



Relatório de Estágio

Mestrado em Engenharia Informática – Computação Móvel

4Forces Smart Teams – Inteligência nas Forças de Segurança

André Filipe Silva Leal

Leiria, Março de 2015



Relatório de Estágio

Mestrado em Engenharia Informática – Computação Móvel

4Forces Smart Teams – Inteligência nas Forças de Segurança

André Filipe Silva Leal

Relatório de Estágio realizado sob a orientação da Professora Catarina Isabel Ferreira Viveiros Tavares dos Reis, Docente da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria

Leiria, Março de 2015

Dedicatória

À Minha Família

Agradecimentos

Gostaria em primeiro lugar de agradecer à minha orientadora de estágio, Professora Doutora Catarina Reis da ESTG Leiria, pela confiança, apoio, disponibilidade, compreensão e enorme paciência que teve comigo durante todo o período de estágio. Mesmo nos momentos mais complicados e com as minhas falhas, o ambiente foi sempre positivo e com total apoio.

Um agradecimento também para o meu responsável da empresa onde realizei o estágio, ao António Marcelo da Tecmic, pela confiança e apoio depositado em mim.

É também devido um agradecimento aos meus amigos Joel Rodrigues, Eduardo Santos, João Belo e Miguel Afonso que estiveram sempre prontos a ajudar quando fosse necessário.

Gostaria também de agradecer a várias pessoas fora do âmbito do estágio e da ESTG Leiria, que de uma ou outra forma me deram apoio, ou simplesmente força durante esta fase da minha vida.

Próximo do fim, gostaria de agradecer aos meus pais por todo o apoio e muita paciência, que tiveram comigo desde sempre, mas principalmente neste período. Obrigado Diamantino e Rosa Leal.

Resumo

O presente estágio curricular foi realizado no âmbito da Unidade Curricular de Estágio pertencente ao mestrado de Engenharia Informática – Computação Móvel, lecionado na Escola Superior de Tecnologia e Gestão (ESTG), pertencente ao Instituto Politécnico de Leiria (IPLeiria), e teve como objetivos colocar em prática todos os conhecimentos adquiridos ao longo do percurso académico numa empresa da área, aprender no que consiste o mundo do mercado e adquirir novas competências para a vida profissional.

O estágio teve uma duração de nove meses, correspondendo a um ano letivo, na empresa Tecmic S.A. onde, durante esse período participei no desenvolvimento e remodelação de quatro projetos. Três destes projetos já existiam e foi necessário uma análise mais profunda para se verificar as funcionalidades e serviços já existentes, como também as suas dependências. Estes três projetos têm como nome “Download Data”, “Emergências Virtuais” e “Safer”. O “Download Data” baseia-se num serviço que realiza regularmente o *download* de ficheiros meteorológicos a partir de um servidor via FTP. O projeto “Emergências Virtuais” consiste num módulo que permite ajudar na formação, ou seja, através de um mecanismo de pergunta e possíveis respostas pretende-se avaliar o nível de preparação do utilizador. No caso do projeto “Safer”, nasceu da necessidade de incluir um novo grupo de utilizadores: as Forças de Segurança onde se pretende gerir as mesmas.

Por último, o projeto “4Forces Smart Teams” foi um projeto desenvolvido de raiz que consistia no desenvolvimento de uma aplicação *mobile*.

Em todos os projetos, a fase inicial consistiu no levantamento dos requisitos pretendidos, dos serviços e das funcionalidades já implementadas e por implementar. No final de cada projeto foram realizados vários testes (testes unitários e de integração), de forma a verificar se a aplicação estava de acordo com todos os requisitos anteriormente planeados, tanto a nível técnico como a nível de segurança.

Após a conclusão dos projetos (fase de prototipagem e demonstração), conseguiu-se no projeto “4Forces Smart Teams”, que é o principal foco deste relatório, obter uma aplicação *mobile* que consegue gerir tarefas e equipamentos de um utilizador, permite visualizar o ritmo cardíaco do utilizador, reportar a posição do utilizador e comunicar com a aplicação *desktop* (“Safer”).

Para o futuro já foram pensadas algumas melhorias e novas funcionalidades para o projeto “Safer” e “4Forces Smart Teams”. Em relação ao projeto “Safer” era interessante e benéfico existir um sistema de notificações, ou seja, que fosse possível notificar o utilizador através de (email, sms, etc.) com informação relevante. Outra mais-valia deste sistema era alertar a equipa de desenvolvimento quando ocorrer um problema, permitindo assim um melhor e mais rápido suporte. Outra implementação pensada é de um “sistema de limpeza” de dados removidos. Neste momento os dados não se encontram a ser removidos, para preservar o histórico de relações entre objetos. Este novo sistema de “limpeza” irá limpar esses dados, movendo os dados removidos para um sistema à parte.

Noutra vertente, temos a aplicação mobile “4Forces Smart Teams” onde já foram pensadas diversas funcionalidades novas das quais se destaca a integração de um mapa na aplicação, permitindo assim visualizar as missões e os meios. Outra funcionalidade nova passaria por permitir integrar funções de administrador, ou seja, muitas das ações feitas da aplicação “Safer” poderiam ser possíveis na aplicação *mobile*. Finalmente, integrar mecanismos como o de *upload* de imagens e notificações.

Palavras- Chaves: *Integração, Estágio, Planeamentos, Projetos e Testes.*

Abstract

This traineeship was made in the context of the Internship course, a part of the master's degree in Computer Engineering - Mobile Computing, at the School of Technology and Management (ESTG), from the Polytechnic Institute of Leiria (IPLeiria). It's main objectives were to put in practice all the gathered knowledge alongside the academic stage in a company within this field of practice.

The traineeship lasted for nine months, the equivalent of a school year at Tecmic S.A where, during that period, i was involved in the development and improvement of four projects. Three of them already existed and further analisys was required in order to verify existing services and functionalities as well as its dependencies. "Download Data", "Emergências Virtuais" and "Safer". The "Download Data" project is based on a service that regularly fetches weather files from a remote server using FTP. "Emergências Virtuais" is a module that aids in the preparation of the user by a question and answer mechanism which enables the evaluation of the user readiness. The "Safer" project was born by the need to include a new group of users: Security Forces.

At last, the "4Forces Smart Teams" project was developed from scratch and involved the creation of a mobile application.

All projects started with an initial phase with the main purpose of gathering the requirements of the services and functionalities already implemented or the ones to be implemented. At the end of each project several tests where made (unit and integration testing), in order to check if the application met the previously established requirements both in a technical and security level.

Upon the completion of the projects (demonstration and prototyping stages), the "4Forces Smart Teams" project, produced a mobile app that was able to manage the tasks and equipment of a certain user and visualize his heart rate, report his position and communicate with the desktop application ("Safer").

Some improvements have been thought for the future work as well as new functionalities for the projects "Safer" and "4Forces Smart Teams". In relation to the "Safer" project it would be beneficial to add a notification system through email (or sms) with relevant information. Another valuable function of this system would be to notify the development team of incoming issues, thus allowing a faster and better

support. In addition to the already mentioned functionalities a cleaning mechanism for removed data was also considered. At the moment, data is not being removed in order to keep history of relationships between objects. This cleaning system will wipe the mentioned data and relocate it to a separate area.

The mobile application “4Forces Smart Teams” will also benefit from several new capabilities been thought to add such as the integration of a map allowing for both mission and assets visualization. Integrating the administrator functions is another feature one that could make possible to execute “Safer” commands on the mobile application. Finally the last would be to integrate image upload mechanism and notifications.

KeyWords: *Integration, Planning, Projects, Tests and Traineeship.*

Índice

Dedicatória	v
Agradecimentos.....	vii
Resumo	ix
Abstract	xi
Índice	xiii
Índice de Figuras.....	xvii
Índice de Tabelas	xxi
Lista de Siglas	xxiii
1. Introdução.....	1
1.1. Tecmic	1
1.2. Motivação e Objetivos	2
1.3. Estrutura do Documento	4
2. Metodologia e Planeamento.....	5
2.1. Planeamento	5
2.2. Metodologia	6
3. XTran 4 Forces.....	11
3.1. Projetos	12
3.1.1. “Simulações”	15
3.1.2. “Emergências Virtuais”	17
3.1.3. “Safer”	17
3.1.4. “4Forces Smart Teams”	18
3.2. Atualizações ao 4Forces	18
3.2.1. Download Data	20
3.2.2. Emergências Virtuais.....	21
3.2.3. Safer	24
4. “4Forces Smart Teams”	41
4.1. Descrição do Projeto	42
4.2. Análise Comparativa de Plataformas Móveis	43
4.3. Implementação do Projeto	44

4.3.1.	Modelo de Domínio	44
4.3.2.	Protótipos.....	45
4.3.3.	Layout	46
4.3.4.	Testes Realizados	47
4.4.	Resultado Smart Teams - Mobile	48
5.	Conclusões	51
5.1.	Trabalho Futuro	52
	Bibliografia	53
	Anexos.....	55
	ANEXO A – Vital Jacket	57
	ANEXO B – Emergências Virtuais	61
1.	Templates de Simulação	61
2.	Templates de Situação.....	63
3.	Templates de Ações Esperadas	64
4.	Pessoas.....	65
5.	Cargos	67
6.	Executar Exercício (Modo de Administração)	67
i.	Consultar Exercício de Simulação Executado	71
ii.	Executar Exercício (Modo de utilizador).....	72
iii.	Tempo de Sincronização.....	74
	ANEXO C.....	75
1.1.	Plataformas Móveis	75
1.1.1.	Android	75
1.1.2.	iOS.....	77
1.1.3.	Windows Phone.....	77
1.2.	Comparação entre Plataformas Móveis	78
1.3.	Quota de Mercado	81
1.4.	Arquiteturas	82
1.4.1.	Cliente/servidor.....	82
1.4.2.	Peer-to-peer.....	83
	ANEXO D.....	85

ANEXO E.....	89
ANEXO F.....	91

Índice de Figuras

Figura 1 – Quadro geral do projeto XTraN4Forces na ferramenta Trello	7
Figura 2 - Metodologia de Desenvolvimento.....	8
Figura 3 - Diagrama global - XTraN 4 Forces	13
Figura 4 - Simulações disponíveis no 4 Forces.	16
Figura 5 - Arquitetura de alto nível do XTran 4 Forces	18
Figura 6 - Arquitetura Emergências Virtuais	22
Figura 7 - Modelo de domínio Emergências Virtuais	23
Figura 8 - Arquitetura Safer.....	25
Figura 9 - Funcionalidades resumidas do sistema 4 Forces [3].	25
Figura 10 - Ecrã principal da versão anterior	26
Figura 11 - Ecrã principal versão atual	27
Figura 12 – Treeview e Context Menu	28
Figura 13 - Exemplo de um user control de uma missão.....	29
Figura 14 - Gráfico com o número de ocorrências, missões e tarefas no sistema.....	29
Figura 15 - Diagrama do modelo de domínio antigo [3].....	30
Figura 16 - Diagrama do modelo de domínio novo.....	31
Figura 17 - Entidades Templates	31
Figura 18 - Entidade Requisition	32
Figura 19 - Entidade StatusHistory / VitalJacket.....	32
Figura 20 - Métodos da class MissionsController	33
Figura 21 - Processo simples de update de uma entidade.....	34
Figura 22 - Exemplo de relações entre entidades.	35
Figura 23 - Class BaseEntityExtension.....	36
Figura 24 - Parte do método PrepareToSend da entidade missão	38
Figura 25 - Ecrã Gestão de Meios - Agente.....	39
Figura 26 - Ecrã Gestão de Tempates - Missões Predefinidas	39
Figura 27 - Arquitetura 4Forces Smart Teams.....	41
Figura 28 - Funcionamento do projeto "Smart Teams".....	42
Figura 29 - Tendência dos próximos anos [21].....	43
Figura 30 - Modelo de domínio da aplicação mobile	44
Figura 31- Protótipo do ecrã de Login.....	45
Figura 32 - Protótipo do ecrã das missões	46
Figura 33 - Ecrã de login	48
Figura 34 - Ecrã de missões do agente	49
Figura 35 - Ecrã pedir um equipamento.....	49
Figura 36 - Ecrã vital jacket.....	50
Figura 37 - T-shirt.....	57
Figura 38 - Eléctrodo.....	58

Figura 39 - Dispositivo Eletrônico.....	58
Figura 40 - Cartão SD	59
Figura 41 - Carregador.....	59
Figura 42 - Painel das Emergências Virtuais	61
Figura 43 - Árvore de navegação de uma simulação.	62
Figura 44 - Conteúdo de uma simulação.	62
Figura 45 - Janela de Situações.....	63
Figura 46 - Janela de edição de Situações.....	64
Figura 47 - Janela de Ações Esperadas.	65
Figura 48 - Janela de edição das Ações Esperadas.....	65
Figura 49 - Janela de Pessoas.....	66
Figura 50 - Janela de edição de dados das pessoas.....	66
Figura 51 - Janela de Cargos.....	67
Figura 52 - Janela com exercícios em preparação e prontos a executar.	68
Figura 53 - Janela de execução de um Exercício de Simulação (em modo de preparação).	68
Figura 54 - Alterar o estado da Simulação.....	69
Figura 55 - Iniciar Simulação.....	69
Figura 56 - Lançamento de Situações (manual)	69
Figura 57 - Alterar estado da ação esperada.....	70
Figura 58 - Janela de execução de um Exercício de Simulação (em modo de execução).	70
Figura 59 - Criar template de uma situação.	70
Figura 60 - Janela para consulta dos exercícios de simulação executados.....	71
Figura 61 - Janela de consulta de Exercício de Simulação já executado.....	71
Figura 62 - Janelas de exercícios prontos a executar.	72
Figura 63 - Janela com exercício de simulação pronto a executar.....	72
Figura 64 - Janela de exercício de simulação em execução	73
Figura 65 - Janela de Exercício de Simulação em execução (vista do aluno).	73
Figura 66 - Informação sobre números relativos de dispositivos ativos a correr Android (Janeiro de 2014) [12]	76
Figura 67 - Informação sobre números relativos de dispositivos ativos a correr Android (Janeiro de 2015) [13]	76
Figura 68- Distribuição das versões do iOS [14].....	77
Figura 69 - Exemplo da arquitetura cliente/servidor nos SmartPhones.....	82
Figura 70 - Exemplo de uma arquitetura peer-to-peer nos SmartPhones	83
Figura 71 - Entidade Problem	85
Figura 72 - Entidade Scenario.....	85
Figura 73 - Entidade Occurrence	86
Figura 74 - Entidade Mission.....	86
Figura 75 - Entidade Team.....	87
Figura 76 - Entidade Group / Unit / Team	87

Figura 77 - Entidade PersonalInfo.....	88
Figura 78 - Entidade User	88

Índice de Tabelas

Tabela 1 - Cronograma com as principais atividades e projetos desenvolvidos	5
Tabela 2 - Funcionalidades dos projetos	14
Tabela 3 - Detalhes dos <i>downloads</i> a serem efetuados diariamente	20
Tabela 4 - Alterações no projeto Emergências Virtuais	24
Tabela 5 - Referência dos testes realizados no projeto "4Forces Smart Teams"	47
Tabela 6 - Kit de desenvolvimentos das principais plataformas móveis.....	78
Tabela 7 - Principais características do Android vs iOS vs WP.....	79
Tabela 8 - Diferenças das stores do Android vs Apple vs Microsoft.....	80
Tabela 9 – Quota de mercado no ano de 2012	81
Tabela 10 – Quota de mercado no ano de 2013	81
Tabela 11 - Teste Unitários do projeto “4Forces Smart Teams”.....	89
Tabela 12 - Testes de Integração do projeto “4Forces Smart Teams”	91

Lista de Siglas

Sigla	Descrição
4Forces	XTraN 4 Forces
API	Application Programming Interface
CLR	Common Language Runtime
FS	Forças de Segurança
FST	4Forces Smart Teams
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
JSON	Java Script Object Notation
KML	Keyhole Markup Language
MOHID	Modelo Hidrodinâmico
MVC	Model View Control
SAFER	System for Advanced FiEld opeRation
SIGEL	Sistema Integrado de Gestão de Emergência e Logística
SQL	Structured Query Language
TECMIC S.A.	Tecnologias de Microeletrónica, S.A.
UC	User Control
UCInfo	User Control Information
UI	User Interface

1. Introdução

Nos últimos anos conseguimos observar um aumento exponencial no uso das tecnologias que cada vez mais, aliadas a uma grande diversidade de equipamentos, nos permitem aceder e interagir com as situações de diferentes formas e a partir de qualquer local.

Uma das áreas de gestão cujo domínio se foca no melhor aproveitamento de recursos e diminuição dos impactos resultantes de incidentes de cariz grave que podem comprometer a segurança e saúde pública é a gestão de emergências. A gestão de emergências trata do processo de acompanhamento e gestão de situações que possam ocorrer tais como incêndios, cheias, sismos, derrames, etc.

É na área referida acima que se aplica um sistema que auxilia na área operacional, na administração, bem como na formação. O sistema é denominado de 4Forces Smart Teams – Inteligência nas Forças de Segurança (FST).

Para poder oferecer um melhor suporte ao nível de interação e funcionalidades este sistema caracteriza-se por estar em constante evolução. Com estes avanços, o FST fica habilitado a agilizar a tarefa de gestão das emergências assim como ajudar na formação sustentável de futuros colaboradores.

1.1. Tecmic

A entidade de acolhimento do estágio Tecmic S.A. - Tecnologias de Microeletrónica, S.A. fundada em 1988, cresceu como membro de um dos maiores grupos em tecnologias de informação, a AITEC (www.aitec.pt), de quem é neste momento acionista de referência. Foi originada num instituto de investigação e desenvolvimento de renome nacional e internacional, o INESC (www.inesc.pt). Tendo a sua origem num processo de transferência de tecnologia Universidade – Indústria (envolvendo a maior universidade portuguesa de engenharia, o IST – Instituto Superior Técnico, (www.ist.utl.pt), a Tecmic manteve parcerias e instalou capacidade própria de desenvolvimento e inovação de forma contínua, sendo hoje, líder nas soluções que oferece.

A Tecmic é uma empresa com soluções integradas de engenharia e sistemas de informação que otimizam a gestão dos recursos das organizações, sejam eles externos (como por exemplo frotas de veículos, meios de combate a emergências, etc.) ou internos (como por exemplo pessoas, bens, etc.), em diversos setores de atividade.

Tendo comemorado em 2009 o seu vigésimo aniversário, a Tecmic é na atualidade uma referência no mercado de sistemas de gestão de frotas, meios e ocorrências - as suas soluções equipam empresas nacionais, tendo na carteira de clientes, empresas como a Companhia Carris de Ferro de Lisboa, o Grupo Luís Simões, os Horários do Funchal, os CTT, a Motorola, a Esegur, o ACP, e muitas outras. Com presença em sete países e um Volume de negócios anual de mais de 4 milhões de euros.

Dando continuidade à sua estratégia de desenvolver soluções com um elevado grau de inteligência aplicacional, a Tecmic conta na atualidade com plataformas únicas a nível mundial, nas áreas de Gestão de Emergências (XTraN 4Forces), na Gestão Integrada da Recolha de Resíduos (Ecogest) e soluções de Gestão de Transporte, nomeadamente no transporte público de passageiros (XTraN Passenger).

São mantidas com Universidades, Institutos de Investigação e Escolas Técnicas e Politécnicas, portuguesas e internacionais, parcerias de investigação, inovação e formação avançada, e é com base nelas que a Tecmic evolui a sua oferta e inova nas diversas vertentes organizacionais, para além de crescer e renovar a sua equipa técnica e de gestão.

A Tecmic tem a sede situada em Leiria, estando as áreas operacionais a funcionar em Lisboa e Brasil.

1.2. Motivação e Objetivos

Cada vez mais a prevenção e estar preparado para as situações de emergência e catástrofe pode fazer toda a diferença a prevenir inúmeras baixas e prejuízos. Devido a vários fatores, como por exemplo, a falta de veículos ou recursos humanos é impossível estar em todo o lado ao mesmo tempo e é necessário organizar os recursos para onde a atenção é mais necessária em determinado momento ou para onde pode causar maior efeito no combate ao problema. É extremamente importante que haja uma gestão eficaz

dos recursos para que não haja recursos em falta, zonas sem serem atendidas ou recursos duplicados.

Com este pensamento foram desenvolvidos vários projetos para auxiliar o automatismo do processo de gestão para agilizar as tarefas diárias e prever situações. De um leque vasto de projetos desenvolvidos, os projetos das “Simulações”, “Emergências Virtuais”, “Safer” e “4Forces Smart Teams” destacam-se, pois, foram os projetos no qual tive a oportunidade de trabalhar.

O projeto das “Simulações” tem como objetivo prever e preparar a reação a diversos tipos de ocorrências ou catástrofes. Pode-se dar como exemplo a evolução de um incêndio para fazer frentes de fogo ou, em caso de naufrágio ou afogamento, saber as mais prováveis localizações onde um corpo pode estar depois de cair à água.

Noutra vertente, temos o projeto das emergências virtuais que tem como principal função a formação. Esta formação é transmitida através de um sistema de perguntas/respostas, ou seja, o formador coloca uma situação perante o formando e este terá de indicar quais as ações que faria para obter um melhor resultado. Podemos dar como exemplo, uma situação de “Alerta de Bomba”, onde o formando tinha à escolha um leque de ações que poderia tomar, mas a resposta correta seria “Evacuar Zona”.

É necessário que haja um local onde a informação possa ser guardada e gerida. Este cenário foi criado através do projeto “Safer”, que permite enquadrar na área de gestão de emergências, uma área especializada para as forças de segurança. Este projeto pretende englobar toda a parte de gestão de recursos, meios, utilizadores, tarefas e planeamentos diários.

Por último, temos o projeto “4Forces Smart Teams” que tem como objetivo auxiliar o utilizador em qualquer lugar. Para isso foi desenvolvido uma aplicação *mobile* que permite visualizar as tarefas diárias, lista de equipamentos ou informações relevantes como o estado atual do utilizador ou ter acesso a serviços como monitorizar a sua pulsação cardíaca.

1.3. Estrutura do Documento

O presente documento encontra-se organizado de forma a permitir que o leitor compreenda a evolução e o progresso de todo o estágio. Estas fases vão desde a introdução, passando pelo trabalho desenvolvido e terminando nas conclusões.

Esta secção descreve a organização deste documento, sendo que os restantes capítulos deste relatório estão organizados do seguinte modo:

- Capítulo 2: Aqui encontra-se descrita a metodologia utilizada para o desenvolvimento do projeto, assim como o planeamento efetuado e consideração sobre a gestão de projeto.
- Capítulo 3: Neste capítulo refere o âmbito do estágio que apresenta o estado do projeto Xtran 4 Forces tal como existia no início do estágio e que enquadra um pouco dos projetos a serem abordados.
- Capítulo 4: Este capítulo descreve em pormenor todo o desenvolvimento realizado para este projeto começando por explicar as opções tecnológicas e a arquitetura do sistema existente, passando para o modelo de domínio e implementação das aplicações servidora e cliente. Neste capítulo é abordado o tema central deste relatório de estágio, ou seja, 4Forces Smart Teams.

Por último, são discutidos resultados e são tecidas considerações a ter em conta em trabalho futuro.

2. Metodologia e Planeamento

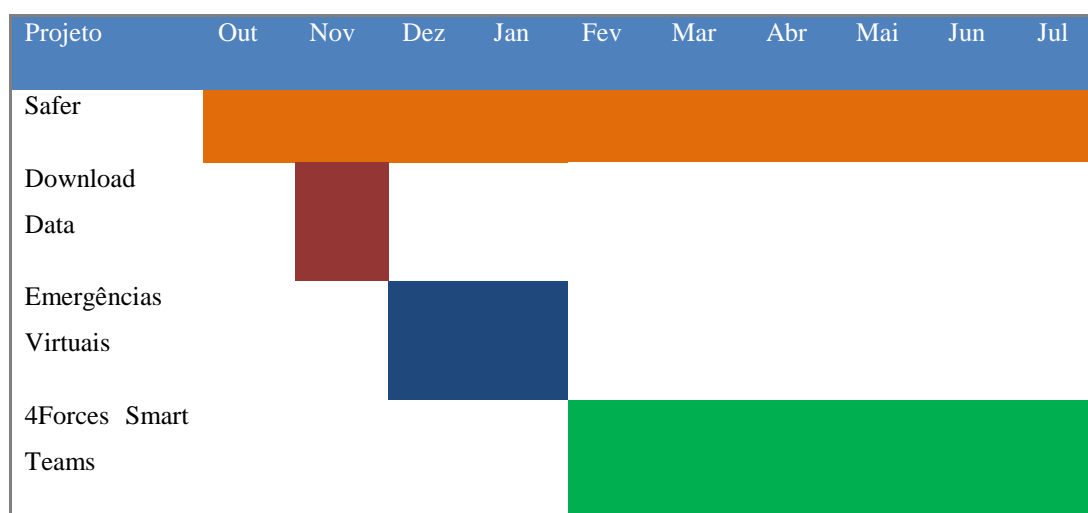
Este capítulo descreve a metodologia concebida para utilizar no desenvolvimento do projeto “4Forces Smart Teams”, assim como no planeamento de outros projetos realizados (“SAFER”, “Emergências Virtuais”, “Download Data”).

2.1. Planeamento

O estágio teve início a 11 de Outubro de 2013 e terminou a 31 de Julho de 2014. Tendo o estágio decorrido num contexto empresarial, foi necessária a existência de um período de adaptação, com a finalidade de integração na equipa de desenvolvimento e nas instalações da empresa, em Leiria.

As primeiras semanas consistiram na instalação de *software* de forma a criar uma base de trabalho sólida, com todas as ferramentas de trabalho necessárias, assim como na formação interna acerca dos vários produtos, já implementados ou ainda em desenvolvimento, que constituem o repertório da Tecmic. Este período permitiu obter uma melhor perceção acerca do modelo de negócio da empresa.

Tabela 1 - Cronograma com as principais atividades e projetos desenvolvidos



Na Tabela 1 podemos observar o tempo despendido em cada um dos projetos que tiveram mais ênfase neste estágio.

Inicialmente foi feito um levantamento dos requisitos existentes e de funcionalidades implementadas no projeto Safer, e após uma análise decidiu-se alterar vários aspetos no qual se destaca o nível visual. Estas alterações serão abordadas no capítulo 3.

Paralelamente existiu a necessidade de desenvolver e remodelar outros projetos, como o Download Data e as Emergências Virtuais. No caso do Download Data, a sua remodelação teve início em Novembro e acabou no início de Dezembro e consistiu em criar um serviço capaz de descarregar ficheiros que posteriormente serão utilizados no projeto das simulações. Relativamente ao projeto das Emergências Virtuais, teve início em Dezembro e terminou no final de Janeiro. É importante destacar que apenas a primeira fase foi terminada e novos requisitos irão ser apresentados futuramente.

Por último, temos o projeto 4Forces Smart Teams, onde se pretende desenvolver uma aplicação para telemóvel para suportar funcionalidades que interajam com o projeto Safer. Este projeto foi uma autoproposta apresentada ao gestor de projeto através de um protótipo com algumas funcionalidades, que após reunião, foi validado e aprovado. Teve início em Fevereiro e terminou em Agosto no final do estágio.

É de salientar que o projeto Safer e Smart Teams foram sendo desenvolvidos paralelamente, pois grande parte da informação disponibilizada no projeto Smart Teams pode ser consultada no projeto Safer.

2.2. Metodologia

Devido à existência de inúmeras tarefas diárias e de ideias que iam surgindo ao longo do desenvolvimento, foi utilizada uma ferramenta *online*, Trello [1] que possibilita uma grande organização do trabalho. Numa forma simplista o que o Trello permite fazer é organizar as tarefas com cartões onde cada cartão tem a hipótese de se inserir descrição, imagens, comentários, listas, pessoas designadas para determinada tarefa, entre outros.

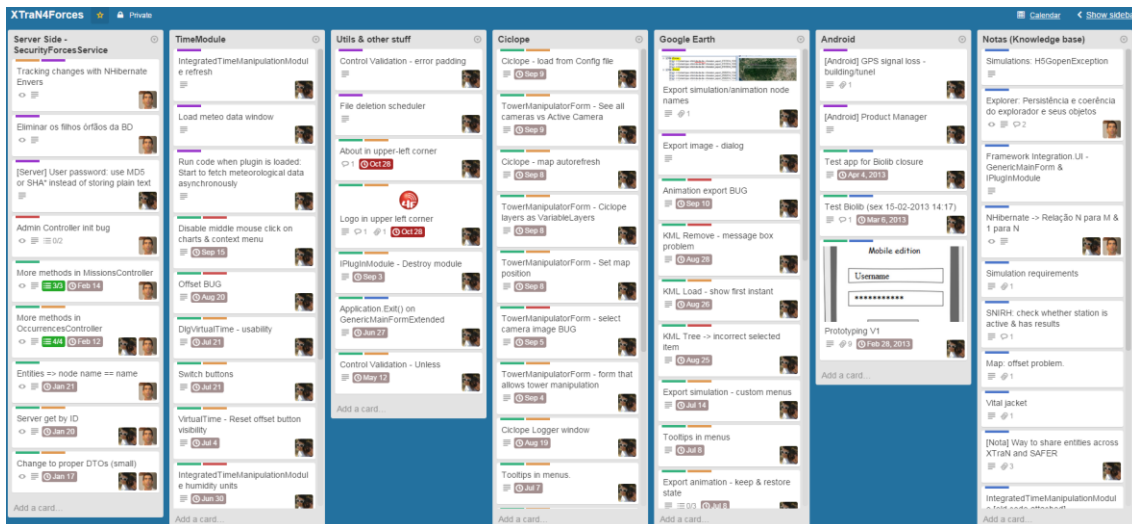


Figura 1 – Quadro geral do projeto XTraN4Forces na ferramenta Trello

Na Figura 1 é ilustrado o quadro geral do projeto XTraN4Forces na ferramenta Trello, onde é possível visualizar os vários cartões criados com descrições e imagens. A cor (roxo, laranja, verde, amarelo, vermelho, azul) por baixo dos títulos dos cartões corresponde aos estados das tarefas, ou seja, por exemplo, a cor verde corresponde a uma tarefa completa, laranja que é uma *feature* ou roxo a uma ideia. Além disso pode-se agrupar as tarefas por grupos, colocar alarmes de *e-mails* entre outras inúmeras funcionalidades que esta ferramenta disponibiliza.

Para auxiliar o desenvolvimento do projeto, foi necessária a definição de uma metodologia de suporte ao desenvolvimento. A metodologia seguida é baseada no modelo de desenvolvimento em cascata (*waterfall*), que define que etapas relativas ao desenvolvimento de *software* ocorrem de um modo sequencial [2]. Optou-se por esta abordagem porque é a metodologia de desenvolvimento de *software* adotada na empresa. No entanto, houve a necessidade de algumas adaptações na metodologia, que permite a navegação bidirecional entre etapas de desenvolvimento. Esta navegação bidirecional é justificável pelo contexto empresarial em que se insere o projeto, ambiente em que as alterações de requisitos são frequentes, de forma a adaptar-se às várias necessidades dos clientes. Nenhuma das etapas é imune a erros, e a navegação entre etapas de forma unidirecional implicaria acarretar esses mesmos erros para etapas futuras, o que não é desejável, principalmente quando os erros ocorrem numa fase inicial do projeto.

Na Figura 2 pode-se visualizar as etapas da metodologia utilizada.

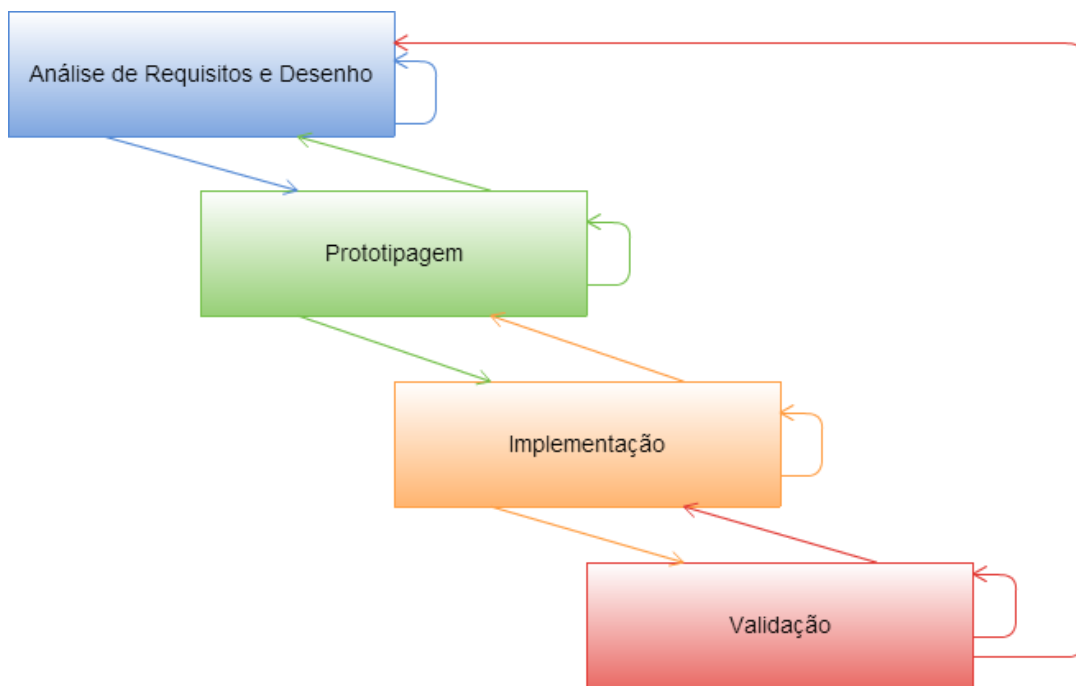


Figura 2 - Metodologia de Desenvolvimento

Inicialmente fez-se uma análise de requisitos onde se apurou as necessidades para os projetos. Passando, de seguida para o redesenho do sistema. A prototipagem foi o passo seguinte, onde se construíram protótipos.

Após ter o conhecimento dos vários módulos do projeto e como este deve interagir, seguiu-se para a parte da implementação. Esta é a fase em que é escrito o código das aplicações que fazem parte do sistema final. O código escrito está todo em inglês, segue as normas de escrita para as linguagens de programação em utilização e o código tem sempre comentários descritivos, num formato de sumário/javadoc para todas as classes, propriedades e métodos. Este processo garante que o código está sempre documentado. É também na fase de implementação que é realmente validado o resultado da etapa da prototipagem.

Por último, temos a etapa de validação, que consiste maioritariamente em testes de aceitação internos, realizados aquando da implementação de novas funcionalidades para validar o estado do desenvolvimento.

Cada uma das etapas tem a parte de refinamento, ou seja, após a conclusão parcial de cada etapa ou no início da etapa seguinte, por vezes, torna-se necessário pequenas alterações que não foram detetadas aquando a sua construção.

O ciclo desta metodologia é repetido semana a semana, pois é realizada uma reunião semanalmente com o gestor de projeto onde é feita uma validação sobre o que está feito e são agendadas novas tarefas e melhoramentos. Deste modo, se a validação for bem-sucedida passa-se para uma nova tarefa onde é iniciado um novo ciclo da metodologia. Caso exista algum problema na validação, faz-se uma análise do que é preciso alterar e se necessário retrocede-se em etapas para que os objetivos sejam alcançados.

3. XTran 4 Forces

A Tecmic possui uma solução de um sistema XTraN para gestão de frotas onde nasceu o sistema XTranN 4 Forces. Este sistema é uma evolução de um outro sistema denominado SIGEL – Sistema Integrado de Gestão de Emergência e Logística. Este nome provém das quatro forças naturais que compõem a natureza: fogo, água, ar e terra [3].

Uma boa organização e gestão de recursos diminui riscos e permite uma resposta mais rápida e eficiente. Foi com esta visão que foi desenvolvido o XTran 4 Forces, permitindo ter acesso a informação, como por exemplo, quando é necessário um agente descansar ou quando foi a sua última refeição, podendo assim, auxiliar e assegurar um bom funcionamento das tarefas a desempenhar pelos agentes (bombeiros, socorristas, exército, proteção civil, marinha, forças de segurança, etc), assim como evitar a duplicação de agentes no terreno por parte de diferentes entidades.

O XTran 4 Forces possui um módulo que consiste num modelo baseado em ameaças (perigo de ocorrência) que recorrendo a diversas metodologias (processos matemáticos, manipulação computadorizada de diferentes tipos de dados – mapas, cartas militares, etc. - e algoritmos de modelação) possibilita a previsão e o acompanhamento da evolução dos mais diversos fenómenos naturais como é o caso de cheias, correntes marítimas, incêndios ou dispersões atmosféricas. Esta é sem dúvida uma característica fundamental para se ter acesso às condições ideais que permitam dar a melhor resposta no menor intervalo de tempo.

O recurso a sistemas externos como fonte de informação adicional na elaboração das referidas modelações é muitas vezes uma mais-valia que conseguindo acesso a uma maior quantidade de fatores tem como consequência uma melhoria substancial nos resultados obtidos. Como exemplo, pode ser referida a previsão de condições meteorológicas para um determinado intervalo de tempo.

Intervalo este que é configurado pelos utilizadores através de uma data inicial e uma data final e que é tido como base na realização de todos os cálculos para a obtenção dos resultados que podem ser visualizados em mapas bi e tridimensionais.

Relatórios e *dashboards* são assim obrigatórios, para se poder ter uma visão precisa da *performance* na gestão de recursos, devido às grandes quantidades de informação que são geradas.

3.1. Projetos

Hoje em dia é extremamente importante assegurar que um sistema tenha a capacidade de crescer, não pondo em causa o seu funcionamento. Deste modo, este sistema foi pensado para ter a capacidade de escalabilidade que permite assim desenvolver novas funcionalidades e customizar as já existentes sem consumir um enorme esforço em trabalho. Com este princípio a Tecmic criou um sistema de *plugins* que permite incorporar várias aplicações numa mesma UI (*User Interface*).

O sistema de *plugins* é uma das grandes vantagens XTraN 4 Forces. Esta abordagem permite incorporar várias aplicações numa só UI. Estas aplicações podem ser bastante distintas em modo de funcionamento e funcionalidades oferecidas, sendo que o único requisito para que uma aplicação possa ser considerada *plugin* é a conformidade com a estrutura de *plugins* utilizada [3].

Como é possível observar na Figura 3, o sistema XTraN 4 Forces abrange diversos projetos dos quais se destacam: Simulações, Emergências Virtuais, Safer e o 4Forces Smart Teams. Dentro dos retângulos com a cor esverdeada (Simulações, Emergências Virtuais, Safer, 4Forces Smart Teams) temos os projetos construídos ou remodelados neste estágio. Alguns desses projetos têm alguns serviços associados que são destacados com a cor rosa (Download Data, Vital Jacket, Serviço GPS, Serviço Vital Jacket). Os projetos e serviços dentro dos círculos azuis (Simulações, Emergências Virtuais, Safer) correspondem ao que foi remodelado, e, dentro do círculo verde (4Forces Smart Teams) está o que foi construído de raiz.

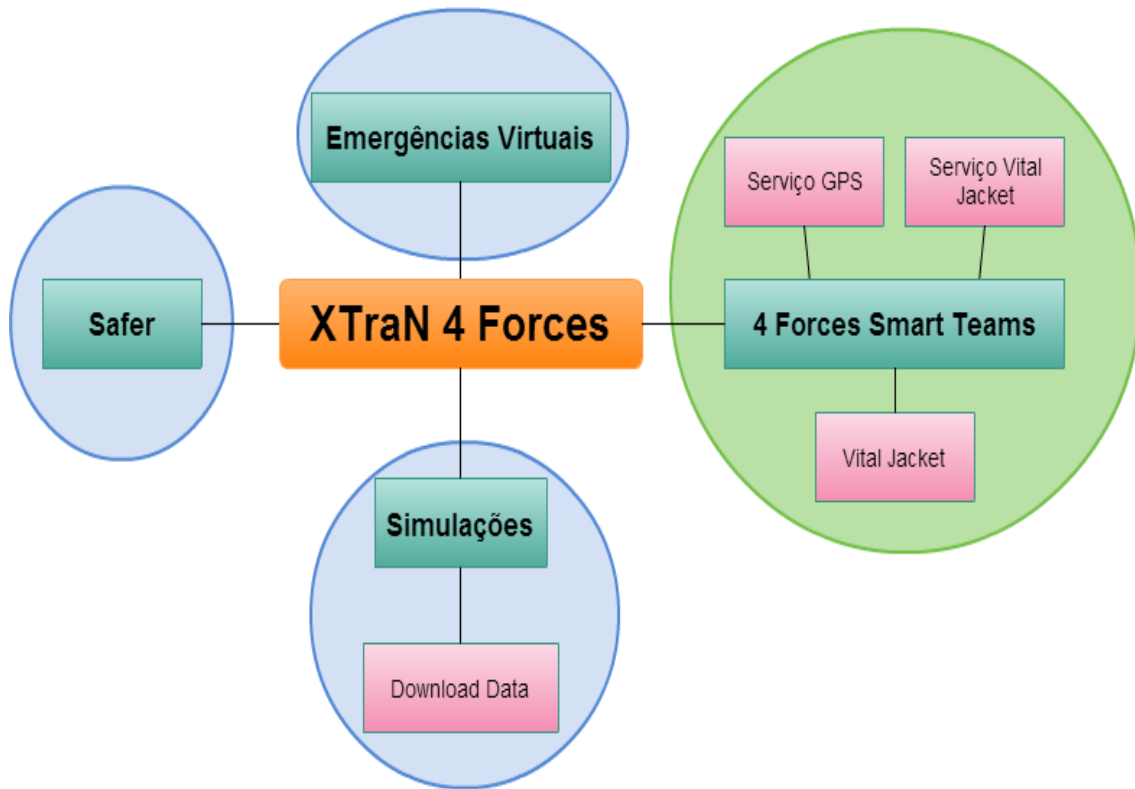


Figura 3 - Diagrama global - XTraN 4 Forces

Todos os projetos mencionados em cima sofreram alterações realizadas por mim, no entanto alguns deles já existiam. Na Tabela 2, consegue-se observar o que existia e o que foi feito em cada um dos projetos.

Tabela 2 - Funcionalidades dos projetos

Nome Projeto	Existente	Novo
Simulações	Modelos matemáticos	
	Download Data modo teste	Download Data como serviço
	Diversos tipos de simulações	
Emergências Virtuais	Estrutura existente mas não compilava	Alterações de modo a permitir compilar e utilizar as funcionalidades
	Projeto isolado	Integrar com o sistema de <i>plugins</i>
Safer	Modelo de domínio	Novas Entidades
	Servidor	DTO - Data Transfer Object
	Cliente	Aspeto gráfico Novas funcionalidades
4 Forces Smart Teams		Aplicação desenvolvida de raiz.

No caso do projeto “Simulações”, o trabalho desenvolvido foi o serviço Download Data que é composto por um serviço que efetua regularmente o *download* dos ficheiros meteorológicos (hdf5) a um servidor via FTP.

No projeto “Emergências Virtuais” pretendeu-se integrar funcionalidades já existentes no projeto XTran4Forces nas quais se destaca a gestão de testes de formação. Estes testes são criados numa base de perguntas e possíveis respostas, que neste caso correspondem a situações e ações. Para alcançar o objetivo de integração foi necessário fazer várias alterações que passaram por atualizar bibliotecas, remodelar a base de dados e alterar mecanismos internos já implementados.

O projeto “Safer” foi o que teve mais alterações a nível de *design* e modelo de domínio. De uma forma simplista, o que se manteve foi a estrutura base e os mecanismos de comunicação com o servidor.

Por último, temos o projeto “4Forces Smart Teams”, que ao contrário dos outros foi desenvolvido de raiz. Este projeto consiste numa aplicação móvel com funcionalidades que permitem aos agentes monitorizarem as suas tarefas diárias, obter informações ou requisitar equipamentos.

Primeiro irá ser feito um *overview* sobre os projetos identificando as suas principais funcionalidades e, de seguida, irá detalhar-se as alterações realizadas a cada um deles.

3.1.1. “Simulações”

O 4Forces está munido de múltiplas técnicas de simulação que possibilita o acompanhamento da evolução de vários cenários, como incêndios, cheias, descargas de corpos ou dispersões atmosféricas. As simulações, depois de executadas, podem ser visualizadas em mapas bidimensionais e tridimensionais, providenciando de forma fácil e intuitiva ao utilizador (agente ou responsável) informação acerca das possíveis evoluções de determinada emergência.

O sistema 4Forces faz-se acompanhar de várias ferramentas de carácter simulatório que permitem a visualização da progressão de diversas situações de catástrofe como é o caso de cheias, descargas de corpos, dispersões atmosféricas ou incêndios. O resultado das simulações é representado sob a forma de mapas bidimensionais e tridimensionais que fornecem ao utilizador informação relativa a uma emergência sobre os seus possíveis estados evolutivos.

De seguida (Figura 4) apresentam-se diferentes tipos de simulações cujo desenvolvimento ficou a cargo de entidades não pertencentes à Tecmic como é o caso da Action Modulers [4] e BlueCape [5]. Com base no Modelo Hidrodinâmico (MOHID) resultaram as simulações da Action Modulers relativas a cheias, urbanas e fluviais, localização de corpos humanos na água e derrames petrolíferos. A BlueCape usando algoritmos que têm em consideração fatores como a topografia do terreno ou a velocidade do vento é responsável pelas simulações de incêndios assim como de dispersões atmosféricas baseadas na modelação CALPUFF [6].

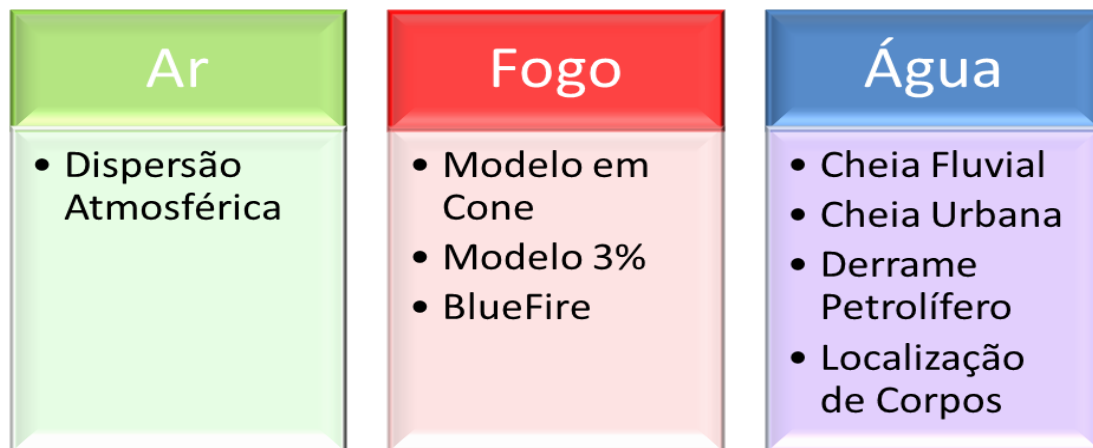


Figura 4 - Simulações disponíveis no 4 Forces.

A realização das simulações tem como requisito o uso de dados meteorológicos atualizados que se traduz na obtenção destes com regularidade. Sendo assim e para realizar a tarefa mencionada acima existe um serviço que agrega toda essa informação necessária e atualizada periodicamente permitindo assim que o sistema possa efetuar as simulações da forma mais eficiente e precisa.

A visualização dos resultados das simulações é feita através de um mapa existindo ainda a possibilidade de os resultados poderem ser vistos em função do tempo (intervalo especificado pelo utilizador) usando para isso um *plugin* de controlo temporal. Como exemplo, se se tiver em conta uma simulação para um intervalo de tempo de 2 horas é possível ao utilizador obter os resultados para qualquer altura desse respetivo intervalo podendo-se assim ter um acompanhamento da progressão da simulação ao longo tempo pois os resultados destas são em função do tempo.

É possível ainda aceder aos resultados das simulações através do *software* Google Earth no formato (KML – Keyhole Markup Language) recorrendo para isso a um *plugin* que incorpora ainda o próprio Google Earth embutido no ambiente gráfico 4Forces. Assim pode-se visualizar os resultados obtidos em mapas 3D que com a adição de funcionalidade do *plugin* de controlo temporal pode reproduzir a informação de forma animada aquando da definição por parte do utilizador de intervalos de tempo muito curtos.

3.1.2. “Emergências Virtuais”

Este projeto foi iniciado pela empresa Action Modulers que visa dar formação, simulando virtualmente emergências virtuais. Este projeto existia isolado e autónomo antes de se pensar em integra-lo na aplicação 4 Forces simultaneamente com a integração do projeto SAFER (que consiste na gestão das Forças de Segurança (FS)). O *plugin* consiste num módulo de treino para agentes em que é possível avaliar o nível de preparação individual para dar resposta a possíveis emergências. Apesar de não ter influência direta no projeto SAFER, será uma mais-valia para as FS, permitindo treino regular, dentro do próprio escritório, de forma a manter sempre os agentes o melhor preparados possível para situações de emergência.

O modo de funcionamento é simples: é criado um exercício (ou cenário de emergência), composto por situações (que podem ser consideradas ocorrências) que o agente deve ser capaz de resolver corretamente (missão – ou conjunto de tarefas de resposta). O exercício é então executado, havendo duas vistas diferentes: uma para o agente a ser avaliado e outra para o examinador. Durante o exercício, o examinador pode adicionar ou remover em tempo real novas situações, de forma a avaliar a capacidade do agente em se adaptar às mudanças súbitas de cenário no mundo real.

3.1.3. “Safer”

Existiu a necessidade de expandir o projeto XTraN 4 Forces de modo a abranger mais entidades, de modo a suportar a lógica de gestão relacionada com as forças de segurança. Com este objetivo nasceu o projeto SAFER (System for Advanced FiEld opeRation) que se enquadra na área de gestão de emergências, uma área especializada derivada da gestão de ocorrências. A gestão de emergências trata da logística que engloba uma situação que possa ocorrer, tais como, incêndios, sismos, acidentes de viação, etc. Este sistema destina-se a corporações como forças de segurança, bombeiros, proteção civil, etc.

3.1.4. “4Forces Smart Teams”

Hoje em dia, ter acesso à informação e de uma forma imediata pode ajudar a salvar vidas, como também pode prevenir que estas entrem em perigo. Com esta visão, pretende-se utilizar meios móveis que permitam aos profissionais obter diversos tipos de informação para que possa existir uma resposta com maior prontidão em situações de emergência e/ou catástrofe.

O 4Forces Smart Teams (FST) foi pensado para auxiliar quem está no terreno, permitindo ao utilizador visualizar informação relativa às suas tarefas diárias. Além destas funcionalidades, esta aplicação contém uma funcionalidade que comunica com um dispositivo (Vital Jacket) para permitir ter acesso a informação relativa à pulsação do utilizador, podendo assim ser monitorizado. O Vital Jacket é um dispositivo médico certificado que combina tecnologias e soluções de engenharia biomédica tradicionais.

O 4Forces Smart Teams (FST) como foi dito anteriormente, foi um projeto proposto que resulta numa aplicação *mobile*.

3.2. Atualizações ao 4Forces

Nos próximos subcapítulos, é feita a descrição de forma detalhada do desenvolvimento e atualização dos projetos já mencionados, focando as principais características dos mesmos. O último projeto, pela relevância que assume no âmbito do estágio é descrito num capítulo dedicado (capítulo 4).

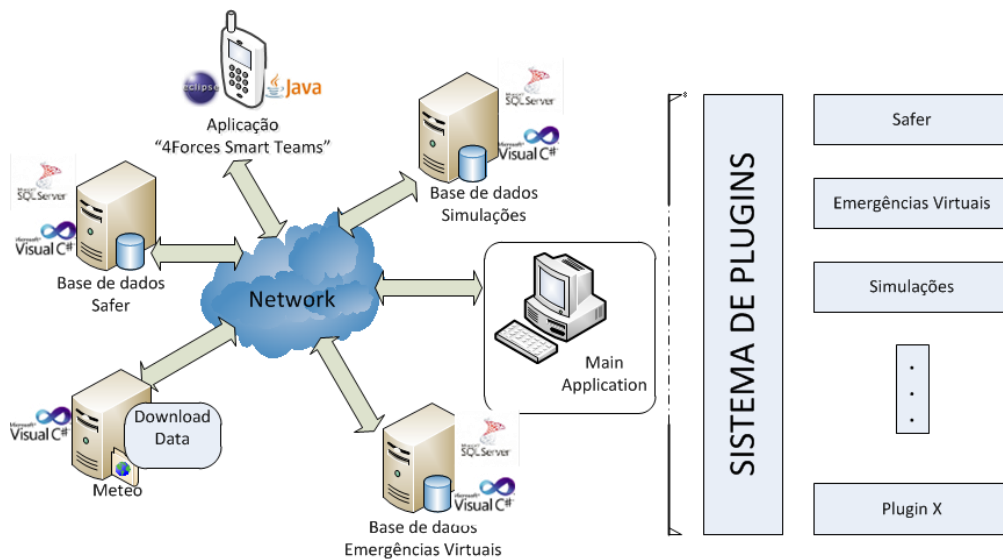


Figura 5 - Arquitetura de alto nível do XTran 4 Forces

Na Figura 5 podemos observar a arquitetura de alto nível que consiste no conjunto de vários tipos de servidores, tecnologias utilizadas, a aplicação nuclear do Xtran 4 Forces e os seus *plugins*.

Este sistema distribuído é facilmente escalável devido ao sistema de *plugins*. O sistema de *plugins* é uma das grandes vantagens do sistema XTraN 4 Forces. Esta abordagem permite incorporar várias aplicações numa só UI. Estas aplicações podem ser bastante distintas em modo de funcionamento e funcionalidades oferecidas, sendo que o único requisito para que uma aplicação possa ser considerada *plugin* é a conformidade com a estrutura de *plugins* utilizada. Através de um ficheiro de configuração podemos indicar à aplicação a ordem e que *plugins* pretendemos importar para a aplicação, obtendo com isto uma forma fácil e rápida de acrescentar ou retirar funcionalidades à aplicação.

Um aspecto relevante sobre as melhorias realizadas ao sistema foi que no início deste estágio a solução do XTran 4 Forces continha 137 projetos e no final, após as remodelações e reconstruções, ficou com um número significativamente menor, 68 projetos. Esta diminuição deveu-se grande parte à reestruturação de classes/projetos e remoção de dependências circulares e não utilizadas.

Tecnologias Utilizadas

Como é possível verificar na Figura 5 há várias tecnologias envolvidas no sistema.

No caso dos projetos “Safer”, “Emergências Virtuais” e “Download Data” foram desenvolvidos utilizando o recurso Integrated Development Environment (IDE) Visual Studio (“Microsoft Visual Studio,” n.d.), sistema proprietário da Microsoft (“Microsoft,” n.d.). Estes projetos utilizaram a linguagem de programação C#, existindo todavia alguns projetos desenvolvidos utilizando Visual Basic .NET. Uma vez que todo o código escrito em qualquer linguagem da Framework .NET é convertido para Common Language Runtime (CLR) – a linguagem intermediária executada pela Virtual Machine da plataforma .NET – é garantida a compatibilidade entre projetos apesar das diferentes linguagens de programação utilizadas.

Noutra vertente temos o projeto “4Forces Smart Teams” que foi desenvolvido com o Eclipse [7] utilizando a linguagem de programação JAVA.

3.2.1. Download Data

Este projeto é composto por um serviço que efetua regularmente o *download* dos ficheiros meteorológicos (hdf5) a partir de um servidor via FTP. Os ficheiros (hdf5) contêm informação relativo a diversos fatores de meteorologia, com por exemplo, humidade do ar, precipitação ou velocidade do vento.

O serviço é composto por diversos *jobs* que podem ser configurados através de um ficheiro de configuração. *Jobs* são tarefas que podem ser agendadas para serem executadas num futuro, ou seja, como se pode observar na Tabela 3 – coluna “Hora de Atualização”, a cada hora é executada uma tarefa referente ao minuto ao qual foi agendada. O serviço pode ser executado em modo de consola, bastando para isso adicionar o argumento “-console”.

Tabela 3 - Detalhes dos *downloads* a serem efetuados diariamente

Modelo	Número de Ficheiros/dia	Tamanho Ficheiro	Hora de Atualização
Costa de Portugal (Mar) [MOHID_Portugal_yyyymmdd_yyyymmdd.hdf5]	1	~32 Mb	xxH00m
Costa de Portugal (Atmosfera) [MM5_Portugal_yyyymmdd_yyyymmdd.hdf5]	1	~38 Mb	xxH05m
Portugal (malha 9km) [D3_yyyymmddhh.hdf5]	4	~210 Mb	xxH10m
Portugal (malha 3km) [WRF-PT-3km_yyyymmddhh.hdf5]	4	~1,1 Gb	xxH20m
WRF (Praia de Carcavelos) [yyyy-mm-dd_yyyy-mm-dd_WRF_w3.hdf5] *	1	~1 Mb	xxH40m
Water Properties (Praia de Carcavelos) [yyyy-mm-dd_yyyy-mm-dd_WaterProperties_w3.hdf5] *	1	~27 Mb	xxH45m
Hydrodynamic (Praia de Carcavelos) [yyyy-mm-dd_yyyy-mm-dd_Hydrodynamic_w3.hdf5] *	1	~31 Mb	xxH50m

* - yyyy-mm-dd_(yyyy-mm-dd)+1d

Na Tabela 3 podem ser encontradas os detalhes dos *downloads* efetuados para cada um dos resultados de modelo meteorológico. Nos detalhes é possível ver o nome do modelo (assim como o nome genérico do ficheiro), o número de resultados que são publicados por dia, o tamanho aproximado de cada ficheiro e os horários a que são realizadas as verificações de novos ficheiros e atualizações dos mesmos.

Todos os ficheiros que são obtidos do servidor, via *FTP*, ficam guardados em disco. Se estes ficheiros forem apagados do disco, não serão apresentados os resultados quando pedidos.

Para o funcionamento deste serviço, foi pensado utilizar as ferramentas de agendamento do Windows que permitem criar uma tarefa. Essa tarefa deverá ser executada diariamente onde terá como objetivo executar o serviço do *Download Data*.

Inicialmente o projeto era utilizado apenas para obter informação estática (relativamente a dois ou três dias) para que existisse informação para testes. Foi necessário adaptar o projeto para ser utilizado como um serviço que trabalhasse independentemente da aplicação. Além disto, foram acrescentados alguns novos tipos de ficheiros para serem utilizados nas simulações dos quais se destaca o relativo a dados de radar.

Devido à alteração de tipos de ficheiros e modelo de domínio nos projetos onde estes dados iriam ser utilizados, foi também necessário fazer alterações na parte onde a informação era guardada, ou seja, quando é realizado um *download* de um ficheiro, este não é apenas copiado para uma pasta física do servidor mas também é guardado numa base de dados. Essa informação guardada e tratada permite um aumento de *performance* quando a informação é requisitada.

Resultado

Com a conclusão deste projeto a parte de gestão de ficheiros e o seu armazenamento deixou de ser um problema.

3.2.2. Emergências Virtuais

Para muitos dos seres humanos a capacidade de aprendizagem é muito mais elevada e com uma maior eficácia quando se pratica uma. Desta ideia surge o projeto

“Emergências Virtuais” permitindo assim aos utilizadores uma formação mais orientada à prática.

Podemos visualizar na Figura 6 a arquitetura do projeto “Emergências Virtuais”, e como este está integrado no sistema de *plugins*.

O primeiro objetivo deste projeto consistiu no desenvolvimento de uma ferramenta que permite criar séries de ocorrências fictícias, a partir das quais se pode gerar emergências virtuais a serem utilizados em exercícios. Para além da ocorrência e dos dados que a tipificam existe um conjunto de variáveis de carácter físico (bloqueios, zonas inacessíveis), meteorológico e respetiva evolução temporal que serão alteradas de forma não controlável pelo utilizador potenciando assim a evolução da imprevisibilidade dos cenários e respetivo impacto da ocorrência.

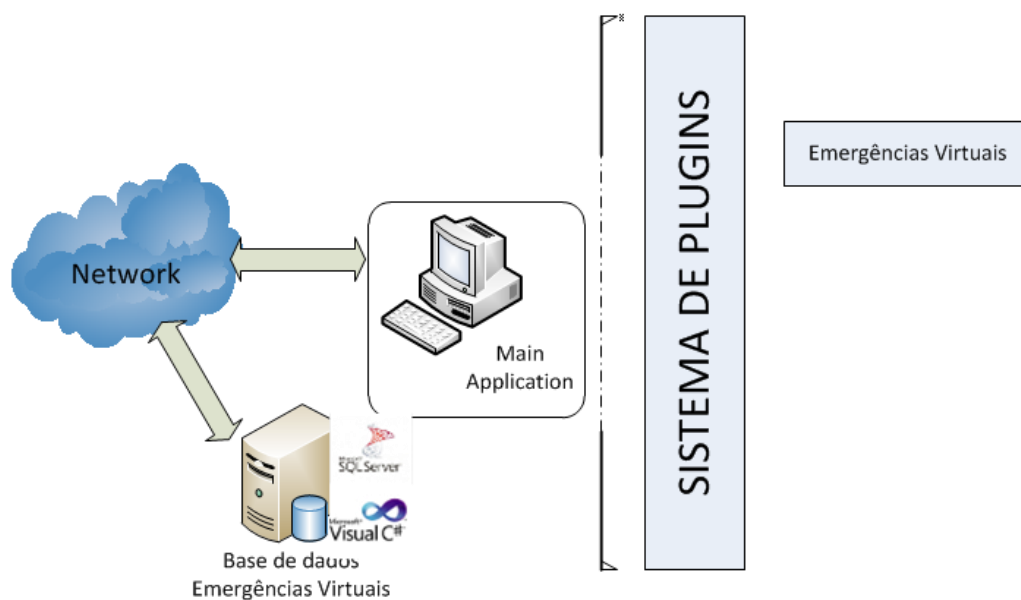


Figura 6 - Arquitetura Emergências Virtuais

Com este objetivo os criadores dos cenários têm ao alcance uma ferramenta que permite estruturar temporal e espacialmente todas as ocorrências que compõem o exercício. Isto permitirá atingir o segundo objetivo, que é a monitorização do exercício ao longo do tempo, permitindo analisar e comparar as reações dos operadores da plataforma.

O terceiro objetivo foi de ser de tal forma genérica que as emergências geradas pudessem ser utilizadas independentes do grau da complexidade do exercício, ou seja desde o simples exercício em gabinete até aos exercícios que envolvem a total mobilização dos meios ao terreno, neste sentido, e para além das variáveis já referidas, o

total de meios (e tipo) ao dispor é outra das entradas que o simulador de cenários deverá alterar de forma não constante.

Para alcançar os objetivos propostos foi desenvolvida uma ferramenta computacional que permitiu criar e estruturar emergências virtuais.

A sua utilização será semelhante à utilização dos simuladores de voo. Nos simuladores do voo é treinado o piloto enquanto através do módulo de emergências virtuais são treinados as forças de segurança. As falhas (e.g. falha de motor) e/ou as condições adversas (e.g. ventos fortes) simuladas a pedido do instrutor (ou geradas aleatoriamente) correspondem às ocorrências (e.g. ruína de edifícios, incêndios) criadas pelos criador dos cenários.

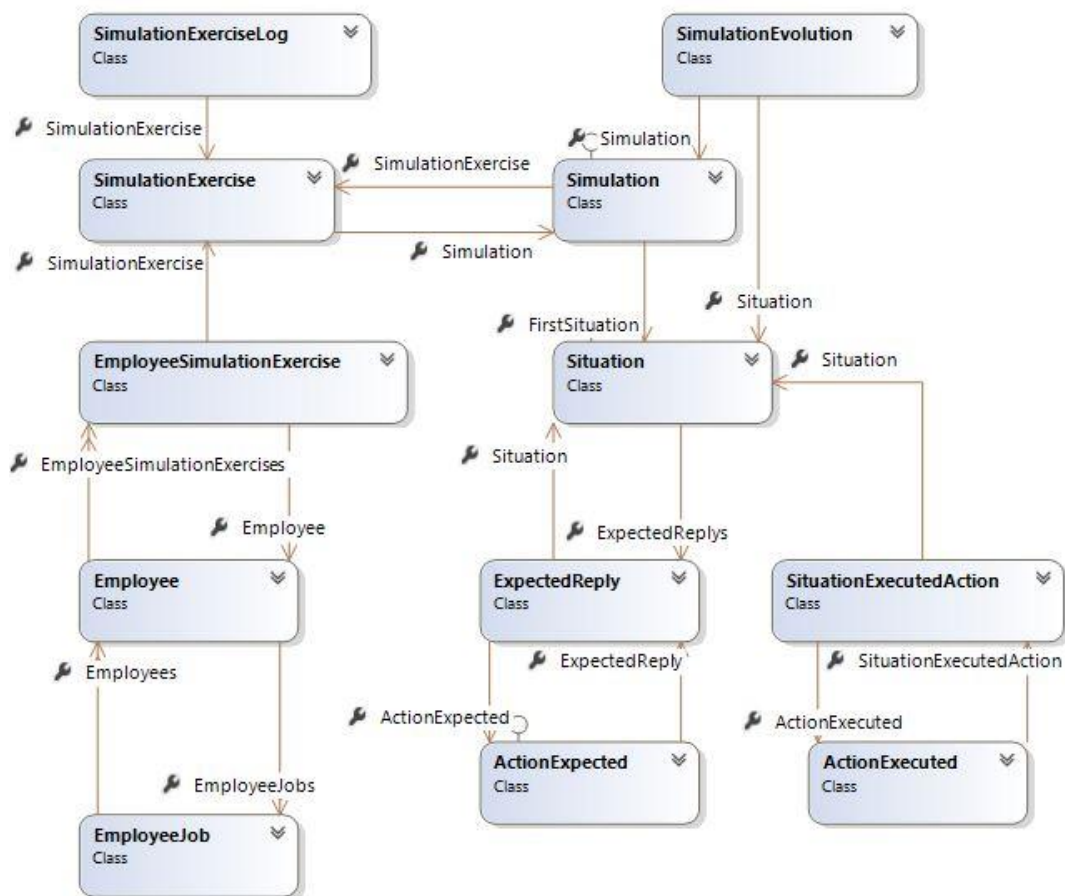


Figura 7 - Modelo de domínio Emergências Virtuais

Na Figura 7 podemos observar o modelo de domínio já existente das emergências virtuais.

Neste projeto é possível planejar, configurar e executar exercícios de simulação. As simulações são compostas por diversas situações que vão ocorrendo ao longo do

período do exercício de simulação. As situações ocorridas podem ser meramente informativas ou podem requerer ações esperadas por parte do formando.

O objetivo de trabalho era colocá-lo em funcionamento, integrado com o sistema de *plugins*.

Tabela 4 - Alterações no projeto Emergências Virtuais

Como encontrado	Alterações
Estrutura existente mas não compilava	Alterações de modo a permitir compilar e utilizar as funcionalidades
Projeto isolado	Integrar com o sistema de <i>plugins</i>

Apesar do foco do estágio não ter sido neste projeto, ainda foi despendido algum tempo para fazer as alterações necessárias com objetivo de cumprir o planeamento. Como se pode observar na Tabela 4, essas alterações numa primeira fase consistiu na atualização de bibliotecas, referências e *namespaces* que permitiu a compilação do projeto. Após compilado e verificado que o projeto funcionava corretamente iniciou-se a integração com o sistema de *plugins*. Com esta integração permitiu juntar o projeto das Emergências Virtuais à aplicação onde já constam os projetos Safer e Simulações.

Resultado

Para um maior detalhe sobre janelas e funcionamento do projeto visualizar o anexo B.

3.2.3. Safer

O que diferencia uma equipa das outras são as suas individualidades, as suas necessidades específicas e os seus atributos. Apesar de, tanto um bombeiro, como um agente terem como objetivo assegurar a segurança dos cidadãos, os meios para alcançarem esse objetivo é diferente. No XTraN 4 Forces também houve a necessidade de se analisar essas diferenças e construir algo que se adequasse a cada equipa, para isso surgiu o projeto SAFER - System for Advanced FiEld opeRation – Response que nasceu com o objetivo de estender o 4 Forces de forma a incluir um novo grupo de utilizadores: as Forças de Segurança (FS).

Podemos visualizar na Figura 8 a arquitetura do projeto “Safer”, onde este está integrado no sistema de *plugins*.

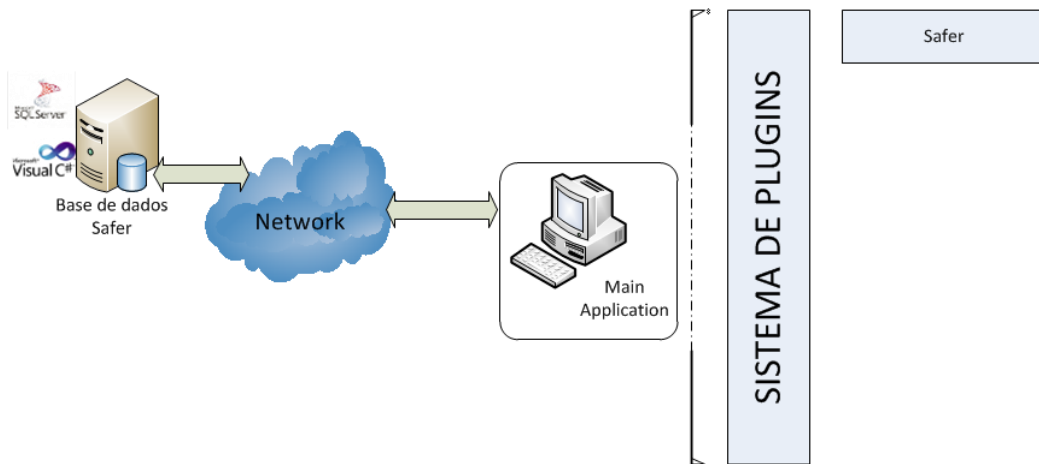


Figura 8 - Arquitetura Safer

Este projeto foi iniciado no ano 2013 onde se pretendia a “construção de um sistema com capacidade de servir vários clientes um conjunto de funcionalidades comuns sem prejuízo das necessidades específicas de cada entidade [3].” De uma forma simplista, o SAFER tem como base as funcionalidades do sistema 4 Forces onde se pode ver ilustrado na Figura 9.



Figura 9 - Funcionalidades resumidas do sistema 4 Forces [3].

O SAFER vem, assim, disponibilizar uma vertente capaz de gerir as necessidades das Forças de Segurança, dando apoio em tarefas importantes como missões e tarefas quotidianas que, quando se tratem de emergências – cheias, incêndios, etc., podem ser geridas com auxílio ao arsenal de técnicas de modelação que o 4 Forces já possui (como previsões, simulações, visualização em tempo real, etc.).

Este projeto foi pensado para poder ser alcançado por outros sistemas futuros ou aplicações, para isso, foi construído por dois tipos de componentes, a componente *desktop* e a componente *web*. A componente *desktop* consistiu na integração de um *plugin* na aplicação 4 Forces cuja sua função foi a capacidade de gestão, ou seja, ter a capacidade de gerir toda a informação das Forças de Segurança, desde informação geral respetiva à esquadra, até ocorrências, missões, agentes, equipas, equipamentos e veículos. Noutra vertente, temos a componente *web* que foi desenvolvida para disponibilizar *webservices* para serem acedidos por aplicações cliente. Este sistema foi desenvolvido para ser orientado a serviços, de modo a ser escalável para o futuro.

Atualização da User Interface

Na Figura 10 pode-se observar a versão antiga do Safer. Esta janela está dividida em três secções distintas: área de lista de entidades, área de visualização/edição de informação e área de menus.

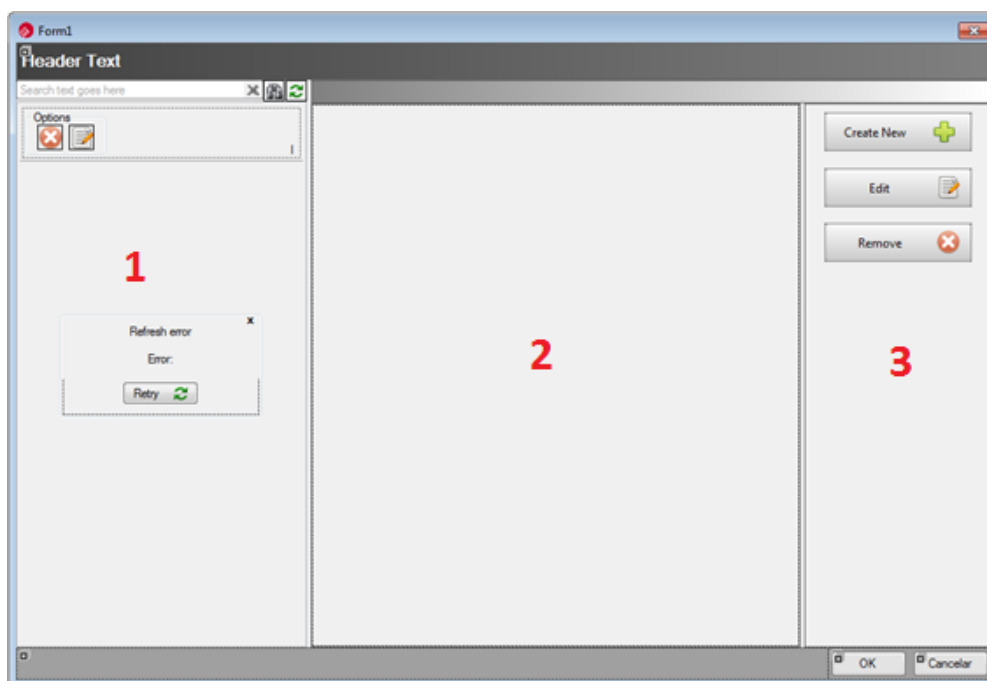


Figura 10 - Ecrã principal da versão anterior

Do lado esquerdo, representado com o número 1 existe uma lista, destinada a apresentar as entidades existente, sendo possível filtrar os resultados através de uma barra de pesquisa. Ao lado dessa mesma barra temos acesso a um botão que permite atualizar os dados da lista. Por vezes a obtenção dos dados pode demorar alguns segundos dependendo da sua fonte (pode variar, exemplo: ficheiro ou base de dados). Deste modo, quando existe demora para obter informação, aparecerá uma janela para dar *feedback* ao utilizador com a opção de “retry” para forçar essa ligação. Cada item pode ser visualizado em detalhe, editado ou removido, sendo definidos botões no *template* para o efeito (é possível configurar quais destes botões são efetivamente mostrados). Do lado direito são mostrados botões para as principais funcionalidades, como adicionar, editar e remover um objeto que se encontra a ser visualizado. Por último, a área do meio é ocupada por três painéis sobrepostos que se destinam a mostrar a informação ao utilizador de modos diferentes consoante a operação que o utilizador está a realizar sobre as entidades: visualizar, editar ou criar um objeto.

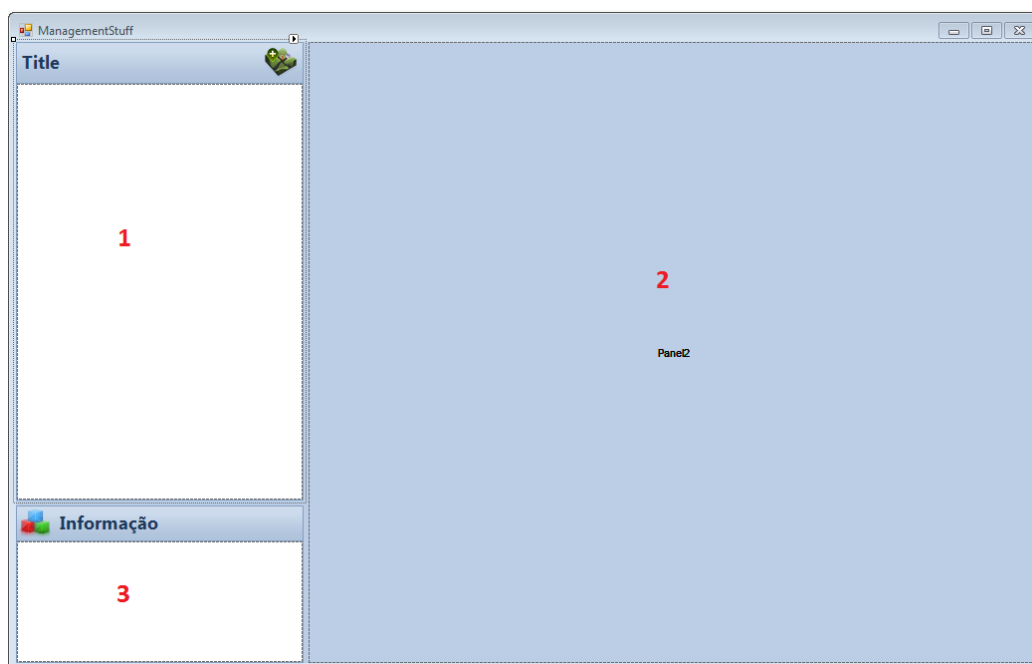


Figura 11 - Ecrã principal versão atual

Com a evolução do projeto foi necessário ir ao encontro de outras necessidades nas quais se realça uma *User Interface* mais “user friendly” e ir ao encontro de outros *plugins* já existentes. Para atingir estes fins o *design* foi modificado como demonstra a Figura 11 onde vários componentes foram adicionados. Entre eles podemos destacar o *TabNavigator*, *TreeView*, *UC (UserControl)*, *UCInfo (UserControlInformation)*. Esta

janela (Figura 11) foi dividida em três partes distintas: TabNavigator, UCInfo e UC. Representado com o número 1 temos a *TabNavigator*. Este separador é um componente *krypton* [8] que permite agrupar diversos separadores e navegar entre elas possibilitando gerir grandes quantidades de informação de forma organizada. Dentro deste componente temos um dos componentes mais importantes deste projeto que possibilita a interação com as nossas entidades, esse componente pode existir entre 1 a N e chama-se *TreeView*. Através dele iremos ter acesso a uma grande variedade de informação como visualizar, adicionar, remover, editar, agrupar, entre outros. Na imagem ilustrada em baixo (Figura 12) pode-se observar um exemplo da informação mostrada no ponto 1.

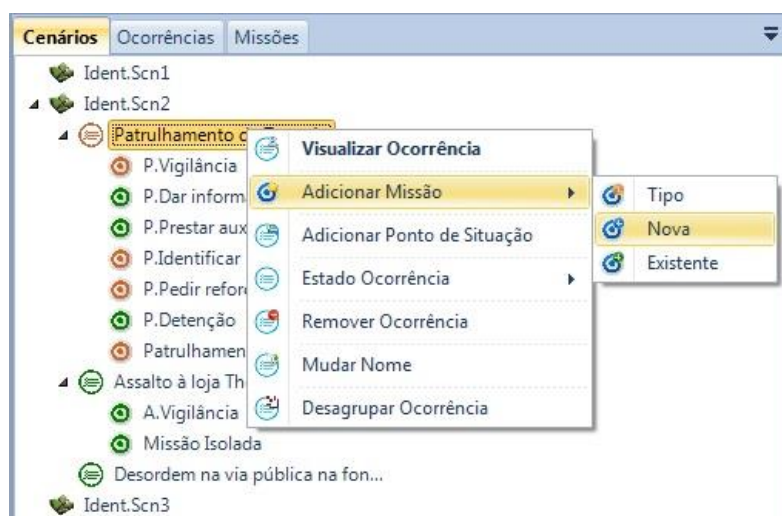


Figura 12 – Treeview e Context Menu

O componente com o número 2 é um *User Control* onde se irá manipular as entidades, isto é, adicionar e editar os objetos. Esta UC é acedida através da escolha de uma opção no *context menu* de um nó da *treeview* ou através da seleção do botão na *tabNavigator*. Na Figura 13 pode-se observar um exemplo de um *user control* com a de uma missão.

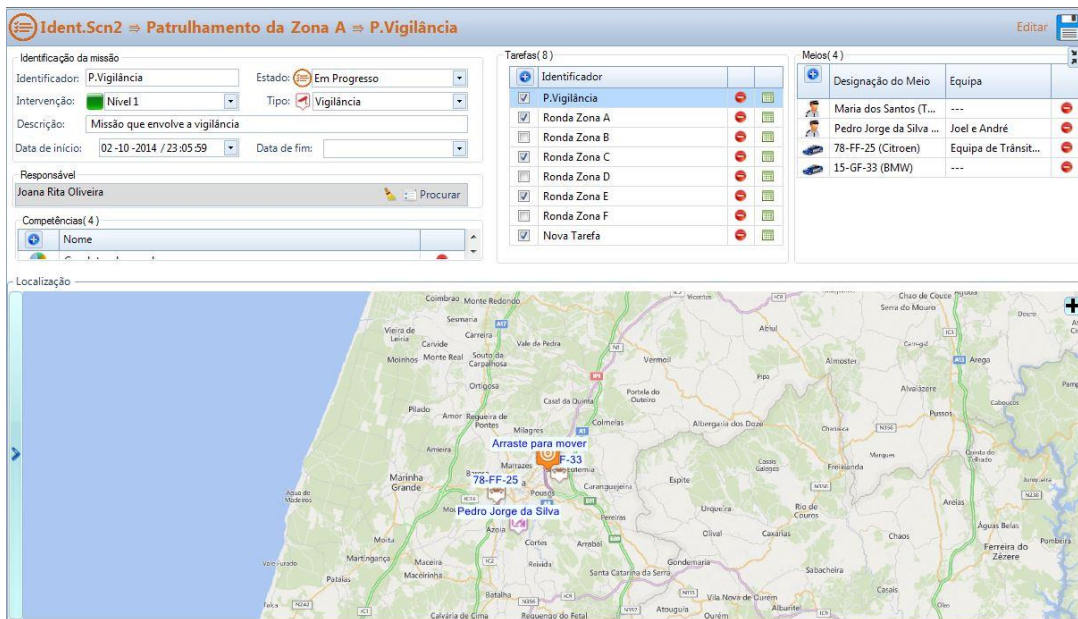


Figura 13 - Exemplo de um user control de uma missão

Finalmente, representado com o número 3 temos um User Control que contém informação relativamente aos nós e à própria *treeview*. Também tem o propósito de mostrar futuramente informação importante, notícias ou *logs* do sistema. Como exemplo, temos a Figura 14 que demonstra um gráfico com o número de ocorrências (4), missões (9) e tarefas no sistema, que neste caso ainda não existem.



Figura 14 - Gráfico com o número de ocorrências, missões e tarefas no sistema

Atualização do Modelo de Domínio

O diagrama geral do modelo antigo está representado na Figura 15 enquanto o diagrama atual pode ser visualizado na Figura 16. As entidades representadas estendem, em última análise, da classe EntityBase. Esta classe disponibiliza uma base para qualquer entidade do sistema. Salientam-se os campos Id, data de última modificação e uma *flag* que indica se o objeto se encontra removido ou não. Assim, com o auxílio da classe EntityBase, garante-se que todas as entidades seguem estas normas, sendo possível a implementação de comportamentos genéricos nos clientes. A título de exemplo, quando um cliente obtém dados pode comparar os dados anteriores, pelas propriedades Id e LastModificationDate, de modo a decidir se atualiza a interface gráfica ou não.

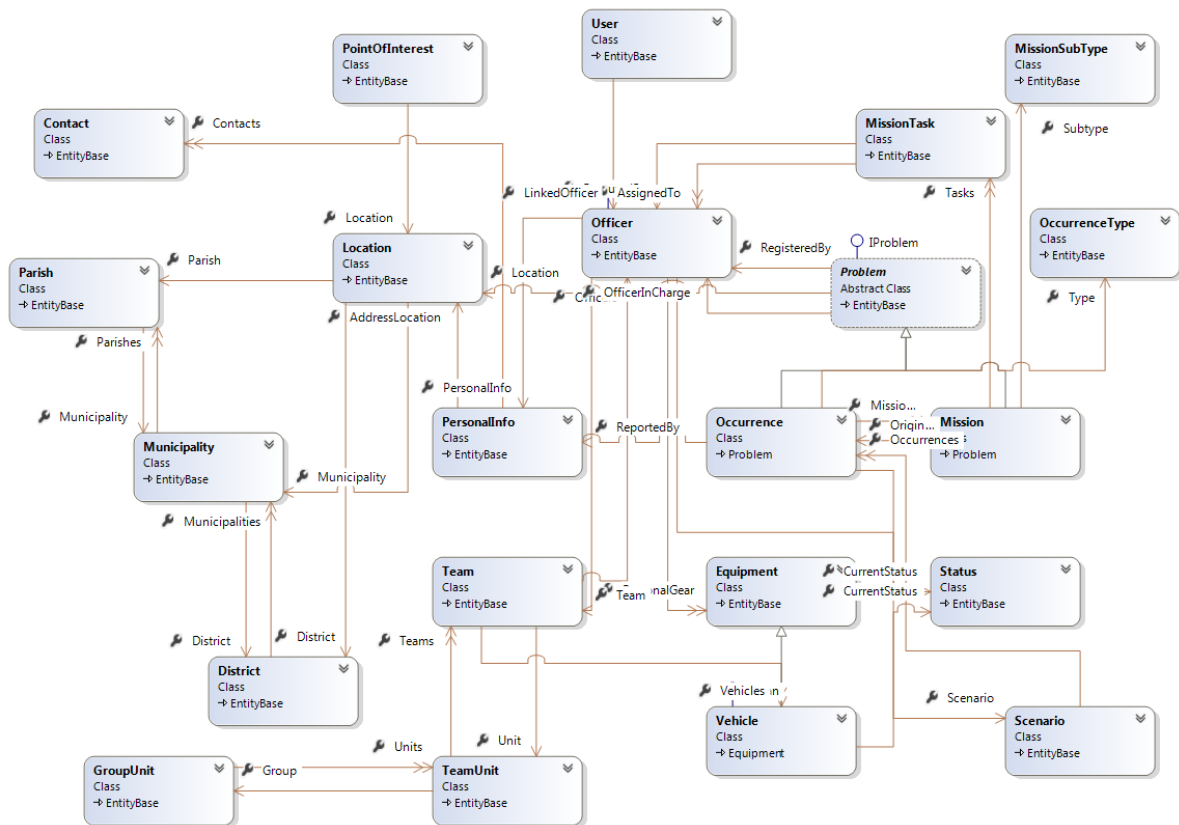


Figura 15 - Diagrama do modelo de domínio antigo [3].

Na Figura 15 é ilustrado um diagrama do modelo de domínio encontrado com o projeto Safer.



Figura 16 - Diagrama do modelo de domínio novo

Como é natural, o novo modelo de domínio, como se pode observar na Figura 16, sofreu várias alterações passando por criação de novas entidades, alterações de atributos e relações nas entidades já existentes.

Há a referir que por uma questão de simplicidade não se apresentam todas as associações entre entidades, e foram seleccionadas algumas que serão descritas a seguir.

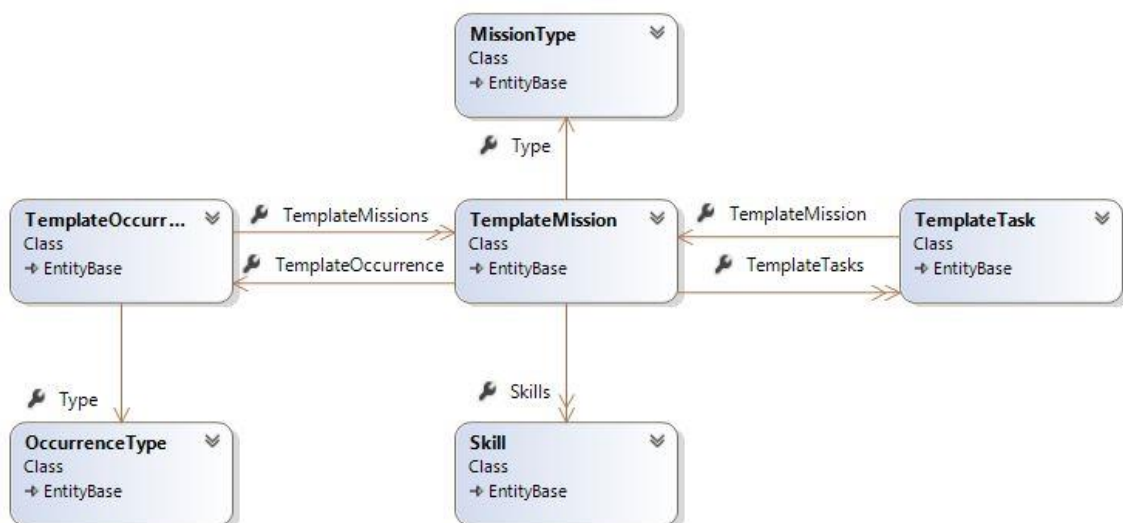


Figura 17 - Entidades Templates

Surgiu a necessidade construir *templates* para que a inserção de entidades fosse mais rápida. Deste modo surgiram as entidades `templateOccurrence`, `templateMission` e `templateTask`.

- `TemplateOccurrence` – Representa a entidade occurrence.
- `TemplateMission` – Representa a entidade mission.
- `TemplateTask` – Representa a entidade missionTask.

Na Figura 17 é ilustrada a associação entre estas entidades.

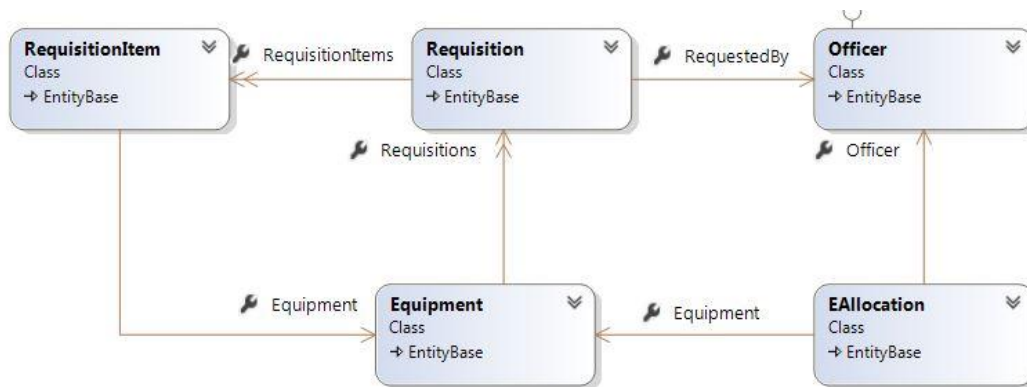


Figura 18 - Entidade Requisition

No sistema existe a entidade `equipment` que representa um equipamento (v.g. “Colete L”, “M4”). Os agentes têm a necessidade de requisitar equipamentos e para esse fim foi criado a entidade `requisition` e `requisitionItem`, ou seja, uma requisição contém vários itens que por sua vez está associado a um equipamento e a uma quantidade (Figura 18). Deste modo é possível distinguir o que foi requisitado e o que foi aprovado.



Figura 19 - Entidade StatusHistory / VitalJacket

Devido ao serviço utilizado na aplicação mobile (`vitaljacket`) existiu a necessidade de guardar essa informação. Como é importante reduzir o número de pedidos ao servidor, optou-se por cada envio de informação, em vez de só enviar o estado do `vitaljacket`, enviar também outros dados, como estado atual no agente ou coordenadas gps. Para este

fim surgiu a entidade statusHistory que além de conter a informação do vitaljacket ainda permite guardar outros dados relevantes. Assim, nasceu a associação descrita na Figura 19.

Poderá ter acesso a mais informação sobre as outras entidades e ao modelo de domínio atual no Anexo C.

Atualização dos controladores

Por omissão, um pedido ao endereço [endereço do servidor]/api/[nome da entidade no plural] realizado via GET é mapeado para um controlador com o nome da entidade no plural, ou seja, se o endereço fosse 192.168.1.2/api/missions/ o pedido era mapeado para um controlador com o nome “MissionsController”. Tal acontece devido à convenção utilizada para nomear os controladores. Os controladores (*controllers*) são as classes que lidam com os pedidos *http* ao servidor, traduzindo os pedidos em ações. Caso não exista uma classe que herde de *ApiController* com esse nome, o servidor rejeita o pedido com o código 404 – recurso não encontrado.

```
/// <summary> ...  
[HttpPut]  
public HttpResponseMessage Rename(Mission obj)...  
  
/// <summary> ...  
[HttpPut]  
public HttpResponseMessage StateInProgress(Mission obj)...  
  
/// <summary> ...  
[HttpPut]  
public HttpResponseMessage StateFinalize(Mission obj)...  
  
/// <summary> ...  
[HttpPut]  
public HttpResponseMessage ProgressStatus(Mission obj)...  
  
/// <summary> ...  
[HttpPut]  
public HttpResponseMessage InsertExisting(Mission obj)...  
  
[HttpGet]  
public HttpResponseMessage Officer()...
```

Figura 20 - Métodos da class MissionsController

Com este mecanismo, facilmente acedemos aos métodos com a *tag* [HttpPut], [HttpGet], [HttpDelete] e [HttpPost] criados nesse controlador, como se pode observar na Figura 20.

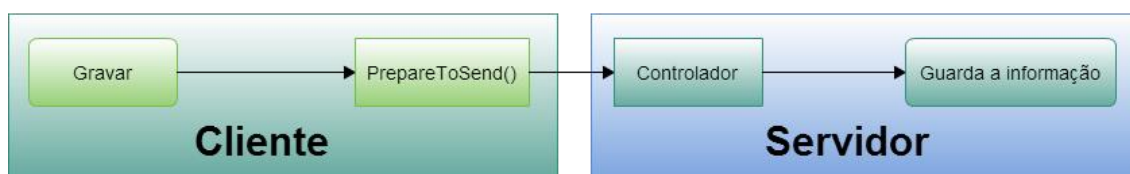


Figura 21 - Processo simples de update de uma entidade

Como ilustrado na Figura 21 o processo para gravar uma entidade começa no lado do cliente por seleccionar um botão de “gravar” onde é despoletado um método chamado “PrepareToSend” que prepara a entidade para ser enviada para o servidor. No servidor é chamado o controlador da própria entidade que processa e valida a informação a ser guardada no servidor.

Atualização do acesso aos dados - DTO

Quando se trabalha com interfaces remotas é importante assegurar que apenas se fazem pedidos de informação quando é estritamente necessário, pois cada pedido vai ter um peso para o sistema. Como resultado iríamos ter que transferir muito mais informação em um só pedido que tornaria o sistema mais lento e moroso.

Para minorar o problema criamos um módulo Data Transfer Object (DTO) que consiste em um objeto que transporta dados entre os processos, a fim de reduzir o número de pedidos. Uma das principais razões para a utilização do DTO é conseguir-se juntar vários pedidos remotos em um único, no entanto vale a pena mencionar que outra das vantagens é sua capacidade de encapsular o mecanismo de serialização para transferência de dados. Ao encapsular a serialização, os DTOs mantêm a lógica separada do resto do código, além de permitir alterar a serialização quando se pretender.

Por outro lado, o DTO apresenta como desvantagem o esforço necessário para a sua implementação, isto é, é imperativo que para cada entidade seja definido que informação se pretende enviar. Neste contexto, por vezes, são criados vários conjuntos da mesma informação consoante o pretendido (pouca, média ou toda) informação sobre a entidade.

De seguida, será apresentado um caso em concreto para que se possa entender melhor este procedimento.

O projeto Safer pode ser considerado um projeto já com alguma complexidade e dimensão onde foi detetado um problema na quantidade de informação que era transferida por pedido.

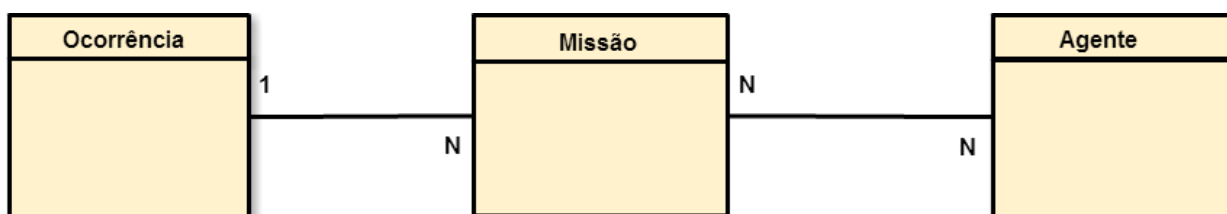


Figura 22 - Exemplo de relações entre entidades.

Na Figura 22 pode-se observar um exemplo de relações entre entidades onde a entidade ocorrência contém uma lista de missões que por sua vez, cada missão conhece a ocorrência a que pertence. O mesmo se aplica na relação de uma missão e os seus agentes, diferenciando no espectro que cada agente pode estar em várias missões em simultâneo, enquanto uma missão apenas pode ter uma ocorrência.

Estas relações bidirecionais originam o problema que foi detetado anteriormente, pois se cada entidade conhece os seus pais e filhos, podemos-nos deparar com situações de ciclos infinitos e redundância. Para colmatar este problema utilizou-se a opção DTO onde permitiu definir que quantidade de informação era desejado transferir.

Para esta solução foi criada uma classe com métodos específicos para cada entidade que permitiu definir que tipo e qual informação a ser enviada.

```

public static EntityBase GetReducedDTO(this EntityBase entity)[...]
private static User GetReducedUserDTO(User clone)[...]
private static Scenario GetReducedScenarioDTO(Scenario clone)[...]
private static Occurrence GetReducedOccurrenceDTO(Occurrence clone)[...]
private static Mission GetReducedMissionDTO(Mission clone)[...]
private static MissionTask GetReducedTaskDTO(MissionTask clone)[...]
private static Group GetReducedGroupDTO(Group clone)[...]
private static Unit GetReducedUnitDTO(Unit clone)[...]
private static Team GetReducedTeamDTO(Team clone)[...]
private static Officer GetReducedOfficerDTO(Officer clone)[...]
private static Vehicle GetReducedVehicleDTO(Vehicle clone)[...]
private static Equipment GetReducedEquipmentDTO(Equipment clone)[...]
private static TemplateOccurrence GetReducedTemplateOccurrenceTypeDTO(TemplateOccurrence clone)[...]
private static TemplateMission GetReducedTemplateMissionDTO(TemplateMission clone)[...]
private static TemplateTask GetReducedTemplateTaskDTO(TemplateTask clone)[...]
private static PersonalInfo GetReducedPersonalInfoDTO(PersonalInfo clone)[...]
private static PointOfInterest GetReducedPointOfInterestDTO(PointOfInterest clone)[...]
private static ProgressStatus GetReducedProgressStatuskDTO(ProgressStatus clone)[...]
private static MissionType GetReducedMissionTypeDTO(MissionType clone)[...]
private static OccurrenceType GetReducedOccurrenceTypeDTO(OccurrenceType clone)[...]

public static EntityBase GetFullDTO(this EntityBase entity)[...]
private static User GetFullUserDTO(User clone)[...]
private static Scenario GetFullScenarioDTO(Scenario clone)[...]
private static Occurrence GetFullOccurrenceDTO(Occurrence clone)[...]
private static Mission GetFullMissionDTO(Mission clone)[...]
private static MissionTask GetFullTaskDTO(MissionTask clone)[...]
private static Group GetFullGroupDTO(Group clone)[...]
private static Unit GetFullUnitDTO(Unit clone)[...]
private static Team GetFullTeamDTO(Team clone)[...]
private static Officer GetFullOfficerDTO(Officer clone)[...]
private static Vehicle GetFullVehicleDTO(Vehicle clone)[...]
private static Equipment GetFullEquipmentDTO(Equipment clone)[...]
private static TemplateOccurrence GetFullTemplateOccurrenceDTO(TemplateOccurrence clone)[...]
private static TemplateMission GetFullTemplateMissionDTO(TemplateMission clone)[...]
private static TemplateTask GetFullTemplateTaskDTO(TemplateTask clone)[...]
private static PersonalInfo GetFullPersonalInfoDTO(PersonalInfo clone)[...]
private static PointOfInterest GetFullPointOfInterestDTO(PointOfInterest clone)[...]
private static ProgressStatus GetFullProgressStatuskDTO(ProgressStatus clone)[...]
private static MissionType GetFullMissionTypeDTO(MissionType clone)[...]
private static OccurrenceType GetFullOccurrenceTypeDTO(OccurrenceType clone)[...]

```

Figura 23 - Class BaseEntityExtension

Na Figura 23 podemos observar dois tipos de métodos diferentes para a mesma entidade que tem como principal diferença a quantidade de informação que é retornada.

Por exemplo, o método *GetReducedOccurrenceDTO* contém a informação essencial sobre a entidade *occurrence*, como o id, identificador e as suas missões. Normalmente este tipo de método é chamado para preencher tabelas, *treeview*, ou componentes onde os requisitos de informação sejam mínimos.

Em outra vertente, temos o método *GetFullOccurrenceDTO* que contém toda ou quase toda a informação relativa à entidade. Este método, geralmente é usado quando se pretende mostrar detalhes ou na resposta de um *insert* ou *update*. É de salientar que mesmo a versão que supostamente devolve toda a informação de uma entidade pode sofrer melhorias de desempenho, isto é, por exemplo, querendo toda a informação relativa a uma ocorrência, esta tem uma lista de missões que por sua vez conhece a ocorrência. Não faz sentido que todas as missões contenham a informação da ocorrência

pois esta é a entidade que está a ser tratada, então é necessário fazer que as missões não apontem para a sua ocorrência prevenindo assim *loops* e redundância.

Neste momento, se o projeto Safer não contasse com o mecanismo DTO não era possível executar pedidos pois os erros de memória devido aos *loops* gerados eram uma constante.

Otimizações

Como em todos os projetos, é necessário a implementações de técnicas de otimização para que seja possível extrair o melhor rendimento possível do projeto, como por exemplo, comunicação cliente/servidor, inserção de dados na base dados, entre outros. O “Safer” não é exceção, como tal, de seguida serão descritos algumas dessas otimizações.

HandleDependencies: Quando ocorre um *insert*, *update* ou *delete* num objeto é preciso garantir que a informação que o utilizador está a visualizar está correta e atualizada. Para este fim, sentiu-se a necessidade de criar alguns métodos com a capacidade de informar as entidades das suas alterações. Poderemos dar como exemplo um cenário que contém ocorrências e cada ocorrência contém várias missões. Quando o utilizador elimina uma ocorrência, esta precisa de verificar todas as entidades relacionadas com ela para que sejam tomadas as devidas operações. Neste caso, todas as missões relacionadas com esta ocorrência passaram a ser isoladas, os ícones do mapa referentes à ocorrência serão eliminados e o cenário ficará com menos uma ocorrência.

PrepareToSend : É um método que está presente numa classe (entidade) *Extension*. Este *extensionMethod* é utilizado quando existe uma comunicação entre cliente e servidor, ou seja, quando é inserido, atualizado ou removido alguma entidade e imediatamente antes da informação ser enviada para o servidor, irá passar por este método fazendo que dados redundantes não sejam passados pela rede.

O problema original deveu-se à quantidade de informação que era transportada pela rede e que não iria ter utilidade no lado do servidor. Podemos dar como exemplo uma missão com uma lista de agentes. Se estamos a tratar de uma missão, não necessitamos da informação relativa aos agentes, apenas queremos que a lista tenha o ID dos agentes. Com esta ideia implementou-se este método que trata destes problemas, como se pode visualizar na Figura 24.

```

/// <summary> ...
public static Mission PrepareToSend(this Mission mission)
{
    var ret = (Mission)mission.Clone();

    //Check StatusMessage
    var listMessages = new List<ProgressStatus>();
    listMessages.AddRange(ret.StatusMessages);

    ret.StatusMessages = new List<ProgressStatus>();
    foreach (var item in listMessages)
    {
        ProgressStatus item1 = item;
        var existingItem = WebserviceUtil.GetList<ProgressStatus>().FirstOrDefault(o => o.Id.Equals(item1.Id)
            && item.IsRemoved == false);

        var obj = new ProgressStatus();
        obj = existingItem != null ? new ProgressStatus { Id = item1.Id } : item;
        ret.StatusMessages.Add(obj);
    }

    //Check Officers
    var listOfficers = new List<Officer>();
    listOfficers.AddRange(ret.AssignedTo);

    ret.AssignedTo = new List<Officer>();
    foreach (var item in listOfficers)
    {
        Officer item1 = item;
        var existingItem = WebserviceUtil.GetList<Officer>().FirstOrDefault(o => o.Id.Equals(item1.Id)
            && item.IsRemoved == false);

        if (existingItem != null)
        {
            var obj = new Officer { Id = item1.Id };
            ret.AssignedTo.Add(obj);
        }
    }
}

```

Figura 24 - Parte do método PrepareToSend da entidade missão

Inicialmente é feito um clone da informação a ser transportada pela rede, de seguida iremos verificar que informação é necessária, ou seja, se for uma lista de objetos, criaremos uma nova apenas com a informação fundamental e essencial para ser tratada no lado do servidor. O mesmo mecanismo é utilizado para os objetos.

Resultado

Após a conclusão dos primeiros testes e tendo um versão estável, deu-se por concluído a atualização à aplicação Safer. Os resultados deste projeto foram bastante positivos, ficando o protótipo pronto para ser apresentado. Como se pode ver pelos dois *screenshots* apresentados nas Figura 25 e Figura 26.

Na Figura 25 é ilustrado um ecrã com a informação relativa a um agente.

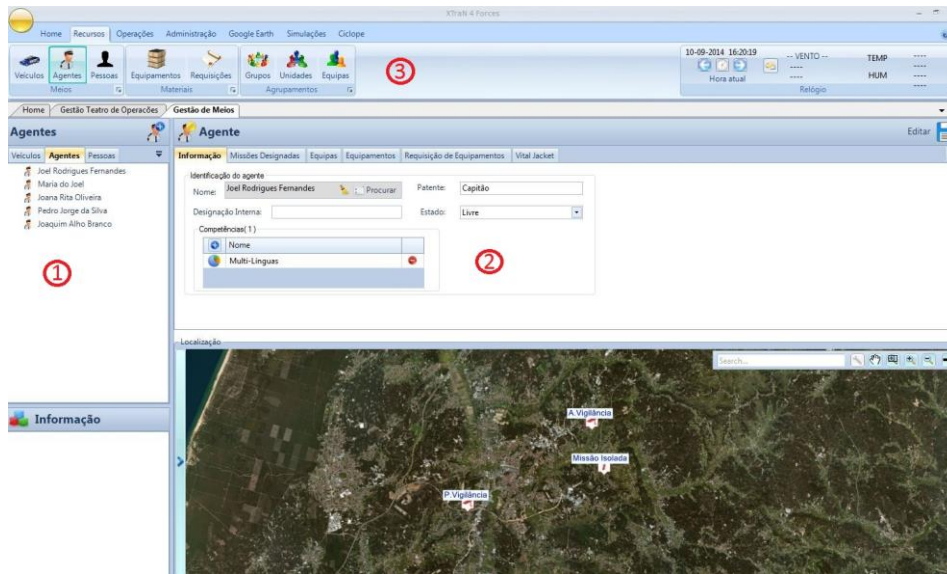


Figura 25 - Ecrã Gestão de Meios - Agente

Esta figura pode ser dividida em três partes, em que cada uma corresponde a um número. O número 1 corresponde a uma lista dos agentes existentes na base de dados. De seguida, com o número 2 temos acesso aos detalhes do agente, esta informação é obtida através da seleção de um dos agentes da lista apresentada junto ao número 1. Por último, temos o número 3 que contém os submenus do separador aberto (neste caso, o separador corresponde a “Recursos”).

Na Figura 26 podemos observar o ecrã relativo às missões predefinidas do separador de gestão de *templates*,

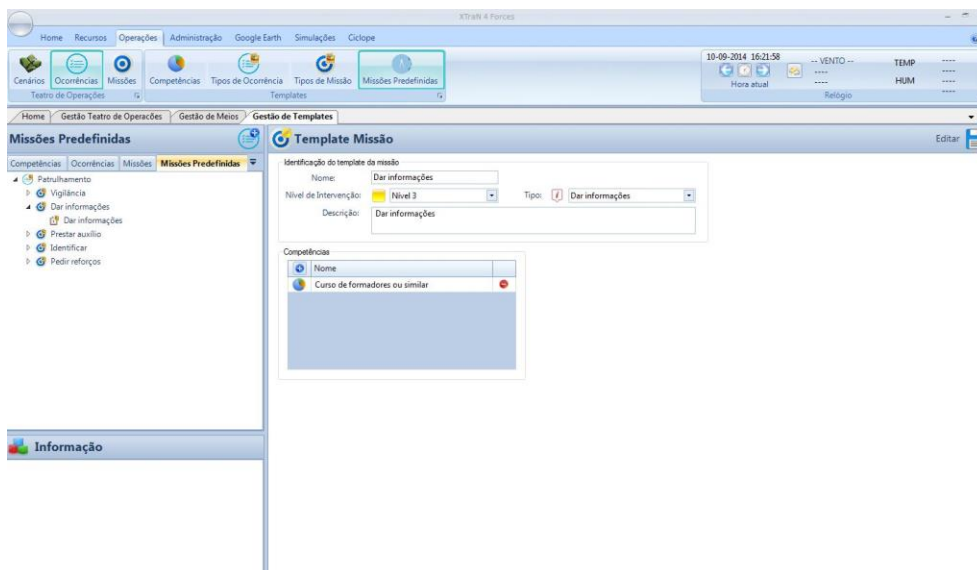


Figura 26 - Ecrã Gestão de Tempates - Missões Predefinidas

4. “4Forces Smart Teams”

Quando o projeto “Safer” estabilizou, iniciou-se o projeto “4Forces Smart Teams” começando por se tentar implementar algumas das funcionalidades já existentes no projeto “Safer”, nomeadamente alocação de tarefas. Foi essencial ter o projeto “Safer” a funcionar pois assim poderíamos monitorizar de uma forma mais realista e prever *features* para implementar no futuro.

A Figura 27 ilustrada abaixo representa a arquitetura “4Forces Smart Teams”.

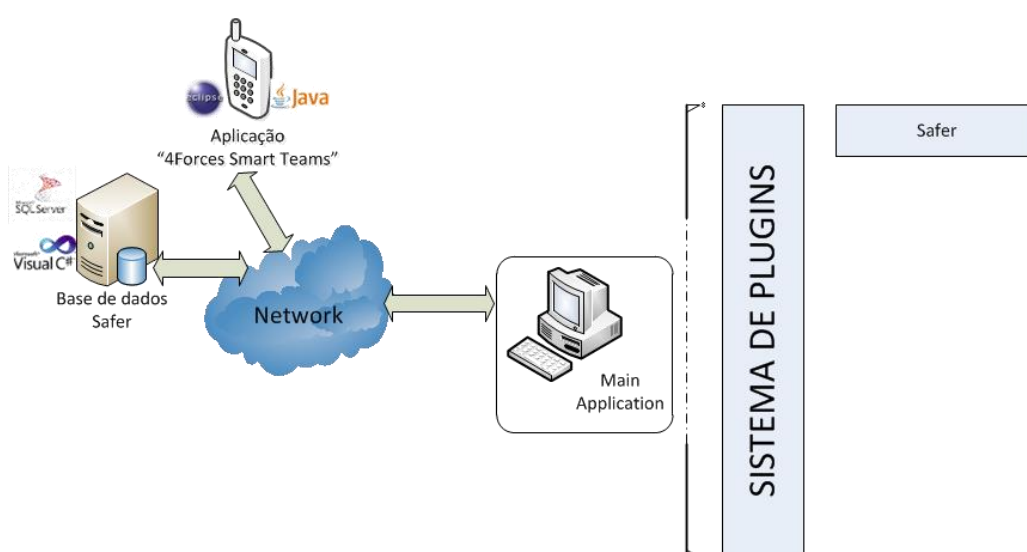


Figura 27 - Arquitetura 4Forces Smart Teams

O projeto “4Forces Smart Teams” surgiu da necessidade que os agentes têm de ter acesso a determinadas informações em qualquer momento e em qualquer lugar. Para isso, este projeto oferece uma aplicação móvel, que lhes permite aceder a essa informação para o trabalho do seu dia-a-dia. O tipo de informação disponibilizada baseia-se em missões e tarefas designadas ao utilizador, como também informação pessoal e possibilidade de ter acesso ao ritmo cardíaco do agente.

Podemos analisar como exemplo prático um acidente de automóvel, onde daríamos ao nome da missão “Acidente automóvel, 22 Maio”. Através da aplicação o agente verifica que tem uma missão onde existe uma tarefa que lhe foi atribuída e que corresponde em falar com os intervenientes do acidente.

Essa tarefa tem a informação sobre como proceder, local e alguns detalhes sobre o acontecimento, além disto o agente após completar a tarefa atribuída poderá alterar-lhe o estado indicando que esta foi finalizada. Neste momento, os outros agentes com a mesma missão, como o centro de controlo têm a informação que aquela tarefa foi realizada e assim podem tomar decisões.

Quando a necessidade de uma aplicação mobile se tornou evidente foi encontrado um problema sobre que plataforma de desenvolvimento utilizar, uma vez que estão disponíveis várias. Desta forma, existiu a necessidade de alguma análise e pesquisa descrita nos subtópicos seguintes.

4.1. Descrição do Projeto

O desenvolvimento desta aplicação consiste em dar suporte aos utilizadores no terreno para as suas tarefas diárias, como também ajudar a tomar decisões. Normalmente, as tarefas são atribuídas consoante uma missão, ou seja, quando acontece uma ocorrência, como por exemplo um acidente de carro, ou um assalto, é despoletado um mecanismo que permite mover agentes para um determinado local e atribuir-lhes funções para resolver a situação.

A Figura 28 descreve o funcionamento do projeto, integrando com os vários elementos.

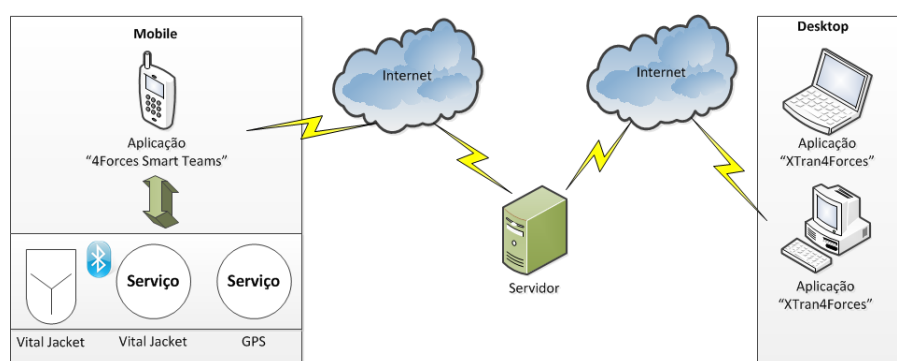


Figura 28 - Funcionamento do projeto "Smart Teams"

Como se pode observar na figura anterior, a aplicação móvel vai comunicar com o servidor, a fim de obter informação relevante para o utilizador, nomeadamente, informação relativa às tarefas executadas ou por executar, lista de equipamentos do utilizador ou ter acesso ao serviço Vital Jacket que permite saber informações relativamente ao estado cardíaco do utilizador.

A informação obtida pela aplicação móvel é gerida e alimentada pela aplicação desktop (Safer), onde aí os administradores ou quem tiver acesso para tal pode introduzir cenários, ocorrências, missões, tarefas, equipamentos, entre outros.

Além da aplicação móvel e aplicação desktop, existiu a necessidade de desenvolver alguns serviços que trabalham de forma independente e servem maioritariamente para a recolha de informação. No caso do serviço Vital Jacket e *gps*, estes serviços têm como objetivo enviar informação relativa à localização de um agente, pulsação e informação relevante para análise. Esta informação é enviada com uma periodicidade de por exemplo, trinta em trinta minutos. Estes serviços dispõem de mecanismos capazes de comunicar com outras aplicações, neste caso, com aplicação móvel que lhe permite visualizar em tempo real a pulsação do utilizador.

4.2. Análise Comparativa de Plataformas Móveis

Cada vez mais vivemos num mundo movido por tendências e projeções, onde cada decisão afeta não só o presente mas também o futuro. Para uma melhor análise das tendências dos sistemas operativos podemos visualizar no anexo C informação relativa às diversas plataformas móveis, como comparações entre elas. Também é referido o posicionamento destas plataformas no mercado e é abordado o tema das arquiteturas dos *smartPhones*.

Na Figura 29, podemos observar uma imagem que ilustra uma das possíveis tendências dos sistemas operativos para os próximos anos. Segundo a Figura 29 o Android terá um grande aumento de vendas em comparação ao iOS ou Windows.

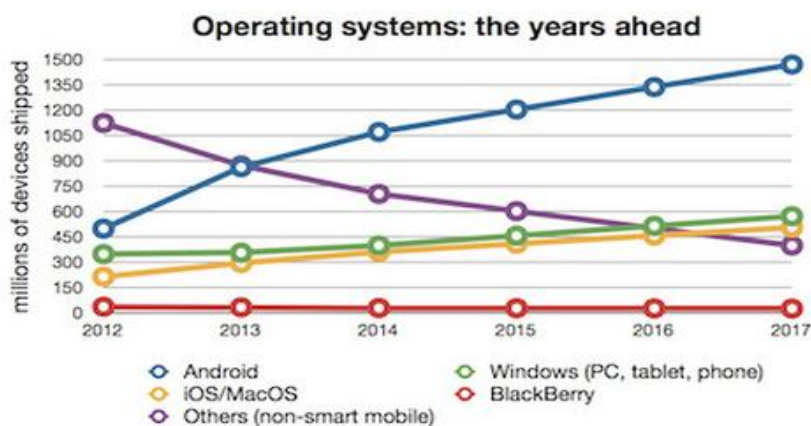


Figura 29 - Tendência dos próximos anos [21].

Esta tendência já se começa a verificar em alguns países onde os consumidores estão a migrar o uso da internet nos seus computadores pessoais (PCs) para os seus telemóveis, de acordo com o mais recente lançamento da International Data Corporation (IDC) Worldwide New Media Market Model. Esta tendência é mais notória atualmente nos Estados Unidos e Europa Ocidental, uma vez que nos EUA, o número de pessoas a aceder à internet por meio de um computador pessoal vai reduzir de 240 milhões de utilizadores (ano 2012) para 225 milhões em 2016. Ao mesmo tempo, o número de utilizadores móveis vai aumentar de 174 para 265 milhões.

Em 2015, pela primeira vez na história, haverá mais utilizadores norte-americanos que acedem à Internet por meio de dispositivos móveis do que por meio de computadores.

4.3. Implementação do Projeto

A fase de implementação foi composta por diversas fases, começando pela realização dos protótipos, de seguida pela programação da aplicação, usando o IDE Eclipse, juntamente com o Software Development Kit (SDK) do Android, através da linguagem de programação JAVA, sendo finalizada pela realização de testes do mesmo.

4.3.1. Modelo de Domínio

Na Figura 30 é possível visualizar o modelo de domínio da aplicação FST onde é notória uma grande parecença com o modelo de domínio do Safer. Visto que ambas as aplicações estão a trabalhar com o mesmo tipo de informação tomou-se a opção de replicar os modelos de domínio permitindo assim acelerar o processo de desenvolvimento.



Figura 30 - Modelo de domínio da aplicação mobile

A única classe diferente entre o modelo de domínio do Safer e do 4Forces Smart Teams é a classe VitalJacketMin que tem como objetivo guardar os dados do agente quando o sistema vital jacket está ligado.

4.3.2. Protótipos

Antes de iniciar o desenvolvimento da aplicação móvel, recorreu-se a protótipos, onde estes foram mostrados ao gestor de projeto para aprovação.

Esta primeira fase pode ser visualizada através de imagens de protótipos e descrições dos mesmos ilustrados em baixo.

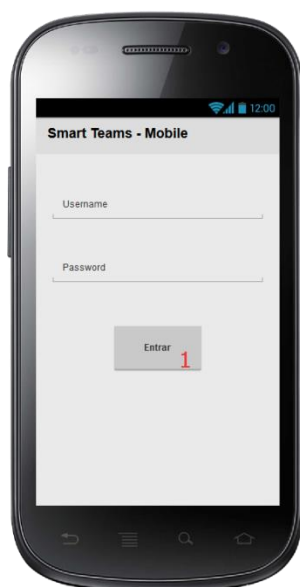


Figura 31- Protótipo do ecrã de Login

Na Figura 31 podemos observar o ecrã onde o utilizador se pode autenticar na aplicação. Este ecrã permite que o utilizador insira o seu *username* e *password* e ao selecionar o botão “entrar” (1) irá ser feita a validação dos dados, onde que se for com sucesso, o utilizador será redirecionado para o ecrã das missões. Por outro lado, se não existir uma ligação à internet ou os dados de o utilizador forem inválidos, aparecerá uma mensagem de erro através de um pop-up.

A Figura 32 ilustra o ecrã das missões. Este ecrã é um dos mais importantes na aplicação, pois neste ecrã, os utilizadores têm acesso às suas missões diárias.



Figura 32 - Protótipo do ecrã das missões

Este protótipo permite fazer uma pesquisa por uma determinada missão que se encontra na lista (1), ordenar as missões (4), como fazer um *refresh* à lista ou *logout* do utilizador (2). Através da seleção de uma missão teremos acesso a várias opções como alterar o seu estado, inserir comentário ou ver os detalhes (3).

4.3.3. Layout

Após a aprovação dos protótipos por parte do gestor de projeto iniciou-se uma fase de estudo sobre que tipo de design se iria seguir. Para isso, começou-se por navegar por diversas aplicações como facebook, gestor de contactos do android, gestor de mensagens, gmail, entre outras com a finalidade de perceber como dispor os ícones, botões e listas na aplicação.

Este estudo foi importante, pois as aplicações estudadas são utilizadas por milhões de utilizadores, facilitando assim a integração e a usabilidade de quem futuramente vai utilizar a aplicação “4Forces Smart Teams”.

4.3.4. Testes Realizados

No final do desenvolvimento de cada fase do projeto, foi necessário realizar testes para verificar se a aplicação estava de acordo com o que se pretendia e se as funcionalidades implementadas estavam a funcionar de forma correta. Os testes realizados foram:

- Testes Unitários.
- Testes de Integração.

Os testes unitários, também conhecidos por testes de unidade, são usados para testar uma única unidade do sistema, ou seja, testa de maneira isolada, geralmente simulando as prováveis dependências que aquela unidade tem. Estes testes normalmente são realizados pelos programadores da aplicação e têm como objetivo confirmar se determinados dados são válidos ou inválidos pela via de entrada/saída (I/O [22][23].

Os testes de integração, realizados após os testes unitários, são a fase em que os módulos são combinados e testados. Os testes de integração são alimentados pelos módulos, anteriormente testados pelos testes unitários, agrupando-os assim em componentes, como estipulado no plano de teste, que resulta num sistema integrado. O objetivo dos testes de integração consiste em verificar os requisitos funcionais, de desempenho e de confiabilidade na modelagem do sistema. Com ele é possível descobrir erros de interface, violação de integridade de ficheiros e estruturas de dados globais, problemas de configurações, tratamento de erros, entre outros [24][25].

A fim de verificar que o projeto se encontrava completamente estável e a funcionar de acordo com o pretendido, foi necessário recorrer à realização de testes. Os testes realizados estão mencionados na Tabela 5.

Tabela 5 - Referência dos testes realizados no projeto "4Forces Smart Teams"

Tipo de Testes	Anexo
Testes unitários	E
Testes de integração	F

Todos os testes descritos nos anexos foram executados até se obter o sucesso dos mesmos, verificando assim a inexistência de falhas.

4.4. Resultado Smart Teams - Mobile

Após a conclusão dos testes e consequente a aprovação pelo gestor de projeto, deu-se por terminado o desenvolvimento da aplicação “4Forces Smart Teams”. Os resultados deste projeto foram bastante positivos, ficando o protótipo pronto para ser apresentado.

Em seguida podemos visualizar o resultado de alguns ecrãs da aplicação, onde é ilustrado através da Figura 33 o ecrã de login, pela Figura 34 o ecrã correspondente às missões, pela Figura 35 o ecrã que representa um pedido de um equipamento e finalmente a Figura 36 ilustra os dados cardíacos de um agente, obtidos através do sistema vital jacket.

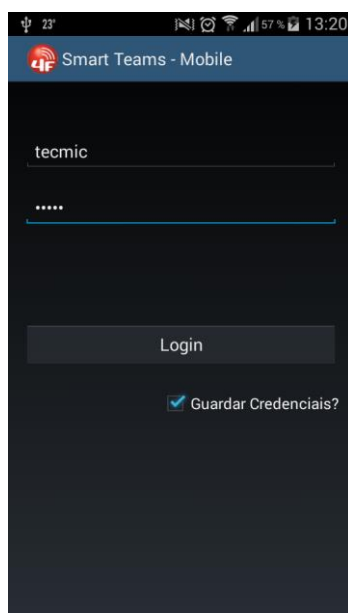


Figura 33 - Ecrã de login

A Figura 33 possibilita o utilizador entrar na aplicação, para isso precisa de inserir as suas credências. Além disso existe a possibilidade do utilizador guardar as suas credências para que não seja necessário colocá-las cada vez que pretenda entrar na aplicação.



Figura 34 - Ecrã de missões do agente

A Figura 34 corresponde a um ecrã que contém as missões de um agente. Estas missões contém várias tarefas onde o utilizador pode marcar como resolvidas ou não. Além disso é possível definir um estado operacional de um agente, podendo assim informar se este está em movimento, a executar uma tarefa ou a almoçar.

As maiores dificuldades neste ecrã deveram-se à construção de uma “custom list”, ou seja, foi necessário construir uma lista com itens, onde cada item tinha uma lista de subitens.



Figura 35 - Ecrã pedir um equipamento

A Figura 35 apresenta o ecrã com a funcionalidade que permite ao utilizador requisitar equipamentos. Para isso, o utilizador tem acesso a uma lista dos equipamentos disponíveis, onde poderá escolher quais pretende requisitar e a sua quantidade.

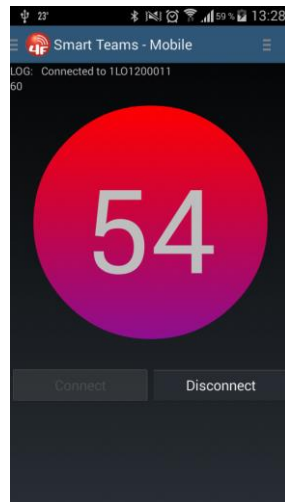


Figura 36 - Ecrã vital jacket

Na figura anterior (Figura 36) temos um ecrã que mostra um número que corresponde à pulsação cardíaca de um agente. Esta informação é acedida através do Vital Jacket (descrito no anexo A).

Para mostrar e atualizar a informação da pulsação foi necessário criar um serviço que corre em paralelo com a aplicação permitindo assim obter os dados necessários sem interferir com o desempenho da aplicação. O mais difícil de concretizar neste ecrã foi conseguir atualizar a informação em “real time”. Para este fim, utilizou-se um “timer” que 3 em 3 segundos vai verificar se algum valor foi alterado e por sua vez atualiza o ecrã.

5. Conclusões

O objetivo deste estágio curricular consistia em desenvolver uma aplicação mobile “4Forces Smart Teams” de raiz e também reconstruir e adaptar projetos já existentes nos quais se destacam os projetos “Safer”, “Download Data” e “Emergências Virtuais”.

Dos projetos nomeados em cima, nem todos tiveram o mesmo ênfase e trabalho durante este estágio pois o nível de funcionalidades e objetivos para cada um deles varia. O “Download Data” consiste num serviço que realiza regularmente o *download* de ficheiros meteorológicos (hdf5) a partir de um servidor remoto e via FTP. Estes ficheiros são utilizados no *plugin* das simulações, podendo assim representar determinados dias e situações, como por exemplo, grande nível de precipitação, ventos fortes, inundações, etc.

O “Emergências Virtuais” consiste num módulo de treino para agentes em que é possível avaliar o nível de preparação individual para dar resposta a possíveis emergências. Através de um mecanismo em tempo real de perguntas e respostas é possível avaliar um agente relativamente à sua capacidade de resposta e de que forma executaria as determinadas situações propostas.

O “Safer” que foi o projeto onde foi investido mais tempo e recursos devido à sua dimensão e complexidade, nasceu com o objetivo de estender o “4 Forces” de forma a incluir um novo grupo de utilizadores: as Forças de Segurança que se enquadra na área de gestão de emergências, uma área especializada derivada da gestão de ocorrências.

O “4Forces Smart Teams” é a aplicação mobile construída de raiz que tem como principal objetivo auxiliar os utilizadores no terreno, permitindo a estes visualizarem e marcarem as suas tarefas diárias, os seus estados, como também pedir equipamentos. Além disto, a aplicação tem outras funcionalidades como reportar a posição do utilizador e através de um dispositivo Vital Jacket monitorizar os dados vitais do mesmo.

5.1. Trabalho Futuro

A primeira etapa dos projetos está concluída, contudo existem sempre vários aspetos e funcionalidades que se poderia ter melhorado ou implementado. Como a maioria dos *softwares* é importante e necessário existir uma evolução e otimização dos sistemas.

Em relação ao projeto “Safer” era interessante e benéfico existir um sistema de notificações, ou seja, fosse possível notificar o utilizador através de (email, sms, etc) com informação relevante. Outra mais-valia deste sistema era alertar a equipa de desenvolvimento quando ocorrer um problema, permitindo assim um melhor e mais rápido suporte. Outra implementação pensada é de um sistema de limpeza de dados removidos. Os dados neste momento não se encontram a ser removidos, de modo a preservar histórico de relações entre objetos. Este sistema de limpeza irá limpar esses dados, movendo os dados removidos para um sistema à parte.

Noutra vertente, temos a aplicação mobile “4Forces Smart Teams” onde já foram pensadas diversas funcionalidades novas das quais se destaca a integração de um mapa na aplicação, permitindo assim visualizar as missões e os meios. Outra funcionalidade era permitir integrar funções de administrador, ou seja, muitas das ações feitas da aplicação “Safer” poderiam ser possíveis na aplicação mobile. Por último, integrar mecanismos como o de *upload* de imagens e notificações.

Bibliografia

- [1] <https://trello.com/>
- [2] W. Royce, "Managing the development of large software systems," Proc. IEEE WESCON, 1970.
- [3] Rodrigues, Joel – Safer. Leiria: [s.n.], 2013. Relatório de Estágio.
- [4] "Action Modulers." [Online]. Available: <http://www.actionmodulers.pt/default.aspx>. [Acedido em 30 de Junho de 2014].
- [5] "BlueCape [Online]. Available: <http://joomla.bluecape.com.pt/>. [Acedido em 30 de Junho de 2014].
- [6] Y. Zhou, J. Levy, J. Hammitt, and J. Evans, "Estimating population exposure to power plant emissions using CALPUFF: a case study in Beijing, China," Atmos. Environ., 2003
- [7] <https://www.eclipse.org>
- [8] <http://www.componentfactory.com/>
- [9] <http://developer.android.com/sdk/index.html>. Acedido em 3 de Maio de 2014
- [10] <http://developer.android.com/sdk/adding-components.html>. Acedido em 3 de Maio de 2014
- [11] http://en.wikipedia.org/wiki/Android_Lollipop. Acedido em 11 de Janeiro de 2015.
- [12] <http://androidandme.com/2014/01/news/google-posts-fresh-android-distribution-stats-jelly-bean-and-kitkat-usage-on-the-rise/>. Acedido em 5 de Maio de 2014
- [13] <https://developer.android.com/about/dashboards/index.html>. Acedido em 4 de Novembro de 2014
- [14] Acedido em 30 de Outubro de 2014, "According to Apple, people have all but stopped upgrading to iOS 8", de <http://9to5mac.com/2014/10/06/ios8-market-share-stagnated/>.
- [15] <http://www.androidmeup.com/en/news/amu-specials/android-vs-ios-vs-windows-phone-7-a-brief-comparison>. Acedido em 3 de Maio de 2014
- [16] <http://www.redmondpie.com/ios-5-vs.-android-4.0-ice-cream-sandwich-vs.-windows-phone-7.5-mango-comparison/>. Acedido em 3 de Maio de 2014
- [17] <https://market.android.com/publish/signup>. Acedido em 3 de Maio de 2014

- [18] <http://developer.apple.com/programs/start/register/create.php>. Acedido em 3 de Maio de 2014
- [19] <https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html>. Acedido em 3 de Maio de 2014
- [20] <http://create.msdn.com/en-us/home/membership>. Acedido em 3 de Maio de 2014
- [21] <http://www.xataka.com.mx/celulares-y-smartphones/conoce-los-datos-de-las-104-millones-de-lineas-moviles-en-mexico>
- [22] Wikipédia (19 de Fevereiro de 2012). “Teste de unidade”. Obtido em 19 de Março de 2012, de http://pt.wikipedia.org/wiki/Teste_de_unidade
- [23] Thiago, André (6 de Abril de 2012). “As Vantagens do Teste Unitário”. Obtido em 19 de Março de 2012, de <http://andrethiago.wordpress.com/2011/04/06/as-vantagens-do-teste-unitario/>
- [24] Wikipédia (17 de Janeiro de 2012). “Teste de integração”. Obtido em 19 de Março de 2012, de http://pt.wikipedia.org/wiki/Teste_de_integra%C3%A7%C3%A3o
- [25] Lourenço, Marcelo (s.d.). “Teste de integração”. Obtido em 19 de Março de 2012, de <http://andrethiago.wordpress.com/2011/04/06/as-vantagens-do-teste-unitario/>
- [26] Acedido em 5 de Maio de 2014. “About Us”, de http://www.vitaljacket.com/?page_id=17
- [27] Acedido em 5 de Maio de 2014, “A Empresa”, de <http://www.petratex.com/Empresa>

Anexos

ANEXO A – Vital Jacket

O VitalJacket é o dispositivo médico certificado que combina tecnologias e soluções de engenharia biomédica tradicionais.

A t-shirt utilizada é uma peça de vestuário fácil de vestir, cardiomonitor muito confortável e confiável que pode ser usada pelo paciente, permitindo, pelo menos, 72 horas de exames contínuos. Deste modo, permite aos médicos fazerem uma avaliação correta de problemas cardíacos num ambiente cotidiano.

O projeto VitalJacket começou na universidade de Aveiro, em 2002, como um colete protótipo capaz de medir vários sinais vitais que se desenvolveram ao dispositivo médico de alta tecnologia têxtil disponível no mercado hoje em dia, graças a uma parceria entre a Biodevices [26] e Petratex [27].

No contexto do projeto Smart-Teams, irão ser utilizados vários equipamentos para a utilização do VitalJacket, nos quais passo a nomear:

- T-shirt
- Eléttodos
- Carregador
- Cartão SD
- Dispositivo Eletrónico



Figura 37 - T-shirt

A Figura 37 é um dos exemplos de uma possível t-shirt utilizada com este serviço, tendo em conta que qualquer tipo de t-shirt poderia ser utilizado, pois estas não influenciam o correto funcionamento do equipamento.



Figura 38 - Eléctrodo

Os eléctrodos são utilizados para obter frequências cardíacas. Estes dispositivos são colocados dentro da t-shirt em contato com a pele e em sítios específicos para que haja a maior fiabilidade com os resultados obtidos. Este equipamento é de fácil utilização e tem um formato circular, como se pode observar na Figura 38.



Figura 39 - Dispositivo Eletrónico

A Figura 39 demonstra um exemplo de um dispositivo que permite adquirir sinais de eletrocardiograma (ECG) e sinais de frequência cardíaca através nos eléctrodos. Para que exista comunicação de dados entre dispositivos, como por exemplo, enviar dados para um *smartphone* é utilizada uma ligação Bluetooth. Para auxiliar o utilizador, este dispositivo tem leds que indicam um determinado estado.

- Luz Vermelha: Frequência cardíaca
- Luz Verde: Dispositivo a carregar
- Luz Azul: Bluetooth ligado



Figura 40 - Cartão SD

Após a informação ser recebida é necessário guardá-la para que futuramente possa ser analisada. Para isso, o dispositivo eletrônico tem a capacidade de utilizar cartões de memória onde guarda a informação obtida. Os cartões normalmente utilizados têm a capacidade de 1 gigabyte e pode-se observar na Figura 40 um exemplo desses.



Figura 41 - Carregador

Por último, na Figura 41 temos um carregador pois o dispositivo eletrônico é alimentado por bateria. A sua autonomia é em média de 36 horas dependendo do número de elétrodos ligados e do tempo que é configurado para a recolha de dados.

ANEXO B – Emergências Virtuais

O *plugin* de emergências virtuais é composto por três categorias: Templates, Pessoas e Exercícios (representado na Figura 42). Na categoria de “Templates” é possível configurar os *templates* relativos às Simulações, Situações e às Ações Esperadas. Os *templates* são compostos por alguns exemplos típicos. Na categoria de “Pessoas” é possível adicionar ou editar a informação relativa às pessoas: cargos, exercícios em que participou e a sua função.

A categoria de “Exercícios” é composta por dois menus, um que pertence apenas aos utilizadores com funções de administrador/instrutor (que poderão editar e executar os exercícios de simulação) e outro para ser apresentado aos utilizadores normais (que apenas podem participar nos exercícios de simulação)

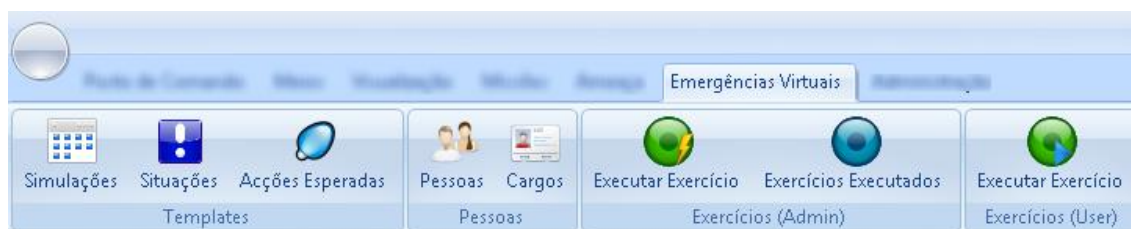


Figura 42 - Painel das Emergências Virtuais

1. Templates de Simulação

Ao selecionar o botão “Simulações” no grupo “Templates” (representado na Figura 42) será apresentada uma janela semelhante à que se encontra representada na Figura 43. Nesta janela serão apresentadas todas as Simulações que são *Template*. As simulações de *Template* podem ser criadas para ajudar a criar um Exercício de Simulação, podendo ser criado de imediato um teatro de operações e prevendo as coisas que devem/podem acontecer (situações) e criar as ações que se espera que os formandos executem, mediante a ocorrência de determinadas situações.

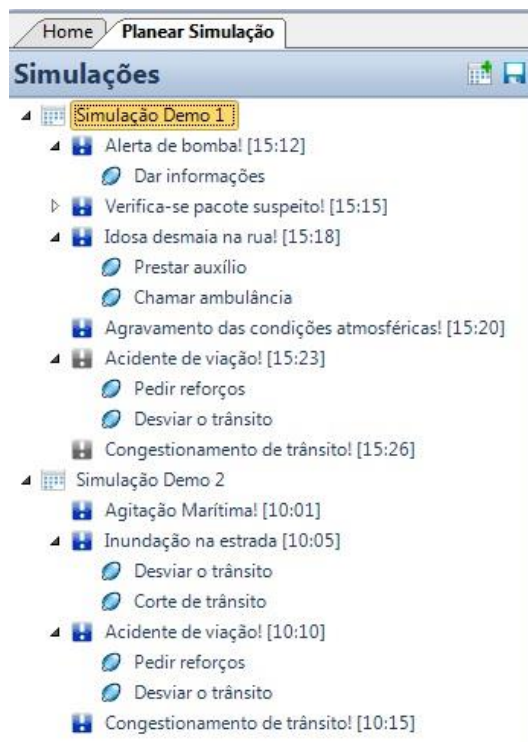


Figura 43 - Árvore de navegação de uma simulação.

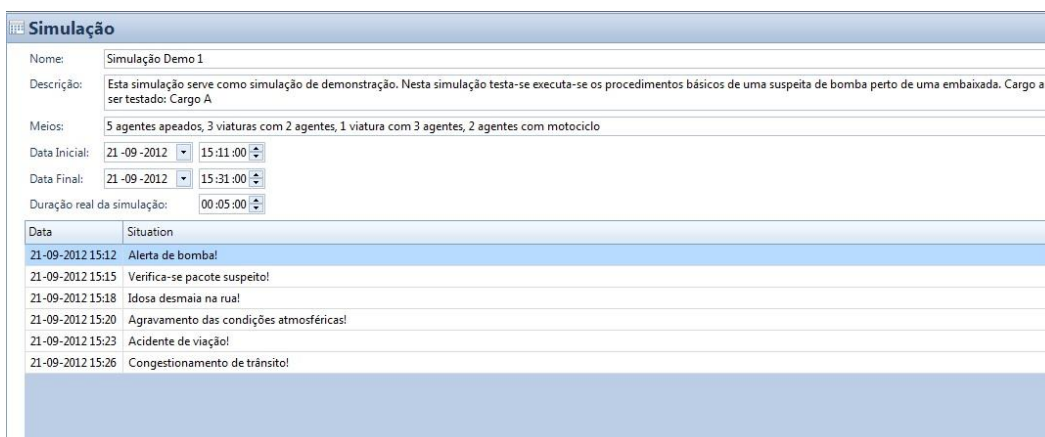


Figura 44 - Conteúdo de uma simulação.

As Simulações *template* podem ser editadas, sendo alterado as características iniciais (data de ocorrência, duração da simulação e os meios (meios podem ser considerados equipamentos, veículos, agentes, etc.)

Através da árvore de navegação, que se encontra como exemplo na Figura 43, é possível criar e editar novas situações assim como as ações esperadas para cada uma delas.

Para guardar as alterações efetuadas nas simulações deve selecionar-se o botão de “guardar”, na barra de Simulações. Para criar novas simulações *template* basta selecionar o botão ao lado do “guardar” e uma nova simulação será criada.

Na Figura 44 é ilustrado um exemplo do tipo de conteúdo que é mostrado após selecionar uma situação ou ação na árvore de navegação.

2. Templates de Situação

Através da janela de Situações (representada na Figura 45) é possível gerir as situações *template* existentes, editá-las ou adicionar novas situações.

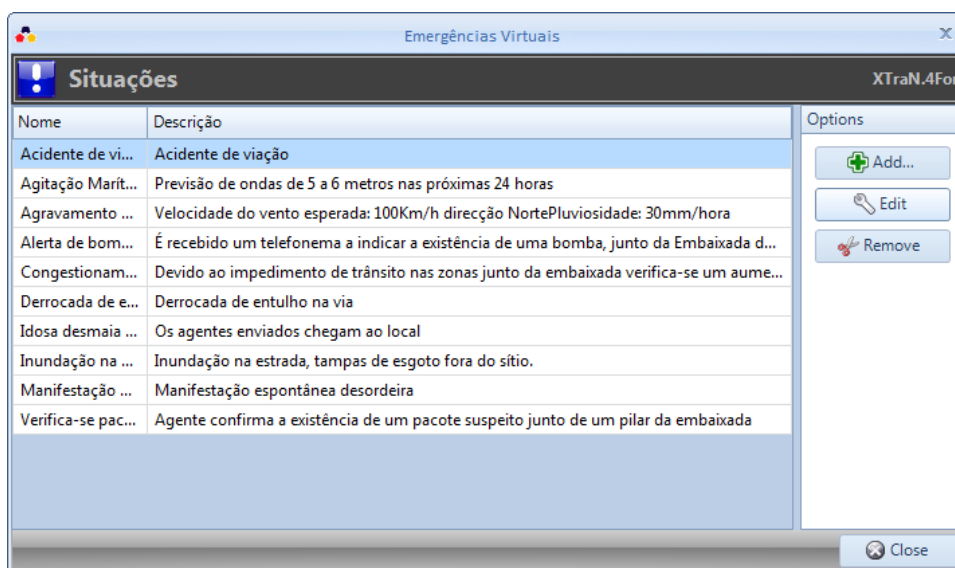


Figura 45 - Janela de Situações.

As situações são compostas por um nome, uma descrição e é possível adicionar uma lista de Ações Esperadas (ver Figura 46).

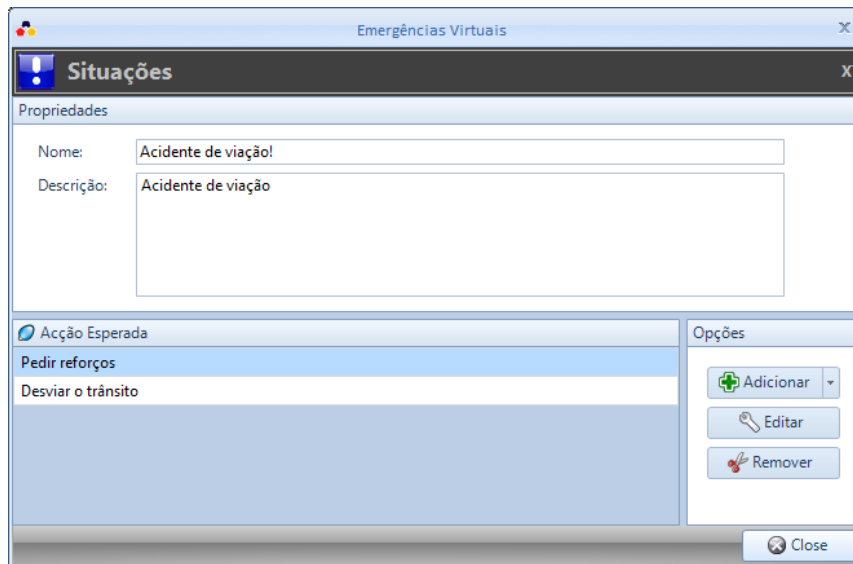


Figura 46 - Janela de edição de Situações.

Para adicionar novas ações basta premir o botão adicionar, onde surgem duas opções: Nova Ação Esperada – onde é possível criar uma nova Ação esperada (será aberta uma janela semelhante à representada na Figura 47); ou Ação Esperada Existente – onde é replicada uma Ação esperada template (Figura 48).

3. Templates de Ações Esperadas

Ao seleccionar o botão “Ação Esperada” no grupo “Template” (representado na Figura 42) será apresentada uma janela semelhante à que se encontra representada na Figura 47. Nesta janela é possível gerir as Ações Esperadas templates existentes. É possível adicionar novas ações (ver Figura 48) ou editar as existentes. Esta janela tem um filtro para ajudar a encontrar as ações pretendidas.

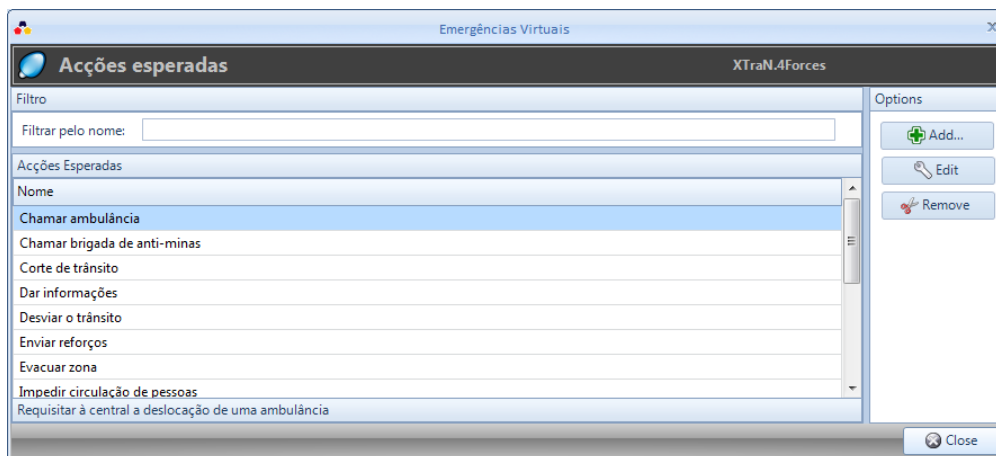


Figura 47 - Janela de Ações Esperadas.

Uma ação esperada é composta por um nome e uma breve descrição, que deve resumir sucintamente todas as medidas que devem ser tomadas para que a mesma seja concretizada com sucesso.

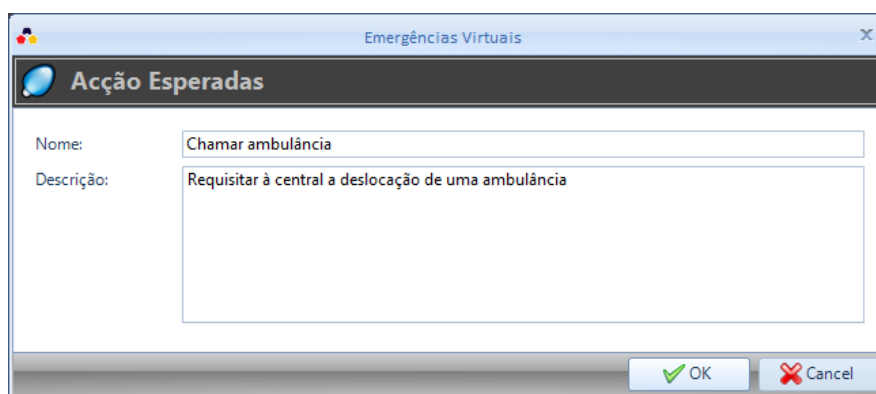


Figura 48 - Janela de edição das Ações Esperadas.

4. Pessoas

Ao premir o botão “Pessoas” no grupo “Pessoas” (representado na Figura 42) será apresentada uma janela semelhante à que se encontra representada na Figura 49. Nesta janela (que teve como objetivo futuro ser ligada à tabela de utilizadores da TECMIC) é possível gerir as pessoas existentes na base de dados. É possível adicionar pessoas à base de dados e editar os dados das pessoas já existentes (representado na Figura 50).

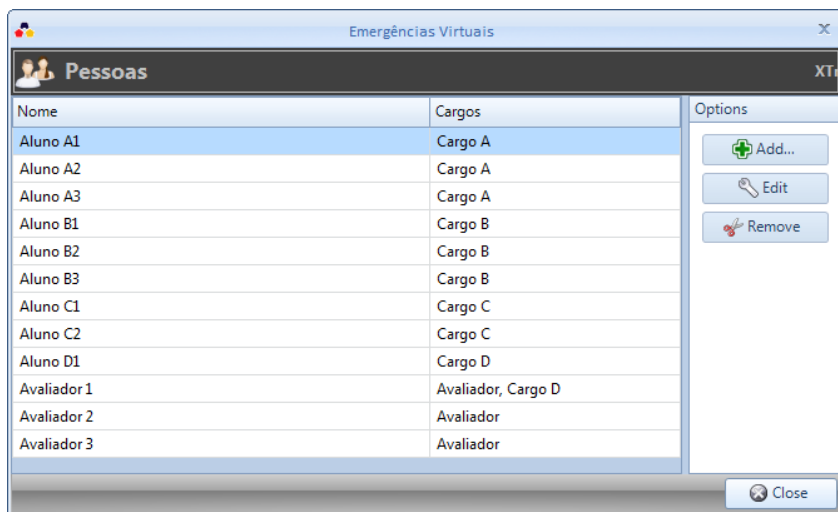


Figura 49 - Janela de Pessoas.

Na janela de edição de dados das pessoas pode-se alterar o nome da pessoa e os cargos que a pessoa possui na empresa. É ainda possível consultar todos os exercícios de simulação em que a pessoa participou, a data de realização do mesmo e qual o papel que a pessoa desempenhou nesse exercício.

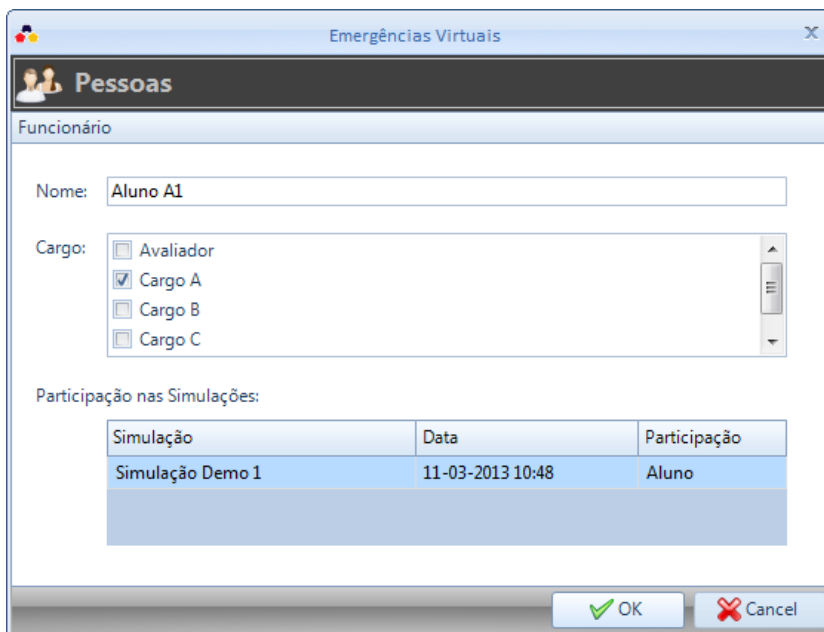


Figura 50 - Janela de edição de dados das pessoas.

5. Cargos

Ao selecionar o botão “Cargos” no grupo “Pessoas” (representado na Figura 42) será apresentada uma janela semelhante à que se encontra representada na Figura 51. Nesta janela é possível gerir os cargos existentes na base de dados. É possível ainda adicionar novos cargos.

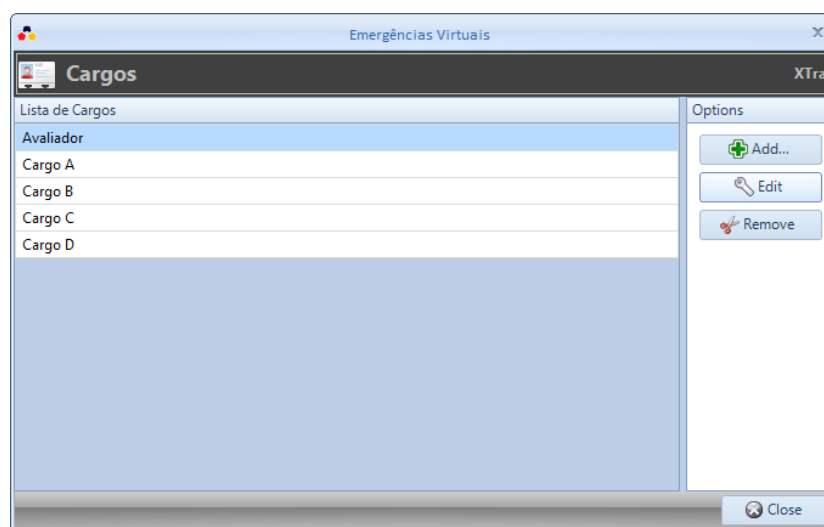


Figura 51 - Janela de Cargos.

6. Executar Exercício (Modo de Administração)

Ao selecionar o botão “Executar Exercício” no grupo “Exercícios (Admin)” (representado na Figura 42) será aberta uma janela que apresenta todos os exercícios que se encontram no modo de preparação ou prontos a executar (representado na Figura 52).

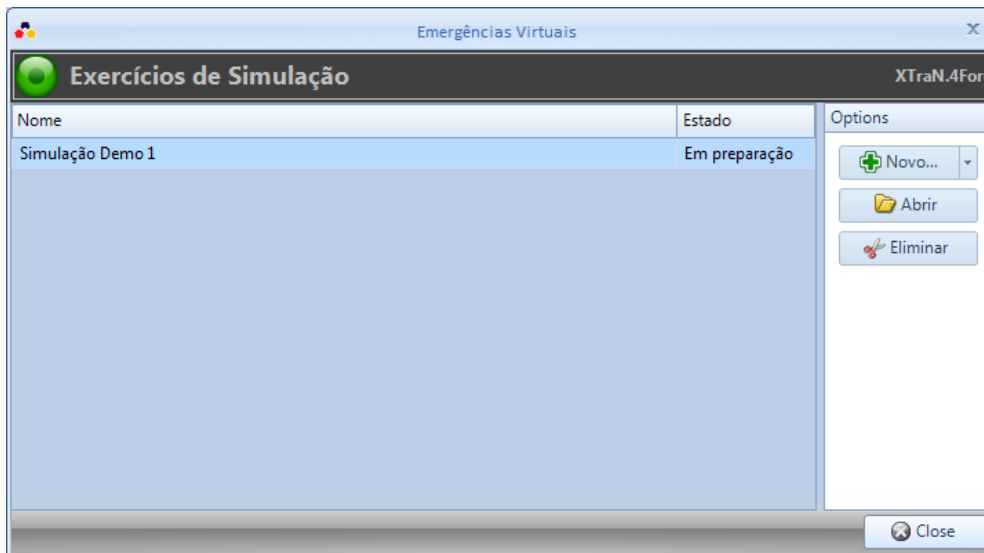


Figura 52 - Janela com exercícios em preparação e prontos a executar.

Após escolher um exercício, ou criar um novo, será apresentada uma janela semelhante à que se encontra representada na Figura 53.

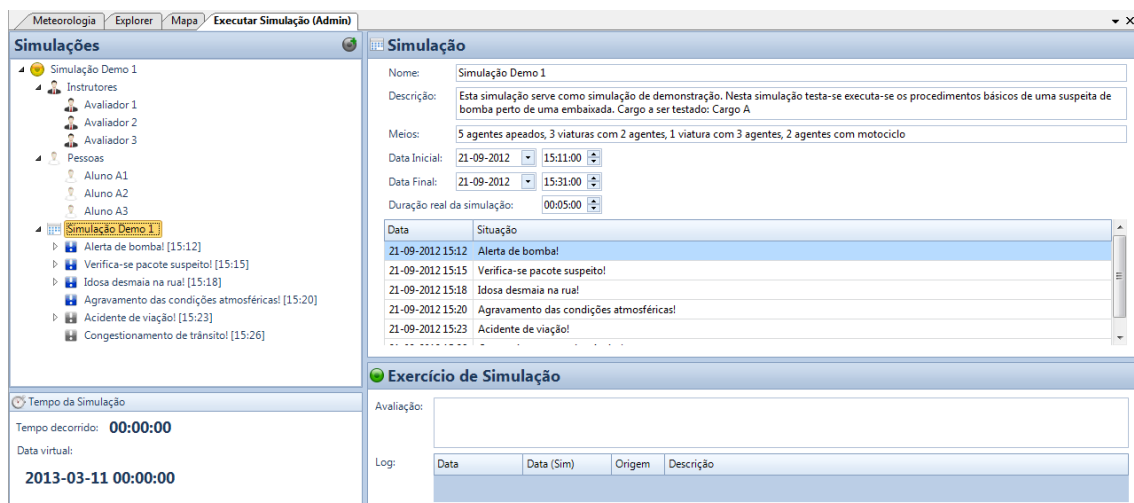


Figura 53 - Janela de execução de um Exercício de Simulação (em modo de preparação).

Na janela representada na imagem a cima ilustrada (Figura 53) é possível criar e gerir todo o exercício de simulação. Através dos menus existentes na árvore é possível alterar o nome do exercício de simulação; alterar o estado do exercício de simulação; adicionar ou retirar instrutores e alunos do exercício de simulação; adicionar/editar/trocar/ de simulação, adicionar/editar/retirar/ativar/desativar as situações; adicionar/editar/retirar as suas ações esperadas.

Quando a preparação do exercício de simulação estiver completa e o mesmo estiver pronto a ser executado, deve alterar-se o estado (premir o botão direito do rato sobre o exercício e escolher a opção: “Pronto a executar” – representado na Figura 54).

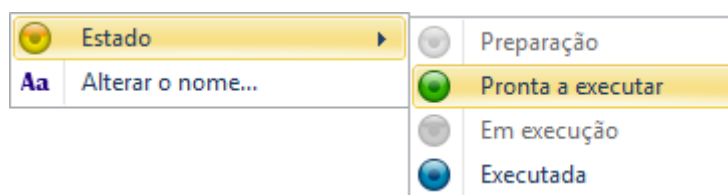


Figura 54 - Alterar o estado da Simulação

Após o (s) aluno (s) estar conectado ao exercício deve dar-se início à execução do exercício de simulação (botão direito do rato sobre o exercício e escolher a opção: “Iniciar simulação” – representada na Figura 55).

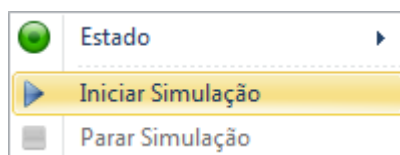


Figura 55 - Iniciar Simulação

Durante a execução do exercício as situações ativas são lançadas automaticamente, chegando a sua hora de lançamento. O instrutor também pode, em tempo real, lançar as situações manualmente, adicionar situações (novas ou existentes) à simulação ou ativar situações que estejam desativas, como é ilustrado na Figura 56.

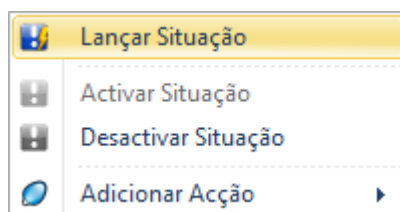


Figura 56 - Lançamento de Situações (manual)

O instrutor deve marcar as ações esperadas como: realizada com sucesso ou Ação realizada tardiamente (ver representação na Figura 57). Se o instrutor não alterar os estados das ações, as mesmas serão consideradas como não executadas.

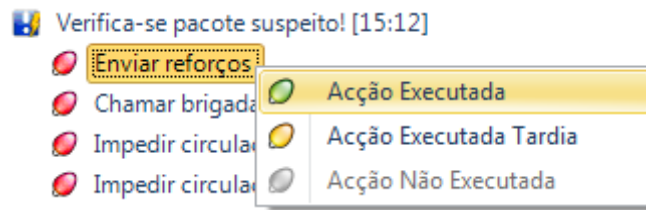


Figura 57 - Alterar estado da ação esperada

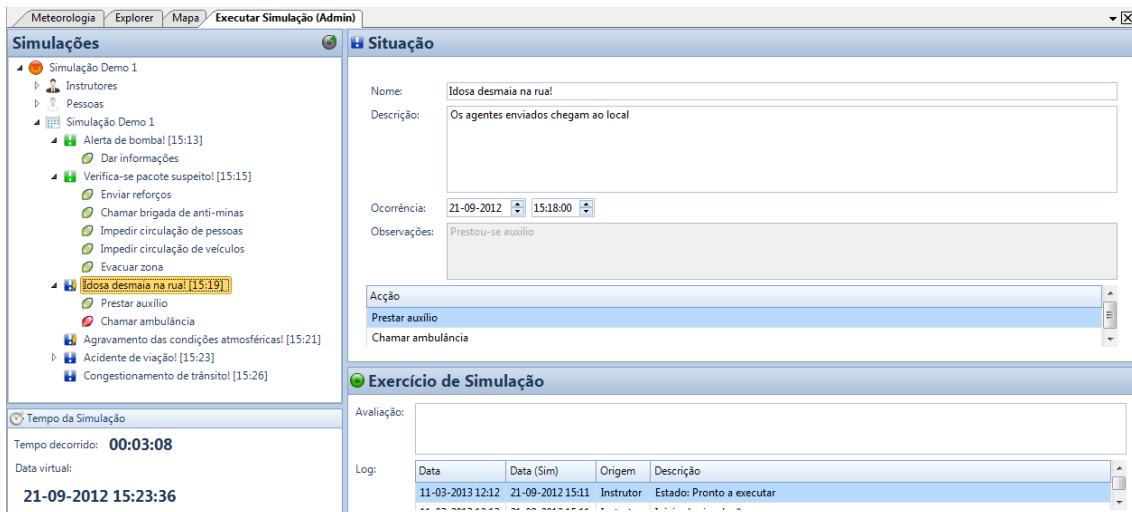


Figura 58 - Janela de execução de um Exercício de Simulação (em modo de execução).

No controlo de “Exercício de Simulação” o instrutor/avaliador pode escrever algumas notas sobre o desempenho do (s) aluno (s) ou outras observações. Neste controlo existe um registo histórico de todo o exercício, onde constam algumas informações chave, como por exemplo: início da simulação, lançamento das situações, situações resolvidas, etc.

Nesta janela é ainda possível a criação de novos *templates*, para a simulação, situações e ações esperadas. Esta funcionalidade permite replicar o exercício realizado de forma a ser realizado por outros alunos, sem ser necessário planear o exercício todo de origem.

Para criar *templates*, basta carregar em cima de um nó (simulação, situação ou ação esperada) com o botão direito do rato e seleccionar a opção: Criar *Template* (representado na Figura 59 para o caso de criar um *template* de uma situação).

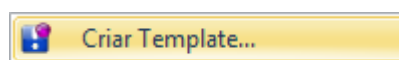


Figura 59 - Criar template de uma situação.

i. Consultar Exercício de Simulação Executado

Ao premir o botão “Exercícios Executados” no grupo “Exercícios (Admin)” (representado na Figura 42) será aberta uma janela que apresenta todos os exercícios que se encontram no modo “Pronta a executar” (representada na Figura 60).

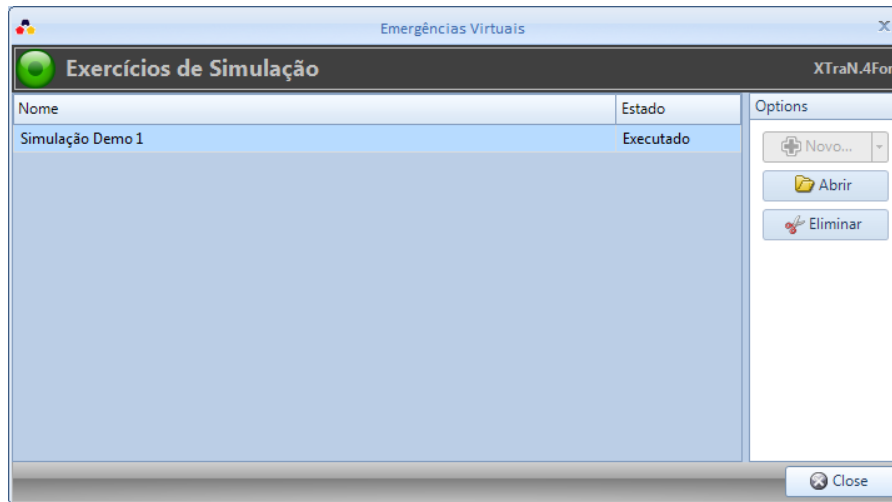


Figura 60 - Janela para consulta dos exercícios de simulação executados

Após escolher o exercício que se deseja consultar, basta premir o botão “Abrir” e automaticamente aparecerá uma janela semelhante à que se encontra representada na Figura 61. Nesta janela é possível analisar o exercício de simulação, vendo as situações que ocorreram, e quando, assim como as ações que foram tomadas pelos alunos, as notas tiradas pelos mesmos e as notas dos instrutores.

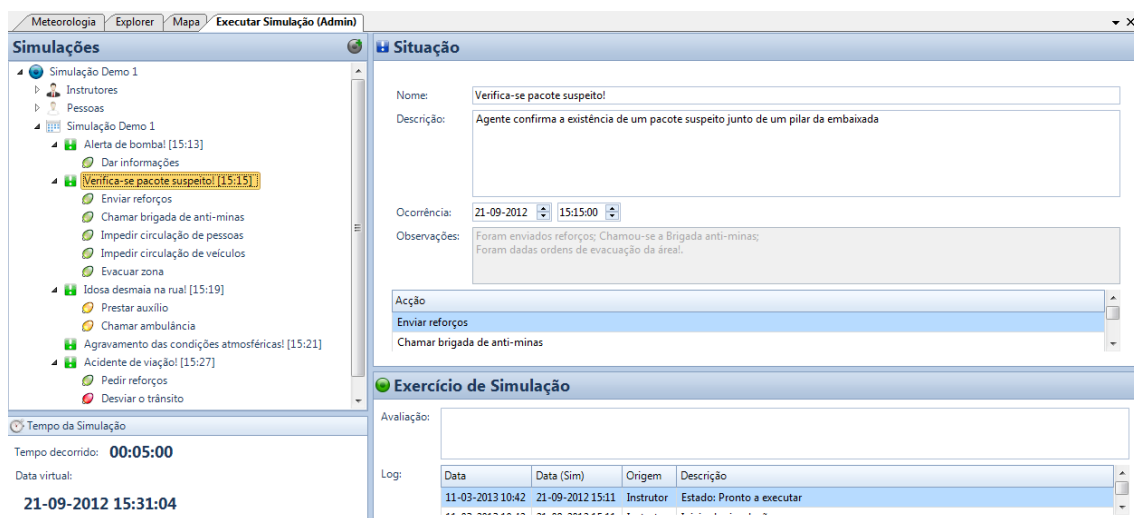


Figura 61 - Janela de consulta de Exercício de Simulação já executado.

ii. Executar Exercício (Modo de utilizador)

Ao seleccionar o botão “Executar Exercício” no grupo “Exercícios (*User*)” (representado na Figura 42) será aberta uma janela que apresenta todos os exercícios que se encontram no modo “Pronta a executar” (representada na Figura 62).

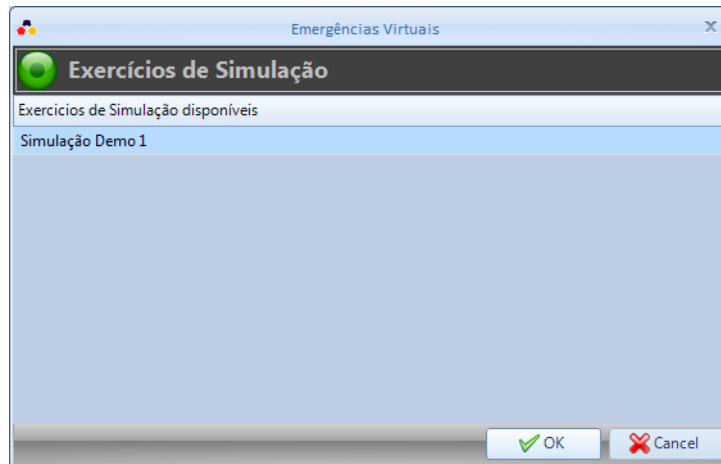


Figura 62 - Janelas de exercícios prontos a executar.

Após escolher o exercício que se pretende executar, será aberta uma nova janela, semelhante à representada na Figura 63. Inicialmente esta janela apenas apresenta a Simulação em teste e a descrição da mesma. À medida do desenrolar do Exercício de Simulação as situações vão sendo apresentadas ao utilizador, tendo o mesmo que tomar algumas medidas.

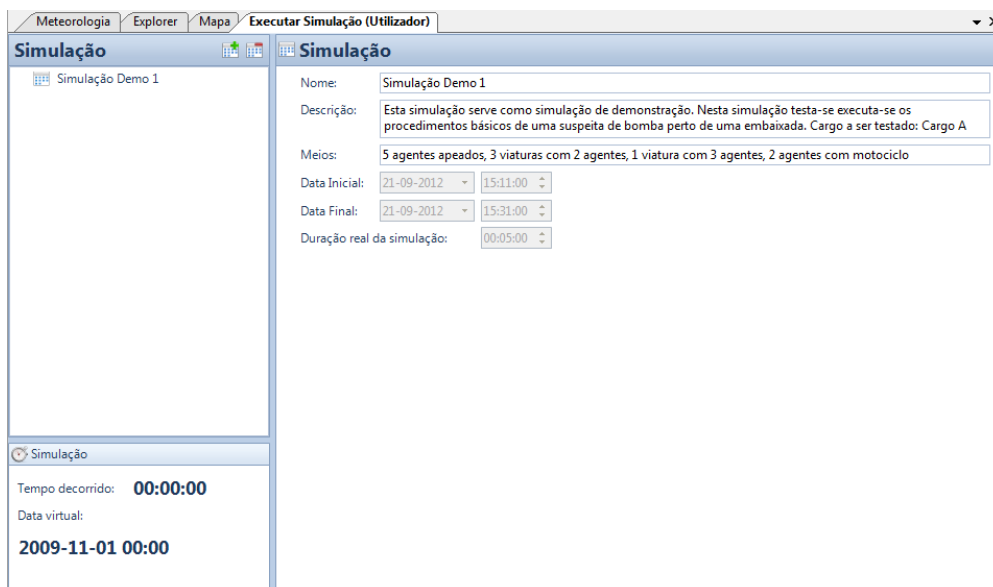


Figura 63 - Janela com exercício de simulação pronto a executar

O utilizador, durante a execução do exercício deve tomar as ações que acha necessárias, escrevendo uma pequena observação ou justificação sobre as medidas que decidiu (representado na Figura 64). O utilizador pode considerar que as ações já se encontram resolvidas, para isso basta selecionar a situação pretendida, premir com o botão direito do rato, e escolher a opção do menu “Situação resolvida”. A mesma será alterada para um ícone verde.

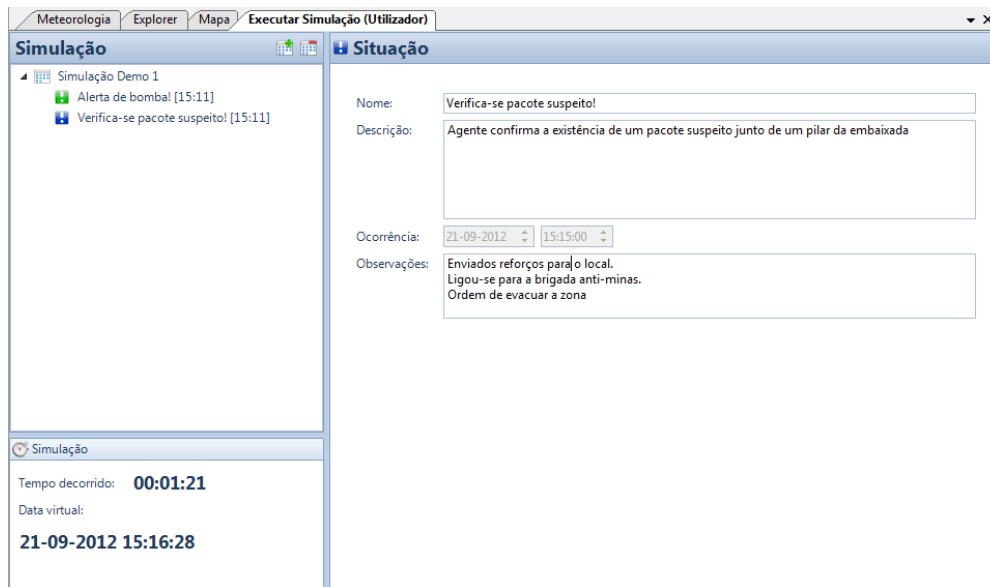


Figura 64 - Janela de exercício de simulação em execução

Após a conclusão do Exercício de Simulação todas as situações devem estar concluídas e o utilizador deverá ter uma janela semelhante à representada na Figura 65.

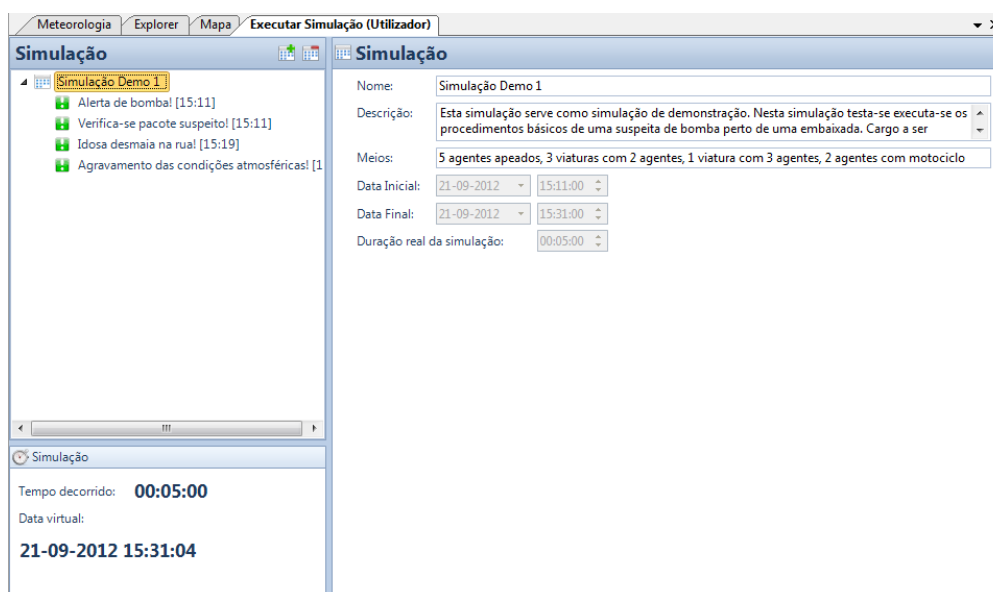


Figura 65 - Janela de Exercício de Simulação em execução (vista do aluno).

iii. Tempo de Sincronização

O tempo de sincronização com a base de dados, enquanto é executado o exercício de simulação é de 30 segundos. Ou seja, o lançamento de situações, criação, ativação ou desativação das mesmas só é persistido na base de dados a cada 30 segundos. O mesmo se aplica do lado do utilizador, só de 30 em 30 segundos é que são atualizadas o lançamento de situações ocorridas e persistidas as alterações que o mesmo efetuou (situações resolvidas e observações sobre as ações tomadas). O tempo total até à sincronização pode ser, no pior caso, de 1 minuto (30 segundos de cada lado).

Este pequeno atraso também pode ser verificado, nalguns casos, no início do exercício da simulação, entre a janela do instrutor e do aluno.

ANEXO C

1.1. Plataformas Móveis

É notório um aumento exponencial de características e funcionalidades que um dispositivo móvel tem vindo a ganhar, possibilitando assim existir alguma variedade de modelos e plataformas, sendo as mais conhecidas e usadas:

- Android
- iOS
- Windows Phone

1.1.1. Android

O Android, tem como principal característica oferecer uma plataforma *OpenSource* e customizável, usando a linguagem de programação JAVA e tendo como base o Linux. Esta plataforma é propriedade da Google.

O seu *kit* de desenvolvimento (SDK) é recomendado ser instalado no IDE Eclipse, mas também é possível ser instalado noutros IDE que suportem a linguagem JAVA sendo que, este SDK oferece ao programador uma interface gráfica para a realização de interfaces, uma ferramenta de *debug* e um emulador Android para o programador testar as suas aplicações sem necessitar de um dispositivo físico [9][10].

A última versão estável que existe desta plataforma é o Android 5 (Lollipop) lançado em Novembro de 2014 [11], contudo apenas alguns dispositivos utilizaram esta versão. Por este motivo a versão 4.4 (Kitkat) torna-se uma opção muito válida mesmo que em semelhança às anteriores ainda contar com poucos dispositivos ativos, como se pode observar na figura abaixo (Figura 66), a maioria da responsabilidade deste fenómeno deve-se à taxa de penetração no mercado ainda ser reduzida.

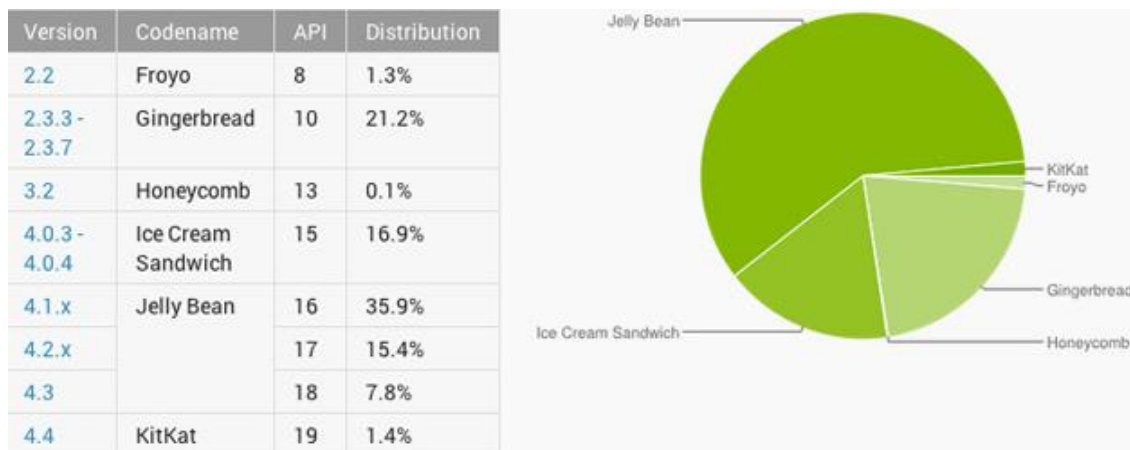


Figura 66 - Informação sobre números relativos de dispositivos ativos a correr Android (Janeiro de 2014) [12]

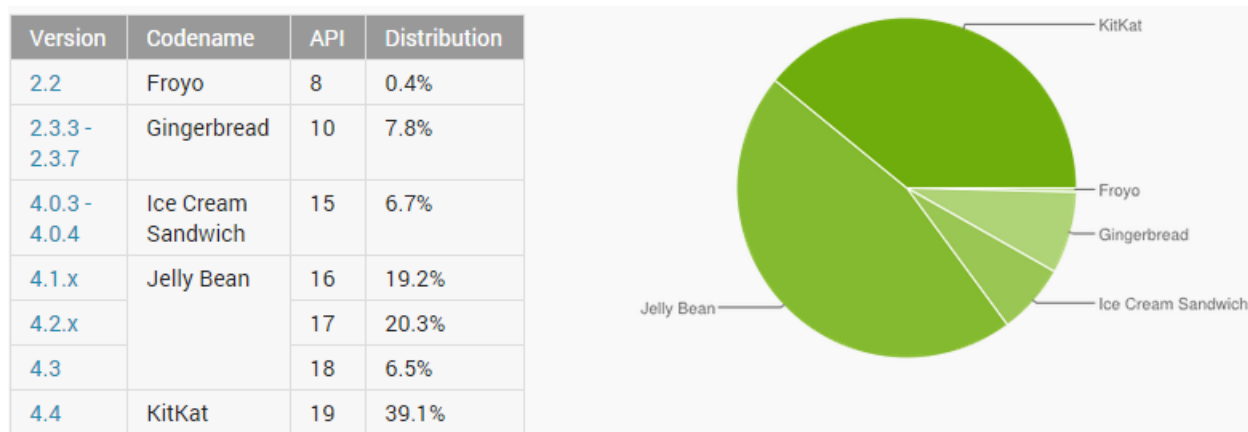


Figura 67 - Informação sobre números relativos de dispositivos ativos a correr Android (Janeiro de 2015) [13]

Nas Figura 66 e Figura 67 pode-se observar que as versões inferiores à 4.0.X tendem a perder utilizadores a uma grande velocidade, visto que de Janeiro de 2014 a Janeiro de 2015 existiu uma quebra de 13.4% nas versões 2.3.3-2.3.7. Por outro lado, a versão 4.4 (KitKat) teve um aumento muito significativo e a tendência é elevar esse número todos os meses.

Na escolha de uma versão para o desenvolvimento optar-se-ia pela versão 4.0.X para a versão mínima, esta opção é fundamentada pela percentagem de utilizadores existentes nesta versão ser elevada e é previsível que quando a *app* estiver pronta para ir para o mercado, o número de utilizadores a utilizar esta versão seja ainda maior.

1.1.2.iOS

O iOS, plataforma móvel proprietário da Apple, baseado do Sistema Operativo Darwin, é destinado apenas aos dispositivos da própria marca (iPhone, iPad, iPod Touch), tornando uma desvantagem em relação a outras plataformas móveis. Esta plataforma contém várias aplicações padrão (browser safari, cliente mail, entre outros) e, os programadores que pretendem criar novas aplicações devem usar o SDK destinado ao iOS, que inclui um IDE (designado por Xcode), uma ferramenta gráfica para a elaboração de interfaces, ferramenta de *debug* e um simulador de ambientes iOS. Este SDK apenas permite ser instalado em computadores Apple (Mac OS X), tornando a sua programação limitada a vários programadores.

A versão iOS mais recente é iOS 8.1 lançado em Outubro de 2014, sendo já muito usada por diversos utilizadores, como se pode visualizar pela Figura 68.

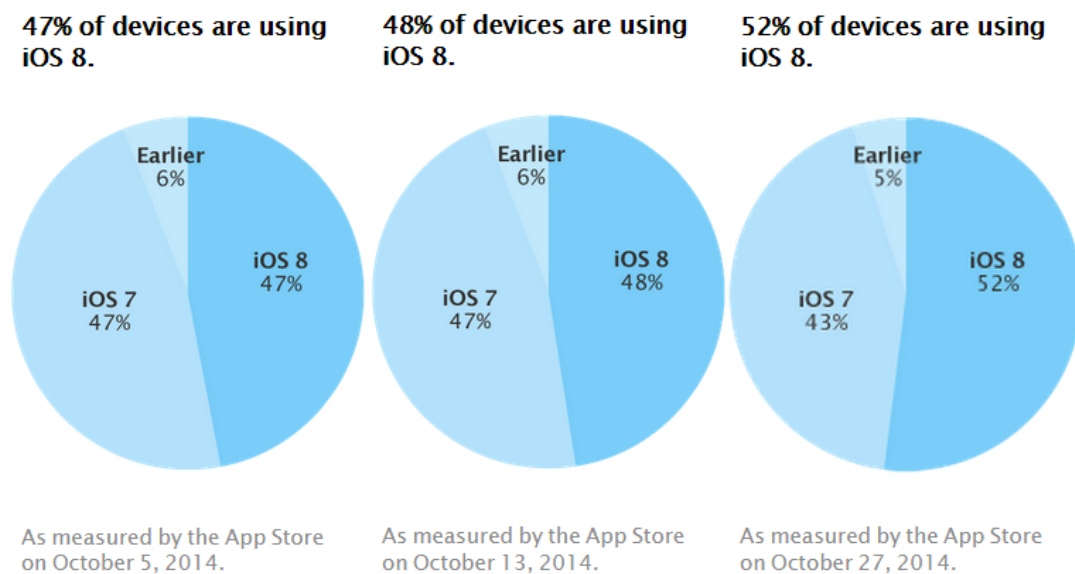


Figura 68- Distribuição das versões do iOS [14].

1.1.3. Windows Phone

O Windows Phone (WP) é a plataforma móvel mais recente no mercado da mobilidade. Esta plataforma móvel, desenvolvido pela Microsoft, apareceu com o objetivo de ser o sucessor do antigo Windows Mobile. Ao contrário do seu ante sucessor, o WP não permite o controlo de todo o *hardware* e como tal as aplicações do Windows Mobile não são suportadas pelo WP, sendo uma desvantagem este tipo de incompatibilidade.

Para usar o respetivo SDK é necessário usar um computador com um Sistema Operativo da Microsoft (Vista/Seven/Eight), em junção com o IDE Visual Studio. O *kit* de desenvolvimento oferece aos programadores uma interface gráfica, uma ferramenta de *debug*, bem como um simulador, para que os programadores possam testar as aplicações sem necessitarem de um dispositivo físico.

A versão 8.1 é a versão mais recente do Windows Phone, lançada em Abril de 2014.

1.2. Comparação entre Plataformas Móveis

A Tabela 6 compara o *kit* de desenvolvimento necessário para a programação de cada umas das plataformas móveis.

Tabela 6 - Kit de desenvolvimentos das principais plataformas móveis

Plataformas Móveis	Android	iOS	Windows Phone
Necessita de SDK	Sim	Sim	Sim
IDE	Eclipse (recomendado)	Xcode	Visual Studio
Plataforma suportada	Linux, Windows e Mac OS X	Mac OS X	Windows Vista / 7 / 8
Linguagem de Programação	Java	Objective C	.NET (C# ou Visual Basic)

Como se pode ver pela Tabela 6 existe uma maior flexibilidade de programação no Android, por ser a única que pode ser programada usando qualquer Sistema Operativo (Linux, Windows e Mac OS X), as restantes duas só podem ser programadas usando o Sistema Operativo proprietário da própria plataforma móvel. Em relação à linguagem de programação, todas elas usam linguagens diferentes, no entanto, são linguagens Orientadas a Objetos. O Android é programado pela conhecida linguagem JAVA, o iOS pelo Objective C e o WP pela linguagem proprietária da Microsoft.

As principais plataformas móveis contêm características muito semelhantes, como por exemplo o suporte de *multi-touch* e de multitarefa, no entanto elas também contêm algumas diferenças significativas, em que a principal é o suporte de JAVA e Flash (só o Android é que suporta) e a questão da segurança (o Android é o mais vulnerável a ataques). Na Tabela 7 estão referidas as principais características/comparações destas três plataformas [15][16].

Tabela 7 - Principais características do Android vs iOS vs WP

Sistemas Operativos Móveis	Android	iOS	Windows Phone
Suporte ecrã tátil	Sim	Sim	Sim
Multi-Touch	Sim	Sim	Sim
Multi-Tarefa	Sim	Sim	Sim
Store	Android Market	App Store	MarketPlace
Suporta Flash	Sim	Não	Não
Updates de firmware	Sim	Sim	Sim
Suporta Java	Sim	Não	Não
Segurança	Sujeito a malware	Sim	Sim
Suporta Tablets	Sim	Sim	Não
Suporte de hardware	Grande variedade de hardware suportado	Só iPhone, iPod touch e iPad	Pouca variedade de hardware suportado

Como se pode observar pela Tabela 7, cada plataforma tem uma Store, onde os programadores podem submeter as suas aplicações. O Android usa o *Android Market* e para se submeter uma aplicação nessa *Store* é necessário realizar um registo e um pequeno pagamento (aproximadamente de 20€). Este valor é de apenas para o registo, não existindo qualquer anuidade para a renovação da conta.

Após o registo e validação da conta, o programador pode publicar qualquer aplicação no *Android Market*. O tempo entre o *upload* da aplicação e este ficar disponível no *Market* é relativamente rápido, podendo ir de minutos a poucas horas. Em comparação a outros *Markets*, este tempo de espera é baixo porque a Google não realiza nenhuma verificação à aplicação submetida (<https://market.android.com>) [17].

As aplicações desenvolvidas em iOS são submetidas na *Apple Store* e, para isso, o programador também necessita de realizar um registo e, neste caso, pagar uma anuidade (aproximadamente de 77€) [18].

Depois da realização do registo e validação da mesma, o programador pode publicar as suas aplicações na *Apple Store*. Ao contrário do que acontece no *Android Market*, na

Store da Apple o tempo entre a submissão da aplicação e ficar disponível ao público pode ser longo (cerca de 14 dias), isto porque a Apple vai verificar a aplicação para ver se está de acordo com as suas políticas de desenvolvimento [19].

Após a validação da aplicação, por parte da Apple e, conseqüente aceitação, a aplicação fica disponível para o público (<http://store.apple.com>)

Em relação às aplicações WP, estas são submetidas no MarketPlace e, o processo requer também um registo e um pagamento de uma anuidade (aproximadamente de 77€) [20].

Após validação do registo, o programador pode submeter as aplicações que quiser para o WP, no entanto, e como acontece na Apple Store, existe um tempo entre a submissão da aplicação e até este ficar disponível ao público (cerca 7 dias), isto porque a Microsoft vai verificar se a aplicação está de acordo com as exigências pedidas pela entidade. Quando a Microsoft validar a aplicação, esta fica disponível para o público (<http://www.windowsphone.com/pt-PT/marketplace>).

A Tabela 8 demonstra as diferenças existentes entre as três *stores*.

Tabela 8 - Diferenças das stores do Android vs Apple vs Microsoft

	Android Market	Apple Store	MarketPlace
Registo	Sim	Sim	Sim
Valor a pagar	≈ 20€	≈ 77€/ano	≈ 77€/ano
Delay upload e disponibilização	Quase imediata	≈ 14 dias	≈ 7 dias
Requer validação por parte da propriedade da App	Não	Sim	Sim

Através da Tabela 8 é possível verificar, que quem programa para a plataforma Android, tem maior liberdade e facilidade em submeter as suas aplicações, além de não ser necessário que o Google faça uma verificação da aplicação, o valor a pagar é muito menor em comparação às restantes *stores*. Um aspeto negativo, refere-se ao aspeto de a Google não realizar nenhuma verificação à aplicação. Assim, a probabilidade de existirem aplicações que possam comprometer SmartPhones, é maior que nas restantes *stores*.

1.3. Quota de Mercado

A luta por uma posição no mercado está em constante mudança. Estas variações devem-se principalmente às características que cada plataforma tem para oferecer, às constantes inovações e preço final de venda. Como podemos observar na Tabela 9, verificou-se que no ano 2012 a plataforma Android foi a mais utilizada obtendo uma quota de mercado de 68.8%, seguido por iOS com 19.4% e colocando em terceiro lugar o Windows Phone com 2.7% de conta mundial.

Tabela 9 – Quota de mercado no ano de 2012

Plataforma	Mercado no ano de 2012 (%)
Android	68.8
iOS	19.4
Windows Phone	2.7
Outros	9.1

No ano seguinte (2013) é notório o crescimento da plataforma Android ao obter uma quota de mercado de 78.9%, aumentando 10% relativamente ao ano anterior. No caso do iOS teve uma ligeira quebra, aproximadamente 4% ficando com uma quota mundial de 15.5%. Por último, o Windows Phone subiu 1% relativamente ao ano anterior para uma quota de 3.6%. Os dados referentes ao ano 2013 podem ser observados na Tabela 10.

Tabela 10 – Quota de mercado no ano de 2013

Plataforma	Mercado no ano de 2013 (%)
Android	78.9
iOS	15.5
Windows Phone	3.6
Outros	2.0

1.4. Arquiteturas

Para que a informação possa ser acessada e atualizada é necessário que exista troca de dados. Para que isso seja possível existem dois tipos de arquiteturas:

- Cliente/Servidor.
- Peer-to-peer.

1.4.1. Cliente/servidor

A arquitetura cliente/servidor baseia na partilha de recursos dos servidores para os seus clientes, ou seja, temos servidores que estão à espera de pedidos por parte dos clientes onde posteriormente responderão com informação relevante para os mesmos. Os servidores e clientes apenas estão ligados entre si através de uma rede de computadores. Podemos ver um exemplo na Figura 69.

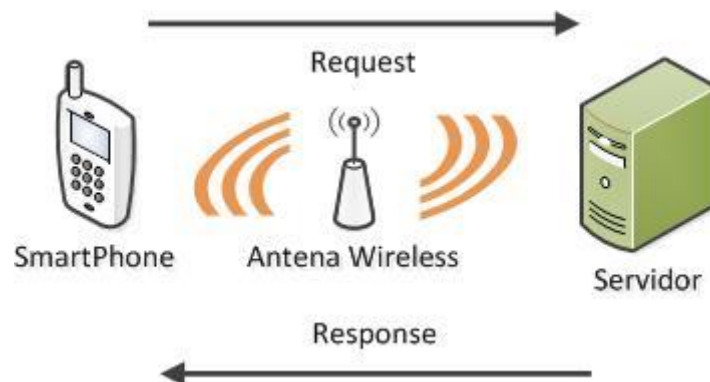


Figura 69 - Exemplo da arquitetura cliente/servidor nos SmartPhones

Ao utilizar esta abordagem, teremos como principais vantagens uma menor probabilidade de a rede ser comprometida e facilidade de sincronização, por outro lado, é necessário ter um servidor com algumas capacidades e características para conseguir responder aos diversos pedidos e respostas no menor tempo possível.

1.4.2. Peer-to-peer

Esta arquitetura também conhecida por P2P é caracterizada por um estilo de comunicação descentralizado entre os participantes de uma rede de dados. Esta permite a ligação ponto a ponto entre dispositivos com funcionalidades semelhantes, ou seja, cada dispositivo pode atuar tanto como cliente como também servidor, como demonstra a Figura 70.

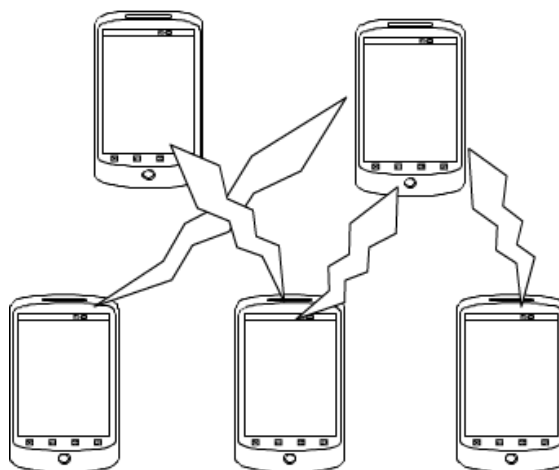


Figura 70 - Exemplo de uma arquitetura peer-to-peer nos SmartPhones

Como a arquitetura cliente/servidor, a *peer-to-peer* conta com as suas vantagens e desvantagens, na qual podemos destacar como vantagem; a troca de mensagens entre dispositivos e o *bottleneck* do servidor é inexistente. No entanto, temos como desvantagens uma maior probabilidade de a rede ser comprometida ou a dificuldade de sincronização entre os dispositivos.

Para o projeto a ser desenvolvido, a plataforma escolhida foi o Android. As razões da escolha foram com base na análise anteriormente referida. Tendo em conta que daqui a uns anos haverá um aumento significativo dos equipamentos com a plataforma Android, como também o aumento do número de utilizadores a usar SmartPhones, esta é a melhor plataforma para se realizar o projeto, bem como o preço dos dispositivos ser razoável, tendo em conta a relação preço/qualidade.

A arquitetura escolhida será cliente/servidor, pois o objetivo é centralizar a informação e disponibilizá-la aos seus clientes.

ANEXO D

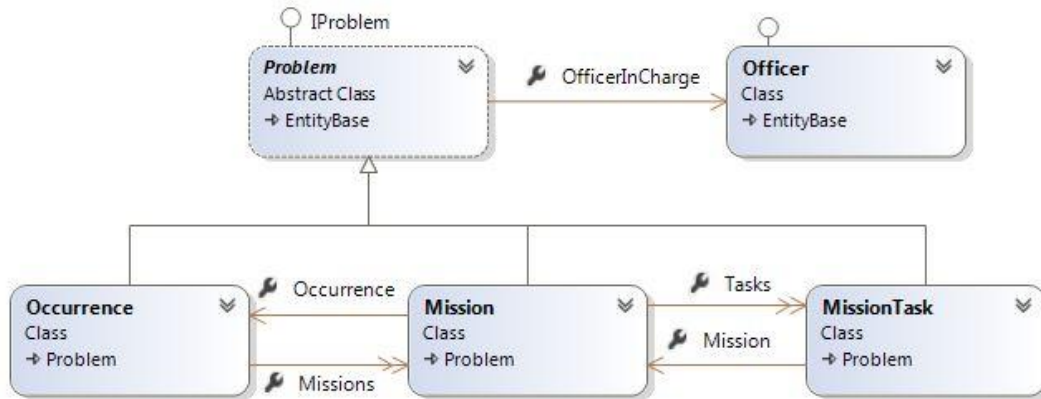


Figura 71 - Entidade Problem

A entidade Problem é uma classe abstrata e encontra-se na base das classes occurrence, mission, missionTask e officer. Esta classe também estende de uma classe que é a base de todo o domínio, que é denominada por entityBase. Na Figura 71 é ilustrado os relacionamentos da entidade problema.

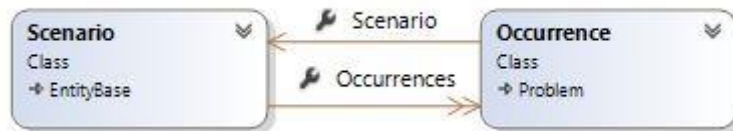


Figura 72 - Entidade Scenario

Um scenario representa uma área que abrange ocorrências que se relacionam. Esta relação pode ser observada na Figura 72.

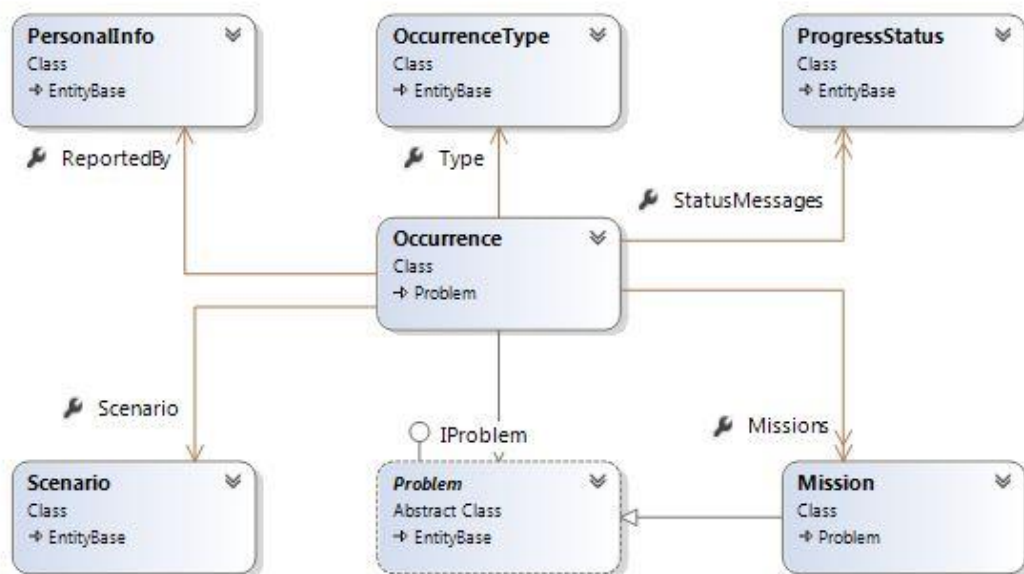


Figura 73 - Entidade Occurrence

Na Figura 73 representa as relações da entidade occurrence. Destaca-se a relação com occurrenceType, que consiste num tipo de ocorrência (v.g. “Assalto”, “Alarme”) e progressStatus que contém o progresso da ocorrência. Uma occurrence normalmente corresponde a um acontecimento.

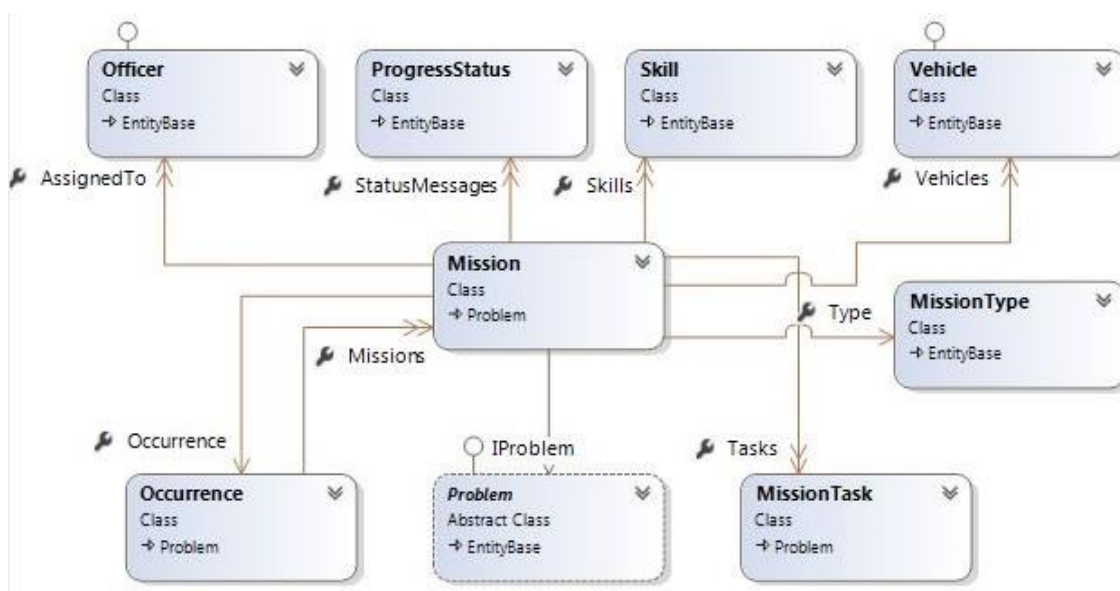


Figura 74 - Entidade Mission

Uma *mission* corresponde a uma ou conjunto de tarefas. Normalmente associada a uma ocorrência mas pode existir independente de uma. Na Figura 74 consegue-se observar as várias relações que a *mission* contém, como os agentes designados para a missão, as competências que a missão necessita, os veículos utilizados, o tipo de missão, a que ocorrência está associada (não precisa de uma) ou que tarefas são necessárias executar.

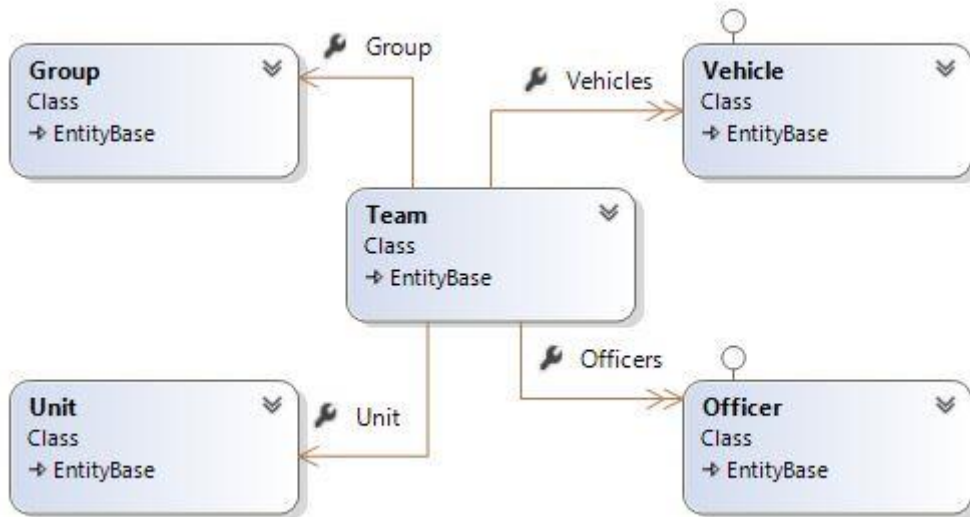


Figura 75 - Entidade Team

A Figura 75 ilustra a entidade team que contém um agrupamento de meios (agentes e veículos) e pode ainda estar associada a uma entidade unit e group.

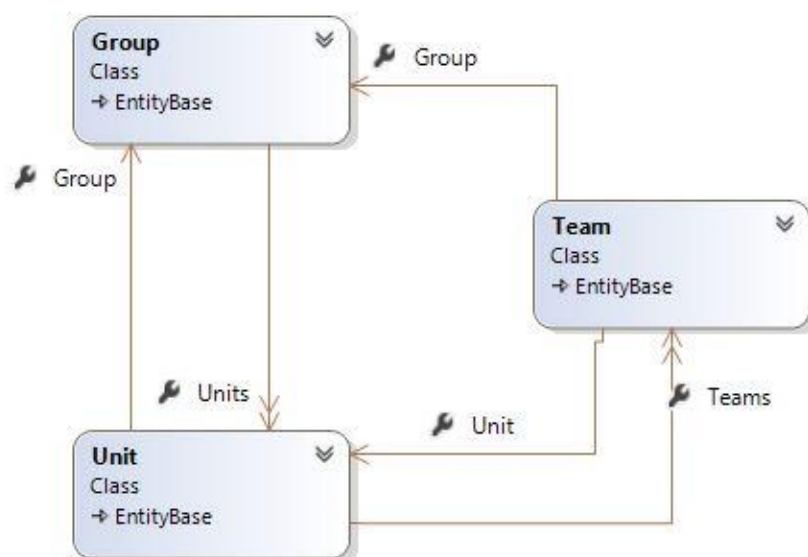


Figura 76 - Entidade Group / Unit / Team

É possível observar na Figura 76 as ligações entre group, unit e team. A entidade group corresponde à “maior área” que abrange unit e team. De seguida temos a entidade unit que é abrange uma área mais pequena que a entidade group mas maior que a entidade team. Por último temos a entidade team que representa uma equipa de meios.



Figura 77 - Entidade PersonalInfo

A entidade personalInfo representa a informação pessoal de uma pessoa. Esta está associada a uma entidade de contactos para que seja possível ter N contactos de uma determinada pessoa. Pode-se observar na Figura 77.

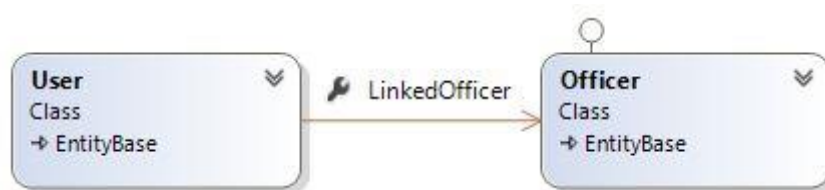


Figura 78 - Entidade User

A gestão de utilizadores é importante para permitir a autenticação no sistema. Um utilizador do sistema pode (ou não) ser um agente conforme representa a propriedade LinkedOfficer no utilizador. Esta associação é visível na Figura 78.

ANEXO E

Tabela 11 - Teste Unitários do projeto “4Forces Smart Teams”

Descrição	Pré-condição	Etapas do Teste	Entradas	Saídas Esperadas
Entrar na aplicação após autenticação bem-sucedida	Ter a aplicação “4Forces Smart Teams” para a plataforma Android instalada	Selecionar o botão “Login”	<i>Username</i> e <i>password</i> válidas	Deverá aparecer o ecrã correspondente à lista de tarefas associadas ao utilizador autenticado.
Selecionar ou desseleccionar uma tarefa de uma missão	Ter a aplicação “4Forces Smart Teams” para a plataforma Android instalada com autenticação	Pressionar a checkbox relativa a uma tarefa	N/A	<ul style="list-style-type: none"> - A aplicação enviará um pedido para o servidor para alteração de estado. - A resposta do servidor deverá ser de sucesso. - Uma mensagem será apresentada ao utilizador com a informação de sucesso.
Alterar o estado operacional do agente	Ter a aplicação “4Forces Smart Teams” para a plataforma Android instalada com autenticação	Selecionar a combobox relativa ao estado operacional do agente	N/A	<ul style="list-style-type: none"> - A aplicação enviará um pedido para o servidor para alteração de estado. - A resposta do servidor deverá ser de sucesso. - Uma mensagem será apresentada ao utilizador com a informação de sucesso.

ANEXO F

Tabela 12 - Testes de Integração do projeto “4Forces Smart Teams”

Descrição	Pré-condição	Entradas	Saídas Esperadas
Entrar na aplicação previamente registado	Ter a aplicação 4Forces Smart Teams para Android instalada.	<i>Username e password</i>	Redireccionamento para o ecrã com a lista de missões.
Tentativa de autenticação na aplicação sem ligação à internet	Ter a aplicação 4Forces Smart Teams para Android instalada.	N/A	Aparece um pop-up com uma mensagem a informar que não existe ligação de dados.
Tentativa de fazer alterações na aplicação sem ligação à internet	Ter a aplicação 4Forces Smart Teams para Android instalada com a autenticação validada.	N/A	Aparece um pop-up com uma mensagem a informar que não existe ligação de dados.
Tentativa de receber informação dos dados cardíacos do utilizador sem ter o aparelho ligado	Ter a aplicação 4Forces Smart Teams para Android instalada com a autenticação validada.	N/A	Aparece uma mensagem no ecrã da aplicação a informar que o aparelho não está ligado.