectivas soluções, evitando a perda de tempo à procura de soluções. Sendo um documento a ser distribuído pelos vários utilizadores finais, este deve ter em conta as características do público-alvo, evitando demasiados termos técnicos e descrição de implementação. O manual não tem de ser exclusivamente distribuído em papel, pode também ser utilizada a ajuda <i>online</i> .

Referências

- [Cor09] Microsoft Corporation. Internet explorer 8, Maio 2009. Microsoft Corporation, disponível em <http://www.microsoft.com/windows/internet-explorer/default.aspx>.
- [dP09] Hugo Manuel Zenha de Pinho. Análise tecnológica. In *Aplicação de registo de autos para dispositivos móveis - Relatório de Análise Tecnológica*. Projecto de Sistemas de Software - Faculdade de Engenharia da Universidade do Porto, Março 2009.
- [dSdI06] Direcção Municipal de Sistemas de Informação. *GIC - Especificação de Requisitos*. Câmara Municipal do Porto, Outubro 2006.
- [dSdI09] Direcção Municipal de Sistemas de Informação. Outros requisitos não-funcionais. In *SGIFM - Especificação de Requisitos*. Câmara Municipal do Porto, Março 2009.
- [eMF09] Mozilla Europe e Mozilla Foundation. Firefox o navegador web, Maio 2009. The Mozilla Foundation, disponível em <http://www.mozilla-europe.org/pt/firefox/>.
- [Lon09] Kevin Loney. *Oracle Database 11g The Complete Reference*. The McGraw-Hill Companies, Inc., 2009.
- [McL08] Michael McLaughlin. *Oracle Database 11g PL/SQL Programming*. The McGraw-Hill Companies, Inc., 2008.
- [W3C09] W3C. World wide web consortium, Maio 2009. The World Wide Web Consortium, disponível em <http://www.w3.org/>.

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Aplicação de Registo de Ocorrências para Dispositivos Móveis

Hugo Manuel Zenha de Pinho

Relatório de Definição de Arquitectura

Mestrado Integrado em Engenharia Informática e Computação

Orientador: João Manuel Paiva Cardoso

Doutor em Engenharia Electrotécnica e Computadores

29 Junho 2009

Conteúdo

1	Introdução	1
1.1	Objectivo	1
1.2	Âmbito	1
1.3	Destinatários do Documento	1
1.4	Estrutura do Documento	2
2	Representação da Arquitectura	3
2.1	Visão Geral	3
2.2	Visão Conceptual	3
2.2.1	SuS - Supervision System	5
2.2.2	MSuS - Mobile Supervision System	5
2.3	Visão Estática	5
2.3.1	SuS - Supervision System	5
2.3.2	MSuS - Mobile Supervision System	7
2.3.3	Comandos	7
2.3.4	Perguntas/Respostas	8
2.4	Visão Dinâmica	8
2.4.1	Interacção com Armazenamento de Dados	8
2.4.2	Interacção entre Sistemas	10
2.5	Visão de Dados	11
2.5.1	Membros	11
2.5.2	Ocorrências	12
2.5.3	Alocação de Trabalho	14
2.5.4	Localização	14
3	Escolhas Tecnológicas	16
4	Testabilidade da Arquitectura	17
	Referências	19

Lista de Figuras

2.1	Visão Física do Sistema.	4
2.2	Camadas de Software.	4
2.3	Visão Geral das Classes do <i>SuS</i>	6
2.4	Visão Geral das Classes do <i>MSuS</i>	7
2.5	Visão das Classes que constituem a definição de Comando.	7
2.6	Visão das Classes que constituem a definição de Pergunta/Resposta a Web services e de Dados.	8
2.7	Exemplo de inserção de dados.	9
2.8	Exemplo de leitura de dados.	10
2.9	Exemplo de interacção entre sistemas.	11
2.10	Visão de Dados - Membros.	12
2.11	Visão de Dados - Ocorrências.	13
2.12	Visão de Dados - Alocação de Trabalho.	15
2.13	Visão de Dados - Localização.	15
4.1	Exemplo de Teste ao <i>IDataStorage</i>	18

Abreviaturas e Símbolos

CRUD	Create, Read, Update and Delete
GIC	Gestão Integrada de Contra-Ordenações
MSuS	Mobile Supervision System
PDA	Personal Digital Assistant
PL/SQL	Procedural Language/Structured Query Language
PSI	Projecto de Sistemas de Informação - FEUP
SQL	Structured Query Language
SuS	Supervision System

Capítulo 1

Introdução

1.1 Objectivo

Este documento tem como finalidade apresentar e especificar a arquitectura da solução a criar no âmbito do projecto final de curso do Hugo Manuel Zenha de Pinho.

No presente documento é descrita a visão global do sistema e do meio envolvente, bem como uma visão detalhada dos diversos componentes de software que constituem a solução e o porquê das escolhas efectuadas, quer a nível das tecnologias a utilizar, quer a nível lógico da arquitectura. A arquitectura, assim como este documento, foi pensada tendo sempre em vista a melhor abordagem para atingir a resolução do desafio proposto.

1.2 Âmbito

A solução a desenvolver consiste em dois sistemas interligados, o sistema de fiscalização e a sua versão móvel. O sistema móvel tem por objectivo colher informação para que depois sincronize com o sistema de fiscalização. Para além da interacção entre estes dois sistemas, é objectivo disponibilizar mecanismos de interacção com aplicações externas de fácil expansibilidade.

1.3 Destinatários do Documento

Este documento destina-se essencialmente aos *stakeholders* do projecto em causa, os quais possuem diversos interesses na informação aqui constante. Por exemplo, para o cliente e os orientadores é uma forma de obterem um ponto de situação e acompanharem o processo de desenvolvimento do projecto, e para a equipa de desenvolvimento é uma forma de obter mais informação sobre o caminho a seguir para a concretização do projecto.

1.4 Estrutura do Documento

Este documento encontra-se estruturado tendo em vista uma compreensão rápida de toda a estrutura da arquitectura, das suas restrições e das motivações que foram catalisadoras para a criação da mesma.

Desta forma, serão inicialmente apresentadas as visões mais globais e simples da arquitectura (Overview e Visão lógica). De seguida será apresentado um conjunto de informações mais detalhado de toda a arquitectura (Visão Estática, Visão Dinâmica e Visão de Dados e Dependências). Após tais apresentações, são analisadas tecnologias passíveis de serem utilizadas na implementação do sistema e justificadas as opções tecnológicas tomadas. No final, serão analisadas as soluções arquitecturais criadas para possibilitar e facilitar os testes do sistema a desenvolver.

Capítulo 2

Representação da Arquitectura

2.1 Visão Geral

O projecto consiste na criação de um Sistema de Supervisão Móvel (*MSuS*) que irá correr numa plataforma móvel, neste caso num *PDA*. Para que este sistema seja possível de administrar, existirá um Sistema de *Back-Office*, Sistema de Supervisão (*SuS*).

A *Figura 2.1* apresenta a visão física do sistema. O *MSuS* sincroniza com o *SuS* através de *Web services*, que este último disponibiliza. Aparte desta sincronização, o *SuS* permite também interacção com aplicações externas, que pode ser através da disponibilização de *Web services* ou consulta destes.

Entre os sistemas externos, pode ser evidenciada a aplicação *GIC - Gestão Integrada de Contra-Ordenações* (para mais informações sobre esta aplicação consultar [dSdI06]), desenvolvida no *PSI*. É também objectivo possibilitar a sincronização com esta aplicação, no entanto, o *SuS/MSuS* não deverá ficar preso a essa sincronização. O *GIC* é apenas mais uma aplicação externa com que estes sistemas podem trocar informação.

2.2 Visão Conceptual

A *Figura 2.2* representa a visão conceptual pensada para os Sistemas de Fiscalização e para a forma como estes interagem.

Primeiro que tudo, é necessário ter em atenção que a forma como a Aplicação Externa (*External Application*) é organizada não é do âmbito do projecto, no entanto foi incluída para que exista um visão global das interacções possíveis.

A troca de informação entre os vários sistemas é realizada através de *Web services* (representados com conexões azuis na *Figura 2.2*). Os módulos denominados por *Responder* disponibilizam serviços, e os denominados por *Querist* consultam esses serviços disponibilizados. Estes módulos

Representação da Arquitectura

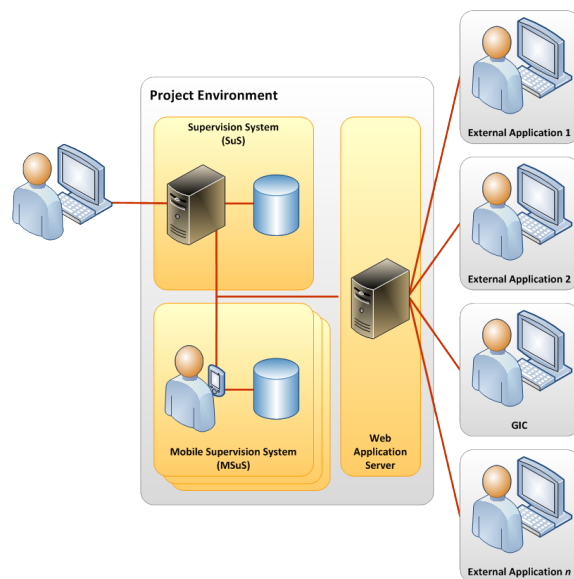


Figura 2.1: Visão Física do Sistema.

são incorporados na aplicação para que seja possível a interação entre o *SuS/MSuS* e aplicações externas. Os desenvolvedores das aplicações externas têm de se preocupar em implementar os mecanismos de consulta de serviços do *SuS*.

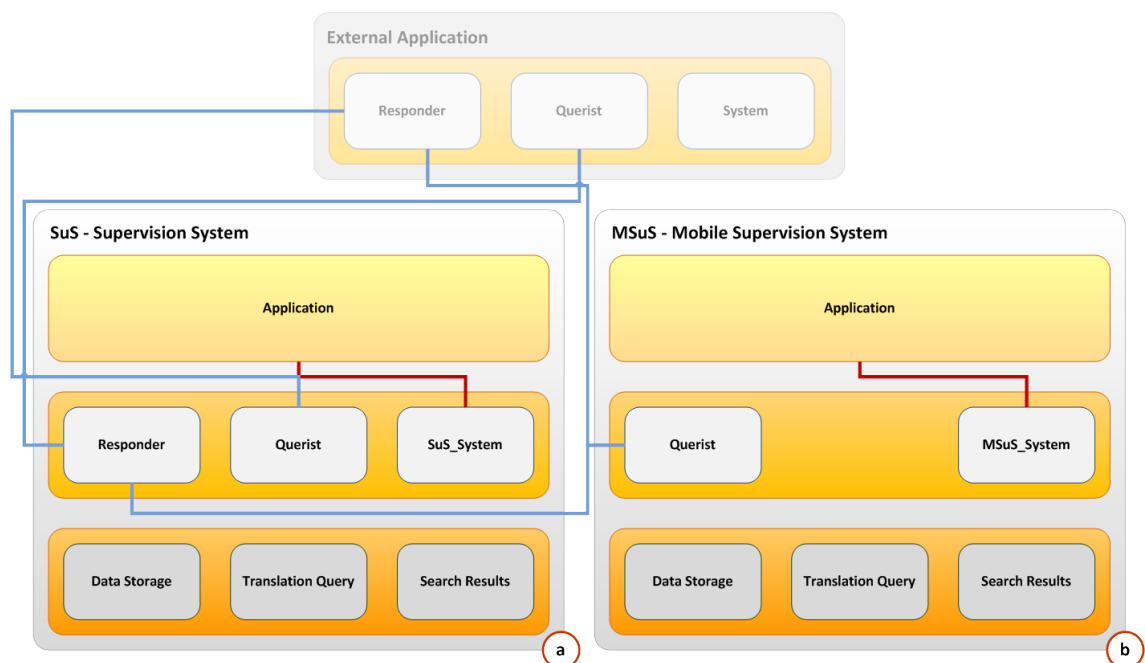


Figura 2.2: Camadas de Software.

De seguida é explicado cada um dos dois sistemas apresentados na [Figura 2.2](#).

2.2.1 SuS - Supervision System

Em relação ao *SuS* (representado na [Figura 2.2 \(a\)](#)) foram definidas três camadas de software:

- *Camada da Aplicação* (camada superior) - Interface de interacção com o utilizador, como é o caso do Administrador do Sistema de Fiscalização, através do qual são disponibilizados os serviços do *SuS*;
- *Camada de Regras ou da Lógica de Negócio* (segunda camada) - inclui todos os mecanismos de interacção com aplicações externas e com o *MSuS* (*Responder/Querist*), e também o sistema responsável pela gestão dos serviços disponibilizados (denominado por *SuS_System*);
- *Camada de Acesso a Dados* (camada inferior) - inclui todos os mecanismos de *CRUD* (*Create, Read, Update and Delete*) do armazenamento de dados (são acedidos através do *SuS_System*). Nesta camada existem o módulo de armazenamento de dados, *Data Storage*, o módulo responsável pelas tradução dos pedidos do sistema para a linguagem do módulo anterior, *Translation Query*, e por fim o módulo responsável pelo tratamento dos resultados, *Search Results*.

De notar que a *Camada de Aplicação* apenas tem acesso aos serviços disponibilizados pelo *SuS* (representado na [Figura 2.2](#) pela conexão vermelha).

2.2.2 MSuS - Mobile Supervision System

Em relação à estrutura em camadas do *MSuS* a ideia é a mesma que no *SuS*. No entanto, existem diferenças no que diz respeito aos *Web services*. Enquanto que o *SuS* disponibiliza e utiliza *Web services* disponibilizados por aplicações externas, o *MSuS* apenas os utiliza, como ilustra a [Figura 2.2 \(b\)](#).

2.3 Visão Estática

2.3.1 SuS - Supervision System

No *SuS*, *SuS_System* é a classe principal. Esta tem a responsabilidade de receber e interpretar pedidos provenientes da camada de aplicação (como apresentado na [Figura 2.2 \(a\)](#)). Para além de tal responsabilidade, também é através desta classe que são disponibilizados *Web services*, assim como mecanismos de consulta de *Web services* disponibilizados por aplicações externas.

A interface *IDataStorage* representa os dados guardados do sistema. Sendo assim, a sua funcionalidade é providenciar os mecanismos de *CRUD* do armazenamento de dados, através do método *readData*, responsável pela leitura de dados, e do método *cudData*, responsável pelas outras acções, ou seja, pela manipulação de dados.

Representação da Arquitectura

As classes do tipo *Responder* disponibilizam *Web services* para que aplicações exteriores possam interagir com o *SuS*. A disponibilização de serviços segue uma estrutura definida através do método *execute*, recebendo *IQuestion* (explicado na *Secção 2.3.4 Perguntas/Respostas*).

Por fim, a classe *Querist* possibilita o acesso a *Web services* disponibilizados por aplicações externas. A forma como tal interacção é realizada terá de ser implementada através do método *execute*, que deverá responder às necessidades específicas de cada *Web service*. As classes definidas para o *SuS* são apresentadas na *Figura 2.3*.

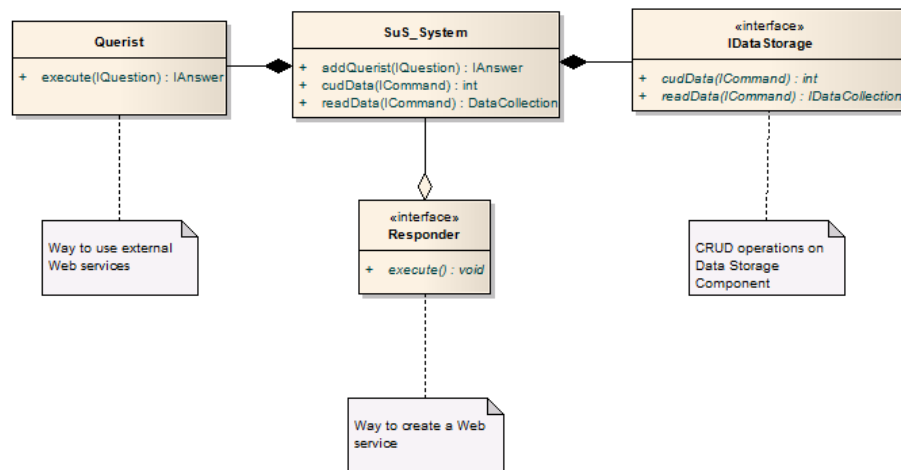


Figura 2.3: Visão Geral das Classes do *SuS*.

2.3.2 MSuS - Mobile Supervision System

O *MSuS* é idêntico ao *SuS* apresentado na secção anterior. A única excepção é a ausência dos mecanismos de disponibilização de *Web services*. Tal alteração pode ser constatada na *Figura 2.4*.

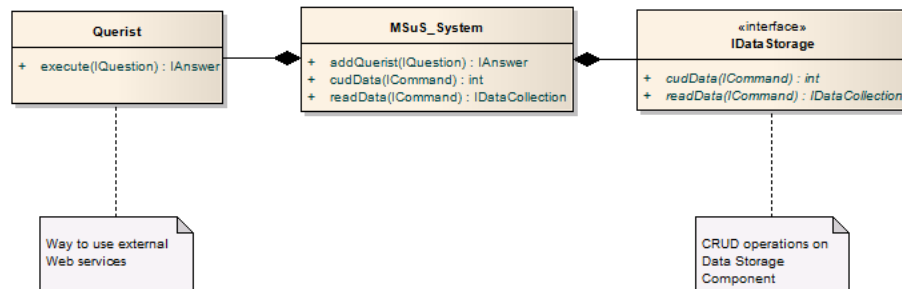


Figura 2.4: Visão Geral das Classes do *MSuS*.

2.3.3 Comandos

A abstracção do conceito de Comandos - interface *ICommand* - aparece devido à necessidade de generalizar a forma de criação de acções sobre o armazenamento de dados, tirando esse esforço da interface *IDataStorage* (ver *Figura 2.2*). Esta interface disponibiliza o método *GetQueryString()* que permite a conversão de pedidos do sistema para a linguagem do módulo de armazenamento de dados. Esta informação está representada na *Figura 2.5*.

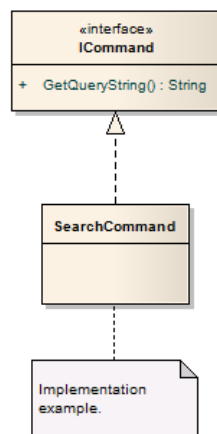


Figura 2.5: Visão das Classes que constituem a definição de Comando.

À partida são criados mecanismos que possibilitam operações *CRUD* que são, basicamente, especializações da interface *ICommand*.

2.3.4 Perguntas/Respostas

Tal como no conceito de Comandos, as Perguntas, Respostas e Dados retirados do armazenamento de dados aparecem como interfaces - *IQuestion*, *IAnswer* e *IDataCollection* - para que exista uma generalização deste conceito, tirando o esforço, quer da camada de aplicação, quer das classes *Querist* e *IDataCollection* (ver Figura 2.2). A Figura 2.6 ilustra estes mecanismos. No caso do *IQuestion* é disponibilizado o método *execute* responsável por realizar o pedido.

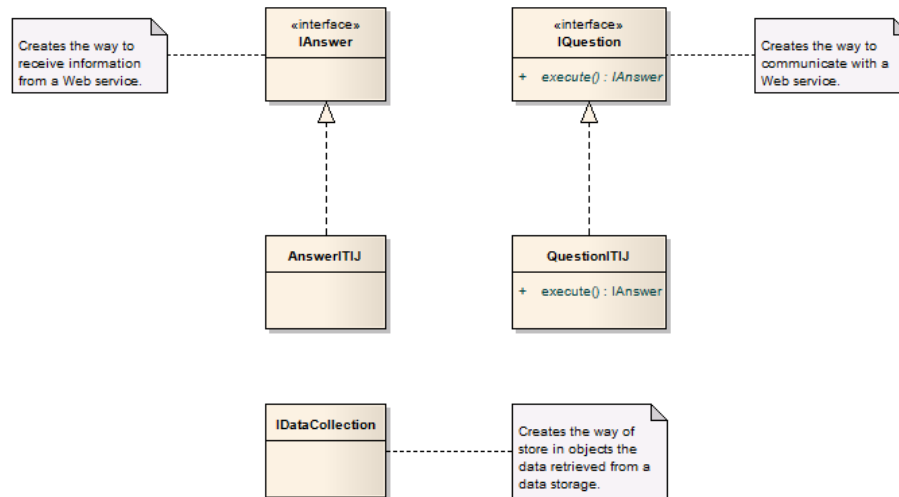


Figura 2.6: Visão das Classes que constituem a definição de Pergunta/Resposta a Web services e de Dados.

2.4 Visão Dinâmica

O comportamento dinâmico dos sistemas abordados pelo projecto é apresentado nesta secção, recorrendo-se a diagramas de sequência. Estes diagramas espelham exemplos de utilização do sistema.

2.4.1 Interacção com Armazenamento de Dados

Quando se fala da interacção com o armazenamento de dados, a questão recai sobre quais os mecanismos disponibilizados para responder aos métodos de *CRUD*. Na arquitectura proposta, existem duas formas de interacção com este armazenamento:

- *Manipulação de Dados* - através do método *cudData(ICommand):int* da interface *IDataStorage* (ver Figura 2.3 e 2.4);
- *Consulta de Dados* - através do método *rData(ICommand):IDataCollection* da mesma interface.

Nas *Figuras 2.7 e 2.8* está representado o *SuS_System*, no entanto, as operações aí apresentadas, que serão explicadas de seguida, são também válidas para o *MSuS_System*.

O exemplo ilustrado na *Figura 2.7* representa a inserção de um novo terminal no armazenamento de dados. Para tal, a camada de aplicação terá de criar uma instância de *ICommand* que reflecta esse pedido para depois pedir ao *SuS_System* para interagir com a implementação do *IDataStorage* através do método *cudData(ICommand):int*. Este método pode por alguma razão falhar. E por isso é transmitida se a interacção foi bem sucedida ou não.

No caso do segundo exemplo, ilustrado na *Figura 2.8*, o processo desenrola-se da mesma forma. No entanto, para pedir informações sobre os terminais (caso retratado neste exemplo) o *ICommand* é diferente do necessário no exemplo anterior, assim como o pedido realizado ao *SuS_System* pela *Camada de Aplicação*. O método chamado *rData(ICommand):IDataCollection* retorna a colecção de dados pedida. No entanto, em caso de erro, a classe retornada contém mecanismos de comunicação que informam dessa ocorrência à camada de aplicação.

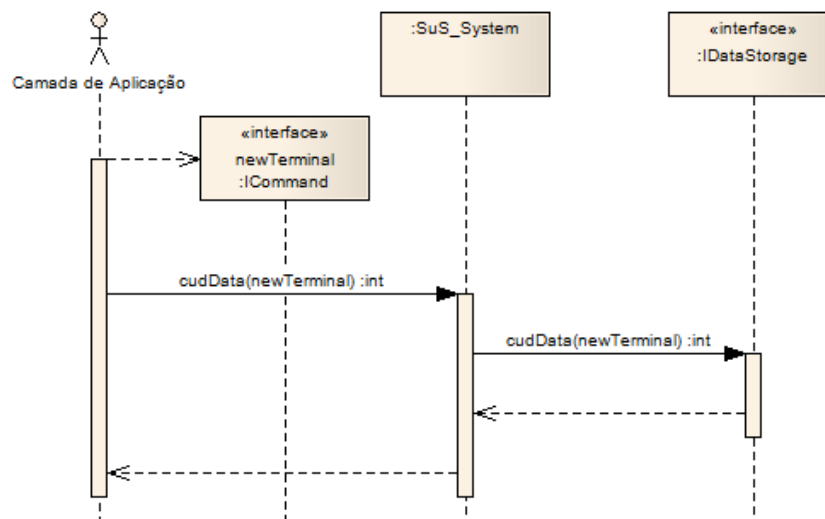


Figura 2.7: Exemplo de inserção de dados.

Representação da Arquitectura

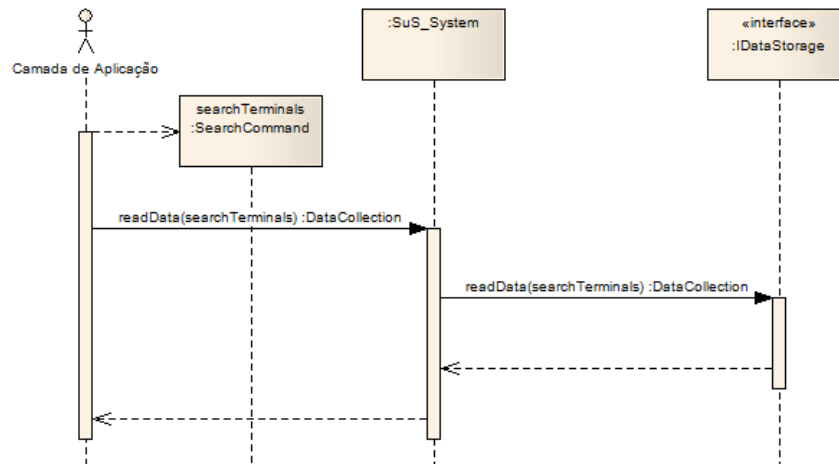


Figura 2.8: Exemplo de leitura de dados.

2.4.2 Interação entre Sistemas

Um pedido de informação por parte do *MSuS* ao *SuS* através de um Web service é ilustrado na [Figura 2.9](#). Tal pedido provém da camada de aplicação. Para tal, uma instância da interface *IQuestion* terá de ser criada. Após tal procedimento, a camada de aplicação pede ao *MSuS_System* para realizar a execução do pedido. Para que isto seja possível, o sistema adiciona um *Querist* com a *IQuestion* criada (*infringementQuestion*, que tem por objectivo pedir ao *SuS* dados sobre uma infracção). *Querist* ao executar tal pedido acede a Web services disponibilizados recebendo a resposta esperada e criando a resposta que será devolvida à camada da aplicação (contém mecanismos para comunicar ocorrências de erro). Esta primeira sequência é válida para pedidos a quaisquer Web services, quer sejam disponibilizados pelo *SuS*, quer por aplicações externas, sendo também válidas no caso da camada de aplicação ser parte integrante do *SuS* e não do *MSuS* (neste caso o *MSuS_System* é trocado pelo *SuS_System*).

Do lado do *SuS*, o Responder recebe os pedidos provenientes do exterior e executa os procedimentos necessários para responder. Estes mecanismos estão explicados na [Secção 2.4.1 Interação com Armazenamento de Dados](#).

Representação da Arquitectura

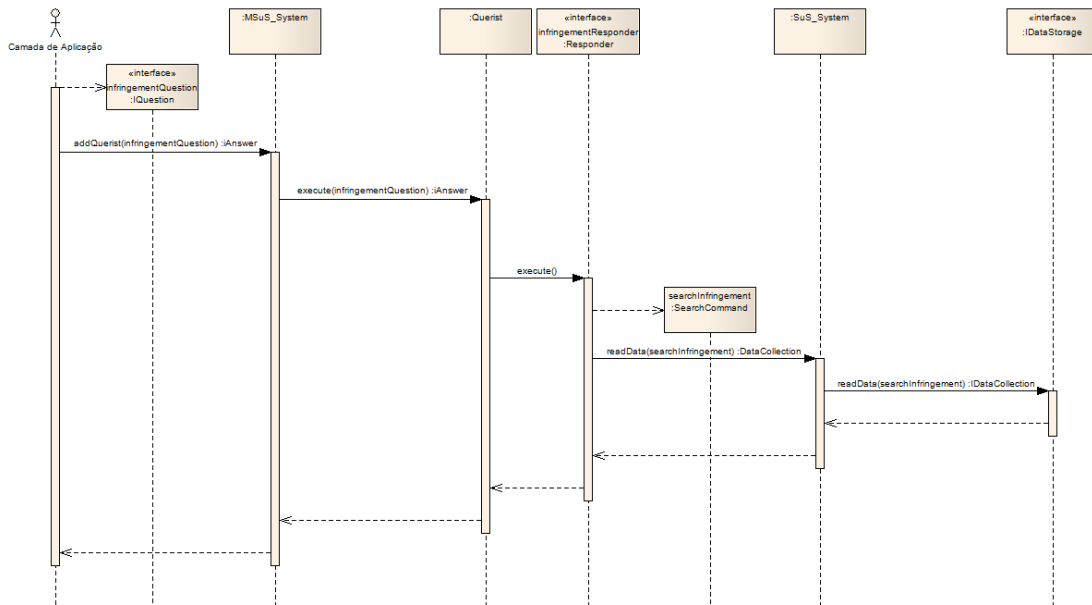


Figura 2.9: Exemplo de interacção entre sistemas.

2.5 Visão de Dados

Nesta secção é apresentada a forma como será guardada a informação dos sistemas. Para tal, serão utilizados diagramas de classes para que os dados necessários sejam mais facilmente entendidos e, também, a forma como estes se relacionam.

Para uma melhor compreensão, o diagrama de classes foi dividido em partes que são apresentadas seguidamente.

2.5.1 Membros

Sendo a solução a desenvolver centrada na mobilidade do trabalho, primeiro que tudo, é necessário criar mecanismos de armazenamento dos dados dos membros - qualquer utilizador da solução. Para tal, a informação geral sobre estes, é guardada sob a forma da classe *Membro*, na medida em que não é necessário guardar dados como o número do bilhete de identidade, entre outros (afinal é uma solução com vista a agilizar a forma de trabalho e não para gerir recursos humanos). Cada membro terá associado a si um *Perfil* - também visto como uma classe - que lhe associará as permissões que terá sobre o sistema. As diversas permissões (escrita, leitura, entre outras) dão origem à classe *Permissao*. Para além da informação associada ao *Membro* até agora descrita, também serão associadas a sua *Categoria* profissional e a *Unidade_Organica* a que pertence.

Cada *Membro* pode pertencer a uma ou várias equipas (classe *Equipa*), cujo objectivo da sua existência será explicado na [Secção 2.5.3 Alocação de Trabalho](#).

Representação da Architectura

Foi criado o conceito de *Login* para que os dados do Membro sejam separados dos mecanismos de autenticação no sistema.

Toda esta informação está representada na *Figura 2.10*, onde se pode visualizar com maior detalhe quais os atributos associados a cada classe.

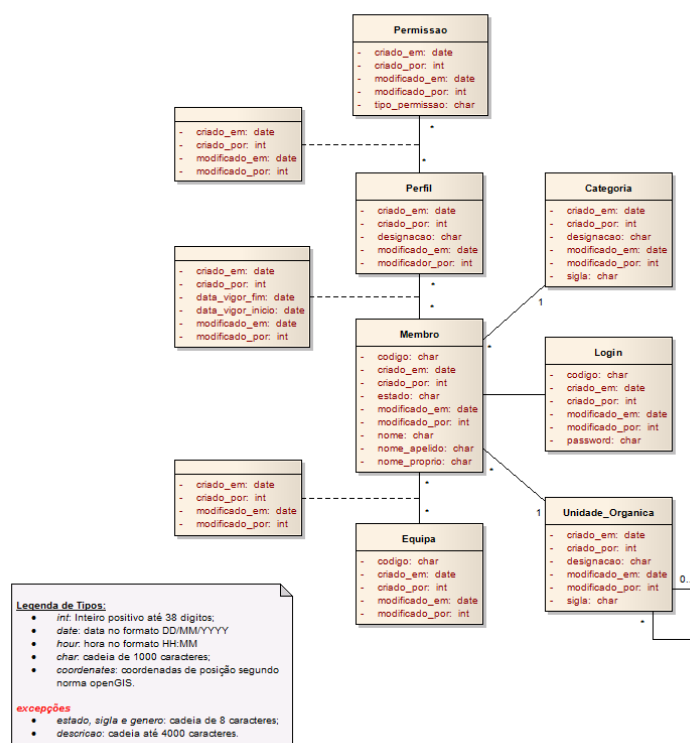


Figura 2.10: Visão de Dados - Membros.

2.5.2 Ocorrências

Para a solução a desenvolver, a parte mais crucial é a forma como serão armazenadas as ocorrências reportadas pelos *Agentes Fiscalizadores*. Assim sendo, a classe *Ocorrencia* aparece naturalmente no modelo de dados proposto, sendo caracterizada pela data e local da sua detecção por parte do *Agente Fiscalizador*, e por determinadas características que dependem do tipo da ocorrência em questão. Logo, a classe *Tipo_Ocorrencia* guarda a informação necessária para ser possível identificar quais as características (representadas pela classe *Caracteristica*) específicas a preencher aquando da detecção de tal ocorrência.

As características podem ser de variados tipos, com texto e números; mas no caso de serem ficheiros uma nova classe foi criada para tratar de tal possibilidade - classe *Ficheiro*. Também foi criada a classe *Lista* para possibilitar a criação de listas de selecção para o valor de uma característica.

Representação da Architectura

Foi criada a classe *Area* para facilitar a procura de tipos de ocorrência. Esta divide os tipos de ocorrência em conjuntos. A classe *Conhecimento* cria uma forma de disponibilização de informação sobre o método de reporte de um *Tipo_Ocorrência* ou um conjunto destas, uma *Area*. Por fim, a ligação destas áreas ao perfil do membro aparece para criar as permissões sobre estas.

Toda esta informação está representada na *Figura 2.11*, onde se pode visualizar com maior detalhe quais os atributos associados a cada classe.

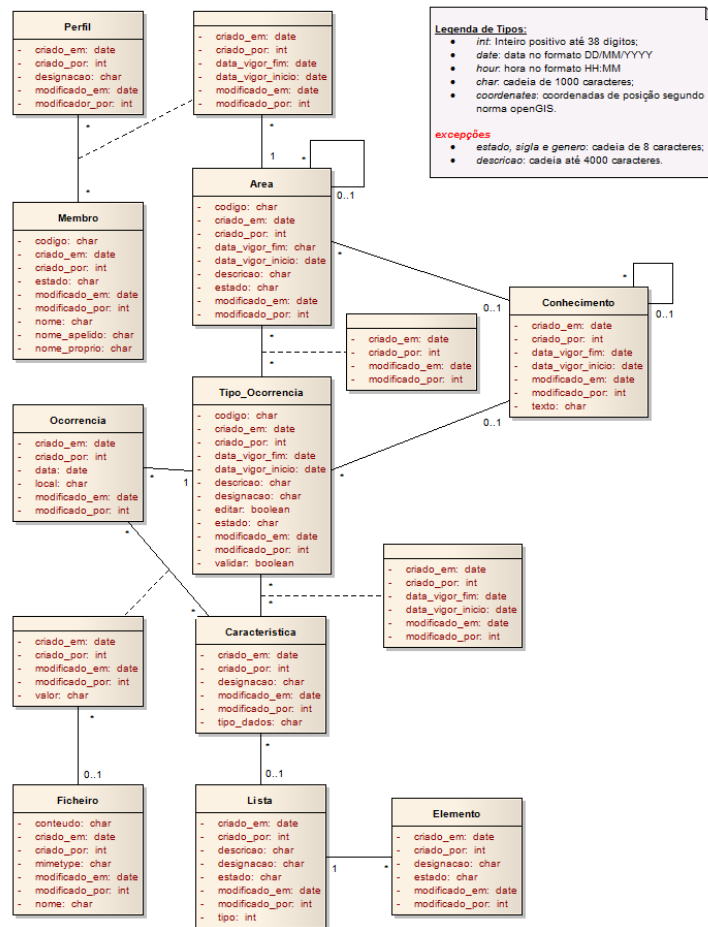


Figura 2.11: Visão de Dados - Ocorrências.

2.5.3 Alocação de Trabalho

A existência de equipas justifica-se pela necessidade da alocação de um ou mais membros a zonas do município. Uma equipa pode ser alocada (através da classe *Alocacao*) a uma zona do município e um membro a uma actividade (classe *Actividade*).

Uma actividade é composta por um conjunto de tarefas (classe *Tarefa*), que dirão se a primeira foi concluída com sucesso ou não. Uma tarefa, para um melhor aproveitamento de informação, é dividida em tipos de tarefas (classe *Tipo_Tarefa*).

Uma actividade é concluída tendo em consideração o seu atributo *genero*. Este pode assumir uma de três situações:

- *Todos*

Todos os membros associados à actividade têm de a ter terminado. O atributo *situacao* na classe associação do membro à actividade transmite este dado;

- *Um*

Quando esta actividade foi de facto completada com sucesso por um dos membros associados;

- *Cancelada*

Quando a actividade for cancelada por um membro superior, neste caso um *Pivot*.

Toda esta informação está representada na *Figura 2.12*, onde se pode visualizar com maior detalhe quais os atributos associados a cada classe.

2.5.4 Localização

Toda a informação contida na solução pretende-se que seja localizada no município. Para tal foram criadas as classes de topónimos e números de polícia organizados em zonas do município. A localização não fica por aqui. É pedida a integração do geo-posicionamento. Assim sendo, uma ocorrência para além de estar associada a um topónimo e número de polícia, também estará a uma localização geo-referenciada, dada pela classe *Localização*. Também as tarefas estarão associadas a uma localização geo-referenciada, enquanto que uma zona será definida por um polígono geo-referenciado. A convenção OpenGIS [OGC09] permite organizar esta informação num só atributo, neste caso é o atributo *coordenadas* na classe *Localização*.

Toda esta informação está representada na *Figura 2.13*, onde se encontra detalhado em mais pormenor os atributos associados a cada classe.

Representação da Architectura

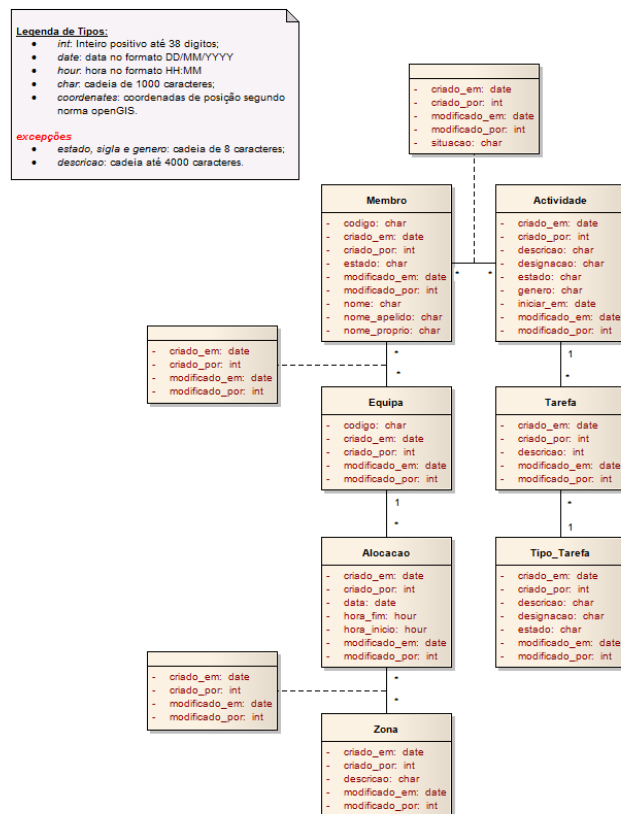


Figura 2.12: Visão de Dados - Alocação de Trabalho.

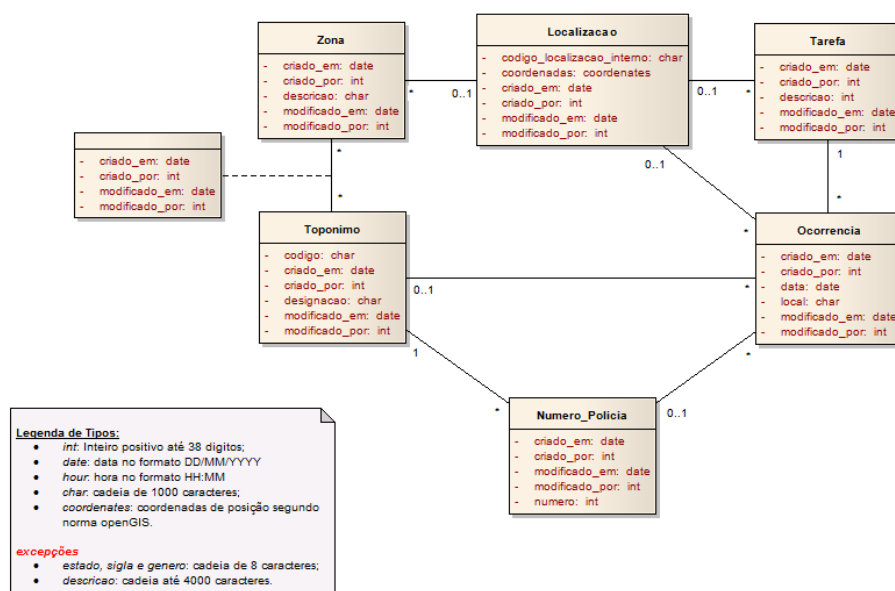


Figura 2.13: Visão de Dados - Localização.

Capítulo 3

Escolhas Tecnológicas

As escolhas tecnológicas foram realizadas após uma exaustiva análise tecnológica. Tal estudo pode ser consultado na Relatório de Análise Tecnológica [[dP09](#)], assim como quais as escolhas realizadas.

Capítulo 4

Testabilidade da Arquitectura

Como qualquer boa solução, a aqui apresentada necessita ser testável, ou seja, a arquitectura deverá permitir um teste facilitado ao bom funcionamento das várias partes que a compõem, bem como às aplicações que a utilizam.

Tendo em conta as especificidades do projecto alguns problemas no teste da solução poderão aparecer. Ao proceder à fase de testes dever-se-á ter em conta as seguintes limitações:

- **Dispositivos de Teste Limitados**

Estão disponíveis para ser utilizados em testes, um *PDA* e uma impressora para ligação a este, tendo também em consideração a possibilidade de utilização do emulador de dispositivos móveis da *Microsoft*. Os testes a grande escala só poderão ser realizados na altura de testes no terreno;

- **Tempo Necessário para um Teste em Ambiente Real**

Devido a ser uma aplicação com a vertente móvel, os testes no terreno poderão demorar mais tempo do que o normal e ser mais complicado proceder às correcções necessárias. Será complicado recriar as condições de erro em alguns casos (principalmente aquando da integração com o geo-posicionamento);

- **Complexidade de Dados**

A complexidade dos dados em questão torna complicada a introdução manual de dados para a criação de testes.

Tendo em consideração as figuras na *Secção 2.3 Visão Estática* torna-se óbvio que os testes da arquitectura apresentada centrar-se-ão na escrita e leitura de dados e para tal as *Interfaces* podem servir de plataformas de criação dos testes necessários para validar esta solução.

Um exemplo para um possível teste à arquitectura é a criação de uma implementação do *IDa-taStorage* para ligação a uma base de dados *Oracle*, como é demonstrado na *Figura 4.1*.

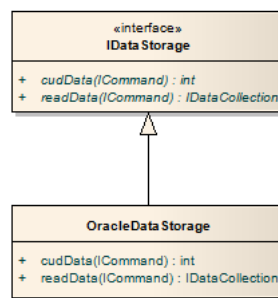


Figura 4.1: Exemplo de Teste ao IDataStorage.

Referências

- [dP09] Hugo Manuel Zenha de Pinho. Análise tecnológica. In *Aplicação de registo de autos para dispositivos móveis - Relatório de Análise Tecnológica*. Projecto de Sistemas de Software - Faculdade de Engenharia da Universidade do Porto, Março 2009.
- [dSdI06] Direcção Municipal de Sistemas de Informação. *GIC - Especificação de Requisitos*. Câmara Municipal do Porto, Outubro 2006.
- [OGC09] Inc. Open Geospatial Consortium. Opengis® standards and specifications, Maio 2009. Wikimedia, disponível em <http://www.opengeospatial.org/standards>.