



ESTGOH

**Escola Superior de Tecnologia e Gestão
de Oliveira do Hospital**

Instituto Politécnico de Coimbra

“Mandown” – Lone Worker Monitoring

**Relatório apresentado com vista à obtenção do grau de Mestre no âmbito da
realização do ciclo de estudos de Mestrado em Informática Aplicada**

Autor:

Fábio Samuel Martins Pais Marques

Orientador:

Professor Doutor Francisco Carlos Afonso

Coorientador:

Professor António Manuel de Brito Paulino

Data: Dezembro de 2015

“O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis”.

José Alencar

À minha família.

Agradecimentos

O trabalho que aqui se apresenta, além da minha dedicação, só foi possível graças à colaboração e apoio de algumas pessoas, às quais não posso deixar de prestar o meu reconhecimento.

Assim, começaria por agradecer aos dois orientadores Prof. Doutor Francisco Carlos Afonso e Professor António Manuel de Brito Paulino, por todo o apoio e acompanhamento. Aos restantes professores, em especial ao Doutor Luís Veloso, agradeço pelo aconselhamento prestado ao longo do percurso académico, e por terem transmitido os seus conhecimentos ao longo destes anos de licenciatura bem como de Mestrado.

Agradeço também, ao meu supervisor de estágio Doutor Jorge Saraiva, pela sua disponibilidade, pela forma como me integrou na empresa, nas tarefas, nos métodos de trabalho e pela forma como sempre esteve disponível para me ajudar a ultrapassar os mais diversos obstáculos encontrados ao longo de todo o projeto no ambiente empresarial.

Um agradecimento a todos os colegas de turma – em especial ao Telmo Viana, Paulo Seixas e Duarte Ormonde – e também a todos os colegas de equipa da “TulaIt” com os quais tive o privilégio de trabalhar durante os meses de estágio: Eng.º João Matos, Eng.º Álvaro Bizarro e Eng.º Filipe Silva pela disponibilidade, companheirismo e espírito de entreatajuda demonstrados ao longo do projeto.

A toda a minha família, em especial aos meus pais e irmão pela forma como sempre me apoiaram e ajudaram a ultrapassar as barreiras que foram surgindo.

Finalmente, um agradecimento muito especial para a minha namorada Joana Almeida pela compreensão, carinho, paciência e incondicional apoio que sempre demonstrou.

A todos, o meu obrigado.

Resumo

Após decisão de realização de estágio para conclusão do Mestrado em Informática Aplicada, seguiu-se a candidatura a empresas onde seria possível a realização do mesmo. Após várias candidaturas, foi eleita a empresa “Tula Move”, para realização do referido estágio.

Em termos práticos e objetivos, a principal finalidade deste projeto foi a produção, em ambiente de contexto empresarial, de uma solução que tem como principal propósito disponibilizar funcionalidades úteis e essenciais cuja correta utilização resulta numa melhoria significativa de segurança, bem como das condições de trabalho, para trabalhadores solitários. Depois de analisadas e enumeradas, estas necessidades e consequentes funcionalidades, foram implementadas e integradas numa só aplicação num equipamento Android.

Esta aplicação foi desenvolvida numa arquitetura cliente-servidor, constituída pelas componentes web e móvel com finalidade de suprir as necessidades acima referidas. A componente web é responsável pela administração e gestão de clientes. É através desta que a administração das empresas que utilizam a aplicação realiza a gestão dos seus funcionários e equipamentos móveis, atribuição de permissões, entre outros. Assim como a monitorização dos serviços realizados pelos mesmos e consequente geração de relatórios de diferentes atividades dos funcionários.

A componente móvel (Android) é utilizada pelos funcionários, sendo através desta que se encontram acessíveis todas as funcionalidades do “cliente” como trabalhador solitário. Ela integra todas as funcionalidades necessárias num só dispositivo, entre as quais se destacam: *tracking* de rondas efetuadas, emissão de alarmes, *walkie-talkie* e relógio de ponto.

Palavras-chave: Node.js, Android, Mumble, Trabalhador Solitário, Monitorização, Posicionamento Interior, Posicionamento Exterior, GPS, Wi-Fi Fingerprint.

Abstract

After deciding to do an internship to complete the Master of Science in Applied Informatics, some applications were made to companies where it would be possible to do so. After several applications, the company “Tula Move” was elected for holding the internship.

In practice, the main purpose of this project was the development, in a business context environment, of a solution that has as its main purpose to provide useful and essential features to significantly improve the safety and working conditions for lone workers. After analyzed and listed, these needs and subsequent features have been implemented and integrated into a single application on an Android device.

This application was developed in a client-server architecture, consisting of web and mobile components in order to meet the needs mentioned above. The web component is responsible for system and customers management. On the web platform, administrators can make the management of employees and mobile devices, assign permissions, among other functionalities. Moreover, it is possible the monitoring of services provided by employees and subsequent reporting of their different activities.

The mobile component (Android) is used by employees. It is through this component that all the features of "customer" as a lone Worker are accessible. It integrates all necessary functions in a single device (e.g. device tracking, alarms, push-to-talk, and time-clock).

Keywords: Node.js, Android, Mumble, Lone Worker, Monitoring, Indoor Positioning, Outdoor Positioning, GPS, Wi-Fi Fingerprint.

ÍNDICE

Índice	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Siglas.....	vii
1. Introdução	1
1.1. Objetivo	1
1.2. O Trabalhador Solitário	2
1.2.1. Definição	2
1.2.2. Perfil do Trabalhador Solitário Acidentado	3
1.2.3. Enquadramento Legal.....	3
1.2.4. Cenários de Trabalho Solitário.....	3
1.2.5. A Avaliação Do Trabalho Solitário.....	4
1.3. Estado da Arte.....	5
1.3.1. Equipamentos Para Situações de Trabalho Solitário.....	5
1.3.2. Aplicações Existentes	6
1.4. Arquitetura do Sistema	7
1.4.1. Comunicação Android-Servidor.....	8
1.4.2. Utilização do Stunnel	9
1.5. Estrutura Multi-Tenant	9
1.5.1. Multi Tenancy	9
1.6. Estrutura Modular	10
1.7. Estrutura do Documento	12
2. Ambiente de Desenvolvimento	13
2.1. Componente Web	13
2.1.1. Bootstrap.....	13
2.1.2. PHP.....	14
2.1.3. Netbeans IDE.....	15
2.2. Componente Móvel.....	16
2.2.1. Eclipse ADT	16
2.2.2. Android.....	17
2.2.3. JAVA.....	19
2.3. Base de Dados.....	20
2.3.1. SQL.....	20
2.3.2. MySQL.....	20
3. Tecnologias Utilizadas	23
3.1. Servidor Web	23
3.1.1. Apache	24
3.2. Atualização automática da aplicação Android.....	25
3.3. WebSockets	25
3.3.1. Protocolo JSON	27
3.3.2. Node.js.....	28
3.4. Mumble.....	35
3.4.1. Mumble Server	35
3.4.1. Mumble Client.....	36
3.5. Localização / Posicionamento	37

3.5.1. Localização em Meios Exteriores	38
3.5.2. Localização em Espaços Interiores	39
3.6. Geofence de dispositivos	42
4. Testes e Resultados.....	45
4.1. Dificuldades Durante o Desenvolvimento	45
4.2. Testes Realizados e Resultados Obtidos.....	46
4.3. Interface gráfica da aplicação	48
5. Conclusão	49
5.1. Trabalho Futuro	50
Referências Bibliográficas.....	53
Anexos.....	57
1. Proposta de Estágio	59
2. Resultados de Testes ao Servidor	61
2.1. Comando ab e respetivos parâmetros para o porto 80	61
2.2. Comando ab e respetivos parâmetros para o porto 3001	62
2.3. Ficheiros Resultantes da execução dos comandos ab	63
2.4. Ficheiro Template a Utilizar na Ferramenta GNUPlot.....	63
2.5. Comando GNUPlot com instrução para utilizar o ficheiro template.....	63
2.6. Ficheiro resultante da execução do comando GNUPlot.....	63
2.7. Conteúdo do ficheiro resultante do processo	64
3. Manual de Utilizador	65
4. Modelo ER da Base de Dados	109

LISTA DE FIGURAS

Figura 1 – Dispositivos Guardian MPS	6
Figura 2 – Software Guardian MPS	6
Figura 3 – Software PSIM – AGORA.....	7
Figura 4 – Diagrama da arquitetura geral do sistema.	8
Figura 5 – Esquema da implementação do servidor e comunicação entre este e os dispositivos Android.....	9
Figura 6 – Esquema da estrutura <i>Multi Tenant</i> da aplicação	10
Figura 7 – Ficheiro ModulesAccess.php com definição para acesso a todos os módulos para o tenant 9	11
Figura 8 – Interface principal da aplicação com acesso a todos os módulos.	11
Figura 9 – Ficheiro ModulesAccess.php com definição para acesso apenas aos módulos core da aplicação para o tenant 2.....	11
Figura 10 – Interface principal da aplicação com acesso apenas aos módulos core da aplicação.	11
Figura 11 - Layout da plataforma web utilizando a <i>framework</i> Bootstrap.....	14
Figura 12 – Estrutura do projeto no IDE Netbeans	16
Figura 13 – Ambiente de desenvolvimento Eclipse ADT.....	17
Figura 14 – Arquitetura Android – Camadas e Componentes	19
Figura 15 – Servidores <i>Web</i> mais utilizados nos últimos anos	23
Figura 16 – Arquitetura Apache <i>Web Seapacherver</i>	24
Figura 17 – Definição do link para ficheiro com dados de update da app	25
Figura 18 – Ficheiro JSON com informação de versão da app.....	25
Figura 19 – Compatibilidade de <i>Websockets</i> e <i>browsers</i> atuais	26
Figura 20 – Comparação de latência entre <i>polling</i> e aplicações <i>Websockets</i>	27
Figura 21 – Protocolo JSON na troca de dados entre cliente-servidor	28
Figura 22 – Módulos utilizados no servidor Node.js	29
Figura 23 – Utilização do módulo Socket.io no servidor Node.js do projeto	30
Figura 24 – Utilização do módulo Express no servidor Node.js do projeto.....	30
Figura 25 – Página de estatística servida pelo módulo express do Node.js	31
Figura 26 – Exemplo da utilização do módulo MySQL no servidor Node.js do projeto....	32
Figura 27 – Comunicação JSON Android-Servidor	32
Figura 28 – Comunicação JSON – Pedido de login na app (App->Servidor).....	33
Figura 29 – Comunicação JSON – Login na app (Resposta do servidor).....	34
Figura 30 – Comunicação JSON – Tratamento da resposta na app (Servidor->App)	34
Figura 31 – Classe PHP responsável pela gestão de servidor Mumble do projeto	36

Figura 32 – Código de ligação do cliente Android ao servidor Mumble	37
Figura 33 – Wi-Fi <i>Fingerprint</i> – Esquema de Arquitetura.....	40
Figura 34 – Pagamento através de NFC	41
Figura 35 – Criação de um edifício através da plataforma web	43
Figura 36 – Criação de uma BoundingBox para Geofence	44
Figura 37 – Verificação se ponto geográfico está dentro da “BoundingBox”	44
Figura 38 – Código de leitura confusa devido ao excesso de callbacks nas funções javascript	45
Figura 39 – Página dedicada à aplicação na secção empresarial do site da Vodafone	50
Figura 40 – Comando ab e respetivos parâmetros para o porto 80 do servidor.	61
Figura 41 – Comando ab e respetivos parâmetros para o porto 3001 do servidor (node.js)62	
Figura 42 – Ficheiros resultantes da execução dos comandos “ab”	63
Figura 43 – Ficheiro template para utilizar na ferramenta GNUPlot	63
Figura 44 – Comando gnuplot com instrução para utilizar o ficheiro template	63
Figura 45 – Ficheiro “.png” resultante da execução do comando gnuplot.....	63
Figura 46 – Conteúdo do ficheiro “.png” resultante do processo.....	64

LISTA DE TABELAS

Tabela 1 – Cenários de enquadramento do trabalhador solitário	4
Tabela 2 – Atividades onde é usual o trabalho solitário	4
Tabela 3 – Empresas clientes da aplicação.....	47

LISTA DE SIGLAS

3G	<i>3(Third) Generation</i>
4G	<i>4(Fourth) Generation</i>
ADT	<i>Android Developer Tools</i>
A-GPS	<i>Assisted GPS</i>
AJAX	<i>Asynchronous Java Script and XML</i>
AP	<i>Access Point</i>
API	<i>Application Programming Interface</i>
APK	<i>Android Application Package</i>
APN	<i>Access Point Name</i>
ASP	<i>Active Server Pages</i>
AVD	<i>Android Virtual Device</i>
BLE	<i>Bluetooth Low Energy</i>
BLS	<i>Bureau of Labor Statistics</i>
CCOHS	<i>Canadian Centre for Occupational Health and Safety</i>
CDN	<i>Content Delivery Network</i>
CGI	<i>Common Gateway Interface</i>
CSS	<i>Cascading Style Sheets</i>
ECMA	<i>European Computer Manufacturers Association</i>
EDGE	<i>Enhanced Data rates for GSM Evolution</i>
ESTGOH	<i>Escola Superior de Tecnologia e Gestão de Oliveira do Hospital</i>
GPL	<i>General Public License</i>
GPS	<i>Global Positioning System</i>
GSM	<i>Global System for Mobile Communications</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
IBM	<i>International Business Machines</i>
ICE	<i>Internet Communications Engine</i>
IDE	<i>Integrated Development Environment</i>
IE	<i>Internet Explorer</i>
IFSEC	<i>International Fire and Security Exhibition and Conference</i>

IMEI	<i>International Mobile Equipment Identity</i>
INRS	<i>Institut National de Recherche et de Sécurité</i>
ISP	<i>Internet Service Provider</i>
JDK	<i>Java Development Kit</i>
JRE	<i>Java Runtime Environment</i>
JSON	<i>Java Script Object Notation</i>
JSP	<i>Java Server Pages</i>
NFC	<i>Near-Field Communication</i>
NPM	<i>Node Package Manager</i>
OHA	<i>Open Handset Alliance</i>
PDA	<i>Personal Digital Assistant</i>
PECL	<i>PHP Extension Community Library</i>
PHP	<i>PHP: Hypertext Preprocessor</i>
PSIM	<i>Physical Security Information Management</i>
QR Code	<i>Quick Response Code</i>
RPC	<i>Remote Procedure Call</i>
RSS	<i>Received Signal Strength</i>
SaaS	<i>Software as a Service</i>
SCGI	<i>Simple Common Gateway Interface</i>
SDK	<i>Software Development Kit</i>
SGBD	<i>Sistema Gestor de Bases de Dados</i>
SMS	<i>Short Message Service</i>
SQL	<i>Structured Query Language</i>
SSI	<i>Server Side Includes</i>
SSL	<i>Secure Sockets Layer</i>
TCP	<i>Transmission Control Protocol</i>
TLS	<i>Transport Layer Security</i>
URL	<i>Uniform Resource Locator</i>
Wi-Fi	<i>Wireless Fidelity</i>
WLAN	<i>Wireless Local Area Network</i>

1. INTRODUÇÃO

A empresa “TulaMove” é um consultor de engenharia global e prestador de serviços em eletrónica e desenvolvimento de *software* para as indústrias aeroespacial, defesa e de fabricação de automóveis. Está sediada em Tallinn (Estónia), e com escritório em Coimbra.

Este estágio teve como principal objetivo, a minha integração num ambiente empresarial aplicando e consolidando conhecimentos adquiridos. Consequentemente resultou numa aquisição de competências, as quais permitem ultrapassar situações complexas, prováveis em futuro contexto profissional.

Relativamente ao tema do projeto – o trabalho solitário – este encontra-se de tal forma disseminado no *modus operandi* da indústria, que de um modo geral passa despercebido. Por este facto, quer a sua definição, o seu enquadramento legal, a sua institucionalização, quer mesmo a sua prevenção, são difíceis de gerir e implementar.

Serve então o presente documento para detalhar o projeto/aplicação realizado assente nesse tema.

1.1. Objetivo

O principal objetivo deste projeto é desenvolver uma aplicação que vá ao encontro da satisfação das principais necessidades dos trabalhadores solitários, fazendo nesta secção uma resenha dos aspetos a considerar na avaliação de risco, das principais atividades, não esquecendo os riscos psicossociais e o modo de proceder em situação de acidente ou de emergência.

Relativamente aos principais objetivos da aplicação propostos pela empresa acolhedora, destacam-se os seguintes:

- Controlo de acesso de utilizadores por *login/password*.
- Controlo de acesso de dispositivos por *IMEI/password*.
- Associação/gestão de dispositivos e utilizadores.
- Registo de utilizadores, dispositivos, trajetos, alarmes e incidentes.
- Visualização de registos de trajetórias e alarmes num mapa Google.
- Manipulação de alarmes pelo utilizador atribuído (e.g. Comentários, validações).
- Registo das posições GPS dos dispositivos/utilizadores.
- Registo de alarmes nas categorias definidas pelo gestor de sistema.
- Detecção de queda através dos sensores (acelerómetros) do smartphone.

- Possibilidade de comunicação através de *push-to-talk* entre utilizadores (*Walkie-Talkie*)

A partir do cumprimento destes, ou seja, da sua implementação prática, foram surgindo novos com finalidade de complementar existentes, colmatar falhas, assim como ampliar funcionalidades da aplicação.

No Anexo 1 encontra-se a proposta de estágio apresentada à escola no início do ano letivo, de onde consta a listagem dos principais objetivos da aplicação.

1.2. O Trabalhador Solitário

1.2.1. Definição

São muitas as definições que podemos encontrar para “trabalhador solitário”. Segundo alguns autores [1], o trabalho realizado de um modo isolado ou quando o trabalhador o realiza sozinho, refere-se a situações em que o trabalhador pode estar exposto a riscos, devido:

- À área em que estão a executar o trabalho ser remota, ou isolada da ajuda dos outros por causa da sua natureza, do tempo ou do local do seu trabalho;
- Ao trabalho envolver operação ou manutenção de instalações perigosas, ou a manipulação de uma substância perigosa;
- Ao trabalho ser perigoso para o trabalhador o realizar sozinho.

Incluem-se nesta definição também, conforme [1] os trabalhadores que trabalham por conta própria na manipulação de substâncias ou mercadorias perigosas e na realização de qualquer outro trabalho que é considerado, com base numa avaliação de risco perigoso para ser realizado por uma pessoa sozinha. A definição utilizada no trabalho desenvolvido com o apoio do INRS [2], caracteriza trabalho solitário como sendo uma tarefa realizada por uma única pessoa num ambiente de trabalho em que não podem ser vistas ou ouvidas por outros, e em que a probabilidade de acesso é baixa. Notar também que trabalhar sozinho é depender apenas de si em caso de situações problemáticas e perigosas, sendo que situações como esta poderão também promover a sensação de solidão. Com isto, concluímos que o trabalho solitário não é apenas uma questão física e pode ser acompanhada por uma reação de isolamento psíquico [2].

Segundo o *CCOHS*, um trabalhador está "só" quando se encontra por conta própria, quando não pode ser visto ou ouvido por outra pessoa, e quando não pode esperar uma visita de outro trabalhador. Trabalhar sozinho inclui a situação em que os funcionários podem passar por um

período de tempo em que não tenham contato direto com um colega de trabalho. Por exemplo, a rececionista de um grande edifício de escritórios pode ser considerada um trabalhador "solitário". Por outro lado, um trabalhador da construção civil, que está a fazer um trabalho num qualquer local em que não pode ser visto pelos colegas de trabalho também pode ser considerado um trabalhador solitário [3].

De acordo com o BLS 2000, “trabalhadores de serviços sociais no sector público, como assistentes sociais, são cerca de sete vezes mais propensos a serem vítimas de ataques violentos no trabalho do que os trabalhadores do sector privado” [3].

Com o aumento da ameaça de violência e probabilidade de ocorrência de incidentes no local de trabalho, nunca foi tão importante garantir a segurança deste tipo de trabalhadores [3].

1.2.2. Perfil do Trabalhador Solitário Acidentado

Segundo alguns autores [2], os trabalhadores solitários que sofrem um acidente, possuem um perfil comum caracterizado pelo facto de serem em 96% das situações do género masculino (contra 94% para os não solitários) e trabalharem nos ramos de atividade da indústria extrativa, cerca de 34,5%, dos quais 17,3% na exploração de pedreiras e os restantes em minas, metalomecânica (14%), condutores de máquinas (35,2%) salientando-se de entre estes em particular na condução de meios de transporte (20,2%) e cerca de 16,6% são operadores não qualificados [2].

1.2.3. Enquadramento Legal

É um vazio legal o que rodeia a situação particular do trabalho desenvolvido de modo solitário. A Lei 102/2009 de 10 de setembro, que regulamenta o regime jurídico da promoção e prevenção da segurança e da saúde no trabalho, de acordo com o previsto no artigo 284.º do Código do Trabalho, não foca de modo específico o trabalho solitário. Podendo o trabalho solitário ser executado e desenvolvido por trabalhadores por conta de outrem ou trabalhando por conta própria, em diversos ramos de atividade, esta situação leva por vezes a contextos de risco acrescido [2].

1.2.4. Cenários de Trabalho Solitário

Em qualquer atividade poderá ser encontrada uma situação de trabalho solitário, o que nos leva a concluir que esta situação de trabalho é transversal a todos os setores. No entanto, nalguns deles, esta situação de trabalho poderá ocorrer com maior frequência e necessitar de uma avaliação mais detalhada. O trabalho solitário poderá aparecer em vários cenários, como por exemplo os apresentados abaixo na Tabela 1 [4].

Tabela 1 – Cenários de enquadramento do trabalhador solitário

Sozinho num local fixo, como uma loja ou uma bomba de gasolina
Separado dos colegas por longos períodos, por exemplo numa fábrica, armazém, num centro de lazer
Fora do horário normal, executando trabalhos de limpeza ou vigilância
Em posições com mobilidade, em setores como a construção, manutenção, engenharia ou agricultura
Com outras funções móveis, visitando casas ou empresas, tais como motoristas, enfermeiros, assistentes
Ao efetuar trabalhos em espaços confinados
Trabalho remoto

Na Tabela 2 apresentam-se exemplos de atividades profissionais onde o trabalho solitário é usual, quer se trate de trabalho desenvolvido por conta de outrem ou por conta própria.

Tabela 2 – Atividades onde é usual o trabalho solitário [5]

Segurança privada	Comércio, por exemplo: oficinas e bombas de combustível
Manutenção e reparação	Estâncias de SKI
Serviços de limpeza	Condução de máquinas por exemplo na limpeza de florestas
Trabalhos com energia elétrica	Pastores de gado

1.2.5. A Avaliação Do Trabalho Solitário

O trabalho solitário por si só não constitui um risco mas apresenta-se como um fator agravante dos riscos, em particular em caso de acidente [6]. O empregador para dar cumprimento às suas obrigações gerais deverá evitar esta situação de trabalho. Não sendo possível evitar o trabalho solitário deverão ser tomadas as medidas de prevenção de riscos profissionais e organizar o alerta, socorro e o resgate em caso de acidente. As medidas a adotar deverão ser adaptadas aos setores de atividades em questão e às situações específicas de trabalho. A sensação de isolamento, sendo um dos principais riscos associados ao trabalho solitário, assume um particular destaque como risco psicossocial. Tendo presente que a avaliação de riscos constitui um dos princípios de prevenção, que não se encontra definida no quadro legislativo atual em matéria de segurança e saúde no

trabalho qualquer referência específica a trabalho solitário leva-nos a concluir que este é um fator subestimado na prevenção.

1.3. Estado da Arte

1.3.1. Equipamentos Para Situações de Trabalho Solitário

Atualmente, as novas tecnologias permitem-nos dispor de equipamentos de segurança para todos os tipos de situações de trabalho solitário, incluindo em ambientes perigosos ou com atmosferas explosivas. O recurso a soluções tecnológicas para alarme e localização de trabalhadores solitários em risco, pode inclusivamente ser uma solução mais económica do que as soluções tradicionais de duplicação de funcionários em todas as tarefas isoladas (em caso de intoxicação por gás o trabalho aos pares pode duplicar o número de vítimas e não resolve o problema).

Existe um leque variado de equipamentos no mercado, desde os alarmes portáteis sonoros ou com envio de sinal para recetor, acionados por botão premido pelo trabalhador, até alarmes automáticos acionados por movimentos bruscos (associados a pânico) ou por tempo de imobilização considerado prolongado (inatividade do trabalhador devido a queda ou mal-estar súbito) [7].

Como exemplo de soluções tecnológicas temos a colocação de balizas ou postos de controlo de passagem, que necessitam de acionamento de chave pelo trabalhador ou aproximação de cartão magnético, a colocação de câmaras de vigilância monitorizadas (exige outro trabalhador a observar os monitores, embora possa monitorizar vários trabalhadores solitários), botoneiras de emergência fixas no local do trabalho solitário, rádios portáteis ou telemóveis utilizados para controlos periódicos [2].

Atualmente, os sistemas mais avançados são os equipamentos de localização pessoal portáteis, de pequena dimensão e transportados pelo trabalhador que permitem o alarme de emergência por botão para o 112 ou qualquer outro número de telefone pré-definido. Estes equipamentos transmitem as coordenadas GPS permitindo saber a localização permanente do utilizador, escutar o som ambiente do local ou comunicar por voz. Os referidos alarmes automáticos possuem tecnologia de posicionamento e leitura de movimentação que possibilitem detetar imobilizações ou movimentos bruscos anómalos. Estes sistemas portáteis estão preparados para resistir a pó, água e quedas em pavimentos duros e no caso de não existir rede GSM (e.g. no interior de edifícios), podem ser ligados por rádio a retransmissores que efetuam a ligação para o exterior [7].

1.3.2. Aplicações Existentes

Para uma maior familiarização com aplicações do género, após algumas pesquisas, foram analisadas as seguintes:

- Guardian MPS – *Mobile Personal Safety*

Como o vencedor do “Prémio de Produto do Ano” IFSEC em 2013, o Guardian MPS é um líder de mercado na área das Soluções Móveis de Segurança Pessoal. Este pode ser utilizado em dispositivos como BlackBerry, iPhone, Android e PDA, como ilustrado abaixo na Figura 1 para realizar tarefas tais como as representadas na Figura 2 [8].



Figura 1 – Dispositivos Guardian MPS

As principais funcionalidades do sistema Guardian são [8]:

- Login de utilizadores;
- Alarmes de Pânico;
- Localizações GPS;
- Durações estimadas de rondas;
- Gerar relatórios de ocorrências e ou atividades;



Figura 2 – Software Guardian MPS [8]

- AGORA – PSIM Made Easy

O AGORA [9], é um *software* PSIM que permite às empresas de segurança privada gerar novas receitas e às empresas descentralizadas reduzir custos operacionais através da centralização das operações de segurança com serviços remotos inovadores e personalizados baseados em vídeo, áudio e GPS.

As principais funcionalidades do Agora são [9]:

- Ronda Remota;
- Portaria Remota;
- Escolta Remota;
- Vídeo-Verificação de Alarmes;
- Supervisão do Controle de Acessos;
- Proteção de “Lone Worker”.

O principal objetivo da análise a estas aplicações foi a visualização e interiorização do conceito de monitorização e da forma como normalmente esta é realizada, assim como a forma de integração das diversas funcionalidades da aplicação, ilustrado abaixo na Figura 3.



Figura 3 – Software PSIM – AGORA [9]

Com isto foram surgindo novas ideias relativas à organização de funcionalidades e disposição de elementos, entre outros aspetos, tendo como resultado uma aplicação mais intuitiva e tendo uma melhor interação e facilidade de utilização por parte dos utilizadores.

1.4. Arquitetura do Sistema

Com o principal objetivo deste trabalho pretendeu-se desenvolver uma aplicação a qual tem como base uma comunicação quase constante entre utilizador (Android) e Servidor (web).

Neste contexto, deverão ser utilizados os sensores e mecanismos disponíveis no dispositivo móvel para recolher informações que representem o ambiente de utilização em que o dispositivo está inserido.

Essas informações deverão ser formatadas de acordo com normas e protocolos existentes e enviadas para um servidor sempre assim se justifique.

O servidor interpretará as informações recebidas e atuará da forma adequada para que o conteúdo seja entregue ao utilizador sob a forma que melhor se adequa às condições de utilização com vista a maximizar a qualidade de experiência do utilizador.

O sistema proposto para a realização do estudo é composto por 3 principais componentes, sendo estes: o servidor, a base de dados e o cliente – Android ou web, como podemos ver representado na Figura 4.

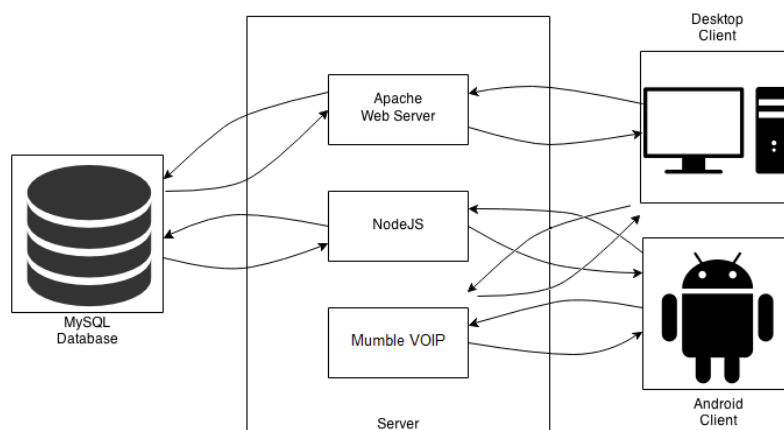


Figura 4 – Diagrama da arquitetura geral do sistema.

1.4.1. Comunicação Android-Servidor

Na utilização da aplicação móvel, é necessária uma constante ligação entre o *smartphone* e o servidor para troca de dados entre ambos.

Esta comunicação, cuja implementação se encontra representada na Figura 5, baseia-se na troca de informação entre servidor e dispositivo móvel Android utilizando o protocolo JSON para o efeito. Esta assenta numa estrutura baseada no servidor Node.js, servidor este que é descrito no capítulo 3.3.2 do presente documento.

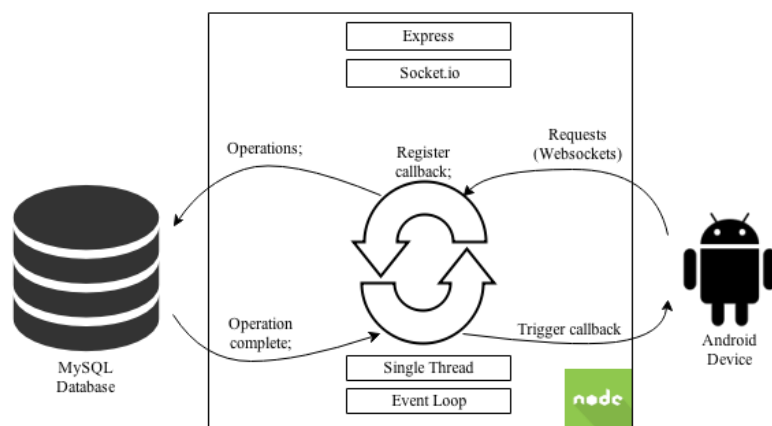


Figura 5 – Esquema da implementação do servidor e comunicação entre este e os dispositivos Android

1.4.2. Utilização do Stunnel

O Stunnel [10] é um software livre distribuído com uma licença GNU GPL v2. Ele é um proxy desenhado para adicionar a funcionalidade de encriptação TLS entre clientes e servidor sem haver necessidade de alterações no código da aplicação. A sua arquitetura é otimizada para segurança, portabilidade e escalabilidade (incluindo balanceamento de carga), tornando-o assim adequado para aplicações de grande escala.

No contexto da aplicação, a utilização do stunnel é uma mais-valia tendo em conta que além de assegurar que toda a troca de informação entre clientes e servidor se realize num processo completamente seguro, sendo encriptada e encapsulada utilizando um túnel (protocolo TLS) também veio resolver o problema relativo ao *proxy* aplicado pela rede móvel, explicado abaixo na secção 4.1-Dificuldades Durante o Desenvolvimento.

1.5. Estrutura Multi-Tenant

Devido à necessidade da aplicação servir diferentes clientes (empresas) em simultâneo, e tendo em conta que cada um destes não tem conhecimento da existência dos restantes, foi adotado um modelo de arquitetura *Multi Tenant* [11].

1.5.1. Multi Tenancy

Tenant é o termo inglês que significa inquilino. No contexto da aplicação o cliente é dono dos seus dados, enquanto que o provedor é o dono do software em si [11].

Um serviço é dito *Multi Tenant* quando uma única instância ou instalação é capaz de atender vários clientes (*Tenants*) simultaneamente, ao invés de ter uma instância/instalação para cada um destes [11].

O que define uma aplicação *Multi Tenant*, portanto, é a capacidade que uma única instância de uma aplicação tem de servir diferentes *Tenants* e respetivos utilizadores, segregando os dados desses *Tenants* de maneira que um utilizador de um *Tenant* não possa ver ou alterar os dados de outros *Tenants*. A Figura 6 mostra em alto nível a estrutura de uma aplicação *Multi Tenant* [11].

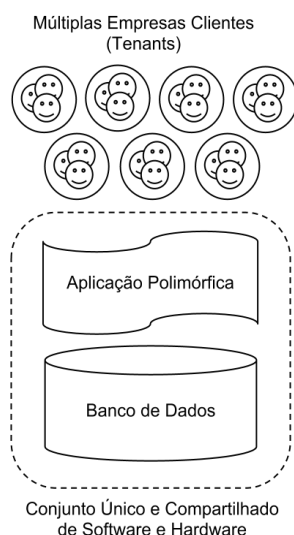


Figura 6 – Esquema da estrutura *Multi Tenant* da aplicação [11]

1.6. Estrutura Modular

Tendo em conta que o *software* tem característica multi tenant, como visto anteriormente, acontece que cada tenant tem as suas necessidades distintas. Para responder individualmente a estas necessidades foi adotada uma estrutura de módulos. Sendo um módulo um componente que utiliza a mesma arquitetura tecnológica do sistema, responsável por atividades que satisfazem um assunto bem definido, estas atividades utilizam tarefas e componentes comuns do sistema. O sistema é então constituído por um *core* - conjunto de funcionalidades básicas e por um ou vários módulos.

No contexto da aplicação, existem casos onde um determinado tenant tem acesso a funcionalidades (módulos) distintas de outro tenant, como podemos verificar nas seguintes figuras (Figura 7, Figura 8, Figura 9 e Figura 10). Este controlo de acesso a módulos por parte dos tenants é realizado na parte de gestão da aplicação, onde o *provider* define num ficheiro (*ModulesAccess.php*) quais os módulos a que cada tenant tem acesso.

```
9 => array( //TULA
    SA_MODULE_ALL
),
```

Figura 7 – Ficheiro ModulesAccess.php com definição para acesso a todos os módulos para o tenant 9

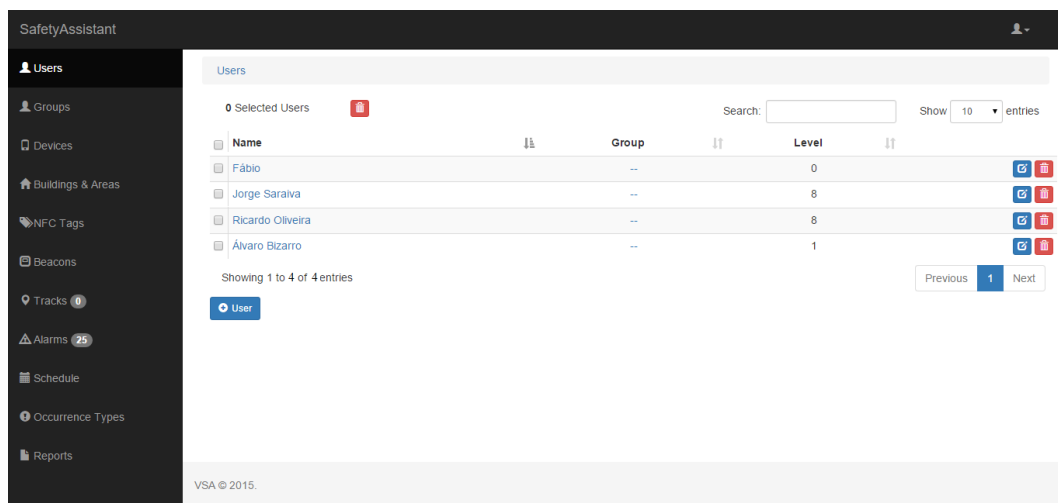


Figura 8 – Interface principal da aplicação com acesso a todos os módulos.

```
2 => array( //PROSEGUR
    SA_MODULE_USERS,
    SA_MODULE_GROUPS,
    SA_MODULE_DEVICES,
    SA_MODULE_BUILDINGS,
    //SA_MODULE_NFC_TAGS,
    //SA_MODULE_BEACONS,
    SA_MODULE_TRACKS,
    SA_MODULE_ALARMS,
    //SA_MODULE_SCHEDULE,
    //SA_MODULE_OCCURRENCE_TYPES,
    //SA_MODULE_REPORTS,
    //SA_MODULE_MOBILE_REPORTS,
    //SA_MODULE_SHIFTS_OCCURRENCES
),
```

Figura 9 – Ficheiro ModulesAccess.php com definição para acesso apenas aos módulos core da aplicação para o tenant 2.

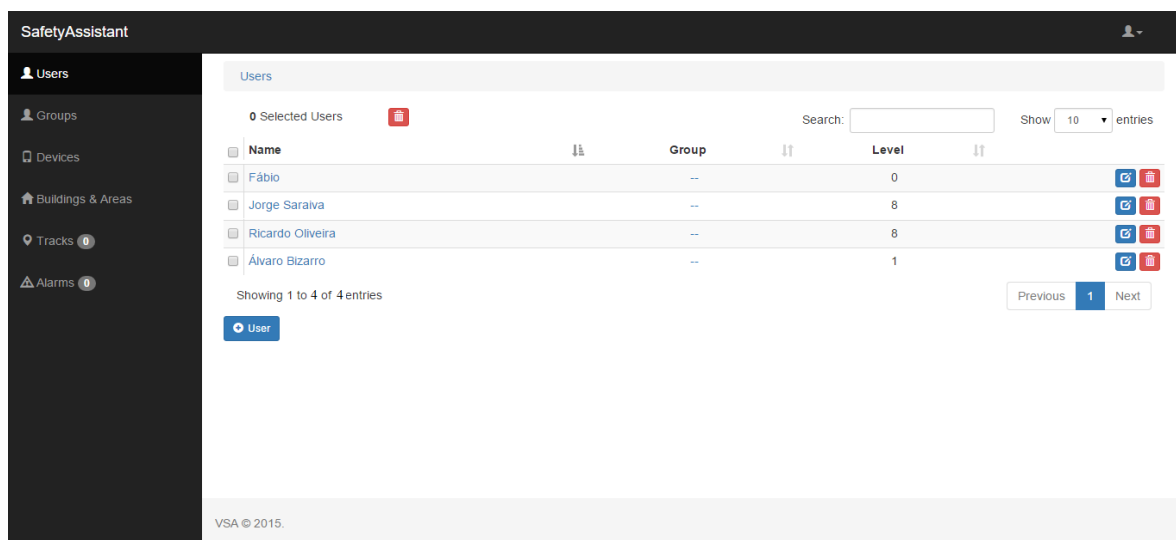


Figura 10 – Interface principal da aplicação com acesso apenas aos módulos core da aplicação.

1.7. Estrutura do Documento

Este documento é composto por 5 capítulos e uma secção de anexos.

O capítulo 1, no qual se insere este texto, encontra-se composto pelo enquadramento do conceito do projeto, definições dos seus pontos base, os seus objetivos, o estado da arte, e a composição da arquitetura do sistema. No 2º capítulo descreve-se o ambiente de desenvolvimento do projeto, dando enfoque nas plataformas de desenvolvimento utilizadas, relativas ao desenvolvimento das componentes web e móvel e também na base de dados. No 3º capítulo identifico as tecnologias utilizadas para o desenvolvimento do trabalho. O 4º capítulo contém uma descrição dos principais problemas encontrados no processo de desenvolvimento do sistema, assim como testes realizados e respetivos resultados obtidos. No 5º e último capítulo é apresentada uma conclusão acerca do projeto e também os pontos que ficam em aberto para trabalho futuro.

Na secção de anexos, estão disponíveis a proposta de projeto, o diagrama de classes do sistema e o modelo ER da base de dados utilizada no projeto.

2. AMBIENTE DE DESENVOLVIMENTO

Para o desenvolvimento do trabalho foram estudadas e utilizadas várias tecnologias, nomeadamente um SDK Android, as linguagens de programação Java e PHP, a linguagem de consulta estruturada SQL, o SGBD MySQL e dois IDE's – *Eclipse* e *Netbeans* como ambientes de desenvolvimento.

O Android é uma plataforma recente e muito prometedora da qual eu tinha experiência apenas da parte lecionada durante o curso de Mestrado de Informática Aplicada, estes conhecimentos adquiridos, permitiram uma maior familiarização com o funcionamento desta tecnologia, tornando assim mais fácil a assimilação e aplicação de novos conceitos que não foram abordados durante o percurso académico.

Aqui faço uma breve referência às principais tecnologias utilizadas neste projeto no que diz respeito ao ambiente de desenvolvimento do projeto.

2.1. Componente Web

2.1.1. Bootstrap

Desenvolvida pela equipa do *Twitter*, o *Bootstrap* é uma *framework front-end opensource*. Ou seja, é um conjunto de ferramentas criadas para facilitar o desenvolvimento de *sites* e aplicações web [12].

Compatível com HTML5 e CSS3, esta *framework* possibilita a criação de *layouts* responsivos, permitindo que seu conteúdo seja organizado em até 12 colunas e que se comporte de maneira diferente para cada resolução [12].

As principais vantagens que levaram à decisão da utilização desta *framework* no projeto foram as seguintes:

- Fácil de utilização;
- Possui documentação detalhada;
- Otimizada para o desenvolvimento de *layouts* responsivos;
- Possui componentes suficientes para o desenvolvimento com interface simples;
- Facilita a criação e edição de *layouts*;
- Funciona em todos os navegadores atuais (Chrome, Safari, Firefox, IE, Opera).

Apesar das muitas vantagens acima listadas, esta *framework*, tem dois pontos menos positivos que são os seguintes:

- O programador terá que ter um conhecimento prévio da *framework* para seguir os padrões de desenvolvimento do Bootstrap;
- Caso seja utilizado um tema padrão e comum do Bootstrap sem serem feitos ajustes visuais (e.g. elementos, cores, disposições, etc...), o projeto vai ser parecido – em termos visuais – com outros que também utilizam o Bootstrap.

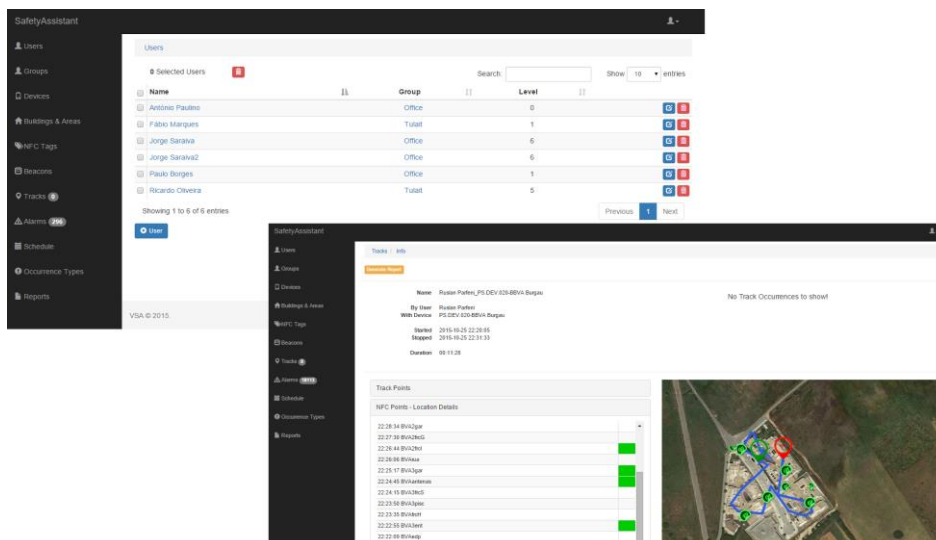


Figura 11 - Layout da plataforma web utilizando a *framework* Bootstrap.

2.1.2. PHP

O PHP é uma linguagem de programação interpretada, livre e muito utilizada para gerar conteúdo dinâmico na Web [13].

Esta é uma poderosa linguagem orientada a objetos, sendo extremamente modularizada, o que a torna ideal para instalação e uso em servidores web. Diversos módulos são criados no repositório “*PHP Extension Community Library*” (PECL) e alguns destes módulos são introduzidos como padrão em novas versões da linguagem [13].

É muito parecida, em tipos de dados, sintaxe e mesmo funções, com a linguagem C e com a C++ dependendo da configuração do servidor. Existem versões do PHP disponíveis para os seguintes sistemas operativos: Windows, Linux, FreeBSD, Mac OS, OS/2, AS/400, Novell Netware, RISC OS, IRIX e Solaris [13].

Construir uma página dinâmica baseada em Bases de Dados é simples com o PHP, devido ao amplo suporte que este oferece para diferentes tipos Bases de Dados.

As principais características do php em geral são:

- Licença gratuita;
- Velocidade, robustez e eficiência;
- Métodos de segurança eficientes;
- Estabilidade e pouco consumo de recursos de hardware no servidor;
- Estruturada e orientada a objetos;
- Portabilidade - independência de plataforma – escreve-se uma vez, e pode executar-se em qualquer lugar;
- Sintaxe similar a Linguagem C/C++ e o PERL.

Até a versão 4 o PHP não suportava todas as características que uma linguagem de programação orientada à objeto deve conter. Na sua versão 5, o PHP implementou o conceito de programação orientada a objetos, o que fez certamente com que muitos programadores vissem o PHP com outros olhos [13].

2.1.3. Netbeans IDE

Após o conhecimento das características do projeto e a definição da linguagem de programação, o passo seguinte foi a escolha do IDE que seria utilizado como base para o desenvolvimento da componente web deste projeto.

O IDE NetBeans é baseado na tecnologia Java, tendo como fundadora (ano 2000) e principal patrocinadora a empresa *Sun Microsystems*. Este oferece todas as funcionalidades necessárias para a edição de código fonte, além de disponibilizar inúmeras outras ferramentas, a exemplo de: formatação de código, dicas e sugestões de códigos automáticos, compilação, depuração, criador visual de formulários, suporte à internacionalização, suporte a diversas linguagens de programação, entre outros. Além disso, podemos ainda integrar novas funcionalidades ao ambiente, através da utilização de *plugins* [14].

Sabendo que existem inúmeros IDEs similares a este, a opção pelo Netbeans deve-se maioritariamente à familiaridade e experiências anteriores com o mesmo.

A Figura 12 ilustra a estrutura das componentes (VSA-Web e VSA-NodeServer) do projeto no IDE Netbeans

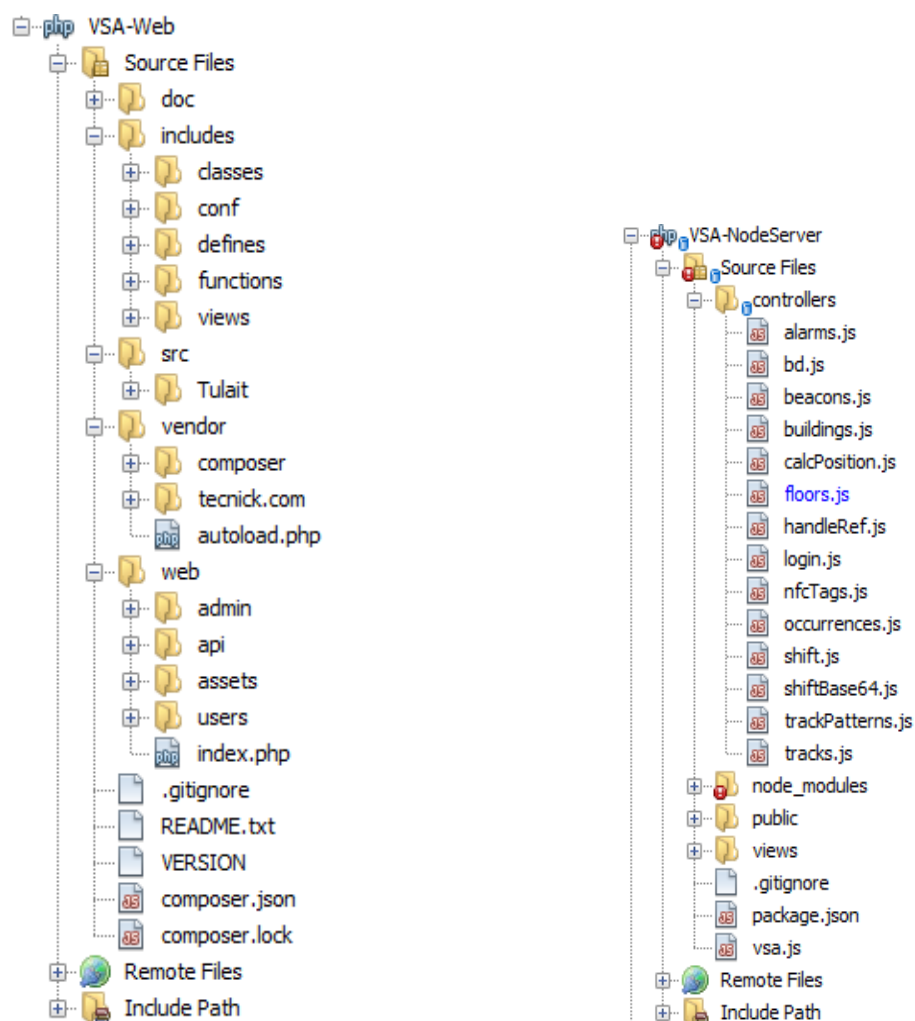


Figura 12 – Estrutura do projeto no IDE Netbeans

2.2. Componente Móvel

2.2.1. Eclipse ADT

Eclipse é uma comunidade *open-source*, cujos projetos são focados em construir uma plataforma de desenvolvimento aberta extensível composto por *frameworks*, ferramentas e *runtimes* para a construção e gestão de *software* em todo o ciclo de vida [15].

A Eclipse Foundation é uma organização sem fins lucrativos que hospeda projetos Eclipse e ajuda a cultivar tanto a comunidade *open-source* e um ecossistema de produtos e serviços complementares [15].

O Eclipse é um dos IDE mais utilizados no desenvolvimento de aplicações Java. Temos pois um editor, compilador, debugger, referência entre outras funcionalidades [16].

Para o desenvolvimento para plataforma Android dispomos de um *Plugin* para inserir no Eclipse, o Android Development Tools (ADT), ilustrado na Figura 13, que é projetado para dar-lhe um ambiente poderoso, integrado, no qual é possível a criação de aplicativos do Android [16].

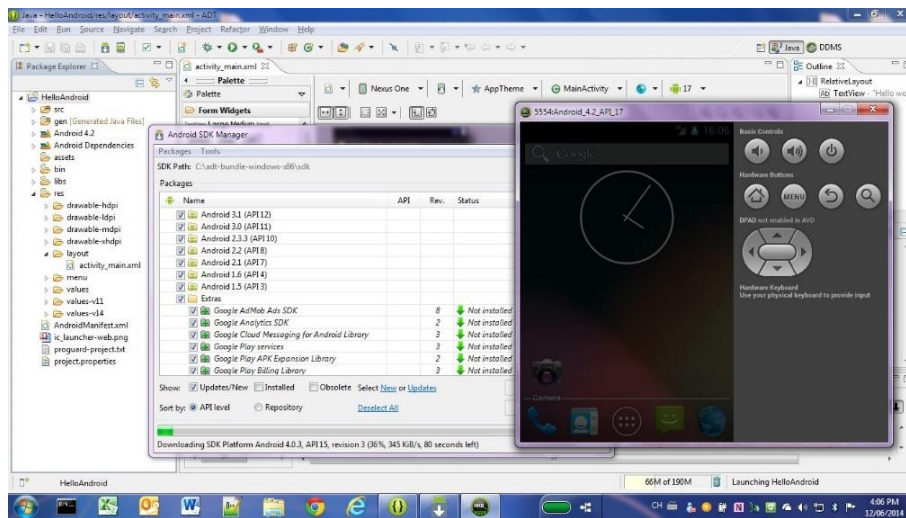


Figura 13 – Ambiente de desenvolvimento Eclipse ADT

2.2.2. Android

Android é um sistema operativo *open-source* com a maior parte do seu código sob uma licença para *software* livre de autoria da *Apache Software Foundation*, este é baseado numa versão modificada do *kernel* do Linux e concebido para dispositivos móveis, em particular para *smartphones*.

Este sistema foi inicialmente desenvolvido por uma *startup* homónima – Android, Inc. Sendo esta pequena empresa posteriormente adquirida pela Google em 2005 como uma estratégia para entrar no mercado móvel, amadurecendo este projeto e tornando-o público em meados de 2007 com o objetivo de apresentar a primeira plataforma *open-source* de desenvolvimento para dispositivos móveis. Atualmente o Android é mantido por um grupo denominado por *Open Handset Alliance* (OHA) que é formado por mais de 40 empresas das quais figuram o próprio Google e outras de importância nos ramos da comunicação, produção de semicondutores (Intel) e fabricação de telemóveis (Motorola), entre outras.

As aplicações Android são escritas em linguagem Java, estas são compiladas em *bytecodes Dalvik* e executadas numa máquina virtual desenvolvida especialmente para utilização em dispositivos móveis denominada Máquina Virtual Dalvik.

A Google disponibiliza o *Android Software Development Kit* (SDK) que proporciona as API's e ferramentas necessárias para o desenvolvimento de aplicações.

2.2.2.1. Android SDK

O Android SDK inclui documentação, código e utilitários para que os programadores consigam desenvolver as suas aplicações de acordo com um padrão de desenvolvimento para o sistema operativo em questão [17].

Neste temos à disposição como principais recursos os seguintes [17]:

- *Application Framework* que proporciona a reutilização de componentes;
- *Dalvik Virtual Machine* que é otimizada para dispositivos móveis;
- Um *browser* integrado baseado no *webkit engine*;
- Gráficos otimizados através de utilização de bibliotecas 2D e 3D;
- *SQLite* para armazenamento de bases de dados estruturadas;
- Suporte multimédia para áudio, vídeo e formatos de imagem;
- Ambiente para de desenvolvimento rico, apresentando emulador de dispositivo, ferramentas de depuração, memória e desempenho.

Os seguintes recursos também estão presentes, porém dependentes de *hardware*:

- Telefone GSM;
- Bluetooth;
- EDGE, 3G, 4G;
- *Wi-Fi*;
- Câmara;
- GPS;
- Compasso, e Acelerómetro.

2.2.2.2. Arquitetura do Android

A arquitetura Android é constituída por várias camadas de componentes como podemos ver na Figura 14. Estas camadas são: núcleo do sistema operativo, bibliotecas, *runtime*, *framework* e aplicações.



Figura 14 – Arquitetura Android – Camadas e Componentes [18].

2.2.3. JAVA

Java é uma linguagem de programação assim como uma plataforma, lançada pela Sun Microsystems em 1995 [19].

É executada em mais de 850 milhões de computadores pessoais no mundo todo, e em milhões de outros dispositivos, incluindo dispositivos móveis e TV [20].

Java é uma linguagem de programação que permite a criação de programas em várias plataformas sem modificação [21].

Contrariamente ao que acontece com as linguagens de programação convencionais, os programas escritos em Java são compilados num código máquina virtual nativo (*bytecode*) que é posteriormente (aquando da execução do programa) transformado em código máquina real sendo então executado por uma máquina virtual [21].

Esta linguagem é utilizada como sendo a linguagem preferencial na programação de plataformas Android. Para o desenvolvimento em JAVA, temos disponível um *Runtime* (JRE) e um *kit* de desenvolvimento (JDK) [22].

2.3. Base de Dados

2.3.1. SQL

Desenvolvida no início dos anos 70 nos laboratórios da IBM, a linguagem SQL é a linguagem padrão para pesquisas em bases de dados relacionais. Isto decorre da sua simplicidade e facilidade de utilização [23].

Esta torna-se diferente de outras linguagens de consulta a bases de dados no sentido em que uma consulta SQL especifica a forma do resultado e não o caminho para chegar a ele. Ela segue um paradigma de programação baseado em programação lógica e funcional contrariamente a outras linguagens procedimentais. Isto reduz a curva de aprendizagem daqueles que se iniciam na linguagem [23].

Permite que os utilizadores definam os dados a serem armazenados na base de dados. Permite também que dados das tabelas sejam acedidos através de consultas e/ou atualizados. Além disso, SQL proporciona facilidades para definição de vistas na base de dados, para a criação e remoção de índices que representam as relações, e para utilização dentro das diferentes linguagens, de forma a permitir o acesso aos dados a partir das aplicações [23].

2.3.2. MySQL

O MySQL é um Sistema Gestor de Bases de Dados (SGBD), que utiliza a linguagem SQL como interface. É atualmente uma das bases de dados mais populares, com mais de 10 milhões de instalações por todo o mundo [24].

O sucesso do MySQL deve-se em grande medida à fácil integração com o PHP incluído, quase que obrigatoriamente, nos pacotes de *host* de *sites* da Internet oferecidos atualmente. Empresas como a Yahoo! Finance, MP3.com, Motorola, NASA, Silicon Graphics e Texas Instruments utilizam o MySQL em aplicações de extrema importância [24].

A principal vantagem do MySQL é o facto de ser *open-source* e funcionar num grande número de sistemas operativos: Windows, Linux, FreeBSD, Solaris, Mac OS X, etc... [25].

As suas principais características são as seguintes [24]:

- Portabilidade (suporta praticamente qualquer plataforma atual);
- Compatibilidade (existem drivers ODBC, JDBC e .NET e módulos de interface para as mais diversas linguagens de programação, como Delphi, Java, C/C++, Python, Perl, PHP, ASP e Ruby);

- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de *hardware*;
- Facilidade na utilização;
- É um Software Livre com base na GPL;
- Pode utilizar vários *Storage Engines* como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid;
- Suporta controlo transacional;
- Suporta *Triggers*;
- Suporta *Cursors*;
- Suporta *Stored Procedures* e *Functions*;
- Replicação facilmente configurável;
- Interfaces gráficas (*MySQL Toolkit*) de fácil utilização cedidos pela MySQL Inc.

3. TECNOLOGIAS UTILIZADAS

Como seguimento do capítulo anterior, neste capítulo são descritas as principais tecnologias utilizadas/integradas na aplicação.

3.1. Servidor Web

Um servidor Web é um programa que é responsável por aceitar pedidos HTTP de clientes Web, na sua maioria navegadores Web, e enviam respostas HTTP que normalmente são páginas Web como documentos HTML e objetos como imagens ou outros ficheiros multimédia [26].

Na prática muitos servidores Web possuem também as seguintes características [26]:

- Autenticação (nome de utilizador e *password*) antes de permitir o acesso aos seus recursos;
- Manipulação de conteúdos estáticos (ficheiros armazenados no sistema de ficheiros do servidor) e conteúdos dinâmicos (SSI, CGI, SCGI, FastCGI, JSP, PHP, ASP, ASP. NET, etc...);
- Suporte para HTTPS (SSL ou TLS) para permitir ligações seguras (cifradas) ao servidor na porta 443 em vez da comum porta 80;
- Compressão dos conteúdos para reduzir o tamanho das respostas (para reduzir o consumo da largura de banda, etc.);
- *Virtual Hosting* para servir várias páginas Web usando apenas um endereço IP;
- Otimização da largura de banda para limitar a velocidade das respostas para não saturar a rede e ser capaz de servir um maior número de clientes.

Na Figura 15 encontra-se uma lista dos servidores Web mais usados ao longo dos últimos anos. Estes valores foram publicados num questionário realizado pela *Netcraft* [27].

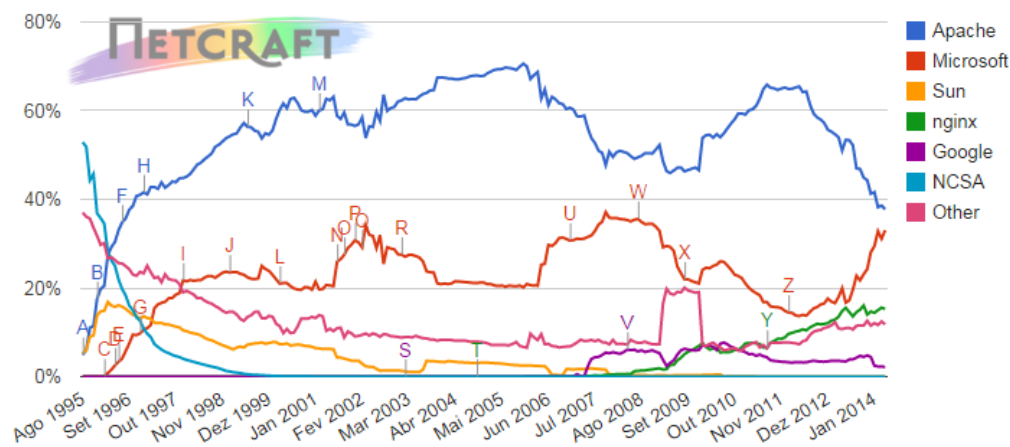


Figura 15 – Servidores Web mais utilizados nos últimos anos [27]

Existem diversos servidores Web disponíveis, sendo alguns deles específicos para algumas situações. Com isto, o facto de um servidor Web não ser muito popular não significa à partida que ele possua muitos bugs ou tenha fraco desempenho.

3.1.1. Apache

Como o projeto está assente numa plataforma Web, existiu a necessidade de escolher então um servidor HTTP. A minha escolha recaiu sobre o servidor Web Apache pela sua robustez, por ser *open source*, por ter a possibilidade de inserir programas externos para manipulação de requisições e também por este ser o servidor HTTP mais utilizado no mundo como mostra a Figura 15 representada acima [28].

Arquitetura do Apache

Em termos de arquitetura, salienta-se que se trata de uma arquitetura modular. Temos duas componentes principais: core ou núcleo e os módulos [28].

- O core é responsável pela implementação das funções básicas despoletando *threads*, processos;
- Os módulos acrescentam ou alteram funcionalidades já existentes no servidor (Figura 16).

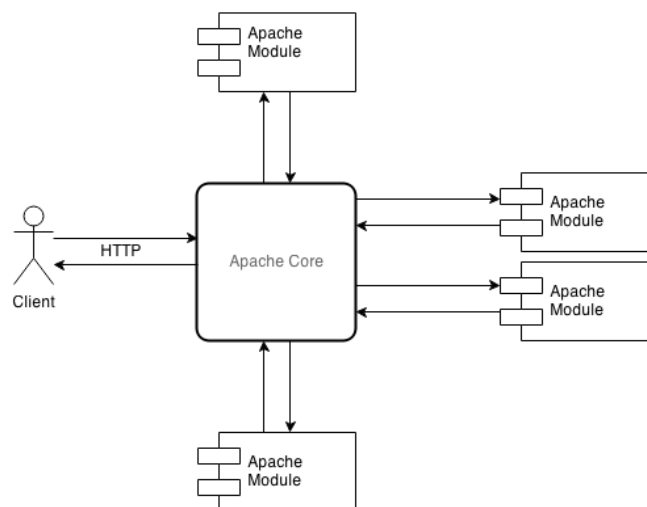


Figura 16 – Arquitetura Apache Web Seapacherver [28]

Os módulos podem ser criados por utilizadores uma vez que o servidor Apache disponibiliza uma API para esse efeito. Exemplificando, o caso desta aplicação em concreto em que foi utilizada a linguagem de programação PHP. Esta linguagem de programação, assim como outras, é adicionada/compilada como um módulo, para que o servidor a possa entender/interpretar [28].

3.2. Atualização automática da aplicação Android

Uma vez que não existiu interesse em colocar a aplicação no Google Play, para que o cliente a tenha sempre atualizada, foi necessário implementar nela um sistema de atualizações automáticas. Para isso foi utilizada uma biblioteca “Updater” disponibilizada no GitHub [29]. Esta biblioteca foi desenhada para permitir auto atualizações da aplicação através da análise de novas versões e *download* do respetivo APK.

A biblioteca utiliza um conjunto de estratégias para permitir a sua integração nas nossas implementações [29]:

1. Uma para determinar se existe alguma atualização disponível;
2. Uma para fazer *download* da atualização;
3. Uma para confirmar se a atualização deve ser descarregada ou instalada (caso já se encontre descarregada).

No caso da aplicação, a verificação de updates é feita após o login, isto implica que sempre que cada utilizador faça um login na aplicação, seja verificada a existência de novas atualizações. Esta verificação de atualizações consiste num pedido HTTP para um link (Figura 17) no servidor onde se encontra um ficheiro no formato JSON com a informação acerca da versão atual assim como a sua localização de rede (link). Este ficheiro está ilustrado abaixo na Figura 18.

```
/**
 * UPDATES URL      </p> Application Version Check </p> See MainActivity class
 */
public static final String APK_URL = "http://cloud01.tulait.eu/vsa-updates/mandown.json";
```

Figura 17 – Definição do link para ficheiro com dados de update da app

```
{
  "versionCode": 9,
  "updateURL": "http://cloud01.tulait.eu/vsa-updates/vsa-apk/TulaItSafetyAssistant_v9.apk"
}
```

Figura 18 – Ficheiro JSON com informação de versão da app

3.3. WebSockets

A criação de aplicações web tem sido efetuada com base no mecanismo pedido/resposta do protocolo HTTP, entre cliente e servidor. Depois da informação de uma página HTTP ser carregada é necessário que exista uma interação do utilizador, como por exemplo, abrir outra página, ou atualizar manualmente a página atual para que haja uma atualização da informação que é mostrada. Alguns destes problemas vieram a ser resolvidos com o aparecimento do AJAX tornando as páginas mais dinâmicas, porém tem que haver sempre uma interação do utilizador ou um temporizador que atualize a página.

As tecnologias push [30] que permitem que o servidor envie dados atualizados mal os recebam, têm sido usadas desde algum tempo. Uma das soluções era o *long pooling*, que permite a abertura de uma ligação ao servidor, até que este receba um pedido. Um dos principais problemas deste tipo de soluções era o *overhead* de HTTP, que poderia prejudicar soluções que não são tolerantes a atrasos. Para além disto, o servidor era obrigado a criar várias ligações TCP para cada cliente, uma para enviar informação e várias para cada mensagem recebida [30].

WebSocket Protocol foi desenhado seguindo o RFC6455 [31] para resolver este tipo de problemas. É um protocolo que fornece uma comunicação bidirecional, é baseado em *sockets* e usa HTTP como camada de transporte. Desta forma é possível usar WebSockets com soluções já existentes, visto que pode trabalhar sob portas HTTP como 80 e 443, o que facilita a integração deste protocolo com as soluções já existentes. Para além disso, este protocolo não está apenas limitado ao HTTP, o que permite fazer uma *handshake* simples sob uma porta sem reinventar o protocolo.

A Figura 19 representa o estado atual da compatibilidade dos *browsers* atuais e da tecnologia de *WebSockets* [32].

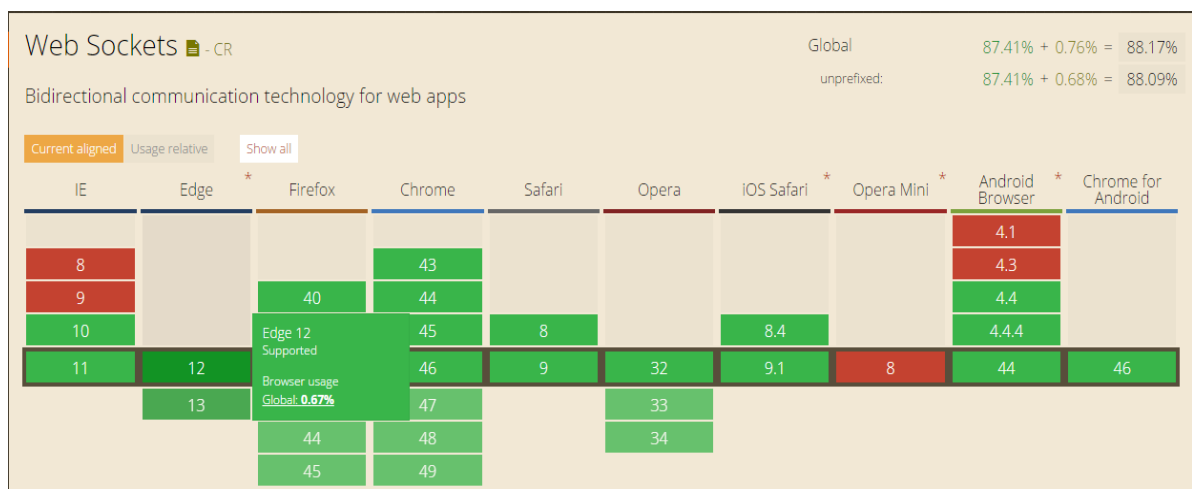


Figura 19 – Compatibilidade de *Websockets* e *browsers* atuais [32]

Na Figura 20 estão representados dois tipos de ligações: *polling* (HTTP) e *WebSockets* [33].

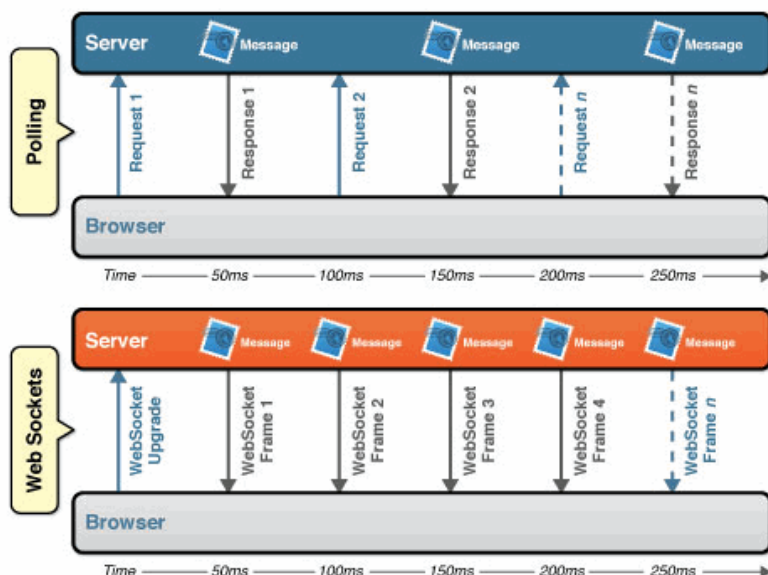


Figura 20 – Comparação de latência entre *polling* e aplicações *Websockets* [33]

Numa ligação *polling* o *browser* necessita de estar sempre a fazer pedidos, pois o servidor só responde no caso de receber esses pedidos. No caso do *WebSocket*, o *browser* faz apenas um pedido e o servidor envia toda a informação em vários pacotes de dados, sem ter que receber qualquer pedido adicional por parte do browser. Esta ligação entre o servidor e *web browser* é designada como uma ligação persistente, pois sempre que um deles precisar de enviar qualquer informação podem fazê-lo a qualquer momento [33].

Existem vários tipos de implementação de um servidor *WebSockets*. Seguidamente será apresentado o servidor Node.js, que foi utilizado no projeto e serão descritas as suas principais características.

3.3.1. Protocolo JSON

O JSON é um formato leve para troca de dados (esquemático na Figura 21) baseado na sintaxe definida no padrão ECMAScript/JavaScript para definição de estruturas de dados, tais como coleções ou listas ordenadas de valores. É baseado em estruturas de dados universais, implementadas por praticamente todas as linguagens de programação modernas [34]:

- Coleções de pares do tipo nome/valor – representados por uma sequência de pares “string|valor”, separados por vírgulas e entre chave. São implementados como objetos em Javascript.
- Listas ordenada de valores (vetores) – representados por uma sequência de valores separados por vírgulas entre parênteses retos. São implementados como *arrays* em Javascript.

Os tipos de dados representáveis em JSON são: *strings*, números, valores lógicos (*true* ou *false*), e a constante *null*, além dos tipos estruturados objetos e arrays mencionados anteriormente [34].

Este protocolo, devido à simplicidade com que é possível interpretá-lo através da linguagem Javascript utilizada pelos clientes Ajax, tem-se mostrado uma grande e válida alternativa ao XML [34].

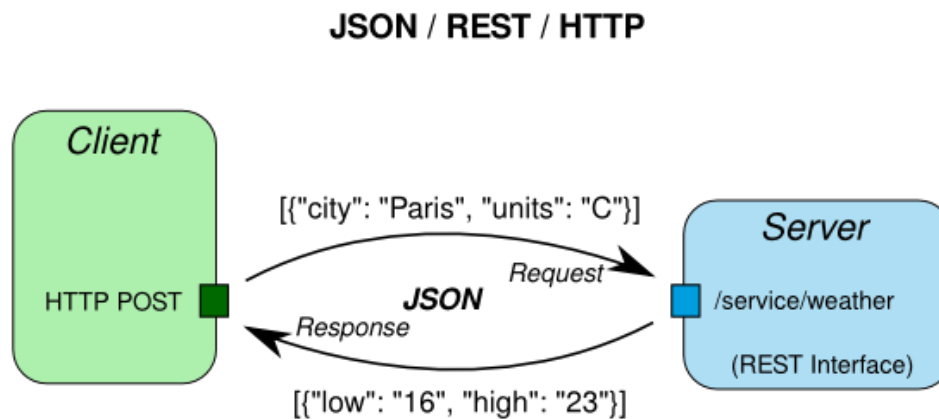


Figura 21 – Protocolo JSON na troca de dados entre cliente-servidor [35]

3.3.2. Node.js

É uma plataforma orientada a eventos, do lado do servidor, que permite a criação de aplicações web escaláveis. Os programas que são desenvolvidos com esta plataforma são escritos na linguagem JavaScript e são assíncronos, o que permite diminuir a sobrecarga e aumentar a escalabilidade, isto com códigos capazes de manipular dezenas de milhares de conexões simultâneas, numa única máquina física [36]. Este utiliza o Node Package Manager (npm) como gestor de pacotes, este é responsável por instalar e gerir os pacotes nos programas node [37]. Contém uma biblioteca de servidor HTTP, que permite correr aplicações web sem usar um servidor típico HTTP (e.g. Apache). Esta plataforma permite a criação de aplicações que tenham por objetivo funcionar completamente em tempo real, como por exemplo, *streaming* de voz ou vídeo, ou então serviços de *chat*, partilha de ficheiros. Implementa também um repositório de módulos, onde cada um tem uma funcionalidade diferente. Cada um destes módulos pode ser instalado a partir do npm. Este, para além de servir para instalar os módulos, permite também tratar da gestão de versões e gestão de dependências. Um dos módulos que permite desenvolver aplicações que atuem em tempo real e que utilizem uma API tipo *WebSockets*, é o “Socket.io”. Este permite que *WebSockets* e tempo real estejam presentes em qualquer *browser*, para além disso ainda fornece *multiplexing*, escalabilidade horizontal e codificação e decodificação em JSON [38].

Os módulos do servidor Node.js (Figura 22) utilizados no projeto foram os seguintes:

```
/**
 * Module dependencies.
 */
var express = require('express');
var http    = require('http');
var io      = require('socket.io');
var mysql   = require('mysql');
```

Figura 22 – Módulos utilizados no servidor Node.js

- **Socket.io:**

O maior problema em desenvolver aplicações real-time é que nem todos os *browsers* suportam o protocolo *WebSockets*, tornando assim inviável o desenvolvimento de uma aplicação 100% *WebSockets*.

Com essa desvantagem, surgiu a *framework* *Socket.IO* cuja sua ideia resolve o problema nos *browsers* antigos, emulando outras técnicas de comunicação real-time de forma totalmente abstrata para o programador. Basicamente, ela possui um script no *client-side*, incluído no código cliente, fornecido através de uma *CDN* [54], que deteta o *browser* e a compatibilidade adequada de comunicação e tenta na seguinte ordem escolher a técnica de comunicação que será viável com o *browser* em questão:

1. *WebSocket*
2. *Adobe® Flash® Socket*
3. *AJAX long pooling*
4. *AJAX multipart streaming*
5. *Forever Iframe*
6. *JSONP Polling*

A vantagem deste processo é o facto de decisões mais complexas, como o exemplo da escolha da técnica de comunicação, serem feitas automaticamente pelo *Socket.IO*, sendo assim abstraídas para o programador [38].

Na Figura 23 ilustrada abaixo está representada a utilização do *socket* no servidor *node*. Na linha 71 são colocados os *sockets* à “escuta” dos pedidos dos clientes, a partir daí são declarados os métodos dos quais estamos à espera receber, como exemplo, podemos verificar o método *login* na linha 77 que recebe os dados, envia-os para uma função interna na linha 82 que retorna a resposta “*userData*”. Esta variável é retornada ao *socket* através do método *emit* presente na linha 87.

```
71 socketio.sockets.on('connection', function(socket) {
72
73   /**
74    * Check login credentials
75    * @param {type} data
76    */
77   socket.on('login', function(data) {
78     console.log("----- LOGIN -----");
79     console.log(data);
80     console.log("-----");
81
82     login.login(data, function(err, userData){
83       // console.log("----- LOGIN RESPONSE -----");
84       if(!err){
85         //console.log("UserData" + userData);
86         socket.emit('login',userData.u,userData.d,userData.g,userData.building,userData.beacons);
87         addUser(socket, userData.u.U_ID,userData.u.U_USERNAME,userData.d.D_ID,userData.u.TENANT_ID);
88       }else{
89         //console.log(" LOGIN ERROR - INVALID LOGIN ");
90         socket.emit('Login.Error', { error: 'invalid' });
91       }
92     }
93     //console.log("-----");
94   });
95 });
```

Figura 23 – Utilização do módulo Socket.io no servidor Node.js do projeto

- **Express.js:**

Express é uma framework minimalista, flexível que contém, entre várias funcionalidades, um robusto conjunto de recursos para desenvolver aplicações web, entre os quais, um sistema *Models-Views-Controllers* intuitivo (MVC) e respetivo sistema de rotas [39]. Na Figura 24 está ilustrada a utilização do express onde é definido o porto, criado o middleware ao chamar a função “express()”, definido o caminho onde se encontram as views, e inicializado o servidor http. Este módulo foi utilizado no projeto de forma a servir umas páginas com fim estatístico para uso interno como mostra a Figura 25.

```
//server port
var port = 3002;

// Setup middleware
var app = express();

// Optional since express defaults to CWD/views
app.set('views', __dirname + '/views');

//Setup Socket.IO Server
var server = http.Server(app);
```

Figura 24 – Utilização do módulo Express no servidor Node.js do projeto

```

{
  - 1: {
    - users: {
      + nd5yxMfQSe--JndqAJRb: {...},
      - Q51sq9yDKwWN2xAFAJSh: {
        host: "::ffff:127.0.0.1",
        connTime: "2015-11-10T10:37:38.131Z",
        socketioId: "Q51sq9yDKwWN2xAFAJSh",
        tenantId: 1,
        deviceId: 27,
        userId: 507,
        userName: "bestrelo",
        signed: true
      },
      + Lk-LhbLiLZbpUvGgAJZU: {...},
      + 1Igl0eqJA5j1bqLUAJeZ: {...},
      + At3_UtcpS9IPDFe3AJgi: {...},
      - peeGMYUm502GEnhQAjHn: {
        host: "::ffff:127.0.0.1",
        connTime: "2015-11-11T11:25:55.471Z",
        socketioId: "peeGMYUm502GEnhQAjHn",
        tenantId: 1,
        deviceId: 128,
        userId: 285,
        userName: "goliveira",
        signed: true
      },
      + e04tr2MaDXteb9McAJhX: {...},
      + XDHha0Elo1XJHKeEAJkd: {...},
      + v04BBTCKLffQetckAJkl: {...}
    }
  },
  + 7: {...},
  + 9: {...}
}

```

Figura 25 – Página de estatística servida pelo módulo express do Node.js

- **MySQL:**

Este módulo é um driver para MySQL escrito em javascript. É utilizado no projeto para aceder e manipular bases de dados MySQL a partir do servidor Node.js. Na Figura 26 está exemplificada a sua utilização no caso do método login da app. Nas linhas 63 e 64, são carregados os módulos que vamos utilizar, o próprio MySQL e o ficheiro bd.js que contém as definições da base de dados e respetivas configurações de ligação. Na linha 71, começa o método de login que recebe as variáveis “json” e “callback_”, a primeira contém os dados recebidos e a segunda é uma função para ser utilizada como callback. Continuando, na linha 71 é inicializado o MySQL e na 72 é feito o parse dos dados recebidos em formato JSON, na linha 81 é executada a query, mais abaixo, na linha 85 é verificado o retorno da consulta, caso este devolva um resultado significa que existe o utilizador na base de dados, e assim sucessivamente até chegar ao ponto de chamar a função callback com a resposta que será então enviada para o cliente com os respetivos dados resultantes do processo de login.

```

60 /**
61  * Module dependencies.
62  */
63 var mysql = require('mysql');
64 var bd = require('./bd.js');
65
66 /**
67  * App Login
68  * @param {[type]} json
69  * @return {[type]}
70  */
71 exports.login = function(json, callback){
72   var mysql = bd.startMySQL();
73   var credentials = JSON.parse(json);
74
75   // Verify if username and password match
76   var query = 'SELECT * FROM USERS '
77             + ' WHERE U_USERNAME = ' + mysql.escape(credentials.username)
78             + ' AND U_PASSWORD = ' + mysql.escape(credentials.password)
79             + ' AND U_DELETED = 0';
80
81   mysql.query(query, function(err, user){
82     if(err){
83       throw err;
84     }else{
85       if(user.length === 1){
86         // Verify if device serial number and password match
87         var query = 'SELECT * FROM DEVICES'
88                   + ' WHERE D_NSERIE = ' + mysql.escape(credentials.serial)
89                   + ' AND D_PASSWORD = ' + mysql.escape(credentials.device)
90                   + ' AND D_DELETED = 0';
91         mysql.query(query, function(err, device){
92           if(err){
93             throw err;
94           }else{
95             if(device.length === 1){
96               //Get the PTISERVER, AND CHANNELID

```

Figura 26 – Exemplo da utilização do módulo MySQL no servidor Node.js do projeto

- **HTTP:**

Este módulo foi projetado para criar de forma simples um servidor http.

Apesar do Node.js possibilitar servir páginas web, neste projeto esta tarefa foi atribuída exclusivamente ao servidor apache já falado anteriormente.

No contexto do projeto desenvolvido, é no servidor Node.js que assentam as trocas de informação realizadas entre clientes Android e Servidor (Base de Dados) utilizando o protocolo JSON como formato de resposta, como ilustrado na Figura 27

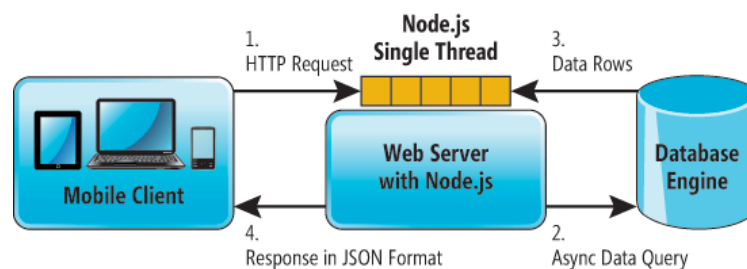


Figura 27 – Comunicação JSON Android-Servidor [40]

Como exemplo do processo de troca de informação entre cliente e servidor, utilizo o caso do login na aplicação Android, ilustrado nas figuras seguintes, onde se utiliza o protocolo JSON da forma exemplificada anteriormente na Figura 27.

```

1 private void requestLogin() {
2     try {
3         if (serverCommunication != null) {
4             socket = serverCommunication.getSocket();
5         } else {
6             return;
7         }
8
9         JSONObject json = new JSONObject();
10
11         json.put("username", username.getText().toString());
12         json.put("password", password.getText().toString());
13         json.put("device", devicePassword.getText().toString());
14         json.put("serial", stDeviceSerialNumber);
15         json.put("imei", stImeiIdentifier);
16
17         socket.emit("login", json.toString());
18
19     } catch (JSONException e) {
20         LogCatMe.LogCat(TAG, LOG_ERROR, " btnLogin", e.getMessage());
21     }
22 }

```

Figura 28 – Comunicação JSON – Pedido de login na app (App->Servidor)

Analisando a Figura 28, na linha 4, após verificação de sua existência, vamos buscar o *socket* para realizar a comunicação. Na linha 9 é criado o objeto JSON, o qual é preenchido, entre as linhas 11 e 15, com os dados a enviar. Chegando à linha 17, é feito o envio do objeto JSON através do método *emit* do *socket* obtido anteriormente na linha 4.

```

1 socket.on('login', function(data){
2     console.log("----- LOGIN -----");
3     console.log(data);
4     console.log("-----");
5
6     login.login(data, function(err, userData){
7         console.log("----- LOGIN RESPONSE -----");
8         if(!err){
9             console.log("UserData" + userData);
10            socket.emit('login', userData.u, userData.d, userData.g, userData.building, userData.beacons);
11            addUser(socket, userData.u.U_ID, userData.u.U_USERNAME, userData.d.D_ID, userData.u.TENANT_ID);
12        }else{
13            console.log(" LOGIN ERROR - INVALID LOGIN ");
14            socket.emit('Login.Error', { error: 'invalid' });
15        }
16        console.log("-----");
17    });
18 });

```

```

----- LOGIN -----
{"username":"fabio","password":"", "device":"tula","serial":"f4ccca6c","imei":"316907061459133"}
----- LOGIN RESPONSE -----
UserData: {"u":{"U_ID":9,"G_ID":23,"U_IDOWNER":7,"U_IDFUNC":"tula003","U_USERNAME":"fabio","U_PASSWORD":"", "U_NOME":"Fábio","U_MORADA":"Morada","U_TELEFONE":"Telef
one","U_EMAIL":"fabio@tulait.eu","U_CANCONFIG":1,"U_CANCHANGEPTTCHANNEL":1,"U_INSERVICE":0,"U_ADMIN":1,"U_DELETED":0,"U_TIMESTAMP":"2015-08-05T16:18:55.000Z","U_LEVEL":
8,"TENANT_ID": , "CREATED":null,"CREATEDBY":null,"UPDATED":null,"UPDATEDBY":null},"d":{"D_ID":56,"D_IDOWNER":7,"D_NOME":"Tulait dev","D_MARCA":"Vodafone","D_MODELO":"Pri
me 6","D_PASSWORD":"", "D_IMEI":"356907061459133","D_NSERIE":"f4ccca6c","D_NTELEFONE":"","D_NSIMCARD":"","D_VERSAOSOFTWARE":"","D_TIMESTAMP":"2015-06-04T11:15:09.000
Z","BUILDING_ID":0,"D_DELETED":0,"TENANT_ID": , "CREATED":"0000-00-00 00:00:00","CREATEDBY":0,"UPDATED":"0000-00-00 00:00:00","UPDATEDBY":0},"g":{"G_ID":23,"PTT_ID":0,"G
_NAME":"Developer","G_PTTIP":"cloud01.tulait.eu","G_PTTIPORT":"","G_PTTIPASSWORD":"","G_PTTSERVERID":"","G_PTTCHANNELID":"","G_TIMESTAMP":"2015-02-13T15:20:
08.000Z","G_DELETED":0,"TENANT_ID": , "CREATED":null,"CREATEDBY":null,"UPDATED":null,"UPDATEDBY":null},"building":{"b_id":0,"ne_lat":0,"ne_lon":0,"sw_lat":0,"sw_lon":0},
"beacons":{"numBeacons":1}}
-----

```

Figura 29 – Comunicação JSON – Login na app (Resposta do servidor)

Na Figura 29, na parte superior, encontramos o código do node.js responsável pelo tratamento e resposta ao evento login. O socket tem o evento “login” à escuta dos dados enviados pelos cliente (linha 1), assim, quando um cliente faz o emit para o evento “login” do socket com os seus dados, este invoca o método login do controller login presente na linha 6. Este método recebe os dados e uma função *callback* que por sua vez devolve uma variável de controlo para erro e os dados do utilizador. No processamento deste *callback*, na linha 8, caso se verifique que este não retornou erro, enviamos de volta para o cliente todos os dados necessários relativos ao utilizador através do método emit para o evento login descrito na linha 10. Em caso da resposta da função *callback* ser erro, entramos na condição else na linha 12 e notifica-se o cliente através de um emit para um método específico “Login.Error” criado para o efeito na linha 14.

```

1 private Emitter.Listener login = new Emitter.Listener() {
2     @Override
3     public void call(Object... args) {
4         Intent intent = new Intent(MainActivity.LOGIN_EVENT);
5         try {
6             JSONObject user = (JSONObject) args[0];
7             JSONObject device = (JSONObject) args[1];
8             JSONObject group = (JSONObject) args[2];
9             JSONObject building = (JSONObject) args[3];
10            JSONObject beacons = (JSONObject) args[4];
11
12            intent.putExtra("user", user.toString());
13            intent.putExtra("device", device.toString());
14            intent.putExtra("group", group.toString());
15            intent.putExtra("building", building.toString());
16            intent.putExtra("beacons", beacons.toString());
17
18            sendBroadcast(intent);
19        } catch (Exception e) {
20            Log.e(TAG, "Failed to parse the json received on event (LOGIN) " + e);
21        }
22    }
23};

```

Figura 30 – Comunicação JSON – Tratamento da resposta na app (Servidor->App)

Na Figura 30, considerando a resposta obtida pelo servidor representada na Figura 29, são recebidos no cliente os dados relativos à resposta do pedido de login, após recebidos, estes são separados e guardados como objetos do tipo JSON (linhas 6 a 10) sendo depois colocados num

intente (linhas 12 a 16) e enviados num *broadcast* (linha 18) que será capturado pelo respetivo *receiver* no local onde este será necessários.

3.4. Mumble

Para as comunicações de voz entre dispositivos, foi utilizado o software open-source Mumble VOIP [41].

Este software tem como principais características as seguintes [41]:

- *Software* livre
- Componente Cliente denominado por “Mumble”
- Componente Servidor denominado por “Mumble Server” ou “Murmur”
- Baixa Latência
- Alta qualidade de som
- Suporte a longo prazo – compatibilidade com versões anteriores
- Fácil customização para diferentes casos de uso

3.4.1. Mumble Server

Por necessidade dos clientes, para que estes tenham disponível a funcionalidade de push-to-talk, foi necessário criar um servidor Mumble dedicado ao projeto.

É neste que os clientes (Android) vão ligar aos respetivos canais de voz e é também sobre ele que os clientes (Web) vão realizar as tarefas administrativas, que permite gerir grupos/canais de voz. Esta gestão implica uma interação com o servidor que é feita através do *middleware* Ice [42]. Este, através da integração com a linguagem PHP, possibilita, através um mecanismo RPC, a execução de scripts no servidor Mumble e a respetiva gestão do mesmo a partir da plataforma web do projeto. Na Figura 31 está representada a classe PHP criada para o efeito.

Entre as linhas 66 e 78 é criado um proxy para o middleware ICE, para tornar possível a comunicação com este servidor a partir do cliente. A partir da linha 82 estão definidos os métodos da interface, estes são os métodos necessários para a gestão pretendida através da interface web. Dando estes as possibilidades de obter, adicionar, editar e remover canais de voz.

```

60 class SaMurmurIce {
61     private $meta;
62     private $version;
63     private $contextVars = array('secret' => '██████████');
64     private $connectionString = "Meta:tcp -h 127.0.0.1 -p ██████";
65
66     public function __construct() {
67         if (Ice_intversion() >= 30400) {
68             $initData = new Ice_InitializationData;
69             $initData->properties = Ice_createProperties();
70
71             $initData->properties->setProperty('Ice.ImplicitContext', 'Shared');
72             $ICE = Ice_initialize($initData);
73
74             $ice_proxy = $ICE->stringToProxy($this->connectionString);
75             $this->meta = Murmur_MetaPrxHelper::checkedCast($ice_proxy);
76             $this->meta = $this->meta->ice_context($this->contextVars);
77         }
78     }
79     /**
80      * Return Murmur ServerProxyInterface
81      * Murmur\Server is not a structure with data/info, but a proxy with function to manage a single virtual server
82      */
83     public function getServerProxyInterface($server_id) {
84
85
86
87
88     public function addChannel($server_id, $channelName) {
89
90
91
92     public function getServerTree($server_id) {
93
94
95     public function getChannel($server_id, $channel_id) {
96
97
98
99
100
101
102     public function updateChannelName($server_id, $channel_id, $channel_new_name) {
103
104
105
106
107     public function deleteChannel($server_id, $channel_id) {
108
109
110
111
112 }

```

Figura 31 – Classe PHP responsável pela gestão de servidor Mumble do projeto

3.4.1. Mumble Client

No projeto foi feita a integração do cliente Android com o servidor Mumble, possibilitando assim a funcionalidade de push-to-talk nos dispositivos dos clientes. Esta integração teve como base um projeto disponibilizado no GitHub [43]. O bloco de código responsável pela ligação está representado abaixo na Figura 32. Na linha 71, está declarada a função `connectToMumble()`, sendo esta a responsável por chamar uma outra função, `registerConnectionReceiver()` que tem como finalidade registar o *observer* que vai fazer com estejamos “alerta” em relação aos eventos dos outros elementos registados no Mumble, e também criar o intent com os dados relativos à conexão (linhas 74 a 79) para então inicializar o serviço Mumble (linha 81). Entre a linha 95 e 104, encontramos a função `unregisterConnectionReceiver()` que realiza o inverso da `registerConnectionReceiver()` anteriormente explicada.

```
71 private void connectToMumble() throws JSONException {
72     registerConnectionReceiver();
73
74     connectionIntent = new Intent(this, MumbleService.class);
75     connectionIntent.setAction(MumbleService.ACTION_CONNECT);
76     connectionIntent.putExtra(MumbleService.EXTRA_HOST, HOST);
77     connectionIntent.putExtra(MumbleService.EXTRA_PORT, PORT);
78     connectionIntent.putExtra(MumbleService.EXTRA_USERNAME, userUsername);
79     connectionIntent.putExtra(MumbleService.EXTRA_PASSWORD, PASSWORD);
80
81     startService(connectionIntent);
82 }
83
84 private void registerConnectionReceiver() {
85     if (mServiceObserver != null) {
86         return;
87     }
88
89     if (mService != null) {
90         mServiceObserver = new ServerServiceObserver();
91         mService.registerObserver(mServiceObserver);
92     }
93 }
94
95 private void unregisterConnectionReceiver() {
96     if (mServiceObserver == null) {
97         return;
98     }
99
100    if (mService != null) {
101        mService.unregisterObserver(mServiceObserver);
102    }
103
104    mServiceObserver = null;
105 }
```

Figura 32 – Código de ligação do cliente Android ao servidor Mumble

3.5. Localização / Posicionamento

Como citado no capítulo 1, o facto de termos conhecimento das posições do utilizador tem um papel preponderante para que se atinjam os objetivos propostos desta aplicação. A exibição da informação pretendida no local correto implica o conhecimento, o quanto mais preciso possível da posição do utilizador.

Em dispositivos móveis existem várias técnicas de localização com diferentes precisões, ou custo de energia, por exemplo. Os resultados podem também ser diferentes conforme que técnicas estejam a ser usadas em meios interiores ou exteriores, uma vez que na primeira situação os sinais usados para determinar a posição do utilizador podem ser muito enfraquecidos devido à existência de paredes, tetos ou outras estruturas e na segunda situação os algoritmos utilizados também não são 100% precisos, apresentando diferentes precisões.

Assim, para determinar a posição, neste projeto foi necessário recorrer a diversas tecnologias dos dispositivos móveis, como GPS, Wi-Fi, GSM, NFC, Bluetooth, entre outros.

3.5.1. Localização em Meios Exteriores

Serão a seguir abordadas algumas das técnicas mais conhecidas para posicionamento em meios exteriores, explicando em alguns casos o seu funcionamento e falando de forma sucinta das suas vantagens/desvantagens.

GPS

O GPS é uma tecnologia que está presente na maioria dos modelos de *smartphones* atuais [44]. Como tal, a sua utilização permite que mais utilizadores tenham acesso às tecnologias que de si são dependentes. O GPS oferece uma excelente precisão no exterior, que é usualmente inferior a 10 metros em mais de 95% do tempo, sendo que alguns autores arriscam que a precisão deste sistema se encontra entre os 6 e os 10 metros, o que é bastante aceitável para o objetivo que se pretende alcançar [44]. Caracteriza-se por ser um sistema de navegação por satélite que fornece a um aparelho recetor móvel a sua posição, a qualquer momento e em qualquer lugar na Terra, desde que o recetor se encontre no campo de visão de pelo menos quatro satélites GPS [44].

Existem vários fatores que condicionam os resultados do GPS em termos de precisão. De acordo com vários autores, três dos mais importantes são [44]:

- Erro causado pelo efeito de multicaminhos;
- Erro causado pelo atraso da propagação do sinal;
- Erro causado por o dispositivo que tem GPS estar a detetar 3 ou menos satélites;

Ora uma vez que as paredes levam ao aparecimento de multicaminho e permitem o atraso de propagação de sinal, conclui-se facilmente que a funcionalidade de GPS é uma fraca opção perante parte dos objetivos que se pretendem atingir (e.g. no caso de posicionamentos no interior de edifícios) [44].

A-GPS

O A-GPS diferencia-se do GPS por uma rede a ajudar nos procedimentos, permitindo assim uma poupança a nível da bateria do dispositivo e redução da quantidade de processamento que é feita no terminal [44].

Wi-Fi

A técnica *fingerprinting* pode ser aplicada para determinar a posição em meios exteriores, da mesma forma que em meios interiores, em sistemas Wi-Fi [44].

A combinação desta técnica com a posição obtida através de GPS, como sugerem vários autores, produzirá resultados mais precisos. Contudo, os gastos essencialmente energéticos que esta combinação acarretará não compensarão uma vez que o GPS ou ainda melhor o A-GPS por si só já produz resultados bastante satisfatórios em meios exteriores [44].

Também o facto de ser uma funcionalidade presente na grande maioria dos *smartphones* facilita o acesso dos utilizadores às aplicações que dependam assim deste sistema [44].

3.5.2. Localização em Espaços Interiores

O posicionamento também pode ter de ser feito em meios interiores como por exemplo centros comerciais, estádios, estacionamento, museus, bancos, etc..., que serão importantes para os objetivos desta aplicação. Assim, é importante estudar os diferentes algoritmos existentes para resolver este tipo de localização [44].

Quando é necessário determinar este posicionamento, uma das alternativas que existe envolve a tecnologia Wi-Fi. Esta tecnologia não foi desenhada com o intuito de determinar a posição de um dispositivo. Contudo, medições de força de sinal do sinal transmitido, seja por pontos de acesso ou por estações, permitem deduzir a localização de qualquer dispositivo móvel, e, conseqüentemente, do seu utilizador [44].

Wi-Fi Fingerprinting

O WLAN RSS Wi-Fi *Fingerprinting* tem-se tornado uma das técnicas mais exploradas em localização interior. É um método relativamente fácil de pôr em execução e é tolerante ao ruído de sinal e, portanto, consegue assim atingir resultados mais precisos [44].

Este processo consiste em 2 fases:

- **Mapeamento *offline*:**

Os RSS de diversos AP's, em diferentes pontos de uma grelha imaginária, representativa da área a ser estudada, são reunidos e guardados numa base de dados de *fingerprints* [44].

- **Determinação *online*:**

O dispositivo *online* vai medir uma amostra de vetores *fingerprint* com os valores RSS de diferentes AP's nas suas posições. A amostra é enviada para um servidor central na infraestrutura. Este servidor compara a *fingerprint* medida com as que tem na base de dados para determinar a posição do dispositivo na grelha. Então, o resultado estimado é devolvido ao dispositivo. A grande desvantagem deste algoritmo é ter de treinar o sistema com o maior número de leituras de sinais de rádio possível (mapeamento descrito anteriormente). Como estes sistemas apenas permitem que a fase *offline* se realize antes do seu verdadeiro uso, os

sistemas só serão precisos enquanto a base de dados dos mapeamentos ainda fizer sentido perante os dados reais, caso contrário, terá de voltar a ser atualizada [44].

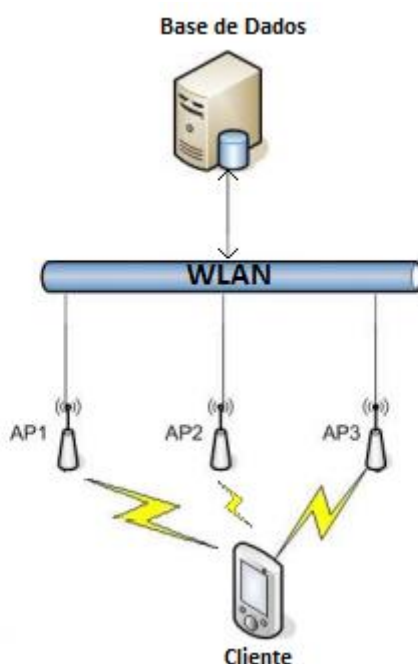


Figura 33 – Wi-Fi Fingerprint – Esquema de Arquitetura

Para tal funcionalidade, neste projeto foram utilizadas as bibliotecas android disponibilizadas para lidar com este tipo de informação wifi – android.net.wifi.*

Tags NFC

O NFC é uma tecnologia que permite a troca de informações sem fio e de forma segura entre dispositivos compatíveis que estejam próximos um do outro [45]. Ou seja, logo que os dispositivos estejam suficientemente próximos, a comunicação é estabelecida automaticamente, sem a necessidade de configurações adicionais. Estes dispositivos podem ser *smartphones*, *Tablets*, crachás, cartões eletrónicos ou qualquer outro dispositivo que tenha um chip NFC [45].

Esta tecnologia já é utilizada atualmente em algumas tarefas como [45]:

- **Compra de bilhetes** (e.g. metro, cinema, entre outros);
- **Carros** – a Hyundai traz esta tecnologia integrada em alguns dos seus produtos, visando a acompanhar a tecnologia envolvendo NFC, no protótipo das chaves, com as quais é possível realizar várias tarefas, como abrir a porta e ligar o motor.
- **Pagamentos** – o pagamento de serviços através do *smartphone* começa a ser uma realidade, com este sistema já é possível realizar pagamentos de estacionamento, máquinas de *vending* (Figura 34), compras nos supermercados, entre outros.

As **principais vantagens** do NFC são [45]:

- Simples – Interações NFC não requerem mais do que um simples toque;
- Versátil – É ideal para uma ampla gama de ambientes e utilizações;
- Baseado em padrões iguais e abertos – As camadas subjacentes da tecnologia NFC seguem padrões universalmente implementados;
- Habilitador – o NFC facilita a configuração rápida e simples de tecnologias sem fio, como Bluetooth, Wi-Fi, etc...;
- Seguro – Transmissões NFC são de curto alcance (de um toque a poucos centímetros);
- Interoperável – NFC trabalha com as tecnologias existentes de cartão sem contato;

As **principais desvantagens** do NFC são [45]:

- Necessidade de equipamentos específicos – equipamentos habilitados com NFC são exigidos para ler as *tags* NFC;
- Custo – Como qualquer avanço tecnológico, existem custos atribuídos ao desenvolvimento e implementação da tecnologia NFC;
- Falta de conhecimento – esta é ainda uma tecnologia relativamente desconhecida, embora exista há algum tempo. Muitas pessoas não sabem o que é NFC nem como utilizá-lo;
- Limitações técnicas – Há limite de velocidade, de quantidade de dados que podemos transferir.

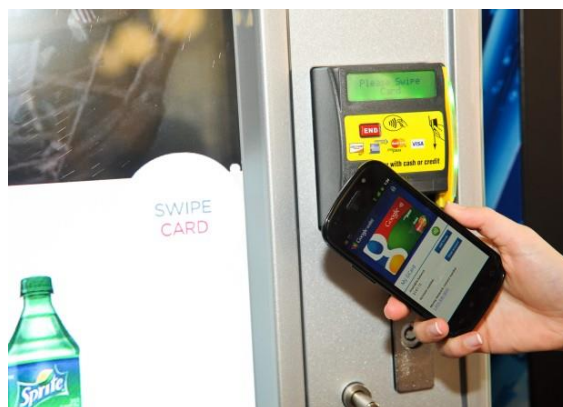


Figura 34 – Pagamento através de NFC [45]

Esta tecnologia foi introduzida na aplicação de forma a substituir a tecnologia dos *QR codes* anteriormente utilizada, com a principal finalidade de colmatar falhas de segurança e fiabilidade sobre os dados obtidos em relação às rondas por exemplo. Na aplicação não foi utilizada nenhuma framework específica para trabalhar com NFC, foi apenas utilizado o código nativo do Android para o efeito, através da inclusão das packages `android.nfc.*`.

BLE's - Beacons Bluetooth

Para o sistema de localização interior, além das tecnologias Wi-Fi e NFC, podem também ser utilizados “Beacons” JAALÉE [46] que utilizam tecnologia rádio bidirecional Bluetooth 2.4GHz conhecida por BLE. Tecnicamente, um JAALÉE Beacon é um pequeno computador provido de um processador ARM Cortex 32-bit e uma memória flash de 256kB. BLE 4.0 é um novo padrão redesenhado e desenvolvido pela Nokia. É apelidado de baixo-consumo visto que estes dispositivos pode executar até vários anos com uma única bateria de lítio, dependendo da taxa e força de transmissão de sinal com que estes estejam configurados [46].

Na aplicação, existe uma funcionalidade que serve para configurar os beacons, ou seja, o cliente atribui o nome da localização a um determinado beacon. Este, depois de configurado, faz broadcast do sinal para os dispositivos ao seu redor que por sua vez vão registando as passagens assim como os nomes dos locais a estes associados.

Para a integração desta tecnologia com o sistema Android, foi utilizado um SDK obtido através do repositório Jaalee no GitHub [47] possibilitando assim ao Android as seguintes operações:

- Encontrar beacons e caso pretendido filtrar os mesmos por valores de atributos associados aquando a sua configuração
- Monitorizar beacons
- Ler e escrever atributos dos beacons (identificadores, frequência de broadcast de sinal, entre outros...)

3.6. Geofence de dispositivos

Uma das funcionalidades implementada a pedido de vários clientes foi o geofence de dispositivos. Este consiste em atribuir um edifício ou área geográfica a um dispositivo. A criação dos edifícios/áreas geográficas é realizada através da plataforma web como ilustrado abaixo na Figura 35.

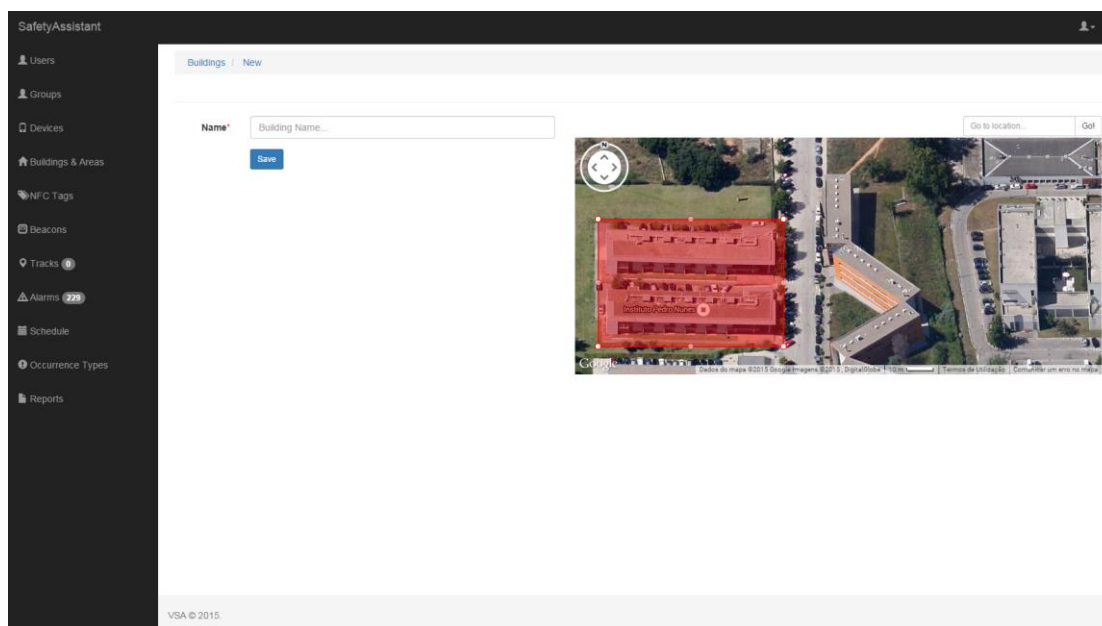


Figura 35 – Criação de um edifício através da plataforma web

Uma vez que se verifique esta atribuição, assim que é feito um login num determinado dispositivo e este se encontre associado a um edifício ou a uma área geográfica, é então criada uma “cerca virtual” que tem como função delimitar o raio de ação do dispositivo. Cada vez que o dispositivo saia do perímetro, o utilizador é alertado através de “bips” sonoros para que possa voltar para a área definida, caso estes alertas sejam ignorados é então emitido automaticamente um alerta para o supervisor através da plataforma web a indicar tal situação.

Apesar de já existirem ferramentas próprias para realizar este processo de Geofence na totalidade, tal como por exemplo a Google que disponibiliza uma API para o efeito [48], não foi utilizada nenhuma específica para o processo completo. Foi apenas feito o uso de uma biblioteca `BoundingBoxE6` presente na package `org.osmdroid.util` da comunidade OSMDroid [49].

Tendo em conta que os edifícios/áreas definidos na web terão sempre a forma de retângulos, ao fazer login na app, caso o dispositivo esteja associado a algum edifício ou área geográfica, este recebe os quatro pontos limitadores da área respetiva. Estes pontos limitadores serão passados ao construtor da classe `BoundingBox` como mostrado na Figura 36 que a partir deles irá criar a respetiva *box* e verificar se as posições do dispositivo capturadas de x em x tempo pertencem ao interior dessa *box*, este método está ilustrado abaixo na Figura 37.

```
29 public MyBoundingBox(GeoPoint upRight, GeoPoint downLeft) {  
30  
31     Bbox = new BoundingBoxE6(upRight.getLatitudeE6(),  
32                             upRight.getLongitudeE6(), downLeft.getLatitudeE6(),  
33                             downLeft.getLongitudeE6());  
34  
35 }
```

Figura 36 – Criação de uma BoundingBox para Geofence

```
86 public boolean isPointInsideBox(GeoPoint point) {  
87  
88     if (point == null || Bbox == null) {  
89         return false;  
90     }  
91  
92     if (point.getLatitudeE6() <= Bbox.getLatNorthE6()  
93         && point.getLatitudeE6() >= Bbox.getLatSouthE6()  
94         && point.getLongitudeE6() <= Bbox.getLonEastE6()  
95         && point.getLongitudeE6() >= Bbox.getLonWestE6()) {  
96  
97         return true;  
98     } else {  
99  
100         return false;  
101     }  
102 }
```

Figura 37 – Verificação se ponto geográfico está dentro da “BoundingBox”

4. TESTES E RESULTADOS

4.1. Dificuldades Durante o Desenvolvimento

Por entre vários problemas que foram surgindo ao longo do desenvolvimento / funcionamento do sistema, dou destaque aos seguintes tendo em conta que exigiram mais tempo e trabalho para resolver:

- Dificuldade com Node.js:

A programação orientada a eventos utiliza bastante o conceito de *callbacks*. Como o javascript permite a passagem de funções anônimas, não será incomum encontrar códigos como o mostrado abaixo na Figura 38.

```
func1(param, function(err, res) {
  func2(param, function(err, res) {
    func3(param, function(err, res) {
      func4(param, function(err, res) {
        func5(param, function(err, res) {
          func6(param, function(err, res) {
            func7(param, function(err, res) {
              func8(param, function(err, res) {
                func9(param, function(err, res) {
                  // Do something...
                });
              });
            });
          });
        });
      });
    });
  });
});

fs.open('results', 'w', function(err, fd) {
  fs.write(fd, results, function(err, written, f) {
    fs.close(fd, function(err) {
      done();
    });
  });
});
```

Figura 38 – Código de leitura confusa devido ao excesso de callbacks nas funções javascript

Devido à sua natureza assíncrona, não há garantia na ordem de execução das chamadas efetuadas. O código acima tem a intenção de manter o controlo na ordem de execução das chamadas. À medida que se vai tentando manter o controlo sobre a ordem de execução, acabamos por perder na legibilidade e manutenção do código, gerando o chamado “Spaghetti Code” ou “Callback Hell”.

- Solução:

Como soluções para este tipo de problema, há uma grande quantidade de módulos desenvolvidos pela comunidade. Entre os quais podem ser citados, Step, Flow-js, Promise, entre outros. Nas suas versões mais recentes, o Javascript inclui o módulo Promise no seu core. Através da sua utilização deste obtém-se maior legibilidade no código, além de uma forma bem simples de executar funções de maneira sequencial [50].

Sendo o node.js uma framework relativamente recente, encontra-se em constante evolução e a cada dia que passa, a comunidade desenvolve novos módulos visando atacar os pontos a serem melhorados.

- Pontos negativos do algoritmo indoor:

A utilização de pontos de acesso Wi-Fi para localização interior trouxe problemas de precisão. Estes problemas surgem pelo facto dos sinais Wi-Fi serem muito instáveis, ou seja, há muitas variáveis externas, relativas ao ambiente que não conseguimos controlar, tais como posições dos pontos de acesso que podem ser modificadas, pontos de acessos móveis que são temporários e podem ser incluídos nas entradas do algoritmo de mapeamento/localização, interferências de outros dispositivos existentes nas proximidades, por exemplo, um micro-ondas a funcionar por perto é o suficiente para baralhar os cálculos de mapeamento/localização.

- Dificuldade com APN (proxy) do ISP Vodafone:

Surgiu durante o desenvolvimento uma dificuldade de ligação ao servidor Node.js por parte dos clientes Android, isto porque estes estão a utilizar a rede GSM da Vodafone. Nesta infraestrutura GSM, é utilizado um *proxy* HTTP, este, assim que detete que é feito um pedido (HTTP request), termina a ligação logo após a resposta (HTTP response), e este cenário não é de todo o pretendido uma vez que a utilização do webSocket necessita de uma ligação permanente.

Como solução foi aproveitada a utilização do *Stunnel* descrito anteriormente no ponto 0. Este túnel seguro entre a aplicação Android e o servidor, encapsula e encripta toda a comunicação entre os clientes Android e o servidor, impossibilitando assim o *proxy* do ISP identificar que tipo de protocolo da camada da aplicação que está a ser utilizada.

4.2. Testes Realizados e Resultados Obtidos

Durante o desenvolvimento do projeto e após constatar que a aplicação estava funcional, esta foi apresentada e proposta a diversas empresas. Esta fase foi importante para o crescimento do sistema pois através desta utilização foram transmitidos *feedbacks* extremamente importantes para tomar decisões de implementação ou melhoria/adaptação de funcionalidades que permitiram melhorar e

tornar a aplicação mais robusta e funcional, tendo em conta que os testes foram feitos num contexto real sob um ponto de vista de utilizador comum.

Das empresas que testaram e/ou utilizam a aplicação, destacam-se as seguintes representadas abaixo na Tabela 3

Tabela 3 – Empresas clientes da aplicação

Cliente	Área de Negócio	Localização Geográfica
Vodafone Portugal	Comunicações	Lisboa
365	Segurança Privada	Coimbra
Charon	Segurança Privada	Lisboa
Prosegur	Segurança Privada	Matosinhos
PowerShield	Segurança Privada	Lisboa
Simlis	Águas	Leiria
Obvispectrum	Comunicações	Lisboa
THR	Soluções Tecnológicas (Segurança)	Faro
Volkswagen Autoeuropa	Indústria Automóvel	Palmela
NOS	Comunicações	Lisboa
ARKO	Segurança Privada	Lisboa
2045	Segurança Privada	Lisboa
Satronel	Indústria Eletrónica	Estónia

Os testes foram portanto maioritariamente funcionais, sendo que a determinada altura foi feito um teste de carga no servidor web e no servidor Node.js utilizando as ferramentas “*ab – Apache Bench*” [51] e “*GNU Plot*” [52], testes estes descritos e representados no ponto 2 da secção de Anexos.

Execução de comandos *ab* para gerar os ficheiros de dados para os testes. Comandos estes que seguem o seguinte padrão:

ab [-**g** *ficheiro GNUPlot*] [-**n** *nº. de pedidos*] [-**c** *concorrência*] [*http[s]://hostname[:porto]/*]

- Comando *ab* e respetivos parâmetros para o porto 80 do servidor.

Comando: *ab -g cloud01-80.tsv -n 1000 -c 50 http://cloud01.tulait.eu:80/*

Resultado: Anexo 2.1.

- Comando *ab* e respetivos parâmetros para o porto 3001 do servidor (Node.js).

Comando: `ab -g cloud01-3001.tsv -n 1000 -c 50 http://cloud01.tulait.eu:3001/`

Resultado: Anexo 2.2.

Após execução dos comandos descritos anteriormente verificamos a existência dos respetivos ficheiros resultantes de cada um dos comandos respetivamente, como podemos verificar na Figura 42 presente no Anexo 2.3.

Seguidamente foi criado um ficheiro *template* “cloud01-benchmark.tpl” para ser utilizado pela Ferramenta GNUPlot, ilustrado na Figura 43 do Anexo 2.4.

Nele estão definidos: o output, que neste caso, será um ficheiro “.png” nomeado por cloud01-benchmark.png, o título, o tamanho, eixos e respetivas *labels*, assim como o tipo de gráficos e respetivas fontes, neste caso utilizamos os dois ficheiros “.tsv” gerados no passo anterior, “cloud01-80.tsv” e “cloud01-3001.tsv” respetivamente.

A execução do comando GNUPlot, representado na Figura 44 do Anexo 2.5, com instrução para utilizar o ficheiro “cloud01-benchmark.tpl” como template, dá origem ao respetivo ficheiro “.png”, representado na Figura 45 do Anexo 2.6, onde está contida a representação gráfica dos ficheiros “.tsv” gerados no início do processo recorrendo à ferramenta “ab”. Representação abaixo na Figura 46 do Anexo 2.7.

Como análise da Figura 46, verificamos que o servidor Node.js lida melhor com o número de pedidos efetuados relativamente ao servidor web, visto que o seu tempo de resposta não aumenta tanto ao longo do tempo em que aumentam os pedidos. Ambos os servidores foram sujeitos a 1000 pedidos com uma concorrência de 50. Em torno dos 600 pedidos nota-se que o tempo de resposta dos servidores começa a aumentar, no caso do servidor web, é perfeitamente visível que aumenta em maior proporção comparativamente ao servidor Node.js, que só sofre um maior pico a em torno dos 800 pedidos.

Foi também criada uma página de visualização de dados de acesso de utilizadores com finalidades estatísticas, como já foi descrito anteriormente na secção Express.js do ponto 3.3.2 Node.js.

4.3. Interface gráfica da aplicação

A interface gráfica da aplicação, quer da componente web como da componente móvel, está ilustrada e explicada em detalhe num manual de utilizador que se encontra em anexo no ponto 3 da secção de anexos.

5. CONCLUSÃO

Esta aplicação pretende contribuir para que empresas nas quais se enquadra a sua utilização, tenham uma noção de como ela pode trazer benefícios, quer em termos práticos, quer em termos financeiros com uma significativa redução de custos. Por exemplo na substituição de rádios existentes pela funcionalidade *push-to-talk* da aplicação, entre outras.

Comparando a proposta de trabalho descrita no plano de estágio com o sistema desenvolvido, tenho que acrescentar que existem algumas diferenças, como novas funcionalidades e novas tecnologias, como causa disso, temos que considerar que na realidade, os projetos raramente seguem à risca o fluxo sequencial que os modelos exemplificam, levando assim à criação de sub-etapas dentro das etapas propostas, à modificação dos requisitos, entre outros pontos relevantes.

Em termos pessoais, este projeto foi uma mais-valia para mim visto que foi, e continua a ser uma grande fonte de aprendizagem nas várias vertentes englobadas, sendo que interiorizei novos conceitos, tecnologias e ferramentas.

Relativamente aos resultados obtidos, este projeto deixou-me muito satisfeito tendo em conta que além de serem atingidos todos os objetivos inicialmente propostos, estes foram expandidos ao longo do tempo, sendo este um sinal claro de crescimento do projeto e satisfação de clientes.

Foi assim um enorme desafio tendo em conta a dimensão atingida até ao momento e também as perspetivas futuras. A parceria de venda com a Vodafone Portugal foi uma aposta que trouxe muitas vantagens, uma vez que é um nome de peso na área em questão e com isto reforçou o potencial da aplicação, facilitando também a sua proximidade com os clientes.

Na Figura 39 está representada a página de destaque da aplicação na secção empresarial do *website* da Vodafone [53].

Figura 39 – Página dedicada à aplicação na secção empresarial do site da Vodafone [53]

5.1. Trabalho Futuro

Sendo a aplicação apresentada uma versão em desenvolvimento, pode ainda possuir muitas funcionalidades, algumas das quais descrevo a seguir com o objetivo de serem incorporadas em trabalhos futuros:

- Melhorias do algoritmo de posicionamento interior.
- Utilização de Webservice PHP responsável pela troca de informação entre cliente e servidor, passando a utilizar o Node.js só para notificações/eventos em “tempo real”.
- Link de *feedback* embutido na plataforma web de forma a possibilitar aos clientes reportar anomalias ou outros assuntos pertinentes relativos à utilização do sistema.
- *Dashboard* inicial da plataforma web, onde constem informações estatísticas tais como por exemplo o número de utilizadores ligados, zona de troca de mensagens, etc...
- Auricular com botão para *push-to-talk*.
- Permitir falar só para um determinado utilizador do *push-to-talk* escolhido da lista.
- Criação de edifícios/áreas geográficas com formas irregulares, não estando estes apenas limitados a retângulos.
- Separação de clientes (tenants) no contexto do servidor Node.js à imagem do que é feito no contexto de clientes web.

- Implementação de módulo de suporte, onde apenas sejam permitidas consultas de dados para fins de helpdesk.
- Plataforma de gestão de Tenants por parte do provider.
- Criar a entidade canal de voz para separar os conceitos de grupo e canal de voz, que atualmente são o mesmo perante o sistema, transformando-os em entidades independentes que poderão estar ou não relacionadas entre si.
- Guardar registo de acesso a módulos por parte dos tenants na base de dados, atualmente está a ser feito diretamente no código da aplicação.
- Possibilidade de criar e gerir padrões de rondas, onde se atribuem pontos de ronda a cada padrão com a finalidade de ser possível visualizar/reportar possíveis incumprimentos relativos à não passagem por determinados de pontos da ronda.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] OHSW&IM. (2012). Working Alone or in Isolation Procedure. University of South Australia, Human Resources Unit of Occupational Health, Safety and Welfare.
- [2] Guillemy, N., Liévin, D., & Pagliero, D. (décembre de 2006). "Travail isolé - Prévention des risques - Synthèse et application". (INRS, Ed.).
- [3] Matos, Luísa (2014) Trabalho Isolado. Um fator subestimado na prevenção. Isolated work. An underestimated factor in prevention. (SHO).
- [4] HSE. (2013). Working alone. Health and safety guidance on the risks of lone working. (Rev 3 ed.). (U. HSE Books, Ed.) Leaflet INDG73. Health and Safety Executive.
- [5] <http://www.ccohs.ca/oshanswers/hsprograms/workingalone.html>
- [6] INRS. (2007). Recommendation 416-Travail isolé et dangereux.
- [7] Gonçalves, R. (Abril a Junho de 2011). Proteção de trabalhadores isolados. Proteger, pp. 46-48.
- [8] Guardian – Mobile Personal Safety: <http://guardianmps.com/>
- [9] AGORA – PSIM Made Easy: <http://observit.pt/agora-psim/>
- [10] STUNNEL – <https://www.stunnel.org/>
- [11] Multitenancy Services – <http://www.devmedia.com.br/transforme-aplicacoes-web-em-servicos-multi-tenant/28881>
- [12] Bootstrap – <http://thiagonasc.com/desenvolvimento-web/desenvolvendo-com-bootstrap-3-um-framework-front-end-que-vale-a-pena>
- [13] PHP – http://php.net/manual/pt_BR/index.php e https://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Introdução/História
- [14] Netbeans – https://netbeans.org/index_pt_PT.html
- [15] Eclipse Foundation – https://pt.wikipedia.org/wiki/Eclipse_Foundation
- [16] *Android developers* – <http://developer.android.com/index.html>
- [17] Android SDK – <http://www.androidpt.info/index.php?title=SDK>
- [18] Android Architecture Layers – <http://novicesuresh.com/category/android-architecture/>
- [19] JAVA, o que é – http://www.java.com/pt_BR/download/faq/whatis_java.xml
- [20] <http://www.rlsystem.com.br/java/noticia/conheca-um-pouco-mais-sobre-a-linguagem-de-programacao-java/>
- [21] JAVA Language – [https://pt.wikipedia.org/wiki/Java_\(linguagem_de_programação\)](https://pt.wikipedia.org/wiki/Java_(linguagem_de_programação))
- [22] JAVA – <https://www.java.com>
- [23] SQL – <http://pt.wikipedia.org/wiki/SQL>

-
- [24] MySQL – <https://pt.wikipedia.org/wiki/MySQL>
- [25] MySQL(vantagens) – <http://www.endcase.com.br/site/node/4>
- [26] Servidor Web – https://pt.wikipedia.org/wiki/Servidor_web
- [27] Inquérito sobre webservers – <http://news.netcraft.com/archives/2014/04/02/april-2014-web-server-survey.html>
- [28] Servidor Web (Apache) - [https://repositorium.sdum.uminho.pt/bitstream/1822/12907/1/Tese de Ricardo André Pereira Freitas.pdf](https://repositorium.sdum.uminho.pt/bitstream/1822/12907/1/Tese%20de%20Ricardo%20Andr%C3%A9%20Pereira%20Freitas.pdf)
- [29] CWAC-Updater GitHub Repo – <https://github.com/commonsguy/cwac-updater/>
- [30] Push Technology – http://en.wikipedia.org/wiki/Push_technology
- [31] The websocket protocol. RFC 6455 <http://tools.ietf.org/html/rfc6455>
- [32] Can I Use (WebSockets) – <http://caniuse.com/#feat=websockets>
- [33] Pooling vs WebSocket – <http://www.codeproject.com/Articles/531698/Introduction-to-HTML-WebSocket>
- [34] JSON – <http://www.json.org>
- [35] JSON Protocol – <http://s11368.blogspot.pt/2014/10/json-j-ava-s-cript-o-bject-n-otacion.html>
- [36] S. Tikov, S. Vinoski. “Node.js: Using Javascript to Build Gugh Performance Network Programs”. Internet Computing, IEEE, 2010.
- [37] NPMJS – <https://www.npmjs.com/>
- [38] Node.js Socketio – <http://udgwebdev.com/node-js-para-leigos-explorando-real-time/>
- [39] Node.js Express – <http://nodebr.com/primeiros-passos-com-express-em-node-js/>
- [40] Real-World Scenarios for Node.js – <https://msdn.microsoft.com/en-us/magazine/jj991974.aspx>
- [41] Mumble VOIP – http://wiki.mumble.info/wiki/Main_Page
- [42] ICE – <http://wiki.mumble.info/wiki/Ice>
- [43] Mumble Android Client GitHub Repo – <https://github.com/pcgod/mumble-android>
- [44] GPS – <http://paginas.fe.up.pt/~ei08031/Dissertacao/Tese.pdf>
- [45] NFC – https://pt.wikipedia.org/wiki/Near_Field_Communication
- [46] JAALÉE – <http://www.jaalee.com/>
- [47] JAALÉE SDK – <https://github.com/jaalee/Jaalee-Beacon-Android-SDK-master>
- [48] Google Geofence API – <https://developers.google.com/android/reference/com/google/android/gms/location/Geofence>
- [49] OSMDROID – <http://osmdroid.azurewebsites.net/osmdroid-android-4.3-javadoc/org/osmdroid/util/BoundingBoxE6.html>
- [50] S. Tikov, S. Vinoski. “Node.js: Using Javascript to Build Gugh Performance Network Programs”. Internet Computing, IEEE, 2010.
- [51] Apache bench – <https://httpd.apache.org/docs/2.2/programs/ab.html>
- [52] GNU Plot – <http://www.gnuplot.info/>
-

[53] Site Vodafone Negócios – <http://negocios.vodafone.pt/aplicacoes-empresariais/solucoes-m2m/vodafone-safety-assistant>

[54] Socket.io.js – <https://cdnjs.cloudflare.com/ajax/libs/socket.io/1.3.5/socket.io.min.js>

ANEXOS

1. PROPOSTA DE ESTÁGIO

Fábio Marques
MIA136002
Proposta de Estágio

Escola Superior de Tecnologia e Gestão de Oliveira do Hospital

Apresento desta forma a minha proposta de estágio:

1. Título do Estágio

- Mandown – *Lone Worker Monitoring*

2. Local do Estágio

- TulaMove.Lda

3. Enquadramento

- Durante o período de estágio, o principal objetivo a cumprir por parte do aluno será o desenvolvimento de uma aplicação que servirá empresas cujo interesse seja monitorização de rondas dos seus funcionários, trajetos realizados, registo de ocorrências, registo de incidentes pessoais, entre outros.
- Esta terá as empresas de segurança como principal cliente alvo.
- A sua estrutura assentará basicamente nos seguintes pontos:
 - Componente administrativa: Onde será feita toda a gestão da utilização da aplicação, utilizadores e eventos aplicação.
 - Componente Android: plataforma em utilização pelo cliente, será a partir de dispositivos com sistema Android que vai ser feita a emissão de registos acima mencionados, trajetos, alarmes, incidentes, entre outros.

4. Plano de Trabalhos 1º Semestre

- Desenvolvimento de plataforma de registo de trajetórias com as seguintes especificações:
 - Controlo de acesso de utilizadores por login/pass.
 - Controlo de acesso de dispositivos por id/pass.
 - Associação/gestão de dispositivos e utilizadores.
 - Criação da estrutura de base de dados para registo de utilizadores, dispositivos, trajetos, alarmes e incidentes.
 - Visualização de registos de trajetórias e alarmes num mapa Google.

- Manipulação de alarmes pelo utilizador atribuído (e.g. Comentários, validações).

5. Plano de Trabalhos 2º Semestre

- Criação de app Android para inserção de dados na plataforma desenvolvida no primeiro semestre com as seguintes especificações:
 - Envio das coordenadas GPS.
 - Envio de alarmes nas categorias definidas pelo gestor de sistema.
 - Detecção de queda através do acelerómetro do smartphone.
 - Possibilidade de comunicação por push-to-talk entre utilizadores (Walkie-Talkie)

6. Condições de Acolhimento

- O aluno irá desenvolver o seu trabalho nas instalações do Instituto Pedro Nunes em Coimbra, onde tem à disposição, um local de trabalho fixo, assim como todos os recursos necessários.
- O presente estágio não será remunerado.

7. Orientador (preferência)

- Prof. António Manuel de Brito Paulino;
- Prof. Francisco Carlos Afonso.

Fábio Marques
MIA136002

2. RESULTADOS DE TESTES AO SERVIDOR

2.1. Comando ab e respetivos parâmetros para o porto 80

```
Benchmarking cloud01.tulait.eu (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:      Apache/2.4.7
Server Hostname:     cloud01.tulait.eu
Server Port:         80

Document Path:       /
Document Length:     259 bytes

Concurrency Level:   50
Time taken for tests: 131.303 seconds
Complete requests:   1000
Failed requests:     0
Total transferred:   529000 bytes
HTML transferred:   259000 bytes
Requests per second: 7.62 [#/sec] (mean)
Time per request:    6565.158 [ms] (mean)
Time per request:    131.303 [ms] (mean, across all concurrent requests)
Transfer rate:       3.93 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    11  131 587.1   29  9578
Processing:  52 5507 6701.8 1609 31640
Waiting:    22 3120 4948.9 1131 31451
Total:      73 5638 6842.8 1645 31967

Percentage of the requests served within a certain time (ms)
 50%    1645
 66%    4376
 75%    6802
 80%    9210
 90%   18604
 95%   21158
 98%   24047
 99%   25982
100%  31967 (longest request)

c:\Users\Fábio\Desktop\benchmark>
```

Figura 40 – Comando ab e respetivos parâmetros para o porto 80 do servidor.

2.2. Comando ab e respetivos parâmetros para o porto 3001

```
Benchmarking cloud01.tulait.eu (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:
Server Hostname: cloud01.tulait.eu
Server Port: 3001

Document Path: /
Document Length: 38 bytes

Concurrency Level: 50
Time taken for tests: 40.028 seconds
Complete requests: 1000
Failed requests: 0
Total transferred: 219000 bytes
HTML transferred: 38000 bytes
Requests per second: 24.98 [#/sec] (mean)
Time per request: 2001.405 [ms] (mean)
Time per request: 40.028 [ms] (mean, across all concurrent requests)
Transfer rate: 5.34 [kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    5   40 171.9   29  3036
Processing: 27 1906 1507.6 1350  9200
Waiting:   16 1111 1190.9   771  5360
Total:     76 1945 1531.2 1376  9232

Percentage of the requests served within a certain time (ms)
50%   1376
66%   1456
75%   1550
80%   1640
90%   4476
95%   5404
98%   6528
99%   8621
100%  9232 (longest request)

C:\Users\Fábio\Desktop\benchmark>
```

Figura 41 – Comando ab e respetivos parâmetros para o porto 3001 do servidor (node.js).

2.3. Ficheiros Resultantes da execução dos comandos *ab*

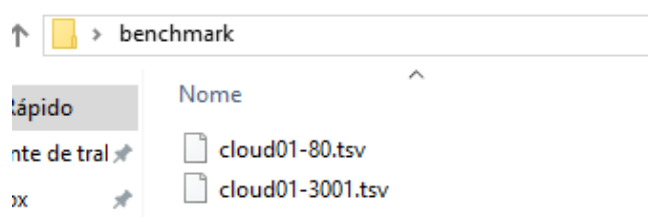


Figura 42 – Ficheiros resultantes da execução dos comandos “*ab*”

2.4. Ficheiro Template a Utilizar na Ferramenta GNUPlot

```
cloud01-benchmark.tpl
1
2 set terminal png
3 set output "cloud01-benchmark.png"
4 set title "Benchmark for cloud01.tulait.eu"
5 set size 1,1
6 set grid y
7 set xlabel 'Requests'
8 set ylabel 'Response Time (ms)'
9 plot "cloud01-80.tsv" using 10 smooth sbezier with lines title "Benchmark cloud01-80.tsv:", \
10 "cloud01-3001.tsv" using 10 smooth sbezier with lines title "Benchmark cloud01-3001.tsv:"
11
```

Figura 43 – Ficheiro template para utilizar na ferramenta GNUPlot

2.5. Comando GNUPlot com instrução para utilizar o ficheiro template

```
C:\Users\Fábio\Desktop\benchmark>gnuplot cloud01-benchmark.tpl
```

Figura 44 – Comando gnuplot com instrução para utilizar o ficheiro template

2.6. Ficheiro resultante da execução do comando GNUPlot

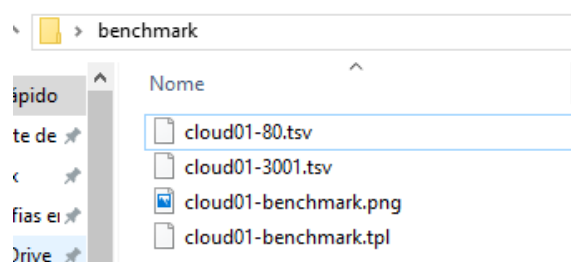


Figura 45 – Ficheiro “.png” resultante da execução do comando gnuplot

2.7. Conteúdo do ficheiro resultante do processo

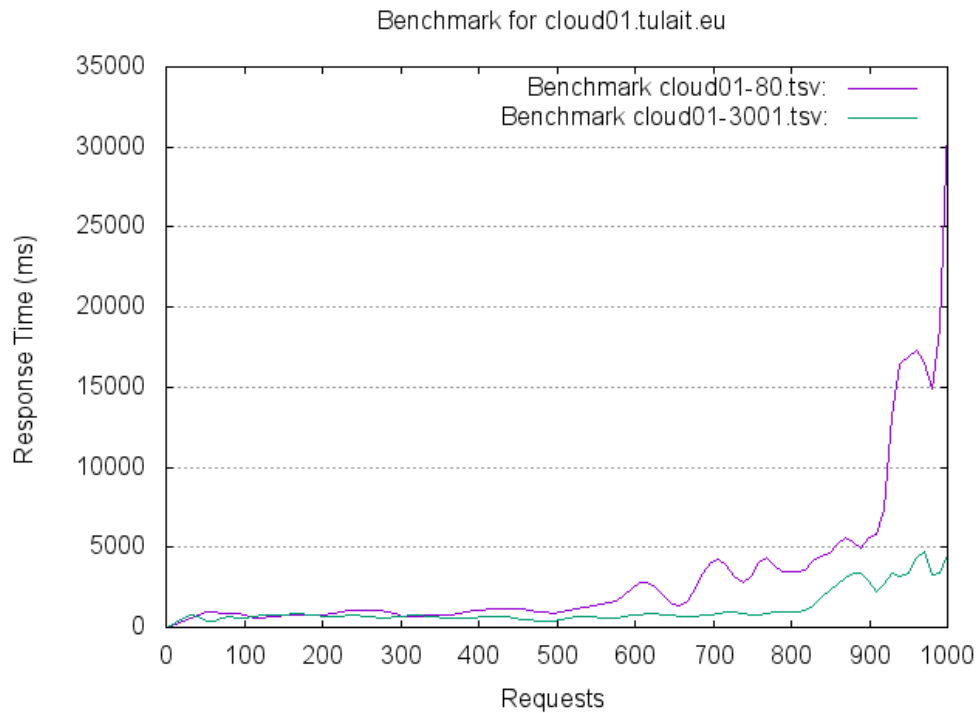


Figura 46 – Conteúdo do ficheiro “.png” resultante do processo

3. MANUAL DE UTILIZADOR

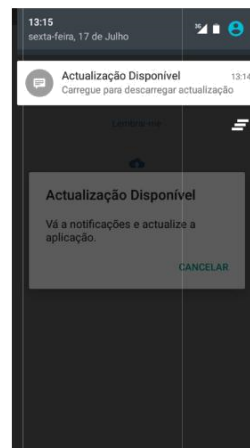




Atualizações da Aplicação

- Atualizações da Aplicação

- Sempre que uma actualização da aplicação estiver disponível, o utilizador, será alertado para a realização da mesma após a ação de login
- Quando isto suceder, o utilizador deverá puxar a barra de notificações (presente no topo do ecrã) para baixo e clicar sobre "Atualização disponível"



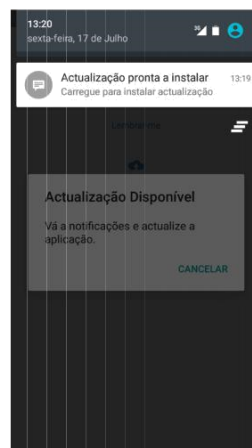
2



Atualizações da Aplicação

- Atualizações da Aplicação

- Após alguns instantes irá aparecer novamente na barra a notificação de actualização pronta a instalar
- Deverá então ser realizado novamente o mesmo procedimento, puxar a barra para baixo e clicar em "Atualização Pronta a Instalar"



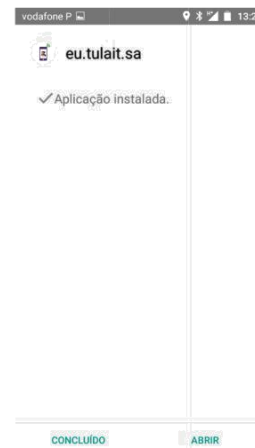
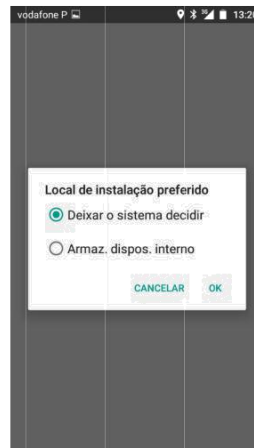
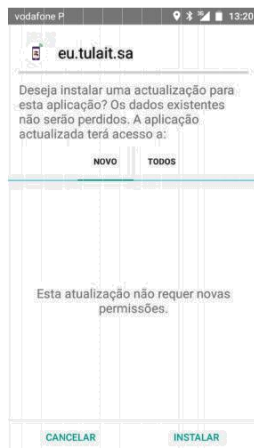
3



Atualizações da Aplicação

- Atualizações da Aplicação

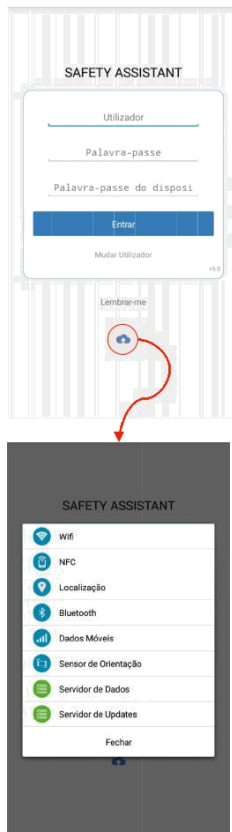
- Seguidamente deverão ser seguidos os passos da instalação normal e a aplicação será então instalada



4

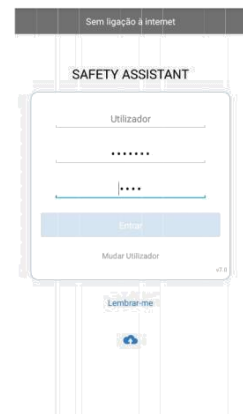


Login na APP – Terminal



- Login no terminal

- Garantir que temos ligação à rede no terminal
- Entrar na aplicação, clicando no ícone no terminal
- Inserir as credenciais de acesso
 - *Username*
 - *Password*
 - Apenas utilizadores autorizados poderão utilizar a APP
 - *Password* do terminal
 - Garante que apenas terminais registados poderão ser utilizados na aplicação
- Janela de *Status* da App
 - Ao clicar no botão status é possível verificar uma janela que nos indica o estado dos componentes necessários à utilização da app, assim como as ligações ao servidor.

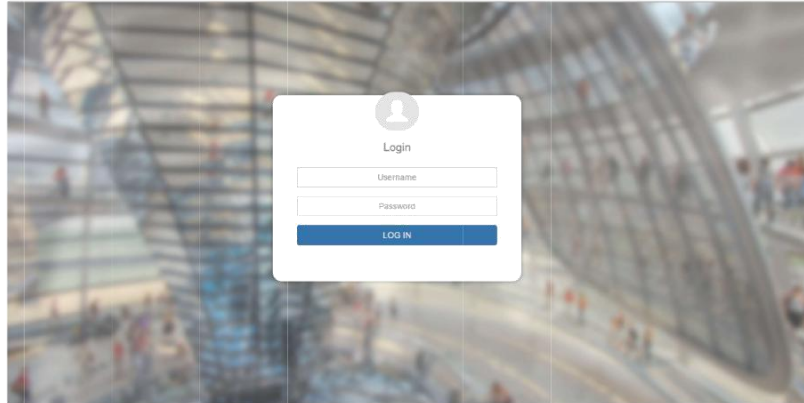


5

Login na APP – BackOffice

- Login no BackOffice

- Num browser internet, ir para o endereço <https://www.vodafonesafetyassistant.vodafone.pt/>
- Inserir as credenciais de acesso



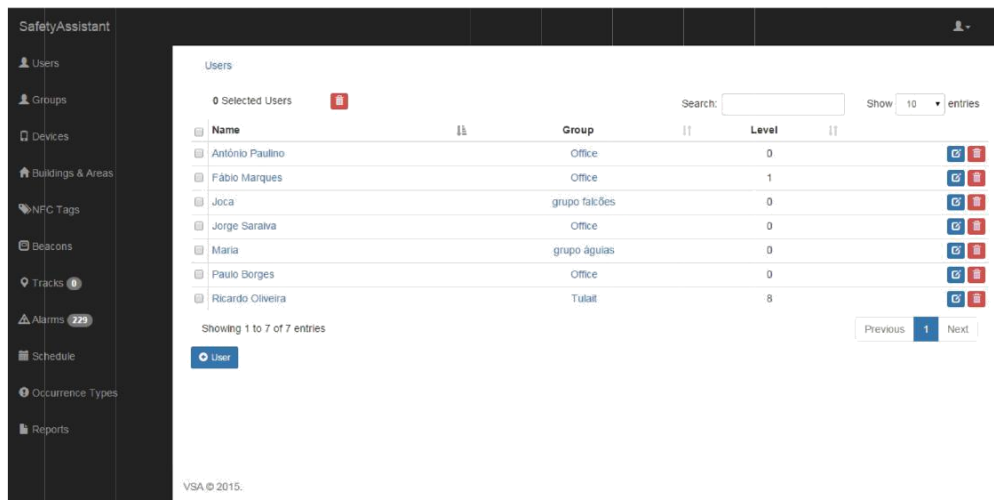
- De notar que estamos perante um site seguro (https)

6

Utilizadores – BackOffice

- Listagem

- Entrar na opção “Users” do menu lateral
- Podemos, assim, visualizar uma listagem de utilizadores existentes no sistema
- Nesta listagem temos acesso a alguma informação dos utilizadores, assim como botões disponíveis para criar, editar e remover utilizadores.



Name	Group	Level
António Paulino	Office	0
Fábio Marques	Office	1
Joca	grupo falções	0
Jorge Saraiva	Office	0
Maria	grupo águias	0
Paulo Borges	Office	0
Ricardo Oliveira	Tuliat	8

7



Utilizadores – BackOffice

• Registrar Utilizadores

- Entrar na opção “Users” do menu lateral
- Clicar no botão presente no fundo da lista
- Preencher o seguinte formulário relativo ao novo utilizador e confirmar no botão
- As opções “Have web access?”, “Can config indoor points and NFC tags?” e “Can switch PTT channel?”, são relativas à possibilidade de dar permissão aos novos utilizadores de terem acesso ao *BackOffice* e de poderem também configurar pontos interiores e etiquetas NFC, e trocar de canal PTT, respetivamente
- Aquando da criação do utilizador, é feita a associação a um grupo (campo “Group”). Esta atribuição pode ser alterada a qualquer altura através da funcionalidade de editar utilizadores.



Utilizadores – BackOffice

• Registrar Utilizadores – Níveis de utilizador

- Aquando do registo de novos utilizadores, é pedido para atribuir um nível (de 0 a 9) ao utilizador em questão.
- É este nível que vai hierarquizar os utilizadores dentro do *BackOffice*.
- Exemplo: Os utilizadores de nível 6, só poderão gerir os utilizadores cujo nível seja inferior a 6.

Utilizadores – BackOffice

• Ver Utilizadores

- Entrar na opção “Users” do menu lateral
- Clicar sobre o nome do utilizador do qual se pretende ver a informação detalhada
- É apresentada a janela com a informação deste utilizador

The screenshot shows the 'Users / Info' page for Fábio Marques. The left sidebar contains navigation options: Users, Groups, Devices, Buildings & Areas, NFC Tags, Beacons, Tracks (1), Alarms (229), Schedule, Occurrence Types, and Reports. The main content area displays the following information:



Employee	tula01	Name	Fábio Marques
Group	Office	Address	n.d.
Username	fabio	Telephone	n.d.
Password	Email	fabio@tulait.eu
	<input type="checkbox"/> Show Password	Registered In	2015-03-26 16:04:07
Have web access	<input checked="" type="checkbox"/>		
Config indoor points	<input checked="" type="checkbox"/>		
Switch PTT channel	<input checked="" type="checkbox"/>		

VSA © 2015.

10

Utilizadores – BackOffice

• Editar Utilizadores

- Entrar na opção “Users” do menu lateral
- Clicar no botão  relativo ao utilizador que se pretende editar
- Editar os campos do formulário onde se pretende alterar os dados do utilizador
- Confirmar as alterações através do botão 

The screenshot shows the 'Users / Edit' page for Fábio Marques. The left sidebar is the same as in the previous screenshot. The main content area displays the following form fields:


- Employee***: tula01
- User Level***: 1
- Name***: Fábio Marques
- Username***: fabio
- Password***:
- Show Password
- Group***: Office
- Have web access?
- Can configure of Indoor points? (Wi-Fi, NFC, Beacons)
- Can switch PTT channel?
- Address**: (empty field)
- Telephone**: (empty field)
- Email**: fabio@tulait.eu

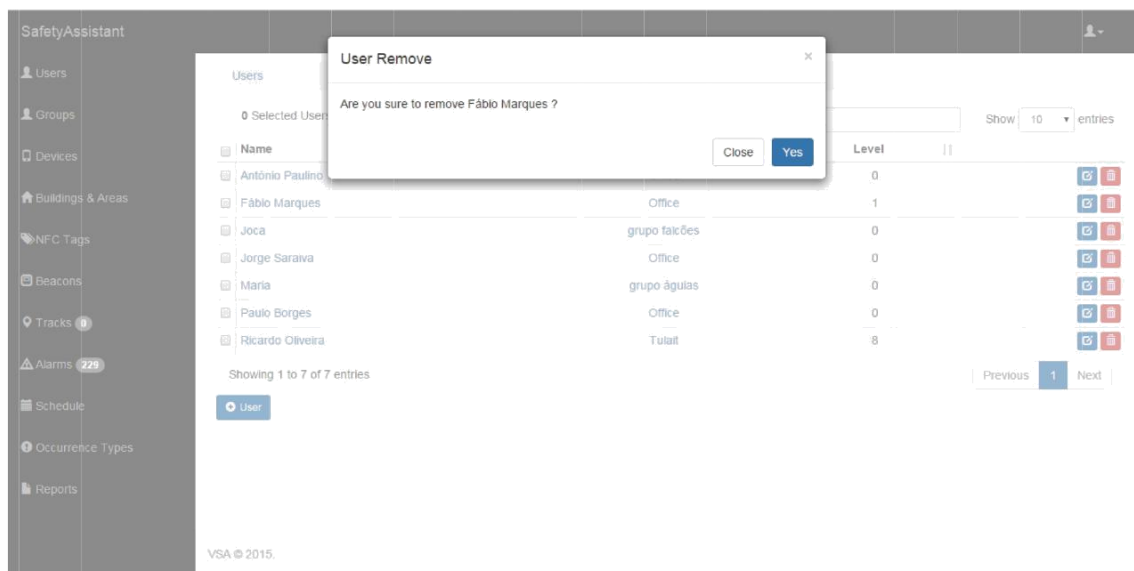
A **Save** button is located at the bottom right of the form. VSA © 2015.

11

Utilizadores – BackOffice

• Remover Utilizadores

- Entrar na opção “Users” do menu lateral
- Clicar no botão  relativo ao utilizador que se pretende remover
- Confirmar a remoção clicando no botão “Sim” da janela de confirmação

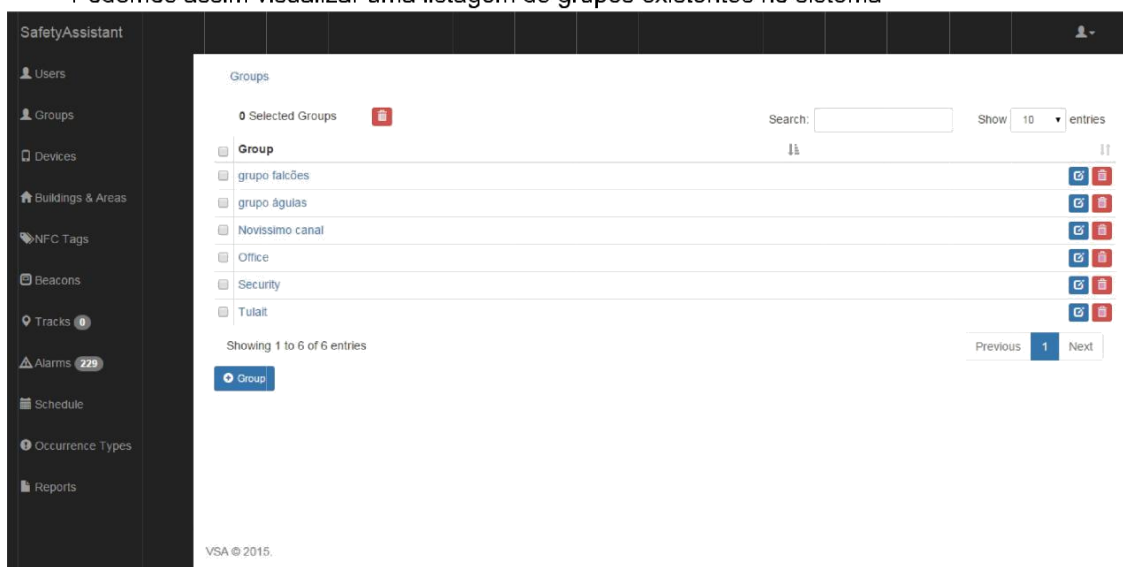


12

Grupos – BackOffice

• Listagem



- Entrar na opção “Groups” do menu lateral
- Nesta listagem temos acesso a alguma informação dos grupos, assim como botões disponíveis para criar, editar e remover grupos.
- Podemos assim visualizar uma listagem de grupos existentes no sistema

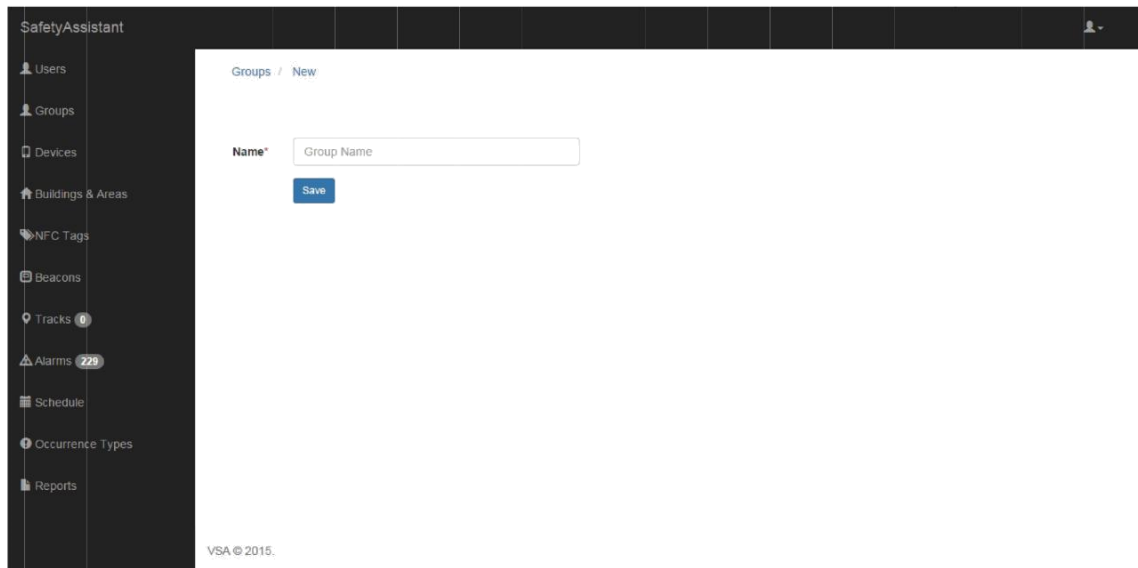


13

Grupos – BackOffice

• Criar Grupos

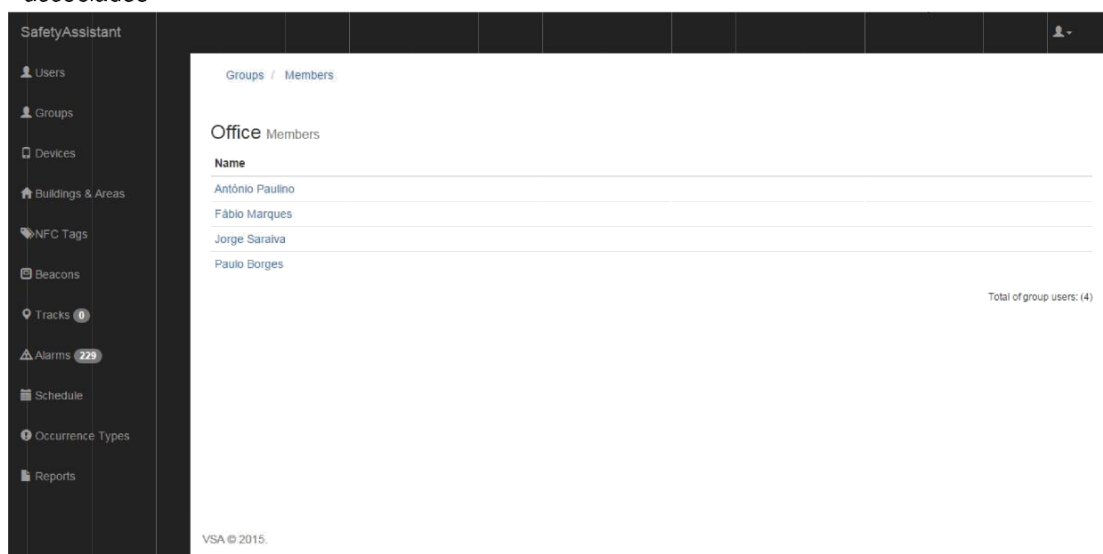
- Entrar na opção “Groups” do menu lateral
- Clicar no botão  presente no fundo da lista
- Preencher o seguinte formulário relativo ao novo grupo e confirmar novamente no botão 



Grupos – BackOffice



• Ver Grupos

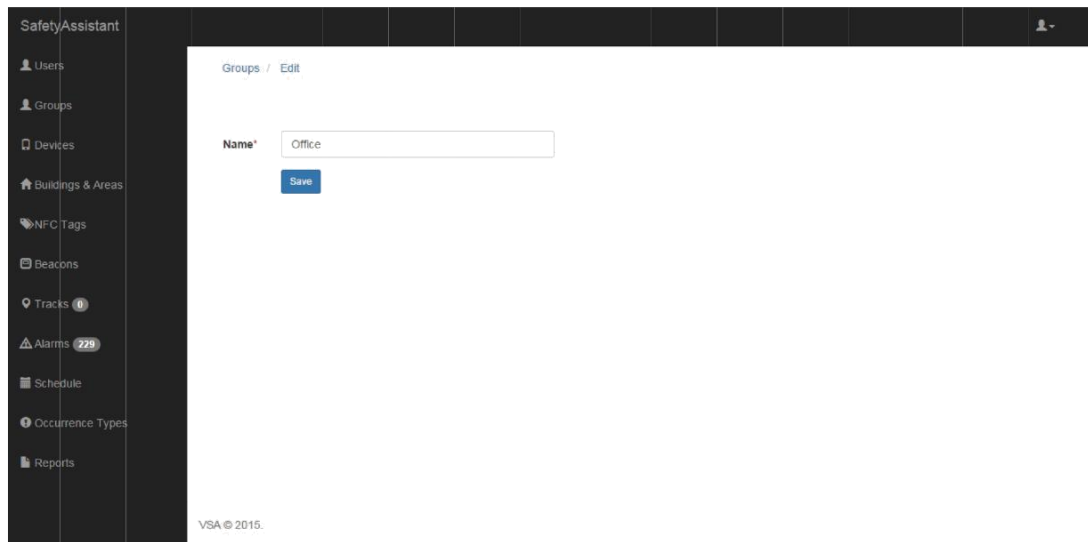
- Entrar na opção “Groups” do menu lateral
- Clicar sobre o nome do grupo do qual se pretende ver a informação detalhada
- É apresentada a janela com a informação deste grupo assim como os utilizadores que estão a ele associados



Grupos – BackOffice

• Editar Grupos


- Entrar na opção “Groups” do menu lateral
- Clicar no botão  relativo ao grupo que se pretende editar
- Editar os campos do formulário onde se pretenda alterar os dados do grupo
- Confirmar as alterações através do botão 

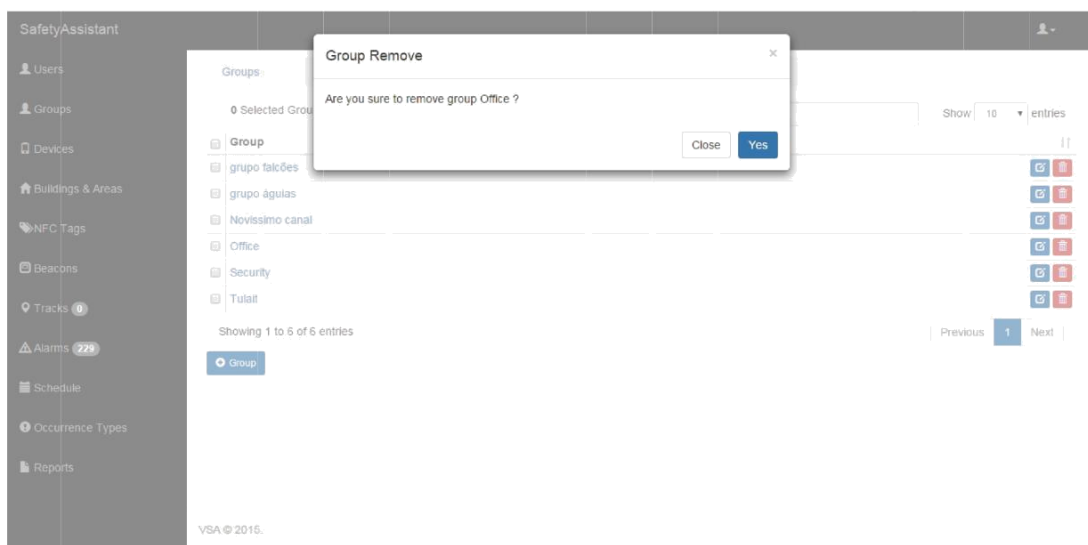


16

Grupos – BackOffice

• Remover Grupos

- Entrar na opção “Groups” do menu lateral
- Clicar no botão  relativo ao grupo que se pretende remover
- Confirmar a remoção clicando no botão “Sim” da janela de confirmação

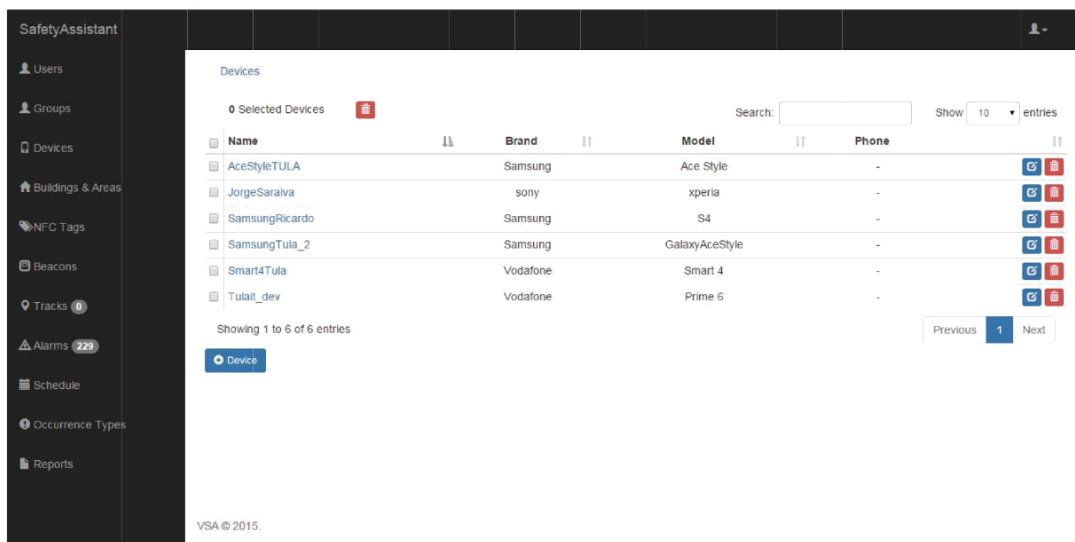


17

Dispositivos – BackOffice

• Listagem

- Entrar na opção “Devices” do menu lateral
- Podemos, assim, visualizar uma listagem de dispositivos existentes no sistema
- Nesta listagem temos acesso a alguma informação dos dispositivos, assim como botões disponíveis para criar, editar, remover os mesmos



18

Dispositivos – BackOffice

• Criar Dispositivo

- Entrar na opção “Devices” do menu lateral
- Clicar no botão Device presente no fundo da lista
- Preencher o seguinte formulário relativo ao novo dispositivo e confirmar novamente no botão Save

19

Dispositivos – BackOffice

• Geofence

- Existe na aplicação a possibilidade de restringir a utilização de um dispositivo a uma área geográfica, podendo esta ser um edifício ou mesmo um terreno ao livre.
- Para realizar esta atribuição, devemos, no formulário de criação/edição de dispositivo seleccionar o edifício / área na zona respetiva.
- Estas zonas são criadas pelo utilizador na secção Buildings & Areas (descrita mais à frente)

- Sempre que algum utilizador se desloque para fora desta zona atribuída, serão lançados uns avisos sonoros para alertar o mesmo deste facto.
- Caso estes avisos sejam ignorados, ao 5º “bip” será emitido um alerta para a central que pode ser consultado na área de alertas no separador *Geofence*. (Ver secção alertas)



Dispositivos – BackOffice



• Ver Dispositivo

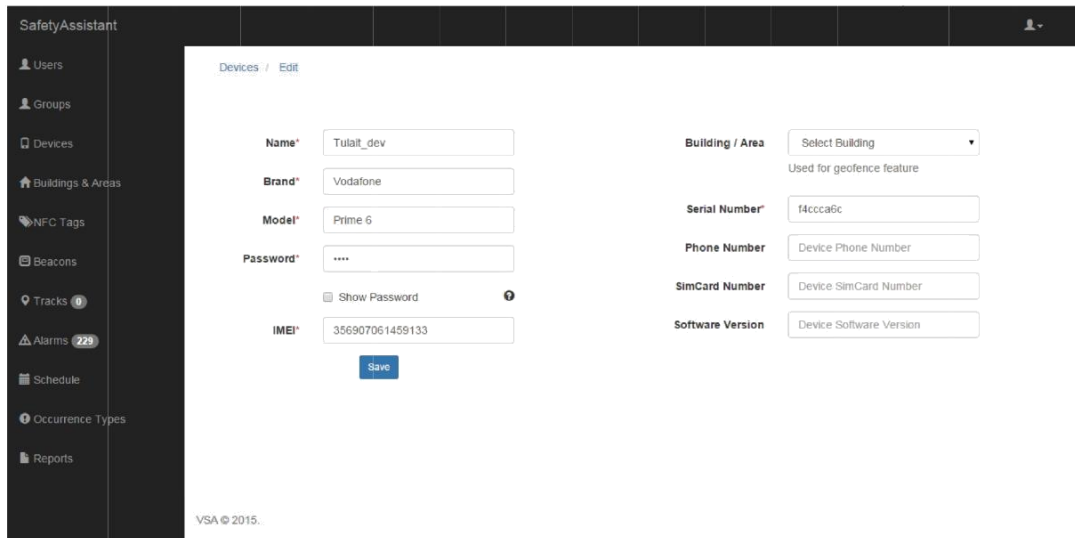
- Entrar na opção “Devices” do menu lateral
- Clicar sobre o nome do dispositivo do qual se pretende ver a informação detalhada
- É apresentada a janela com toda a informação relativa a este dispositivo

 Devices

Dispositivos – BackOffice

• Editar Dispositivos

- Entrar na opção “Devices” do menu lateral
- Clicar no botão  relativo ao dispositivo que se pretende editar
- Editar os campos do formulário onde se pretenda alterar os dados do dispositivo
- Confirmar as alterações através do botão 



SafetyAssistant

Users

Groups

Devices

Buildings & Areas

NFC Tags

Beacons

Tracks

Alarms (22)

Schedule

Occurrence Types

Reports

Devices / Edit

Name* Tulait_dev

Brand* Vodafone

Model* Prime 6

Password* ****

Show Password

IMEI* 356907061459133

Save

Building / Area Select Building

Used for geofence feature

Serial Number* f4ccca6c


Phone Number Device Phone Number

SimCard Number Device SimCard Number

Software Version Device Software Version


VSA © 2015.

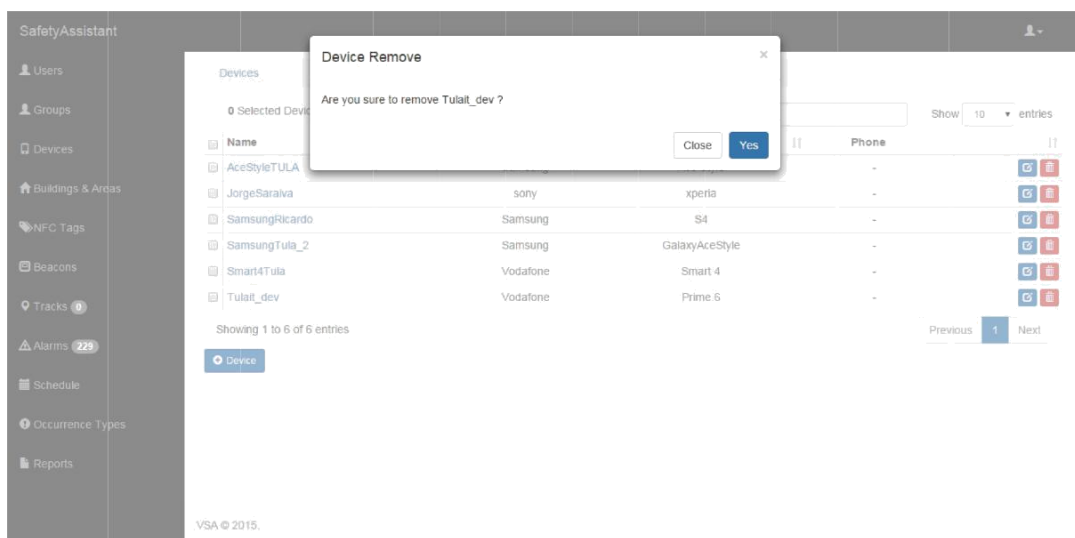
22

 Devices

Dispositivos – BackOffice

• Remover Dispositivos

- Entrar na opção “Devices” do menu lateral
- Clicar no botão  relativo ao dispositivo que se pretende remover
- Confirmar a remoção clicando no botão “Sim” da janela de confirmação



SafetyAssistant

Users

Groups

Devices

Buildings & Areas

NFC Tags

Beacons

Tracks

Alarms (22)

Schedule

Occurrence Types

Reports













Devices

0 Selected Devices

Device Remove

Are you sure to remove Tulait_dev?

Close Yes

Name	Brand	Model	Phone	
AceStyleTULA				 
JorgeSaraiva	sony	xperia	-	 
SamsungRicardo	Samsung	S4	-	 
SamsungTula_2	Samsung	GalaxyAceStyle	-	 
Smart4Tula	Vodafone	Smart 4	-	 
Tulait_dev	Vodafone	Prime 6	-	 

Showing 1 to 6 of 6 entries

Device

Previous 1 Next

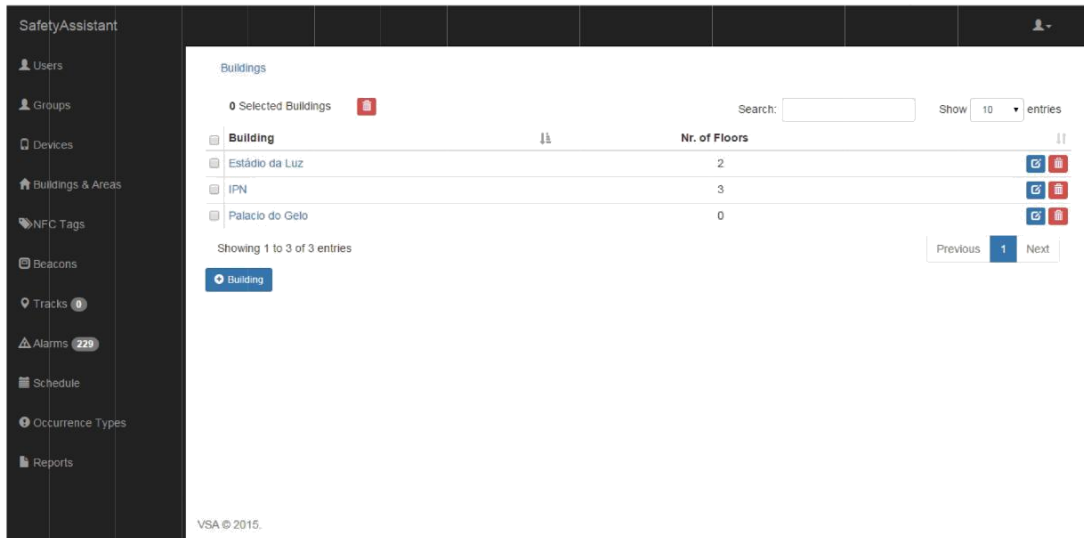
VSA © 2015.

23

Edifícios – BackOffice

• Listagem

- Entrar na opção “Buildings” do menu lateral
- Podemos assim visualizar uma listagem de edifícios existentes no sistema
- Nesta listagem temos acesso a alguma informação dos edifícios, assim como botões disponíveis para criar, editar, remover os mesmos

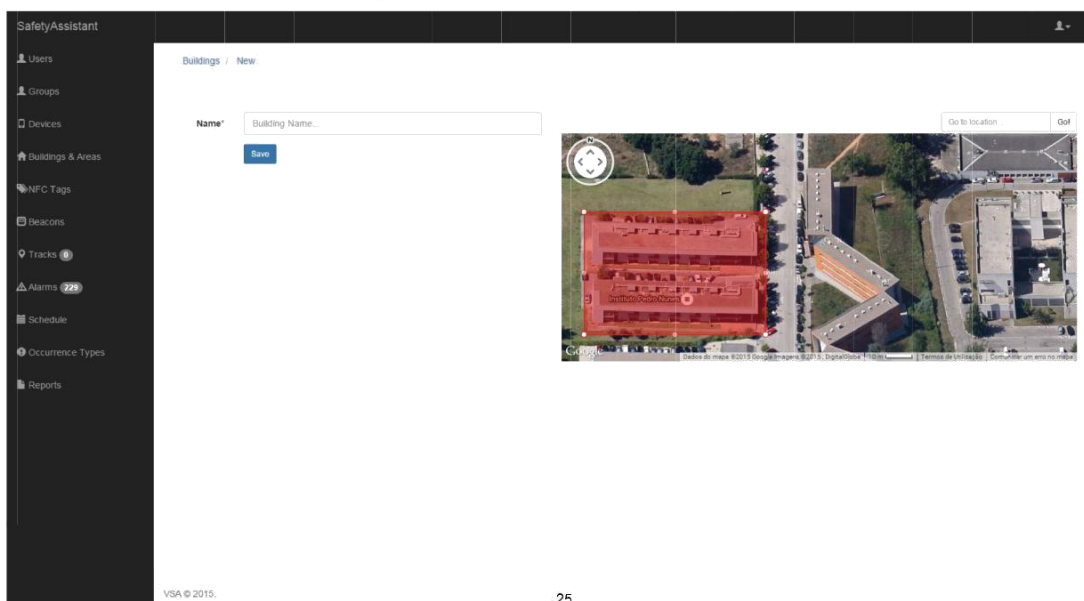


24

Edifícios – BackOffice

• Criar Edifício

- Entrar na opção “Buildings” do menu lateral
- Clicar no botão **+ Building** presente no topo da lista
- Preencher o seguinte formulário relativo ao novo edifício, definir a sua localização no mapa e finalmente e confirmar no botão **Save**

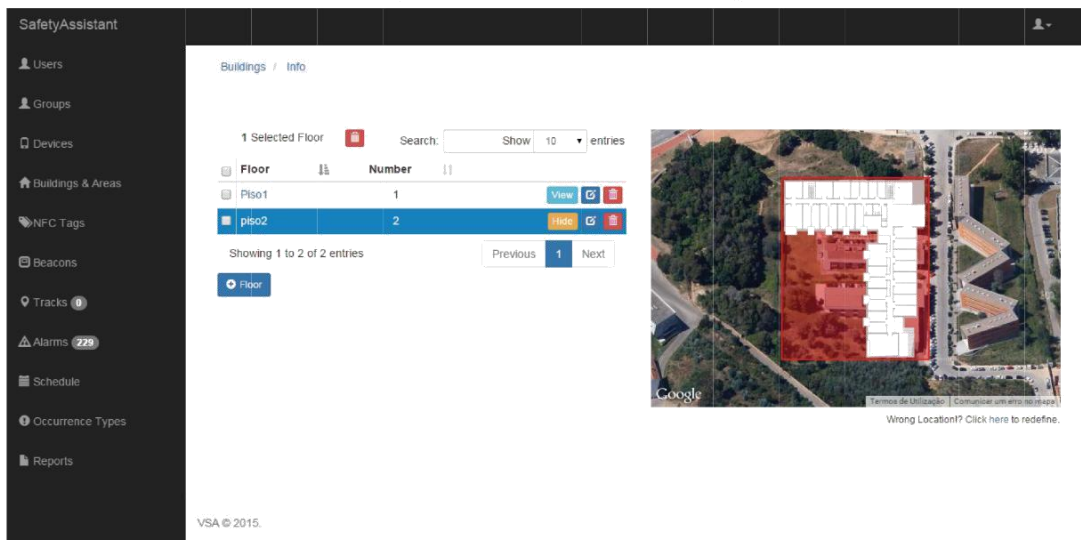


25

Buildings **Edifícios – BackOffice**

• Ver Edifício

- Entrar na opção “Buildings” do menu lateral
- Clicar sobre o nome do edifício do qual se pretende ver a informação detalhada
- É apresentada a janela com a informação deste edifício assim como uma listagem de pisos relacionados com ele. Nessa listagem temos também acesso à gestão desses pisos existentes

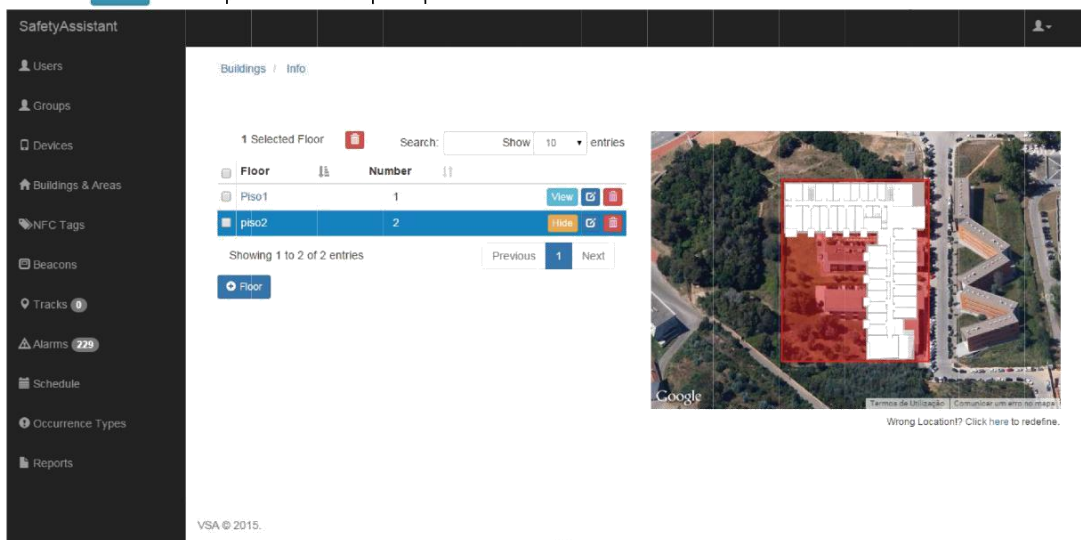


26

Buildings **Edifícios – BackOffice**

• Ver Piso do Edifício



- Entrar na opção “Buildings” do menu lateral
- Clicar sobre o nome do edifício do qual se pretende ver a informação detalhada
- É apresentada a janela com a informação deste edifício assim como uma listagem de pisos relacionados com ele. Nessa listagem temos também acesso à gestão desses pisos existentes
- Caso desejarmos visualizar a planta de algum piso sobre o mapa do edifício, devemos clicar no botão **Show** correspondente ao piso pretendido

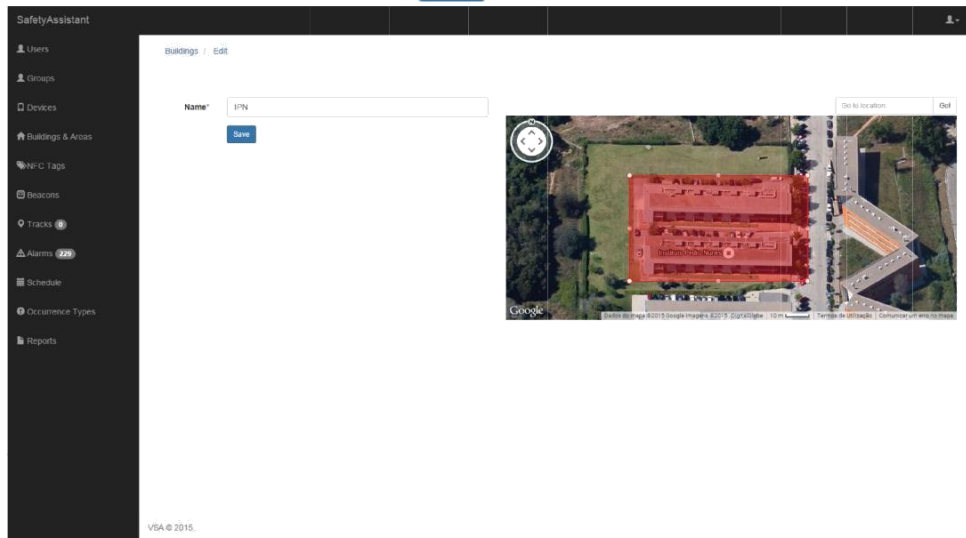


27

Buildings **Edifícios – BackOffice**

• Editar Edifícios


- Entrar na opção “Buildings” do menu lateral
- Clicar no botão  relativo ao edifício que se pretende editar
- Editar os campos do formulário onde se pretenda alterar os dados do edifício ou mesmo a localização geográfica deste
- Confirmar as alterações através do botão 

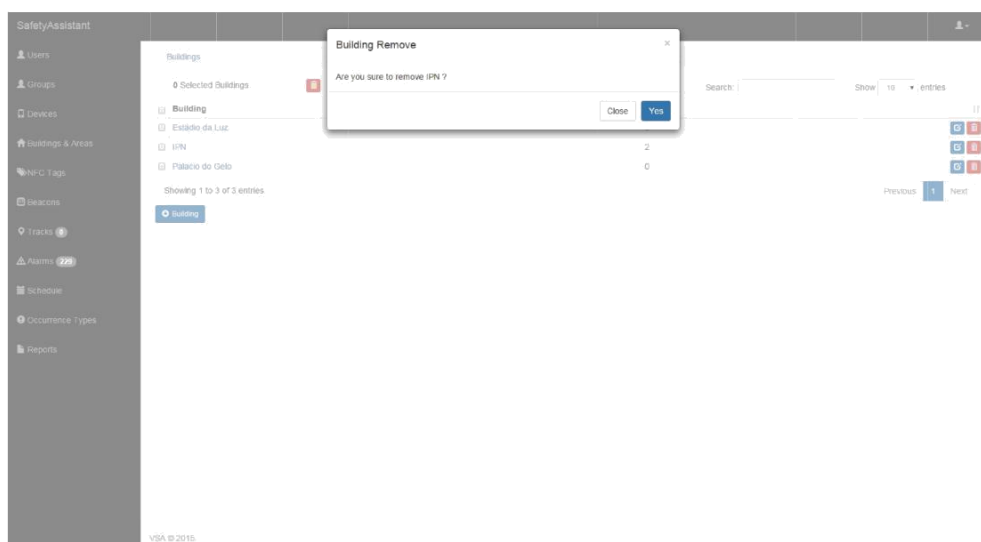


28

Buildings **Edifícios – BackOffice**

• Remover Edifícios

- Entrar na opção “Buildings” do menu lateral
- Clicar no botão  relativo ao edifício que se pretende remover
- Confirmar a remoção clicando no botão “Sim” da janela de confirmação



29

Edifícios – Pisos – BackOffice

• Criar Piso

- Entrar na opção “Buildings” do menu lateral
- Clicar sobre o nome do edifício relativo ao edifício ao qual se pretende adicionar o piso
- Clicar em **+ Floor**
- Preencher o formulário com os dados do piso e com a sua planta
- Confirmar as alterações através do botão **Save**

30

Edifícios – Pisos – BackOffice



• Ver Piso

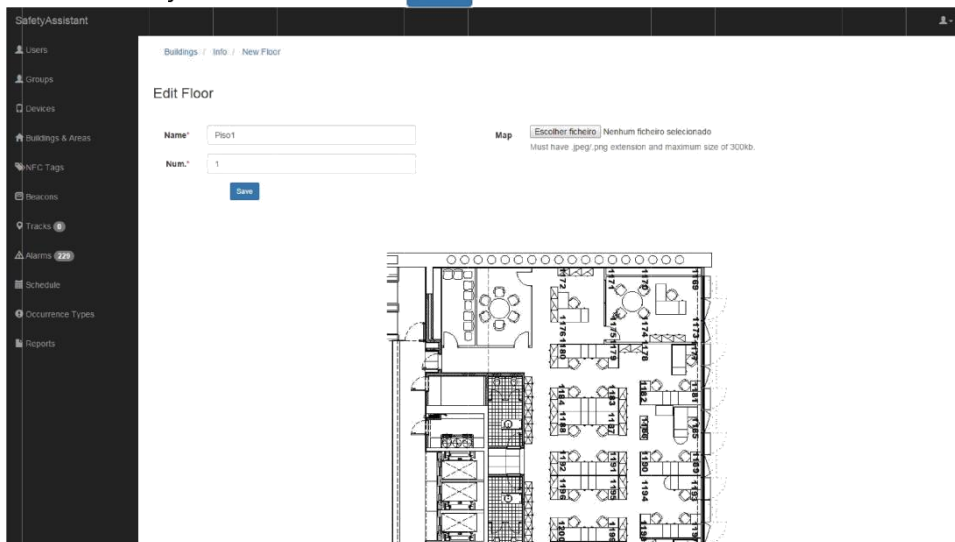
- Entrar na opção “Buildings” do menu lateral
- Clicar sobre o nome do edifício do qual se pretende ver o piso
- É apresentada a janela com a informação deste edifício assim como uma listagem de pisos relacionados com ele. Nessa listagem temos também acesso à gestão desses pisos existentes
- Na listagem de pisos, clicar sobre o nome do piso do qual pretendemos ver a informação detalhada.

31

Edifícios – Pisos – BackOffice

• Editar Pisos



- Entrar na opção “Buildings” do menu lateral
- Clicar sobre o nome do edifício do qual se pretende ver o piso
- Na lista de pisos apresentada à esquerda, clicar no botão  relativo ao piso que se pretende editar
- Editar os campos do formulário onde se pretenda alterar os dados do piso ou mesmo a sua planta
- Confirmar as alterações através do botão 

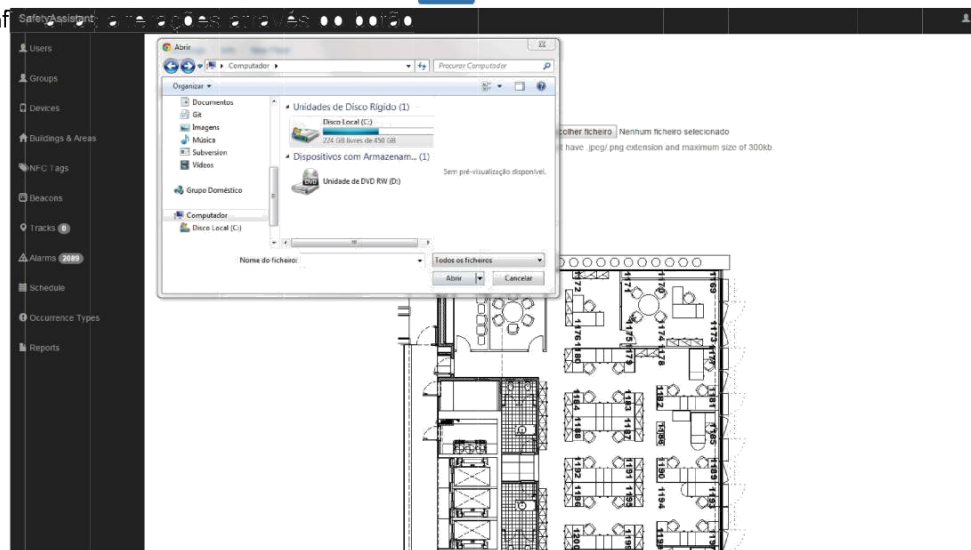


32

Edifícios – Pisos – BackOffice


• Editar Pisos – Alterar plantas

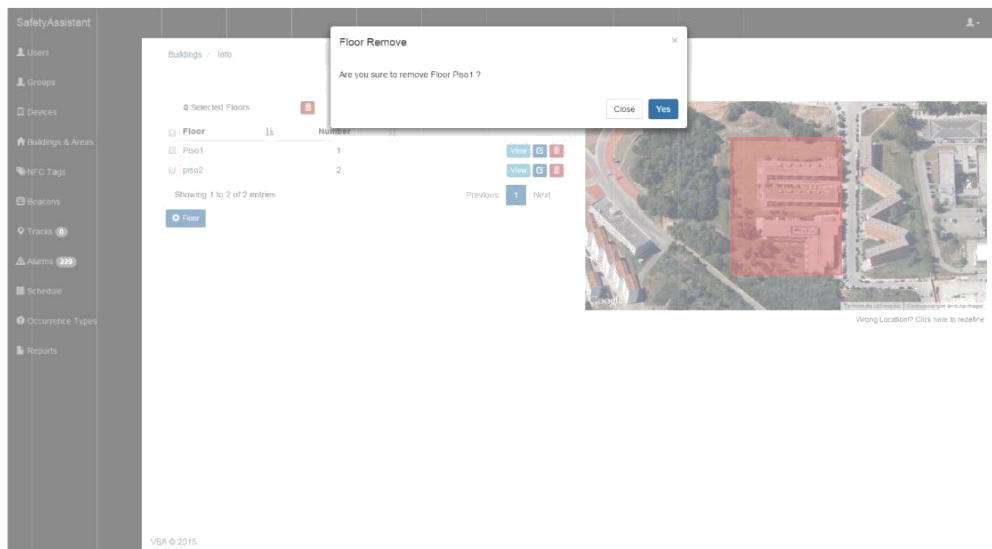
- Entrar na opção “Buildings” do menu lateral
- Clicar sobre o nome do edifício do qual se pretende ver o piso
- Na lista de pisos apresentada à esquerda, clicar no botão  relativo ao piso pretendido
- Clicar em  na secção “Map”, na janela que surgir escolhermos a imagem da nova planta.
- Conf



33

Buildings **Edifícios – Pisos – BackOffice**

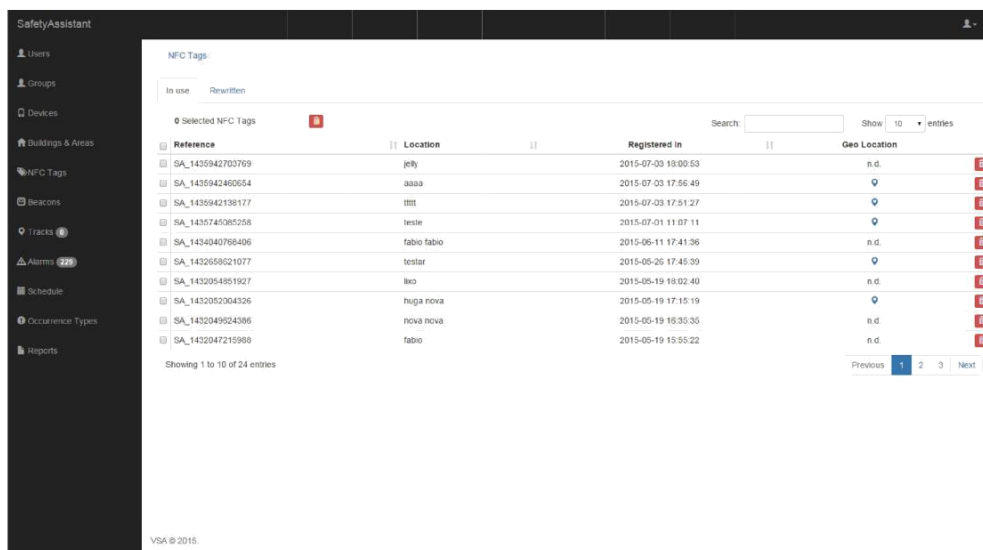
- Remover Pisos
 - Entrar na opção “Buildings” do menu lateral
 - Clicar sobre o nome do edifício do qual se pretende ver o piso
 - Na lista de pisos apresentada à esquerda, clicar no botão  relativo ao piso que se pretende remover.
 - Confirmar a remoção clicando no botão “Sim” da janela de confirmação



34

NFC Tags **Etiquetas NFC – BackOffice**

- Listagem
 - Entrar na opção “NFC Tags” do menu lateral
 - É apresentada uma listagem de etiquetas NFC existentes no sistema – Em uso e/ou reescritas.
 - Nesta listagem temos acesso a alguma informação das etiquetas, assim como um botão disponível para remover as mesmas.



35



Etiquetas NFC – BackOffice

Listagem – Etiquetas reescritas

- Sempre que alguma etiqueta NFC é reescrita, ela passa automaticamente para o estado de “reescrita”.
- Na secção “NFC Tags”, existe um separador que tem como funcionalidade mostrar a lista das tags cujo estado seja “reescrita”.

SafetyAssistant

NFC Tags

In use Rewritten

0 Selected NFC Tags

Search: Show 10 entries

Reference	Location	Registered In	Geo Location
SA_1437136100567	loc anterior	2015-07-17 10:27:47	n.d.
SA_1437136020966	novo local	2015-07-17 10:26:39	
SA_1435940359758	dev grande	2015-07-03 17:18:44	n.d.
SA_1435852549319	api 21	2015-07-02 16:55:19	
SA_1435745158510	lixo	2015-07-01 11:05:28	n.d.
SA_1434966070337	ricardo ricardo	2015-06-22 16:14:25	n.d.
SA_1434983408602	ricardo ipn	2015-06-22 15:30:00	
SA_1431940835962	goodTag	2015-05-18 10:20:06	
SA_1430819051841	ricardo 9	2015-05-05 10:44:17	n.d.
SA_1430816974651	ricardo 8	2015-05-05 10:43:14	n.d.

Showing 1 to 10 of 14 entries

Previous 1 2 Next

VSA © 2015.

36



Etiquetas NFC – BackOffice

Georreferência

- Entrar na opção “NFC Tags” do menu lateral
- Podemos assim visualizar uma listagem de etiquetas NFC existentes no sistema
- Na listagem, caso a etiqueta em questão tenha sido registada com georreferência, é possível ver a sua localização no mapa através do botão presente na coluna “Geo Location”

SafetyAssistant

NFC Tags

In use Rewritten

0 Selected NFC Tags

Reference

SA_14309427037

SA_14305424669

SA_14309421381

SA_14307450682

SA_14305407688

SA_14305686211

SA_14302054851

SA_14302052004

SA_14300496242

SA_14300472119

Showing 1 to 10 of 14 entries

Close

teste

SA_14305424669

SA_14305686211

SA_14302054851

SA_14302052004

SA_14300496242

SA_14300472119

Showing 1 to 10 of 14 entries

Previous 1 2 3 Next

VSA © 2015.

37



Escrita de Etiquetas NFC

• Criar Etiqueta NFC

- A criação de etiquetas NFC só é possível através dos dispositivos móveis. Assim, devemos proceder do seguinte modo nos terminais:



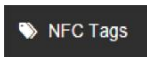
• No terminal

- No menu de opções da aplicação, está disponível uma outra funcionalidade, acessível aos utilizadores com permissão atribuída para tal:
 - Realização da escrita de etiquetas NFC com finalidade de lhes atribuir locais, sejam estes interiores ou exteriores

- Aceder ao menu de opções e selecionar a opção “Escrever etiqueta NFC”




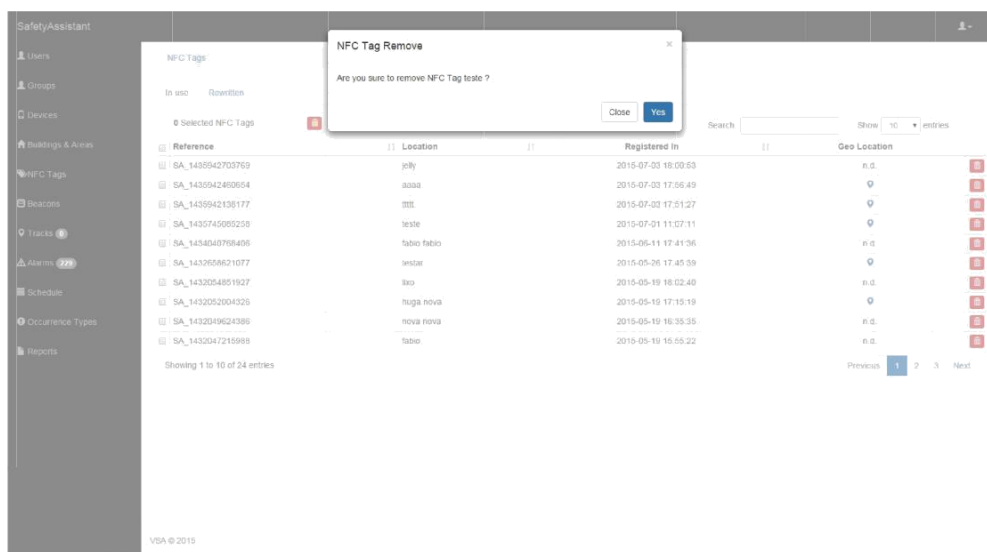
- Na nova janela, devemos escrever o nome da localização e marcar a opção “Adicionar Localização GPS”, caso seja pretendido guardar as localizações GPS do local onde o utilizador se encontra no momento da escrita da etiqueta NFC

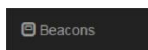


Etiquetas NFC – BackOffice

• Remover Etiquetas NFC

- Entrar na opção “NFC Tags” do menu lateral
- Clicar no botão  relativo à etiqueta que se pretende remover
- Confirmar a remoção clicando no botão “Sim” da janela de confirmação

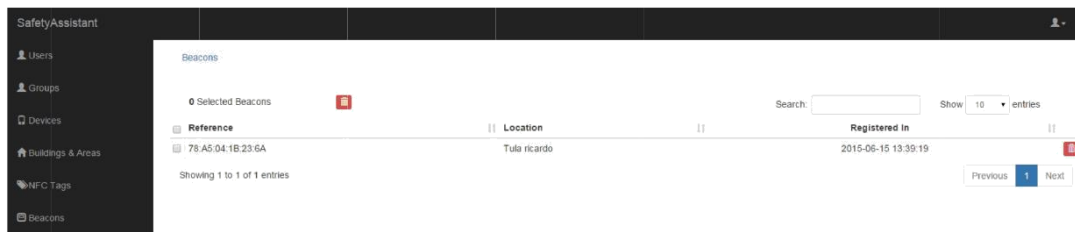




Beacons – BackOffice

• Listagem

- Seleccionar a opção “Beacons” do menu lateral
- É apresentada a listagem de dispositivos “Bluetooth Low Energy”, Beacons, registados no sistema, assim como, toda a informação relevante a cada registo.



– Referência

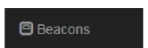
Identificador Único do dispositivo, Media access control address (MAC), identificando a interface de rede responsável pela comunicação no segmento físico da mesma.

– Localização

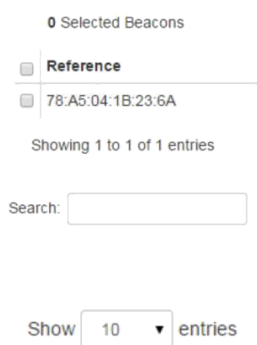
Nome atribuído à localização do beacon, facilitando a sua rápida identificação.

– Registado Em

Data à qual foi efectuado o registo do dispositivo.



Beacons – BackOffice



• Operações Disponíveis

- Selecção de Beacons

É disponibilizada a selecção individual ou múltipla de dispositivos, através de “check-box”.

- Procura de Beacons


Introdução de palavras chave na procura, seguida da apresentação dos dispositivos com informação mais aproximada à chave de procura.

- Entradas na Lista


Seleção no número de dispositivos listados por página. Disponíveis entradas de 10, 25, 50 e 100 dispositivos em simultâneo.



- Remoção de Múltiplos Beacons

É disponibilizada a remoção de múltiplos beacons através do botão  localizado à direita do texto informativo do número de beacons seleccionados. Esta operação prevê a selecção antecipada dos beacons a remover.

- Remoção de um Beacon


Seguindo o exemplo anterior, um único Beacon pode ser removido através do botão  que sucede à informação de cada dispositivo.

Beacons – BackOffice

• Operações Disponíveis

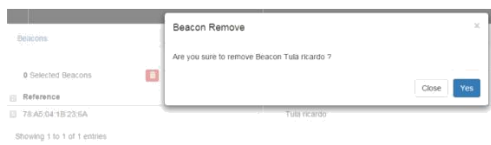
Registered In
2015-06-15 13:39:19

- Remoção de um Beacon

Seguindo o exemplo anterior, um único Beacon pode ser removido através do botão , que sucede à informação referente ao dispositivo.

- Confirmar Remoção

A remoção de um ou múltiplos beacons prevê a confirmação da operação em causa.



42

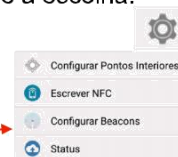


Configuração de Beacons

• No terminal

- No menu de opções da aplicação, está disponível uma outra funcionalidade, acessível aos utilizadores com permissão atribuída para tal:
 - Configuração de dispositivos “Bluetooth Low Energy”. É possível configurar o edifício a que pertencem e ainda atribuir um nome à escolha.

- Aceder ao menu de opções e selecionar a opção “Configurar Beacons”



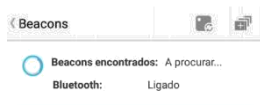
- Na nova janela surgirá a listagem dos dispositivos ao alcance, assim como informações relativas ao número de dispositivos encontrados e estado da ligação Bluetooth.



43



Configuração de Beacons



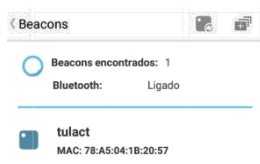
• Listagem de Beacons



– Atualizar lista de dispositivos ao alcance.



– Configurar todos os dispositivos listados.



– É também possível configurar apenas um dos dispositivos listados selecionando o mesmo na lista.

44



Configuração de Beacons



• Configuração do Beacon



• Procurar Beacon

– É possível identificar o beacon a que se está ligado no momento, fazendo soar um “bip” ou acendendo um led, de acordo com as características do mesmo.



• Restaurar Beacon

– Possibilidade de restauro das configurações de fábrica do beacon. Depois de restaurado e sincronizado o Beacon deixará de existir no servidor, possibilitando nova configuração.



• Configurar Beacon

– Configuração do Beacon com as informações introduzidas pelo utilizador. Se o beacon já se encontrar configurado estas configurações serão rescritas e atualizadas.



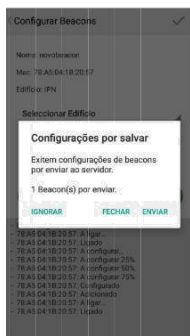
• Sincronizar Beacon(s)

– Sincronização das novas configurações e/ou restauros com o servidor.

45



Configuração de Beacons



- Configuração do Beacon

- Avisos de configuração

- O utilizador é avisado sempre que tente fechar a janela tendo alterado, previamente, a configuração de um beacon sem ter efetuado a sincronização com o servidor.



- Avisos do Terminal

- Se o Bluetooth do terminal for desligado, perdendo a ligação aos beacons, o utilizador é notificado para que sincronize com o servidor todas as modificações efectuadas.

46



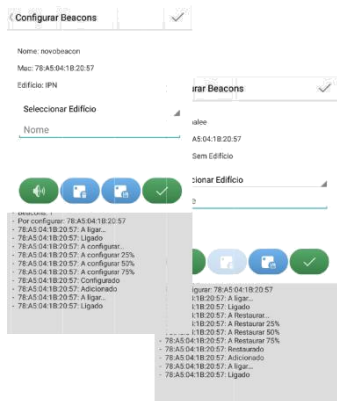
Configuração de Beacons

- Configuração do Beacon



- Sincronizar Beacon(s)

- Também é possível efetuar a sincronização com o servidor através do botão presente na barra do terminal. Este botão surge após configuração ou restauro de um beacon. Este pode ser especialmente útil na presença de múltiplos beacons, facilitando a sincronização com o servidor em qualquer altura não necessitando de configurar todos os beacons para efetuar a mesma.



- Acompanhamento de Configurações

- No decorrer da configuração ou restauro do beacon o utilizador tem presente na janela todas as informações necessárias e atualizadas referentes à evolução da acção.

47



Tracks Rondas Exteriores - Iniciar

• Registrar Rondas

- O registo de rondas só é possível através dos dispositivos móveis. Assim, devemos proceder do seguinte modo nos terminais:




• Para registar as rondas

- Clicar no ícone 
- Selecionar o tipo de ronda “Exterior”
- Confirmar mudança do ecrã para o modo ronda
- O contador regressivo de segurança irá então iniciar; este contador começa com 5 minutos
- O operador, deverá então tocar no botão  antes da contagem terminar, para que esta contagem se reinicie.
- No caso de este tempo se esgotar, será emitida uma notificação para a central




Tracks Rondas Interiores - Iniciar

• Para registar as rondas

- Clicar no ícone 
- Selecionar o tipo de ronda “Interior”, selecionar edifício e piso pretendidos



- Confirmar mudança do ecrã para o modo ronda
- O contador regressivo de segurança irá então iniciar; este contador começa com 5 minutos
- O operador, deverá então tocar no botão  antes da contagem terminar, para que esta contagem se reinicie.
- No caso de este tempo se esgotar, será emitida uma notificação para a central tal como visto anteriormente para o tipo de rondas exteriores

Rondas – Trocar para Exterior / Interiores



• Durante as Rondas

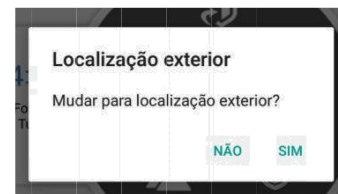
- Existe a possibilidade de, durante uma ronda, trocar entre localização de ronda interior ou exterior
- Para isso basta tocar no respetivo botão e confirmar a troca:



Troca de exterior para interior



Troca de interior para exterior



50



Rondas – Registo de passagem em pontos (etiquetas NFC)



• Durante a Ronda

- Durante uma ronda, seja exterior ou interior, é possível registar passagens por pontos referenciados através de etiquetas NFC
- Para isso basta aproximar a parte traseira do terminal a uma etiqueta e confirmar o “bip” e a mensagem “Localização Registada!”
- A partir deste momento, o ponto de passagem NFC foi registado, dando continuidade à ronda



51



Rondas – Registo de passagem em pontos (etiquetas NFC)



• Validação de Etiquetas e Procedimentos

– Localização Registada

A passagem de uma etiqueta seguida da mensagem “Localização Registada” reflete o correto registo da mesma, não necessitando de qualquer procedimento adicional por parte do utilizador.



– Etiqueta Inválida

Resposta do servidor à passagem da etiqueta. A etiqueta em questão não se encontra registada, não pertence ao utilizador ou poderá encontrar-se danificada. Como procedimento o utilizador deverá entrar em contacto com o responsável ou administrador e reportar o problema.

52



Tracks

Rondas – Registo de Ocorrências (pré-definidas)



• Durante a Ronda

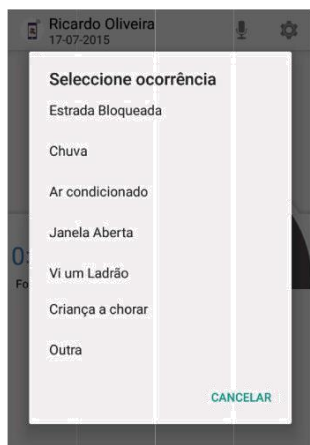
– Para registar ocorrências padrão durante as rondas, sendo estas exteriores ou interiores, devemos proceder da seguinte forma:

– Clicar no ícone



– Selecionar o tipo da ocorrência pretendido

– Caso pretendido pode adicionar uma descrição à ocorrência (não obrigatório)



53

Rondas – Registo de Ocorrências (outras)



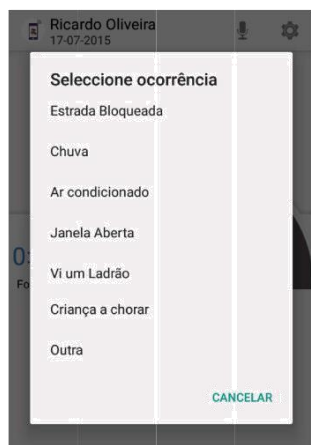
- Durante a Ronda

- Para registar ocorrências que sejam diferentes das ocorrências padrão já existentes, devemos proceder da seguinte forma:

- Clicar no ícone



- Selecionar o tipo “Outra” e escrever a descrição da ocorrência pretendida



54

Rondas – Alertas – Alerta de “Homem Caído”



- Durante a Ronda

- Durante a ronda, continua disponível a possibilidade de registar alertas. Os tipos de alertas disponíveis são os mesmos vistos anteriormente na secção de alertas e o procedimento para os registar é também o mesmo

- Adicionalmente, enquanto estivermos em ronda, está disponível o tipo de alerta de “Homem caído”:

- Este alerta é emitido automaticamente pelo terminal, quando este se encontra na posição horizontal, para prever o caso de queda do operador



Tracks

Rondas – Finalizar



• Durante a Ronda

- Assim que seja necessário finalizar a ronda em curso, basta premir o botão de “fim de ronda”



- E confirmar então a ação



Tracks

Rondas – Rondas em aberto

• Durante a Ronda

- Poderá acontecer, no momento em que se inicia uma nova ronda, que uma ronda anterior não tenha sido corretamente finalizada e continue em aberto

- Neste caso, seremos alertados dessa situação e questionados sobre a continuidade dessa ronda anterior


- Podemos então decidir entre continuar ou terminar corretamente essa mesma ronda

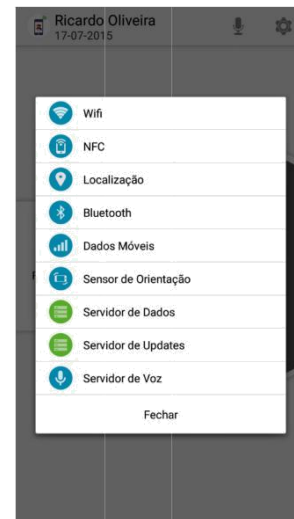


Rondas – Menu Disponível

- Durante a Ronda



- Durante qualquer tipo de ronda, o utilizador tem disponível no menu de opções, o acesso à janela de *status*.
- Para aceder a esta basta aceder a opções  e escolher a opção Status



58

Rondas – BackOffice

- Listagem

- Entrar na opção “Tracks” do menu lateral
- Podemos assim visualizar uma listagem de rondas registadas no sistema – Finalizadas ou em Decurso

Track	Live	Nr. of Occurrences	Registered by	Started	Stopped
Fábio Marques_SmartTula	X	0	Fábio Marques	2015-07-20 15:20:04	2015-07-20 15:25:06
Ricardo Oliveira_Tulst_dev	X	0	Ricardo Oliveira	2015-07-16 12:23:56	2015-07-16 12:23:59
Fábio Marques_AceStyleTULA	X	0	Fábio Marques	2015-07-03 18:01:02	2015-07-03 18:01:25
Fábio Marques_AceStyleTULA	X	0	Fábio Marques	2015-07-03 17:59:28	2015-07-03 17:59:56
Fábio Marques_AceStyleTULA	X	0	Fábio Marques	2015-07-03 17:57:03	2015-07-03 17:57:30
Fábio Marques_AceStyleTULA	X	0	Fábio Marques	2015-07-03 17:51:45	2015-07-03 17:52:32
Ricardo Oliveira_SamsungRicardo	X	0	Ricardo Oliveira	2015-07-01 11:07:22	2015-07-01 11:14:22
Ricardo Oliveira_SamsungRicardo	X	0	Ricardo Oliveira	2015-07-01 11:06:03	2015-07-01 11:06:32
Ricardo Oliveira_SamsungRicardo	X	0	Ricardo Oliveira	2015-07-01 10:05:48	2015-07-01 10:06:21
Fábio Marques_AceStyleTULA	X	0	Fábio Marques	2015-06-26 15:37:02	2015-06-26 15:37:31

59

Tracks

Rondas – BackOffice

- Listagem – Rondas finalizadas

SafetyAssistant

Tracks

Live Tracks 3 Finished Tracks 207

0 Selected Tracks

Search: Show 10 entries

Track	Live	Nr. of Occurrences	Registered by	Started	Stopped
Fábio Marques_Smart4Tula	✘	0	Fábio Marques	2015-07-20 15:20:04	2015-07-20 15:25:06
Ricardo Oliveira_Tulal_dev	✘	0	Ricardo Oliveira	2015-07-16 12:23:56	2015-07-16 12:23:59
Fábio Marques_AceStyleTULA	✘	0	Fábio Marques	2015-07-03 18:01:02	2015-07-03 18:01:25
Fábio Marques_AceStyleTULA	✘	0	Fábio Marques	2015-07-03 17:59:28	2015-07-03 17:59:56
Fábio Marques_AceStyleTULA	✘	0	Fábio Marques	2015-07-03 17:57:03	2015-07-03 17:57:30
Fábio Marques_AceStyleTULA	✘	0	Fábio Marques	2015-07-03 17:51:45	2015-07-03 17:52:32
Ricardo Oliveira_SamsungRicardo	✘	0	Ricardo Oliveira	2015-07-01 11:07:22	2015-07-01 11:14:22
Ricardo Oliveira_SamsungRicardo	✘	0	Ricardo Oliveira	2015-07-01 11:06:03	2015-07-01 11:06:32
Ricardo Oliveira_SamsungRicardo	✘	0	Ricardo Oliveira	2015-07-01 10:05:48	2015-07-01 10:06:21
Fábio Marques_AceStyleTULA	✘	0	Fábio Marques	2015-06-26 15:37:02	2015-06-26 15:37:31

Showing 1 to 10 of 207 entries

Previous 1 2 3 4 5 21 Next

VSA © 2015.

60

Tracks

Rondas – BackOffice

- Listagem – Rondas em decurso

SafetyAssistant

Tracks

Live Tracks 1 Finished Tracks 0

0 Selected Tracks

Search: Show 10 entries

Track	Live	Nr. of Occurrences	Registered by	Started	Stopped
Ricardo Oliveira_Tulal_dev	✔	0	Ricardo Oliveira	2015-07-22 11:44:21	In progress

Showing 1 to 1 of 1 entries

Previous 1 Next

VSA © 2015.

61

Rondas – BackOffice

- Ver informação de rondas
 - Entrar na opção “Tracks” do menu lateral
 - Clicar sobre o nome da ronda da qual se pretende ver a informação detalhada
 - É apresentada a janela com a informação desta ronda assim como todos os eventos com ela relacionada. Sendo estes, pontos de ronda exteriores ou interiores (NFC, WiFi e BLE), assim como ocorrências e sua descrição.

Tracks / Info

Generate Report

Name	Time	Occurrence	Description	Lat / Lon
Ricardo Oliveira_Tulaf_dev	11:45:09	Ví um Ladrão	zona 2	39.7611013 / -8.8089125
By User Ricardo Oliveira With Device Tulaf_dev	11:44:39	Janeira Aberta		39.7611003 / -8.8089092
Started	2015-07-22 11:44:21			
Stopped	2015-07-22 11:45:54			
Duration	00:01:33			

Track Points

Time	Location	Building	Floor	Type
11:45:32	loc nova	-	-	nfc
11:45:29	beacon	-	-	ble
11:45:21	loc nova	-	-	nfc
11:44:59	beacon	-	-	ble
11:44:28	sala 1	zona 2	5	wifi

NFC Points - Location Details

VSA © 2015.

62

Rondas – BackOffice

- Ver informação de rondas
 - Relativamente aos pontos de ronda NFC, podemos verificar a sua localização no mapa desde que a etiqueta NFC em questão tenha sido registada com georreferenciação. Para tal basta aceder à secção “NFC Points – Location Details” (localizada por baixo da lista de pontos) e passar o rato sobre a coluna “Show” relativa ao ponto que desejamos ver no mapa ou clicar sobre o mesmo caso se pretenda fixar este sobre o mapa.

Tracks / Info

Generate Report

Name	Time	Occurrence	Description	Lat / Lon
Ricardo Oliveira_Tulaf_dev	11:45:09	Ví um Ladrão	zona 2	39.7611013 / -8.8089125
By User Ricardo Oliveira With Device Tulaf_dev	11:44:39	Janeira Aberta		39.7611003 / -8.8089092
Started	2015-07-22 11:44:21			
Stopped	2015-07-22 11:45:54			
Duration	00:01:33			

Track Points

Time	Location	Show
11:45:32	loc nova	
11:45:21	loc nova	


NFC Points - Location Details

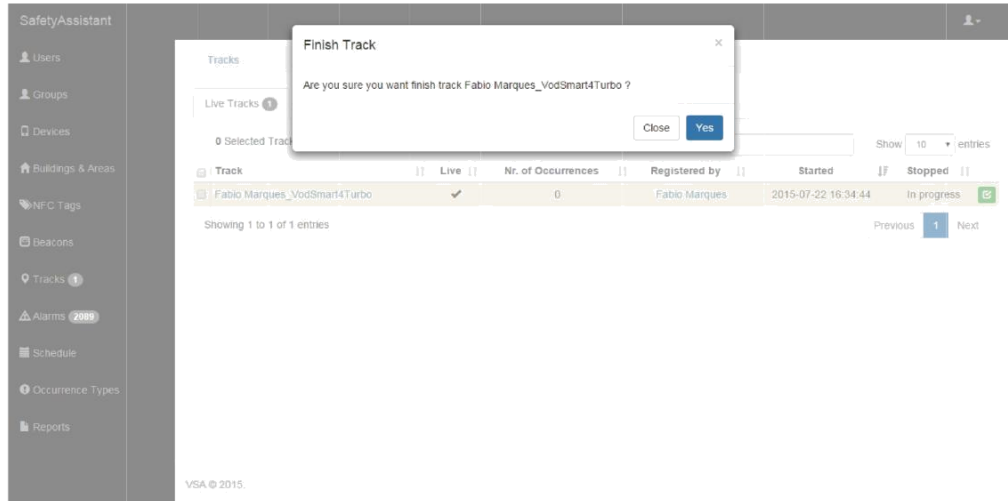
VSA © 2015.

63

Rondas – BackOffice

• Concluir Rondas


- É possível, desde que se tenha permissão para tal, concluir rondas “manualmente” no BackOffice
- Para tal, devemos aceder à opção “Tracks” do menu lateral
- Clicar sobre o separador Live Tracks
- Clicar no botão  da ronda que se pretende concluir.
- Confirmar a conclusão clicando no botão “Sim” da janela de confirmação

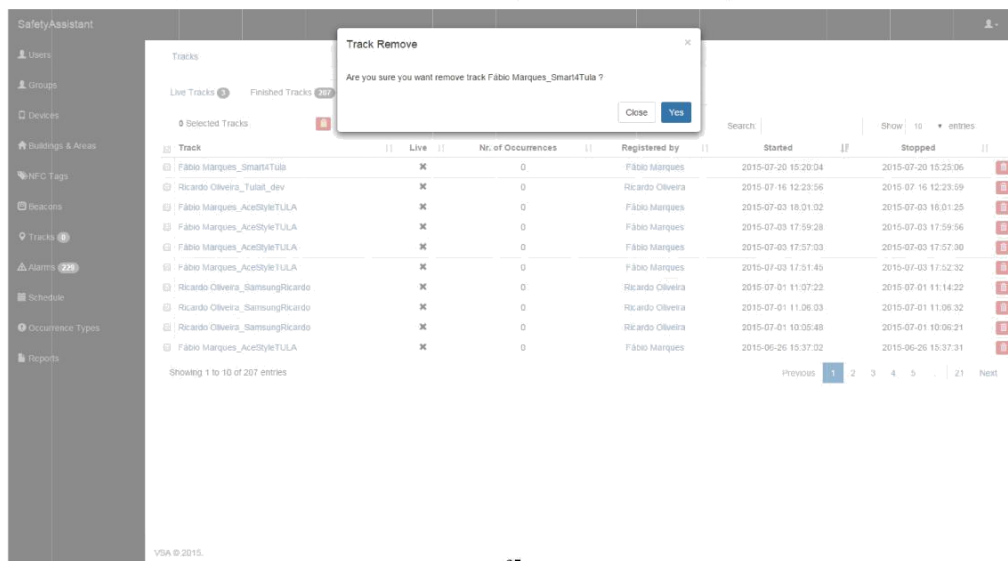


64

Rondas – BackOffice

• Remover Rondas

- Entrar na opção “Tracks” do menu lateral
- Aceder ao separador “Finished Tracks”, visto que só podemos remover rondas já finalizadas
- Clicar no botão  da ronda que se pretende remover.
- Confirmar a remoção clicando no botão “Sim” da janela de confirmação



65

Alertas

• Registrar Alertas

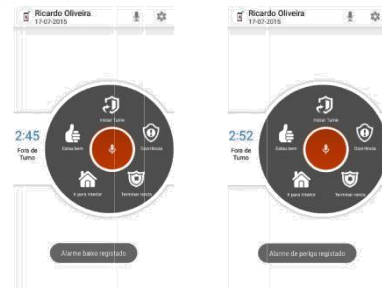
- O registo de alertas só é possível através dos dispositivos móveis. Assim, devemos proceder do seguinte modo nos terminais:



• No terminal

- Existe na aplicação a possibilidade para o utilizador emitir alertas, sempre que ocorra alguma situação que assim o justifique
- Existem dois tipos de alertas possíveis de se emitir, sendo estes:
 - O de baixa prioridade – para situações de menor importância
 - O de alta prioridade – para situações de maior importância
- Para emitir os alertas, basta manter premido durante 3 segundos os botões de volume:
 - Volume (+) – lança alerta de prioridade baixa
 - Volume (–) – lança alerta de prioridade alta

66



Alertas – BackOffice

• Listagem

- Entrar na opção “Alarms” do menu lateral
- Podemos, assim, visualizar uma listagem de alertas registados no sistema – Novos e/ou Validados.
- Nesta listagem temos acesso a alguma informação dos alertas, assim como um botão disponível para remover os mesmos

Name	Registered by	Date / Time	Level
DANGER - Fábio Marques - Smart4Tula	Fábio Marques	2015-07-16 12:11:07	Danger
LOW - Fábio Marques - Smart4Tula - 26.0 - metros network	Fábio Marques	2015-07-16 12:00:09	Low
DANGER - Fábio Marques - Smart4Tula - 59.352 - metros network	Fábio Marques	2015-07-16 11:22:49	Danger
MANDOWN - Fábio Marques - AceStyleTULA - 20.702 - metros network	Fábio Marques	2015-07-03 18:01:20	ManDown
MANDOWN - Ricardo Oliveira - SamsungRicardo - 24.0 - metros network	Ricardo Oliveira	2015-07-01 11:13:23	ManDown
MANDOWN - Ricardo Oliveira - SamsungRicardo - 34.5 - metros network	Ricardo Oliveira	2015-07-01 11:12:23	ManDown
MANDOWN - Ricardo Oliveira - SamsungRicardo - 37.5 - metros network	Ricardo Oliveira	2015-07-01 11:11:15	ManDown
MANDOWN - Ricardo Oliveira - SamsungRicardo - 34.5 - metros network	Ricardo Oliveira	2015-07-01 11:10:24	ManDown
MANDOWN - Ricardo Oliveira - SamsungRicardo - 50.874 - metros network	Ricardo Oliveira	2015-07-01 11:10:08	ManDown
MANDOWN - Ricardo Oliveira - SamsungRicardo - 50.874 - metros network	Ricardo Oliveira	2015-07-01 11:10:01	ManDown

VSA © 2015.

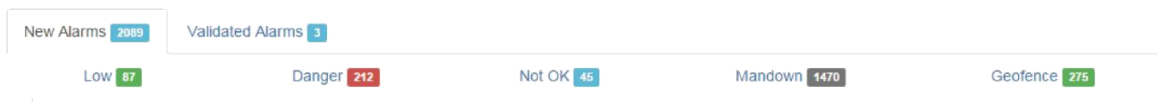
67



Alertas – BackOffice

• Listagem

- Na listagem estão disponíveis uns separadores para filtrar os alertas, sendo estes:
 - Novos alertas
 - Alertas validados




- Dentro de cada um destes separadores, existem filtros para separar todos os tipos de alertas possíveis:
 - *Low*
 - *Danger*
 - *Not Ok* – Emitido automaticamente quando o temporizador de segurança da ronda atinge o final
 - *Mandown* – Emitido automaticamente quando o dispositivo se encontrar na horizontal por mais de 5 segundos, enquanto se efetua uma ronda
 - *Geofence* – Emitido sempre que o utilizador saia do edifício atribuído ao seu dispositivo – caso exista esta atribuição



Alertas – BackOffice


• Ver Alerta

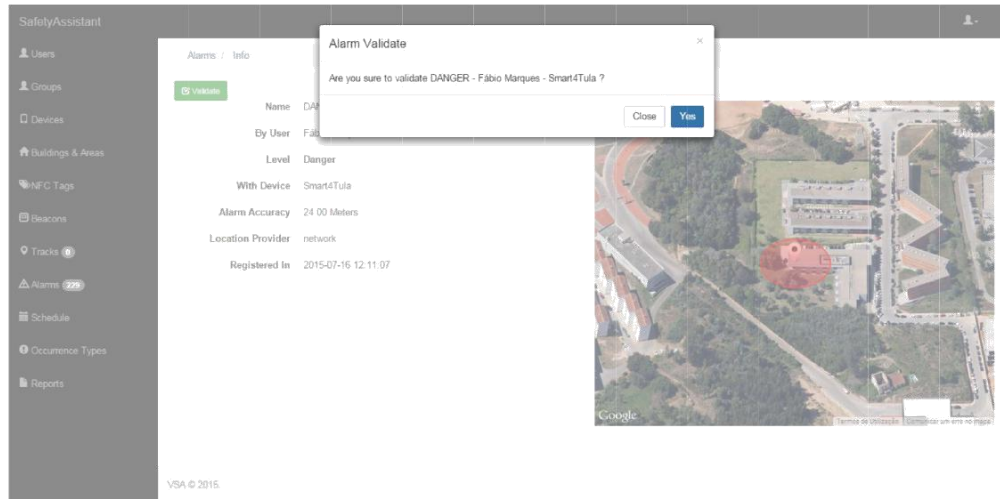
- Entrar na opção “Alarms” do menu lateral
- Clicar sobre o nome do alerta do qual se pretende ver a informação detalhada
- É apresentada a janela com a informação deste alerta bem como a sua localização e também um botão  que permite ao administrador validar os alertas emitidos.

 Alarms

Alertas – BackOffice

• Validar Alerta

- Entrar na opção “Alarms” do menu lateral
- Clicar sobre o nome do alerta do qual se pretende ver a informação detalhada
- É apresentada a janela com a informação deste alerta bem como a sua localização e também um botão  que permite ao administrador validar os alertas emitidos.
- Clicar neste botão e validar assim o alerta




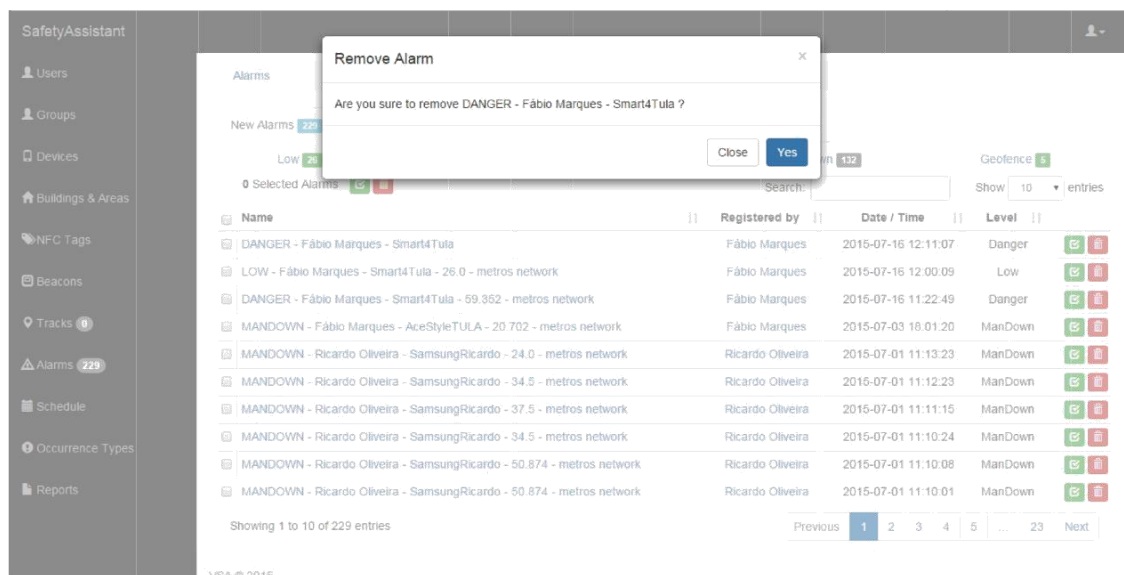
70

 Alarms

Alertas – BackOffice

• Remover Alertas

- Entrar na opção “Alarms” do menu lateral
- Clicar no botão  relativo ao alerta que se pretende remover
- Confirmar a remoção clicando no botão “Sim” da janela de confirmação



71




Entradas / Saídas ao Serviço



• Entrada ao serviço


- Para registarmos uma entrada ao serviço devemos proceder do seguinte modo:

- Clicar no ícone 
- Confirmar a entrada ao serviço



• Saída de serviço

- Para registarmos a saída do respetivo serviço, o procedimento é idêntico:

- Clicar no ícone 
- Confirmar a saída do serviço



72




Entradas / Saídas ao Serviço



• Entradas / Saídas em “aberto”

- Poderá acontecer, no momento em que se inicia a aplicação, que uma entrada ao serviço anterior, que por algum motivo, não tenha sido corretamente finalizada com a respetiva saída e continue então em aberto

- Neste caso, ao iniciarmos a app, estará disponível o ícone  relativo à saída de turno, para que possamos concluir o serviço anterior que se encontra em aberto.

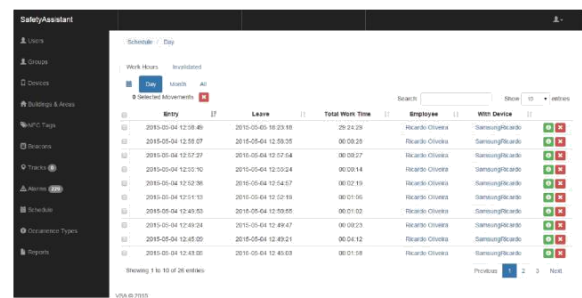
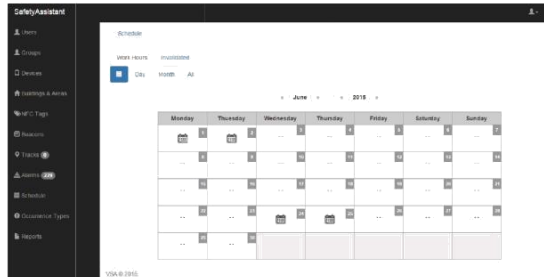


73

Schedule Entradas / Saídas ao Serviço – BackOffice

• Listagem

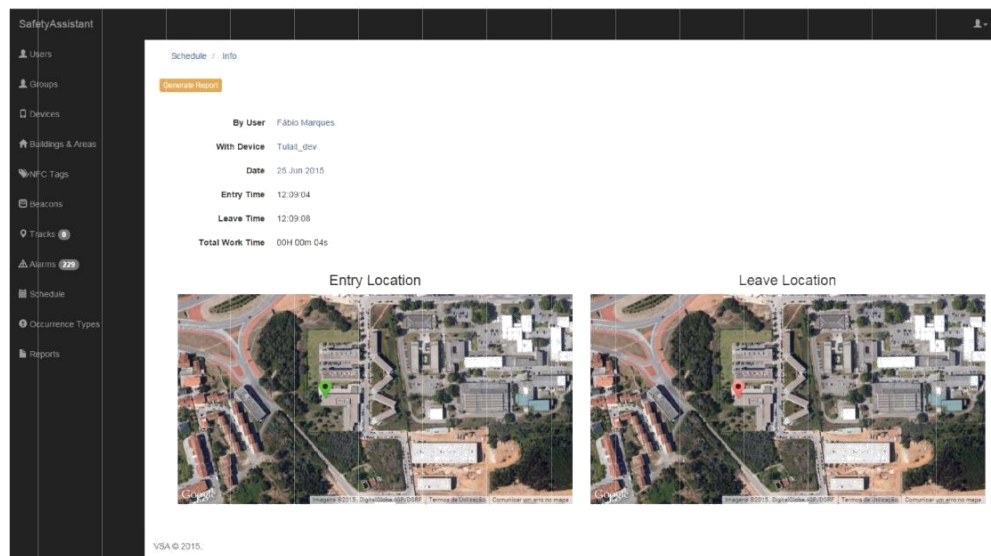
- Entrar na opção “Schedule” do menu lateral
- Clicar no dia atual (divisória com fundo verde)
- Verificar que está registada a entrada, para o utilizador, e na hora correta



Schedule Entradas / Saídas ao Serviço – BackOffice

• Detalhes

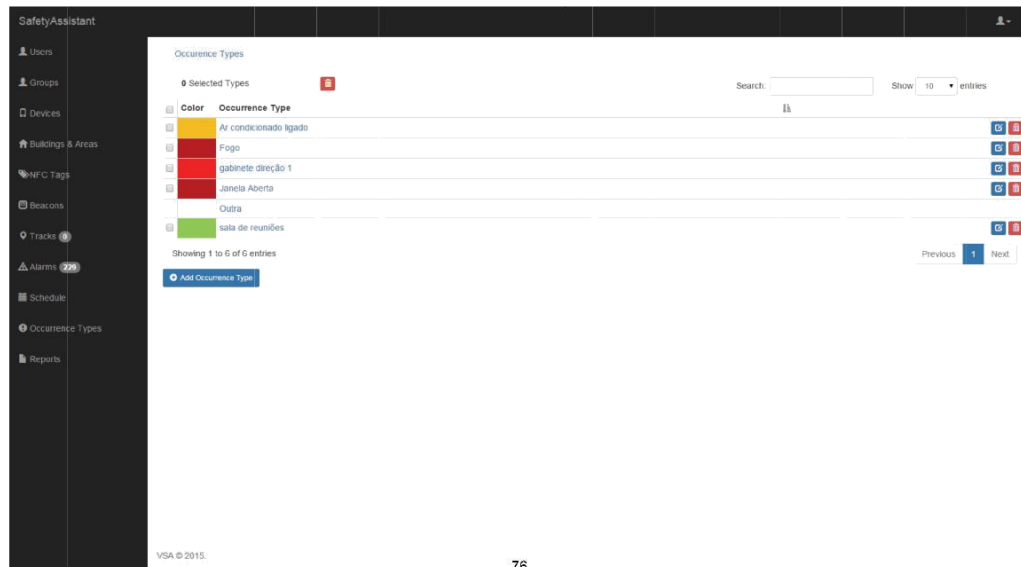
- Para visualizar os detalhes de uma Entrada/Saída, basta clicar no botão “Detalhes” para abrir a informação, onde constam, nome do utilizador, dispositivo utilizado, data e horas de entrada saída, total de tempo de serviço efetuado e localização do(s) respetivo(s) movimento(s)



Occurrence Types Tipos de Ocorrências – BackOffice

• Listagem

- Entrar na opção “Occurrence Types” do menu lateral
- Podemos assim visualizar uma listagem de ocorrências existentes no sistema
- Nesta listagem temos acesso a alguma informação das ocorrências, assim como botões disponíveis para criar, editar, remover as mesmas.

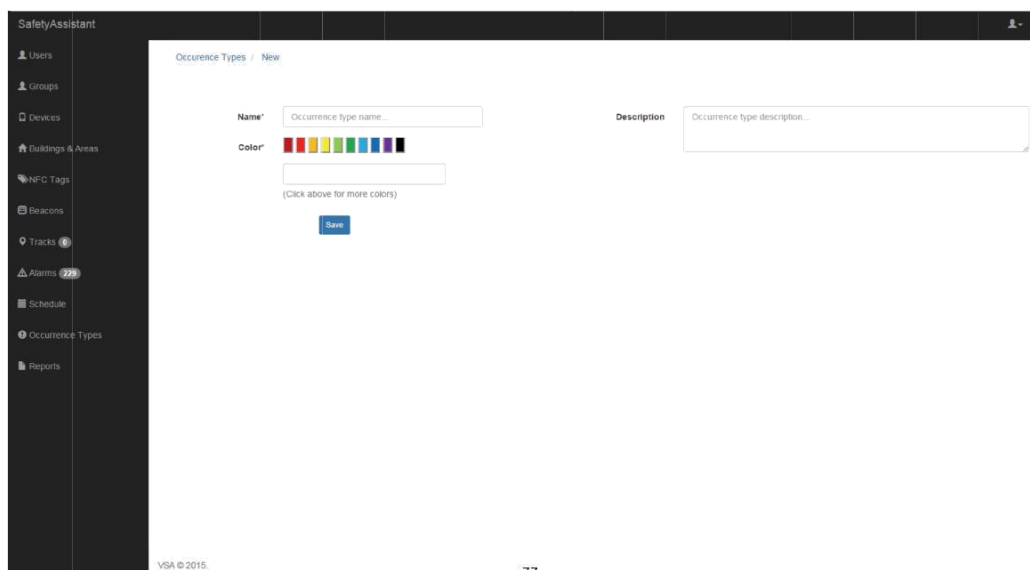


76

Occurrence Types Tipos de Ocorrências – BackOffice

• Criar Tipo de Ocorrência

- Entrar na opção “Occurrence Types” do menu lateral
- Clicar no botão **+ Add Occurrence Type** presente no topo da lista
- Preencher o seguinte formulário relativo à nova ocorrência e finalmente e confirmar através do botão **Save**

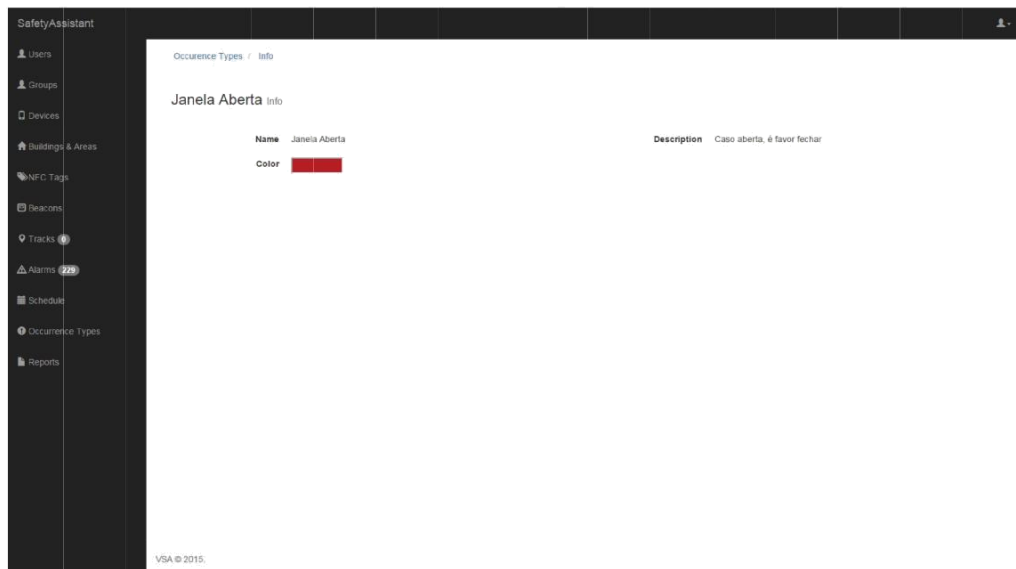


77

Occurrence Types **Tipos de Ocorrências – BackOffice**

• Ver Tipo de Ocorrência



- Entrar na opção “Occurrence Types” do menu lateral
- Clicar sobre o nome da ocorrência da qual se pretende ver a informação detalhada
- É apresentada a janela com a informação desta ocorrência

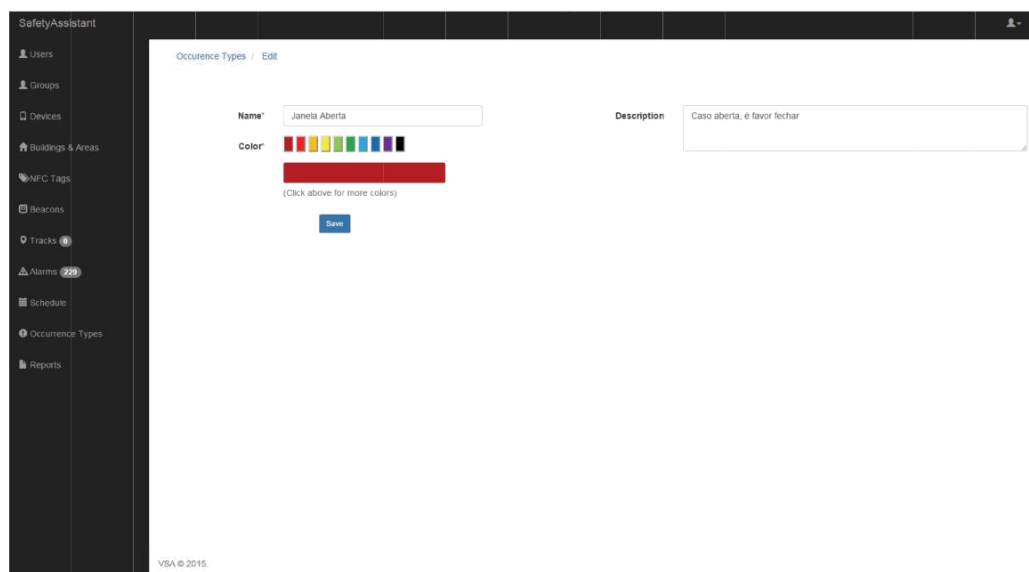


78

Occurrence Types **Tipos de Ocorrências – BackOffice**


• Editar Tipo de Ocorrência

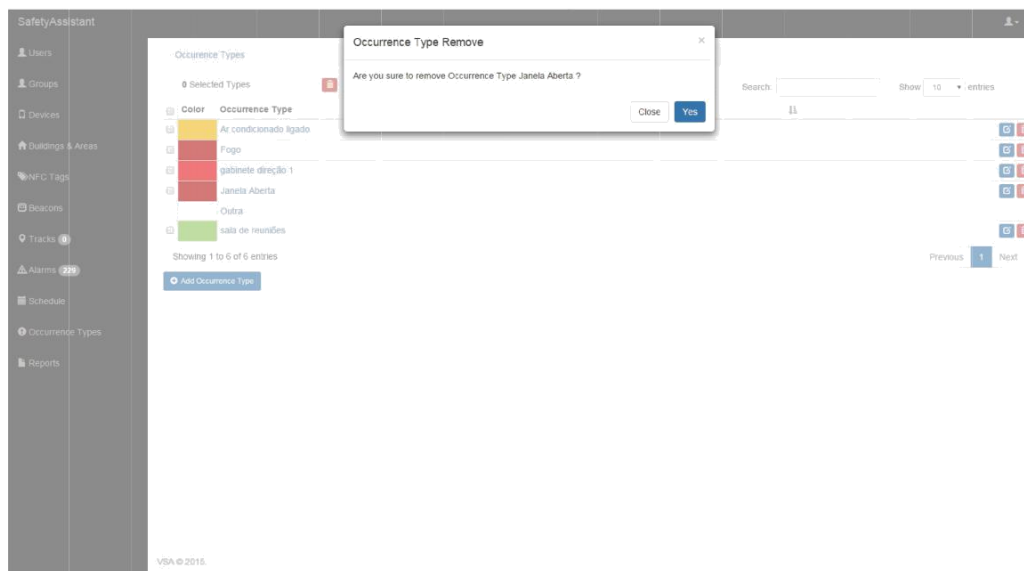
- Entrar na opção “Occurrence Types” do menu lateral
- Clicar no botão  relativo à ocorrência que se pretende editar
- Editar os campos do formulário onde se pretenda alterar os dados do tipo de ocorrência.
- Confirmar as alterações através do botão 



79

Occurrence Types **Tipos de Ocorrências – BackOffice**

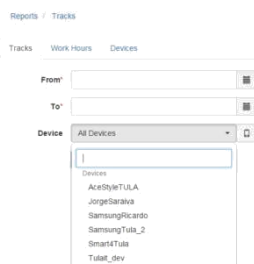
- Remover Tipo de Ocorrência
 - Entrar na opção “Occurrence Types” do menu lateral
 - Clicar no botão  relativo ao tipo de ocorrência que se pretende remover
 - Confirmar a remoção clicando no botão “Sim” da janela de confirmação



80

Reports **Relatórios – BackOffice**

- Relatórios
 - Selecionar a opção “Reports” do menu lateral
 - Possibilidade de gerar Relatórios Individuais para Rondas, Horas de Serviço e Dispositivos.
 - Cada tipo de relatório disponibiliza a seleção entre Datas e Horas sobre as quais deve incidir.
 - Possibilidade de gerar relatórios referenciando determinado dispositivo, um conjunto de dispositivos ou a globalidade dos dispositivos registados.
 - Possibilidade de introdução de datas e horas manualmente ou por seleção no calendário e relógio, respetivamente.
 - Relatórios exportados no formato PDF (*Portable Document Format*).

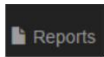


Tipos de Relatórios

• Rondas

- Relatório referente a rondas efetuadas por dispositivos, entre determinadas datas e horas.
- Possibilidade de gerar relatórios por dispositivo(s).
- O relatório gerado reflete os dispositivos em causa, assim como, o utilizador que efetuou a ronda com o/os dispositivos.

81



Relatórios – BackOffice

Reports / Work Hours

Tracks | Work Hours | Devices

From:

To:

User: All Users

- Users
- Ricardo Oliveira
- Fábio Marques
- António Paulino
- Joca
- Jorge Saraiva
- Maria
- Paulo Borges

Reports / Devices

Tracks | Work Hours | Devices

From:

To:

Device: All Devices

- Devices
- AceStyleTULA
- JorgeSaraiva
- SamsungRicardo
- SamsungTula_2
- Smart4Tula
- Tulal_dev

• Tipos de Relatórios

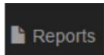
• Horas de Serviço

- Relatório referente a Horas de Serviço por utilizador, conjunto de utilizadores ou a totalidade dos utilizadores.
- Gerado entre Datas e Horas selecionadas.
- O relatório diz, apenas, respeito a Entradas e Saídas de serviço que não tenham sido Invalidadas, ou seja, que se mantenham válidas à data do relatório.

• Dispositivo

- Relatório referente à utilização de determinado dispositivo, conjunto ou totalidade dos dispositivos.
- Gerado entre Datas e Horas selecionadas.
- Compreende informações relativas a utilizadores que usaram o dispositivo(s), rondas registadas e ainda entradas e saídas de serviço registadas no dispositivo(s)

82



Relatórios – BackOffice

• Gerar Relatórios

• Selecionar Data e Hora

Tracks | Work Hours | Devices

From: 2015-07-24 01:54

To: July 2016

Device:

- Introdução Manual
- Seleção no Calendário
- Seleção no Relógio

Get Reports

Download Reports

• Gerar Relatório

- Depois de selecionada a Data/Hora pressionar o Botão “Get Reports” para gerar o relatório(s).

• Exportar Relatório

- Depois de gerados pressionar o botão “Download Reports” para efetuar o Download dos relatórios, selecionando o local onde pretende guardar o ficheiro.

- É gerado um ficheiro de extensão .zip, contendo os ficheiros .pdf correspondentes aos relatórios gerados.

83



Conta e Logout – BackOffice

- Terminar Sessão

- No menu de opções (canto superior direito) estarão disponíveis as seguintes opções: “Account” e “Logout”

- A opção “Account” permite ver/editar as informações de conta de administrador

The screenshot shows the 'Users / Edit' page in the TulaIT Admin interface. The left sidebar contains navigation options: Users, Groups, Devices, Buildings & Areas, NFC Tags, Beacons, Tracks, Alarms (201), Schedule, Occurrence Types, and Reports. The main content area is titled 'Users / Edit' and contains the following fields and options:

- Employee***: tula000
- User Level***: 9
- Name***: TulaIT Admin
- Username***: Username
- Password***: Password
- Show Password**: [checkbox]
- Group***: TulaIT
- Address**: Addr
- Telephone**: Tel
- Email**: tula@tula.eu
- Permissions**:
 - Have web access?
 - Can configure indoor points? (Wi-Fi, NFC, Beacons)
 - Can switch PTT channel?

A 'Save' button is located at the bottom right of the form area. The top right corner of the interface shows the user profile 'TulaIT Admin' with a dropdown menu containing 'Account' and 'Logout' options.

- A opção “Logout” tem como finalidade, a saída da aplicação de forma segura

