

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ENGENHARIA GEOGRÁFICA, GEOFÍSICA E ENERGIA



**Implementação de uma solução de Sistemas de Informação
Geográfica aplicado ao *Customer Relationship Management*
(*CRM*)**

Emanuel Delgado Barros

Mestrado em Sistemas de Informação Geográfica – Tecnologias e Aplicações

Trabalho de Projeto orientado por:
Prof. Doutor João Catalão Fernandes

Agradecimentos

Atinjo agora uma nova fase da minha vida, a conclusão do meu mestrado em SIG. Nesta etapa da minha vida culmina o término da minha fase acadêmica e para o reforço do meu lugar no mercado de trabalho. Embora apenas com puro esforço se atinge esta meta, existem sempre aqueles essenciais que nos ajudam diretamente ou indiretamente durante a jornada deste projeto.

Quero desde já agradecer ao meu tutor de projeto Tiago Xavier, que esteve sempre presente durante todo o desenvolvimento deste trabalho, a sua pronta disponibilidade em ajudar e companheirismo; mostrou-se sempre acessível para eu poder prosseguir em frente na empresa onde presentemente estou colocado. Agradeço também ao meu orientador, Prof. Doutor João Catalão Fernandes que me orientou e influenciou para o melhor caminho possível o trajeto deste trabalho, cujo título foi proposto por mim em conjunto com a *Unipartner IT Services*, a empresa onde estagiei durante o mestrado e que me forneceu todos os recursos necessários para a consolidação e entrega do projeto.

Agradeço também aos meus colegas de mestrado que me acompanharam por todo este percurso e grandes amigos ficaram, são eles Alexandre Rodrigues, João Gil e Fábio Madeira.

Gostaria também de agradecer à minha família pelo constante apoio, duvido que estivesse neste patamar não fosse todo o suporte que me deram até hoje. Em especial à minha mãe, Ema Barros e ao meu irmão, Diogo Barros.

Resumo

Este projeto tem como objetivo desenvolver uma Prova de Conceito (PDC), para dar resposta às necessidades espaciais de georreferenciação de uma câmara municipal. Uma instituição deste género necessita recorrentemente de gestão de recursos no seu território, resolução de ocorrências e de otimização no atendimento ao cliente. A instituição dada como exemplo será a da câmara municipal de Caldas da Rainha e das suas freguesias.

A PDC recorre a uma solução de CRM (*Customer Relationship Management*) da *Microsoft* denominada por *Dynamics 365*. Este software é constituído modularmente por aplicações, tais como *Sales*, *Customer Service*, *Marketing*, entre outras. Cada uma foca-se em funcionalidades na área da qual condiz o seu nome e serve de base para ser complementada e desenvolvida à medida de um futuro projeto. Foi utilizada a aplicação *Field Service*. Esta aplicação está sediada na *cloud* (servidores da *Microsoft - Azure*), incluindo, portanto, as suas bases de dados e *back-end* na *cloud* para armazenar registos assim como o *front-end* da própria aplicação.

Foram implementadas na solução a visualização de incidentes na câmara municipal através de um mapa (fornecido por *BING maps*), assim como a última posição e histórico de posições dos técnicos no terreno da câmara. Com a aplicação de *Field Service*, criou-se uma entidade denominada por “*Ocorrências*” que designa todos os incidentes no terreno, p.e. buracos na estrada, atividades de lazer, obras da câmara, entre outros. Os técnicos poderão marcar o estado (atributo associado a esta entidade) de uma ocorrência (p.e. ativo, concluído, etc) através de uma aplicação móvel no seu smartphone (*RESCO Mobile CRM*), que sincroniza com o CRM da câmara municipal. Esta também permite o envio de outras informações do incidente como fotografias do mesmo para posterior análise de funcionários da câmara. Esta aplicação apenas foi configurada para este caso específico.

Como último desenvolvimento e a pensar na otimização dos serviços ao cidadão, foi criado uma página .ASP que servirá de ilustrativo ao portal da câmara tendo a funcionalidade de dar a um cidadão uma senha para um serviço mais perto ou o que demora menos tempo a partir da posição atual do cidadão no terreno. Estes serviços são por exemplo, pedidos de certidões e informações num dos postos de atendimento ao público, entre outros. No caso do serviço mais perto, é fornecido o tempo total de espera sendo este a soma do tempo de viagem (calculado com a API do *Bing Maps* – o tempo com o tráfego está incluído) com a soma do tempo médio de senhas à frente do cidadão que pretende o serviço. Para o caso do serviço menos demorado, é calculado da mesma forma, porém poderá ser mais vantajoso dado que o caminho até uma freguesia poderá ter menos trânsito e menos fila de espera do que uma freguesia cujo caminho seja mais curto, mas demorado devido ao trânsito e tenha mais pessoas na fila do serviço pretendido. O site comunica com o CRM (tendo um servidor intermediário) da Câmara Municipal, sendo este utilizado pelos funcionários ao balcão para gestão de senhas e atendimento dos serviços respetivos, e o CRM comunica de volta ao site e mostra as melhores alternativas dado a posição do cidadão (obtida através da linguagem de marcação *HTML5*).

Palavras-Chave: CRM, georreferenciação, Bing Maps, Serviço no Terreno, desenvolvimento web.

Abstract

The aim of this project is to show a *Proof of Concept (POC)* which targets the spatial needs of georeferentiation of a town hall. Such an entity has often the need of managing its resources along its territory, fast resolution of incidents and customer service optimization. The example entity of this work will be Caldas da Rainha's Town Hall as well as its parish councils.

This POC uses a *Microsoft CRM* solution (Customer Relationship Management) known as *Dynamics 365*. This software is assembled in modules known simply as *apps*, such as *Sales App*, *Customer Service*, *Marketing*, and *others*. Each one has its focus in the field regarding its name and serves as a basis to be developed to the needs of a future customer. In the present case, the *Field Service Solution* was used. This application is implemented and maintained in the *cloud* along all of its data bases and *back-end* functionality to store records, including also the front-end of the app.

In terms of development, a visualization of incidents in the territory was implemented with the aid of Bing Maps Services, as well as the last known position of the town hall's technicians in the field along with the history of these positions. With this Field Service App, an entity known as "*Ocorrências*" describes all the incidents in the field such as holes in the town hall's roads, recreational activities, construction sites, etc. The field technicians can now set the status of an incident (a record of the entity *Ocorrências*) with the aid of a Mobile App in their smartphone or tablet (*RESCO Mobile CRM* app), which synchronizes with the town hall's CRM. This also enables sending information about the incidents with photos for prior analysis of the town hall's *backoffice* staff. This App was only configured for this specific case and no custom code was implemented.

As last development and with optimization of customer service in mind, an *.aspx* site was created to serve as an illustrative to its portal and enables citizens to know the nearest hall's service or least waiting time from the citizen's current position (The journey time and the service queue time). For the case of the nearest service, the total waiting time is the sum of the journey time (calculated with Bing's API – the waiting time with traffic is included) with the sum of the remaining queue tickets' mean time. For the case of the least time service, it is calculated the same way, though it can be beneficial since the course to a more distant parish council could have less transit and queue than a nearer council which could have a longer queue and more traffic for the same council service. The site is connected to the town hall's CRM (with an intermediary server), and the *backoffice* employees take advantage of using the CRM by managing tickets and regarding customer services. The CRM then gives back information about the queue's status with the best routes according to the citizen geographical position (Obtained with the HTML5 library).

Keywords: CRM, Georeferentiation, Bing Maps, Field Service, Web Development

Índice

| | |
|--|-----|
| Agradecimentos..... | iii |
| Resumo | v |
| Abstract..... | vii |
| Índice..... | ix |
| Capítulo 1 – Introdução..... | 1 |
| 1.1 Motivação..... | 1 |
| 1.2 Objetivos..... | 1 |
| 1.3 Conceito de <i>CRM</i> | 2 |
| 1.4 Conceito de <i>Field Service</i> – Serviço no Terreno..... | 3 |
| 1.5 Conceito de Business Inteligente – Power BI..... | 3 |
| 1.6 Conceito de <i>Cloud Computing</i> – Azure..... | 4 |
| 1.7 Planeamento..... | 5 |
| 1.8 Notação Adotada..... | 5 |
| 1.9 Estrutura da Dissertação..... | 5 |
| Capítulo 2 – Desenvolvimento do projeto..... | 7 |
| 2.1 Tecnologias usadas..... | 7 |
| 2.2 Metodologia..... | 8 |
| 2.3 Arquitetura Geral – Descrição Funcional..... | 9 |
| 2.4 Ambiente de Desenvolvimento..... | 10 |
| 2.5 Página ao público..... | 10 |
| 2.6 CRM – Dynamics 365..... | 15 |
| 2.7 Definição de Entidade..... | 15 |
| 2.8 Definição de <i>Dashboards</i> | 16 |
| 2.9 CRM – Criação de Entidades..... | 16 |
| 2.10 CRM – Configuração de Dashboards..... | 19 |
| Capítulo 3 – Resultados do Trabalho..... | 21 |
| 3.1 CRM da Câmara Municipal depois de todos os desenvolvimentos..... | 21 |
| 3.2 Visualização do Formulário da entidade Ocorrência..... | 22 |
| 3.3 Site da Câmara Municipal..... | 22 |
| 3.4 Dashboard Atendimento..... | 24 |
| 3.5 <i>Dashboard</i> Quadro da Agenda..... | 25 |
| 3.6 <i>Dashboard</i> Métricas..... | 27 |
| 3.7 Georreferenciação de Incidentes..... | 28 |

| | |
|--|----|
| 3.7.1 Ciclo da resolução de ocorrências | 28 |
| 3.8 CRM – <i>Dashboard</i> de Ocorrências | 29 |
| 3.9 Funcionalidades no CRM – Localização dos Técnicos no terreno. | 29 |
| 3.10 Técnicos no Terreno – Aplicação <i>Resco Mobile CRM</i> | 31 |
| 3.11 CRM Mobile | 31 |
| Capítulo 4 – Demonstração do funcionamento do Projeto | 33 |
| Capítulo 5 – Conclusão | 38 |
| 5.1 Competências Adquiridas..... | 38 |
| 5.2 Dificuldades encontradas | 39 |
| 5.3 Trabalho futuro..... | 39 |
| Referências Bibliográficas | 40 |

ÍNDICE DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1.1 – Exemplo de uma visualização PowerBI no Dynamics 365..... | 4 |
| Figura 2.1 – Planeamento do estilo Cascata utilizado neste trabalho | 9 |
| Figura 2.2 – Arquitetura da solução Field Service deste trabalho. Cada letra representa uma forma de comunicação com os componentes da solução..... | 10 |
| Figura 2.3 – Chamada AJAX Javascript | 12 |
| Figura 2.4 – Expressão FetchXML que pesquisa informação sobre os balcões..... | 14 |
| Figura 2.5- Relações entre as entidades através dos campos de pesquisa. Atenção que estas relações não representam relações normais SQL. Cada ligação representa um atributo que estará a apontar para um registo em específico da entidade que aponta. (Campos de Pesquisa). | 19 |
| Figura 2.6 - Janela da Solução UFieldService onde estão os desenvolvimentos deste projeto. Na janela estão 3 dos 4 dashboards, pois o dashboard Quadro da Agenda é nativo e está à parte desta solução no CRM..... | 19 |
| Figura 2.7 - Janela de personalização de um Dashboard. São permitidos gráficos, listas (Ex: Senhas com a categoria de Certificação), Recursos Web entre outros. Este último é o mais interessante pois permite a introdução páginas web com código à medida..... | 20 |
| Figura 3.1 - Página Principal da aplicação Field Service do Dynamics 365. Tem como dashboard por defeito um dos desenvolvimentos feitos neste projeto. | 21 |
| Figura 3.2 - Formulário da Entidade Ocorrência. Cada atributo está distribuído por secções. Existe a possibilidade de colocar páginas html nos formulários. Neste caso, a secção mapa contém uma página html que através do serviço Bing Maps, integra as coordenadas do formulário e representa-as no mapa..... | 22 |
| Figura 3.3 - Resultado mais próximo do utilizador. Cada número representa uma métrica na tabela 3.6. | 23 |
| Figura 3.4 - Site dos serviços da Câmara Municipal. Aqui, o utilizador clica na área azul que contém o destino que deseja | 23 |
| Figura 3.5 - Após clicar no seu destino, o site comunica com o CRM e este cria um agendamento para o utilizador..... | 23 |
| Figura 3.6- Dashboard do atendimento. Está dividido em 4 secções que despoletam as senhas de cada serviço | 24 |
| Figura 3.7 - Formulário de uma senha. Um formulário poderá ser depois customizado para acomodar as necessidades dos serviços da câmara..... | 24 |
| Figura 3.8 - Dashboard do Quadro de Agendamento. | 25 |
| Figura 3.9 - Código do Filtro de balcões (Entidade Recursos Reserváveis)..... | 26 |
| Figura 3.10 – Dashboard de métricas..... | 27 |
| Figura 3.11 - Ciclo de resolução de incidentes proposto. | 28 |
| Figura 3.12 - Dashboard de Ocorrências. Este dashboard é na sua totalidade uma página web..... | 29 |
| Figura 3.13 - Representação da última posição do técnico no terreno. | 30 |
| Figura 3.14 - Representação das últimas posições do utilizador "Emanuel Barros"..... | 30 |
| Figura 3.15 - Vista do CRM Mobile para um técnico no terreno. | 31 |
| Figura 4.1 - Itinerário ao destino mais próximo. | 33 |
| Figura 4.2- Itinerário ao destino mais rápido. | 33 |

| | |
|---|----|
| Figura 4.3 - Ao clicar mais vezes no destino, um utilizador está a tirar mais senhas..... | 33 |
| Figura 4.4 - As novas senhas aparecerão neste dashboard para o atendimento..... | 34 |
| Figura 4.5 - Formulário de uma senha. Aqui o utilizador poderá marcar a senha como concluída depois do atendimento. Esta deixará de aparecer no dashboard anterior..... | 34 |
| Figura 4.6 – As senhas como reservas de recurso reservável (neste caso o respectivo balcão)..... | 35 |
| Figura 4.7 – Criação de uma nova ocorrência no lado do backoffice..... | 35 |
| Figura 4.8 – A nova ocorrência do dashboard..... | 36 |
| Figura 4.9 – Formulário da nova ocorrência..... | 36 |
| Figura 4.10 – Posições registadas durante a viagem do autor..... | 37 |

INDICE DE TABELAS

| | |
|--|----|
| Tabela 1.1 – Tarefas planeadas para este projeto | 5 |
| Tabela 2.1 - Tecnologias Utilizadas no desenvolvimento deste projeto..... | 7 |
| Tabela 2.2 – Ficheiros desenvolvidos pertencentes a esta seção do projeto..... | 10 |
| Tabela 2.3 – Métodos das chamadas REST. Neste projeto apenas foram necessários os dois primeiros, GET e POST | 11 |
| Tabela 2.4 – Chamadas AJAX existentes do lado do site (cliente). | 13 |
| Tabela 2.5 - Métodos principais do lado do Servidor..... | 14 |
| Tabela 2.6 – Descrição dos atributos da entidade Ocorrência. | 16 |
| Tabela 2.7 - Dashboards a serem elaborados neste projeto. Os primeiros três dashboards foram construídos de raiz, sendo que o último é uma personalização dum dos dashboards por defeito, não sendo construído de raiz.. | 16 |
| Tabela 2.8 – Entidades criadas de início para este Trabalho. | 17 |
| Tabela 2.9 - Entidades pré-definidas do CRM aproveitadas para este trabalho. Notar que mais uma vez estas entidades possuem muitos mais atributos, que no fundo não são relevantes para o projeto. | 17 |
| Tabela 2.10 - Ficheiros com os desenvolvimentos efetuados do lado do CRM..... | 20 |

SÍMBOLOS, SIGLAS, UNIDADES E ABREVIATURAS

CRM – *Customer Relationship Management*

HTTP – *Hyper-text Transfer Protocol*

HTML – *Hyper-text Markup Language*

BD – *Base de Dados*

JSON – *Javascript Object Notation*

CSS – *Cascading Style Sheet*

IIS – *Internet Information Services*

API – *Application Programming Interface*

REST – *Representational State Transfer*

ASP – *Active Server Page*

Capítulo 1 – Introdução

Neste capítulo é apresentado o contexto do projeto começando pela motivação, seguida do objetivo, o conceito de *CRM*, o planeamento do projeto e a estrutura do documento.

1.1 Motivação

Uma Câmara Municipal tem como função a administração dum determinado munícipe, sendo associada a subdivisões denominadas por freguesias. A gestão é assegurada pelo presidente da câmara. Cabe à administração procurar servir da melhor forma os munícipes, tanto nos casos de serviços como ocorrências que alterem o normal estado do território abrangida pela câmara. Com o surgir da tecnologia e aplicações móveis, compete à administração tirar o maior partido das mais recentes novidades, de modo a aprimorar o serviço à população melhorando os processos de atendimento de modo a que não sejam tão morosos e incómodos para os cidadãos. Hoje em dia qualquer pessoa tem acesso a um *smartphone*. Este dispositivo permite que em alguns segundos, sejam efetuadas tarefas ou prestados serviços sem que o utilizador tenha de se deslocar aos postos de atendimento onde os serviços são prestados, evitando o dispêndio de energia e tempo nas deslocações e na espera pelo atendimento. Se a partir do site da Câmara Municipal um cidadão pudesse requisitar uma senha a partir do seu *smartphone*, chegar e ser atendido de imediato, seria muito vantajoso para interferir o mínimo possível com a sua vida profissional e pessoal. Melhor ainda, seria saber com exatidão o tempo da viagem dependendo do trânsito até à junta de freguesia mais próxima ou com menos fila somado com o tempo médio que deverá demorar até ser atendido.

Num município, várias ocorrências vão acontecendo pelo seu território, e uma Câmara Municipal poderia tirar partido caso pudesse, a partir de um sistema centralizado, saber a posição exata de ocorrências (buraco na estrada, obras, atividades de lazer), assim como a posição exata dos técnicos no terreno (e histórico de posições) que vão dar conta destas situações. Estes técnicos poderiam enviar imagens sobre uma determinada ocorrência para análise da Câmara Municipal. De modo a saber a presente situação de todas estas ocorrências, estes poderiam enviar informações sobre o estado da ocorrência, georreferenciação e a conclusão destas (neste caso, um *CRM* com a aplicação de *Field Service* e os desenvolvimentos descritos). Tudo isto otimiza e facilita a gestão da Câmara para dar resposta rapidamente às necessidades de todos os munícipes.

1.2 Objetivos

O objetivo deste projeto é desenvolver, com base no software *cloud Microsoft Dynamics 365*, uma prova de conceito que inclui as seguintes funcionalidades:

Funcionalidade 1 - Serviço que devolva a soma de tempo de viagem e tempo médio na fila de espera de um determinado serviço para um cidadão. Será constituído por um site que devolva a partir da posição do utilizador, a soma do tempo de caminho e também o tempo médio da fila de espera de um serviço pretendido por um cidadão.

Funcionalidade 2 - A possibilidade de incidentes poderem ser georreferenciados e tratados centralizadamente numa aplicação de *Field Service* desenvolvida de acordo com as necessidades de uma de uma Câmara Municipal;

Funcionalidade 3 - A georreferenciação da posição dos técnicos no terreno de uma Câmara Municipal, assim como todo o histórico dessas posições;

1.3 Conceito de CRM

“How you gather, manage, and use information will determine whether you win or lose.”

-Bill Gates

Desde sempre que na área de negócios o registo de vendas assim como informação relativa aos clientes foi a chave para a manutenção e crescimento das relações dos vendedores para os clientes. A elaboração e análise de relatórios sobre o histórico da atividade dos negócios possibilita uma gestão melhorada e personalizada para as necessidades de cada cliente, agilizando o retorno dos investimentos e decisões mais ponderadas, tendo como produto uma maior satisfação de cada cliente de uma determinada entidade. O conceito de *CRM (Customer Relationship Management – Gestão de Relacionamento com o Cliente)* surge como resposta a todas estas eventualidades, centralizando toda a informação num sistema desenhado especificamente para tal, automatizando muitos processos morosos e repetitivos em todo o processo de negócio que, devido ao grande volume de dados hoje em dia, seria impraticável sem uma solução do género num cenário de uma grande empresa. Com todas estas funcionalidades, a gestão de uma empresa será consideravelmente mais eficiente na medida que o software *CRM* irá disponibilizar toda a informação de clientes e negócios registados da empresa, automatizar processos, facilitar comunicações com empresas e clientes, rastrear oportunidades de negócio entre muitas outras possibilidades. No caso deste trabalho, o software em específico utilizado é denominado como *Dynamics 365 CRM* da *Microsoft*.

O software mencionado acomoda não só as anteriores situações, mas também entidades que não sejam do sector de vendas. Sejam estas instituições públicas ou organizações sem fins lucrativos, todas elas necessitam de um modo de organização e planeamento, e um software *CRM* (neste caso o *Dynamics 365*) extensível para colmatar qualquer eventualidade de gestão dando assim uma resposta rápida aos envolvidos. O software possibilita também a automatização de tarefas, de agendamentos de recursos ou localização dos mesmos no terreno, e é integrável com os mais variados softwares, assim como possibilita o *input* de vários canais de comunicação externos para registo dos mesmos mediante o seu desenvolvimento. Facultado também com automatização de relatórios e oferecendo possibilidade de desenvolvimento à medida, uma entidade é facilmente dependente de um *CRM* para garantir a sua autonomia.

1.4 Conceito de *Field Service* – Serviço no Terreno

As organizações que pretendem minimizar despesas e aumentar a produtividade de recursos móveis rapidamente procuram soluções de gestão do serviço no terreno. Estas beneficiam da otimização de custos relacionados com o número de técnicos necessários para efetuar tarefas, melhor gestão de comunicações e monitorização, assim como a redução da gestão corporativa. Exemplos mais concretos incluem localização de veículos e técnicos no território, agendamento e envio de trabalho consoante os recursos disponíveis e comunicação entre a entidade e estes recursos no terreno.

Para este caso em concreto, esta área é aplicada na medida em que uma Câmara Municipal necessitará de tratar situações de carácter espacial no seu território por técnicos (ex. piquetes, etc) que se deslocarão até à proximidade destas com o intuito de avaliar e resolver a situação, mantendo o contato com o *backoffice* sobre a atualidade do incidente, utilizando para isso software *CRM* que permite a centralização de registos que podem ser visualizados e manipulados pelos técnicos no terreno e pelo *backoffice* da câmara. A análise destes registos poderá ser fulcral para o desenvolvimento da Câmara Municipal no âmbito do seu território.

1.5 Conceito de *Business Inteligente* – *Power BI*

Irá agora ser dada uma breve introdução neste tema que indiretamente contribuem para o trabalho em questão na medida que existirão referências a eles mais à frente.

O termo *BI (Business Intelligence)* refere-se a aplicações e práticas para a recolha, integração, análise e apresentação de informação de negócio. Esta área é importante na medida que fornece visualizações das operações no negócio atuais, de histórico e preditivas, usando dados que são reunidos numa base de dados (*data warehouse*) de uma empresa. De seguida, com o software de BI apropriado, é possível fazer operações (*drill-down, slice and dice*) que nos dão uma visualização ampliada e detalhada sobre os dados da empresa, trabalhará-los fazendo relatórios estatísticos e *datamining*, onde tudo se irá transmitir em visualizações através de *dashboards* e gráficos que apresenta a informação agregada com o intuito de poder otimizar o retorno do negócio numa próxima escolha estratégica a este nível.

O *Microsoft PowerBI* é uma ferramenta desta área que permite análises de dados das mais variadas fontes de dados. Existem muitas integrações com este software das quais se destaca a integração nativa com o *Microsoft Dynamics 365*. Configurando como fonte de dados a base de dados de uma instância de *Dynamics*, este software oferece aos analistas de uma empresa uma potente e agregada visão geral do negócio da empresa. É implementado como um *dashboard* numa instância de CRM e facilmente configurável sem a necessidade de desenvolvimento.

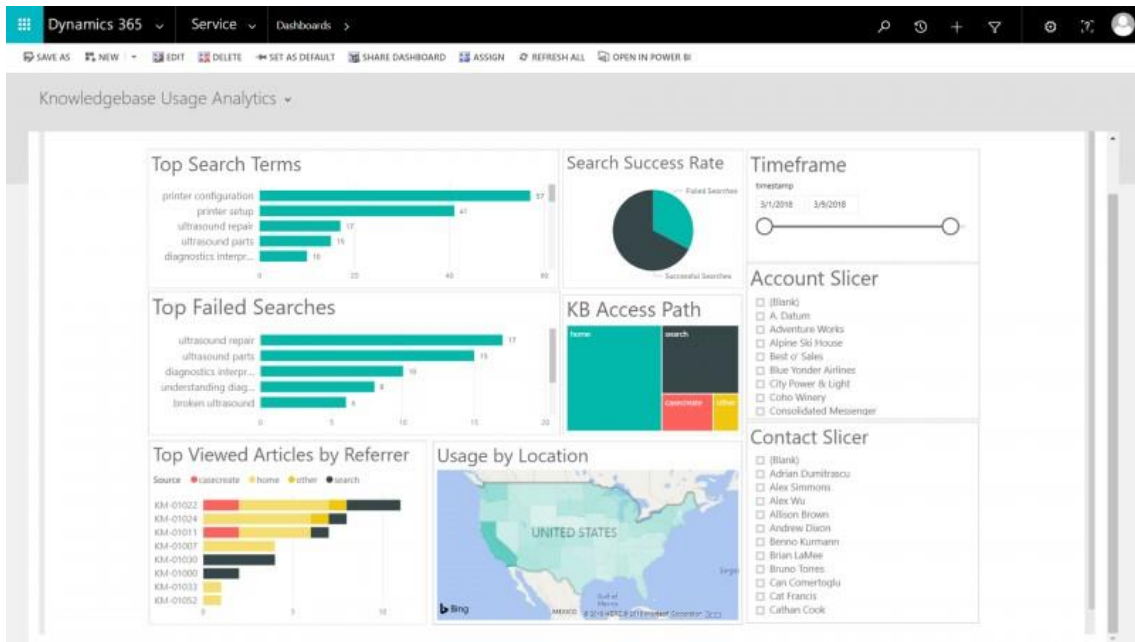


Figura 1.1 – Exemplo de uma visualização PowerBI no Dynamics 365.

1.6 Conceito de *Cloud Computing* – *Azure*

Este conceito traduz-se basicamente na entrega de serviços de computação, incluindo servidores, armazenamento, bases de dados, trabalho em rede, *software*, análises e inteligência artificial pela internet (a *cloud*) de modo a oferecer uma maior inovação, recursos flexíveis e economias de grande escala. Para este tipo de serviços, onde um utilizador não necessita de adquirir o hardware e armazená-lo, são aplicadas, no entanto tarifas de acordo com os serviços e recursos usados, que podem ser ajustados como forma de otimizar estes custos e usar esta infraestrutura de uma forma mais eficiente, assim como escalar as necessidades conforme o crescimento do negócio. Os benefícios deste tipo de computação são imensos, pois para além de não se necessitar de adquirir hardware e espaço físico, assim como tratar de toda a manutenção deste, estes serviços podem ser usados à escala global, existe um maior desempenho e segurança assim como a consequente fiabilidade e produtividade. As *clouds* poderão ser públicas, privadas ou híbridas. Uma *cloud* pública é gerida e mantida por uma entidade que fornece os seus serviços através da internet. Toda a infraestrutura é mantida por esta entidade. Uma *cloud* privada existe quando uma empresa contém toda a sua infraestrutura informática em instalações próprias. Por último, temos a *cloud* híbrida, onde é feita uma combinação destes dois tipos de *cloud* de forma a otimizar custos.

Os usos da *cloud* são imensos, e é possível criar aplicações e sediá-las através da web, fazer *backups*, usar máquinas virtuais, transmitir áudio e vídeo, usar modelos de inteligência artificial entre muitos outros.

O *Microsoft Dynamics 365* está presente no *Microsoft Azure*, que é um serviço *cloud* da *Microsoft*. Existem muitos outros como a *Amazon Web Services*, *Google Cloud*, *IBM*, entre muitos outros. O *Dynamics 365* está disponível exclusivamente na *cloud Azure*, pois são ambos produtos da *Microsoft*.

1.7 Planeamento

Qualquer projeto de longo prazo beneficia de um planeamento antecipado de modo a tentar perceber todos os requisitos necessários, evitar situações imprevistas e completar os objetivos atempadamente. A presente secção mostra uma tabela com todas as tarefas realizadas no projeto, assim como a data de início e fim das mesmas.

Tabela 1.1 – Tarefas planeadas para este projeto

| # | Tarefa | Data de início | Data de fim |
|-----|--|----------------|-------------|
| 1 | Componente Aplicacional | 08/10/2018 | 08/02/2019 |
| 1.1 | Planeamento | 17/10/2018 | 28/10/2018 |
| 1.2 | Estudo das Tecnologias a usar no projeto | 20/10/2018 | 15/01/2019 |
| 1.3 | Contexto e requisitos do projeto | 24/01/2019 | 08/02/2019 |
| 2 | Desenho da Arquitetura Geral | 27/02/2019 | 04/03/2019 |
| 3 | Desenvolvimento aplicacional e testes | 08/03/2019 | 21/06/2019 |
| 3.1 | Codificação das funcionalidades | 03/03/2019 | 11/06/2019 |
| 3.2 | Testes | 20/06/2019 | 21/06/2019 |
| 4 | Relatório do Projeto | 30/07/2018 | 22/10/2019 |

1.8 Notação Adotada

A língua adotada para este documento foi o Português de acordo com o novo acordo ortográfico. Para a bibliografia e citações, foi usado o sistema padrão *Havard*. Todo o vocabulário de outra língua está apresentado em itálico.

1.9 Estrutura do Documento

O presente documento consiste em cinco capítulos onde cada um apresenta os vários conteúdos que fazem parte deste projeto.

Capítulo 1 – O primeiro capítulo deste trabalho descreve a sua motivação e objetivo. Apresenta também o conceito de *CRM*, o planeamento, a notação adotada e a estrutura desta dissertação.

Capítulo 2 – Este capítulo apresenta a arquitetura do software Dynamics 365, bem como uma breve explicação de onde se inseriram os desenvolvimentos deste trabalho.

Capítulo 3 – Neste capítulo é apresentado o projeto tecnicamente, desde a arquitetura geral até às ferramentas usadas, ambientes de desenvolvimento e o funcionar do projeto.

Capítulo 4 – São apresentados testes sobre a aplicação.

Capítulo 5 – Neste capítulo são apresentadas as conclusões relativas ao projeto desenvolvido, assim como o estado atual do projeto, possíveis desenvolvimentos e perspetivas de trabalho futuro.

Capítulo 2 – Desenvolvimento do projeto

Dadas as bases e os conceitos de *CRM*, segue-se agora em detalhe como foi desenvolvido o projeto, metodologias, e tecnologias usadas.

2.1 Tecnologias usadas

Qualquer desenvolvimento hoje em dia assenta sempre em ferramentas ou *APIs* previamente definidas com a tecnologia do estado da arte. Isto garante a minimização de esforço no desenvolvimento e aproxima a melhor experiência de um utilizador.

Tabela 2.1 - Tecnologias Utilizadas no desenvolvimento deste projeto.

| Tecnologia | Finalidade |
|--------------------|---|
| ASP.NET | Framework de Desenvolvimento Web |
| HTML | Desenvolvimento do esqueleto de páginas web |
| CSS | Desenvolvimento do <i>design</i> de páginas web |
| Javascript | Desenvolvimento de <i>user interface web</i> |
| Jquery | Desenvolvimento de <i>user interface web</i> |
| Json | Linguagem de Comunicação entre pedidos |
| FetchXML | Linguagem de query do <i>Dynamics 365</i> |
| Metro | Framework de desenvolvimento de <i>design web</i> |
| Bing Maps | Framework de Mapas da <i>Microsoft</i> |
| Dynamics 365 | CRM utilizado neste trabalho |
| Resco Mobile CRM | App <i>Android</i> para o acesso ao <i>Dynamics 365</i> |
| Visual Studio 2017 | Ambiente de Desenvolvimento integrado |

Segue agora alguma descrição mais detalhada de cada uma destas tecnologias.

ASP .NET – (*Active Server Pages*) Plataforma *open-source* que consiste numa *web application framework* do lado do servidor, e permite colocar numa página web *scripts* que irão ser executados por um servidor. O servidor é conhecido como IIS (*Internet Information Services*), e já se encontra disponível com a instalação desta tecnologia. A página web respetiva tem a extensão *.aspx* e no seu *frontend* está presente código *html* como uma página web normal, que suporta *CSS* e *Javascript*. O código no *backend* é codificado em *C#*.

HTML (*HyperText Markup Language*) – Linguagem padrão utilizada na construção de páginas web para ser lida num browser web. Desenvolvido por Tim Berners-Lee em 1990, é a mais usada linguagem para desenvolver páginas web. A versão atual é a *HTML5*, que foi lançada em 2014.

CSS (*Cascading Style Sheet*) – Linguagem padrão de estilos que permite alterar a aparência dos elementos *HTML* para efeitos de *design* em linguagens *HTML* ou *XML*. O *CSS* descreve como os elementos devem ser renderizados no ecrã. É um padrão *W3C*.

Javascript – Esta linguagem permite a interação entre um utilizador e as páginas web. É essencial para as aplicações web, dado que é a partir desta linguagem que é definido o comportamento de uma página web consoante as ações do utilizador.

jQuery – Consiste numa biblioteca de funções que tem como intuito a simplificação do desenvolvimento em *Javascript*. Das suas características destacam-se as animações visuais assim como as chamadas *AJAX* que permitem a comunicação de páginas web com sistemas externos (ex. servidores).

JSON (*Javascript Object Notation*) – Formato de dados que serve para comunicação destes dados entre diferentes sistemas. Destaca-se a facilidade de leitura humana.

FetchXML – Linguagem de *queries* utilizada na consulta de dados do *Dynamics 365* baseada em *XML*. Pode ser usada tanto em ambiente *front-end* como *backend*.

Metro – Framework *CSS* gratuita que torna o desenvolvimento do design mais rápido. É especialmente baseada no estilo *Metro* nativo do *Microsoft Windows 10*. Para ser usada em ambiente web.

Bing Maps – Sistema web de visualização de mapas desenvolvido pela *Microsoft*. Lançado em 2005.

Dynamics 365 – CRM utilizado neste projeto desenvolvido pela *Microsoft*.

Resco Mobile CRM – App *Android* desenvolvida pela *Resco* que permite aceder ao *Dynamics 365* através do seu *layout mobile*.

Visual Studio 2017 – Plataforma de desenvolvimento de aplicações da *Microsoft*.

Em termos de equipamento, foi utilizado maioritariamente um portátil *Dell Latitude E6430*, i53340M CPU @ 2.70GHz com 8GB de memória com processador 64-bit e o sistema operativo *Windows 10*. Os desenvolvimentos foram elaborados na linguagem principal *Javascript* e *C#* em ambiente *Visual Studio 2017*. Destaca-se também o evidente uso das restantes linguagens web, como *html5*, *CSS3*, *Javascript* e *jQuery*. No desenvolvimento da página web da Câmara Municipal recorre-se também à biblioteca *Metro* de forma a agilizar o estilo da página. O *smartphone* usado tratou-se de um *Xiaomi Redmi note 5*, com 4GB de *RAM*, processador de *Snapdragon 636*. Há que ter em conta o *servidor web* usado para fins de teste, pois o site ao público do projeto apenas foi testado localmente. Este é denominado por *IIS*. É um *servidor web* flexível com uma arquitetura em escala e aberta com o objetivo de armazenar aplicações *Web* e dados. No nosso caso, foi usada a versão *IIS Express*, ferramenta que acompanha o *Visual Studio 2017* que permite testar apenas com um clique todos os desenvolvimentos efetuados.

2.2 Metodologia

De modo a tentar unir duas áreas diferentes que é o trabalho de mestrado e o projeto da empresa em que o autor trabalha, foi necessário encontrar um âmbito comum. Este foi o de *Field Service* que respondia tanto às necessidades do projeto da empresa e ia de encontro ao contexto do mestrado de *SIG*. Nesta empresa, existem clientes que pedem os serviços de consultoria, manutenção e desenvolvimento de *CRM*, no entanto, a *Unipartner* nunca tinha prestado trabalhos a clientes num contexto de *Field Service*. De modo a conciliar os interesses da empresa e a temática do mestrado na área de *SIG*, decidiu-se fazer uma prova de conceito para que fosse divulgada pela empresa e clientes.

No início do projeto, houve a necessidade de aplicar este conceito a algo mais prático e não só demonstrar funcionalidades sem as ter aplicado. Foi decidido aplicar este conceito numa Câmara Municipal devido à sua natureza muito abrangente.

A metodologia usada envolve experiência em projetos académicos anteriores assim como experiência no contexto profissional. O próprio atraso no trabalho evidenciou o avanço na tecnologia. Depois de ser planeada a arquitetura do projeto e estar a meio deste desenvolvimento, já não havia necessidade de colocar uma página aberta ao público num servidor à parte pois a *Microsoft* tinha lançado a tecnologia *Portals*, que permitia o desenvolvimento de uma página web no próprio *CRM*, e esta era ligada nativamente à instância principal.

A metodologia usada baseia-se no modelo cascata, onde cada etapa constitui uma solução à medida. É caracterizada pelas etapas de Planeamento, Análise, Desenho, Desenvolvimento e Testes. A figura 2.1 retrata o progredir destas etapas.

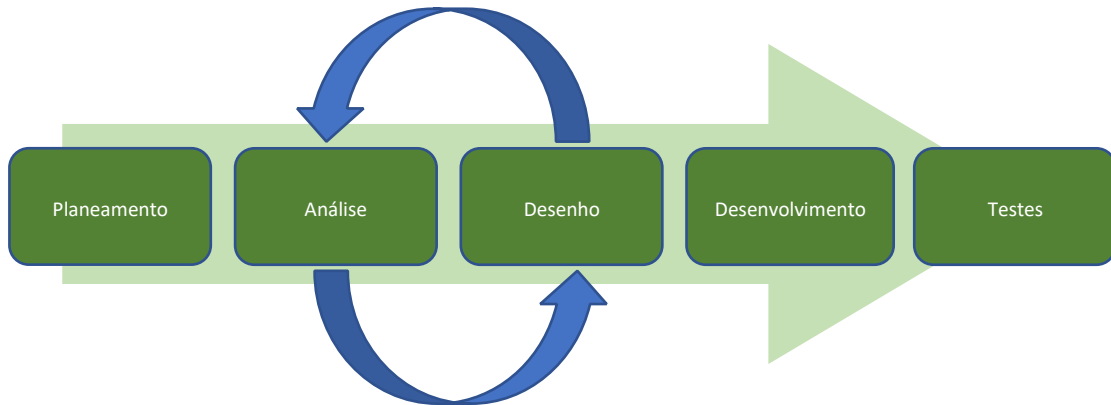


Figura 2.1 - Planeamento do estilo Cascata utilizado neste trabalho.

Devido à natureza sempre evolutiva deste tipo de trabalhos, foi necessário traçar os componentes a desenvolver que permitissem que as funcionalidades do projeto fossem demonstradas. Uma Câmara Municipal poderá ter muitas aplicabilidades que necessitem do âmbito geográfico. Depois de alguma pesquisa das necessidades mais relevantes, traçaram-se as três funcionalidades descritas no projeto. Na fase de análise, pesquisaram-se os requisitos imprescindíveis para possibilitar o desenvolvimento e minimizar as probabilidades de faltarem recursos durante a fase de desenvolvimento. A fase de desenho ajuda a colmatar mais esta situação pois consiste na elaboração técnica do trabalho. Aqui desenhou-se a forma como cada componente irá ser construído e ligado. Na etapa de desenvolvimento, procedeu-se à construção de todo o trabalho ao nível técnico, utilizado as linguagens e ferramentas já descritas. Na última fase do projeto, simulou-se a utilização de um utilizador final passando por todas as características elaboradas.

2.3 Arquitetura Geral – Descrição Funcional

Apresenta-se neste capítulo a descrição dos vários componentes deste projeto. Como referido, o objetivo deste projeto é fornecer uma solução de serviço no terreno de uma Câmara Municipal. A figura 2.2 representa o esquema de todas as comunicações presentes do projeto. Será feita uma primeira análise funcional.

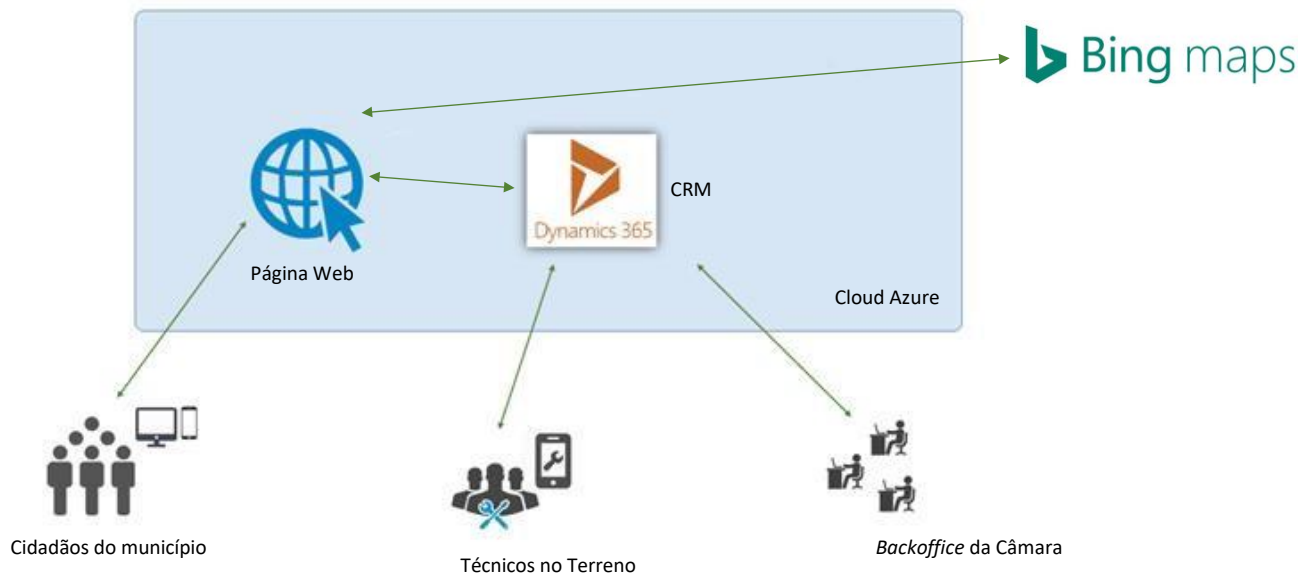


Figura 2.2 - Arquitetura da solução de Field Service deste trabalho.

2.4 Ambiente de Desenvolvimento

Todo o desenvolvimento ao nível do código foi executado na aplicação **Visual Studio 2017 Community** que não tem custos associados. Os restantes foram ao nível do *Dynamics* que consistiam em configurações, criação de registos e entidades assim como personalizações.

2.5 Página Web

Foi criado um projeto do tipo **ASP.NET Empty Web Site**. A página principal do cliente consiste num ficheiro do formato *aspx* que permite ajolar o código de página web como o do servidor. Na tabela 2.2 estão descritos os ficheiros que pertencem ao código fonte.

Tabela 2.2 - Ficheiros desenvolvidos pertencentes a esta seção do projeto.

| Ficheiro | Descrição |
|------------------------------|---|
| PaginaCliente.aspx | Página principal para acesso ao público |
| PáginaCliente.aspx.cs | Código <i>C#</i> do lado do servido |
| GetCRMService.cs | Código que possibilita troca de informação entre a página <i>.aspx</i> e o <i>CRM</i> |
| MainJavascript.js | Código <i>Javascript</i> que permite a dinâmica entre o utilizador e a página. Também permite a comunicação entre a página e o <i>CRM</i> através de chamadas <i>AJAX</i> . |
| MainStyleSheet.css | Código <i>CSS</i> que personaliza a página <i>.aspx</i> |

O *Design* da página foi construído com base num modelo previamente elaborado (*template*). Depois, apenas se personalizou a página ao encontro da vertente municipal. Portanto, existem outros

ficheiros que correspondem ao layout previamente existente, os quais não foram alterados e todas as customizações foram feitas a este nível encontram-se no ficheiro **MainStyleSheet.css**. Ao nível de *HTML*, as personalizações mais relevantes estão na secção Serviços, onde está disposta a funcionalidade principal desta secção.

Decidiu-se por convenção quatro serviços que uma câmara poderá disponibilizar nesta POC. São eles “**Certificações**”, “**Emissão de Certidões**”, “**Informações**”, e “**Registos e Licenças**”. Construiu-se um botão para cada um dos serviços. Depois, foi disponibilizado um mapa através dos serviços *Bing Maps*. Este serviço contém uma *API* que serve para chamar todas as funcionalidades que este produto oferece. Foi aqui que a biblioteca *Metro* foi usada para agilizar o desenvolvimento dos botões. Neste ponto, ainda todas as funcionalidades estão para ser desenvolvidas.

A partir daqui o foco torna-se na linguagem *Javascript*. Antes de mais é necessário estabelecer as ligações entre a página *aspx* e o *CRM*. Esta ligação é feita através de chamadas *AJAX*, e os dados são enviados e recebidos no formato *JSON* e é uma forma de fazer chamadas *REST* em *Javascript*.

2.5.1 Chamadas *AJAX* e estrutura *REST*

Conhecida como *REpresentational State Transfer (REST)*, o objetivo desta *API* consiste na exposição através de um *URL* de dados entre sistemas através de *URIs*. Esta *API* permite 4 tipos de interação, sendo eles designados como **GET**, **POST**, **PUT/PATCH**, e **DELETE**, sendo explicados na tabela 2.3. Através de uma parametrização do *URI* no *endpoint* (*url* que dispõe o serviço) do local na qual se quer fazer troca de dados, o sistema que recebe a chamada (*endpoint* recetor) e devolve/altera os dados requeridos pelo sistema que fez a chamada consoante a parametrização do *url*.

Tabela 2.3 - Métodos das chamadas *REST*. Neste projeto apenas foram necessários os dois primeiros, *GET* e *POST*.

| Método | Descrição |
|------------------|----------------------|
| GET | Obtenção de dados |
| POST | Criação de dados |
| PUT/PATCH | Atualização de dados |
| DELETE | Eliminação de dados |

As chamadas *AJAX* usam uma combinação da estrutura *REST* com a linguagem *Javascript* de forma a que as chamadas possam ser feitas de forma assíncrona, sem que o processamento fique estagnado à espera de resposta e consequentemente parando todo o restante processamento da página. Ao nível do código, a figura 2.3 exemplifica uma chamada *AJAX* com todos os seus componentes. A propriedade ***type*** representa o tipo de chamada a ser efetuado. O parâmetro ***url*** contém o método a ser invocado no lado do servidor. ***ContentType*** representa o tipo de dados a ser enviado assim como o mapa de caracteres. ***Data*** representa os dados a enviar para o lado em que se pretende a comunicação com o parâmetro *Service* (que neste caso representa o tipo de serviço pretendido). ***DataType*** diz respeito ao tipo de dados que o lado do cliente espera receber, neste caso também será *JSON*. O parâmetro booleano ***async*** informa se este pedido deverá ser corrido sincronamente ou assincronamente. Por último, temos o parâmetro ***success*** que contém o código a ser executado depois da chamada ser efetuada. A chamada irá depois consistir no *URI* que concatena todos os dados agora referidos para depois ser interpretado no servidor. Caso o código seja assíncrono, o processamento faz a chamada e processa de imediato o código a seguir a esta estrutura, e só corre a parte de ***success*** quando o pedido estiver concluído. Poderão ocorrer imprevistos nos pedidos ***REST***, e nesse caso existe mais uma propriedade denominada de ***error*** que contém código a correr caso haja um erro durante a execução do código. A variável ***result*** contém a resposta ao código do lado do servidor.

```
async function GetSchedulesButton() {
    LoadingSpinner.Start();

    try {
        document.getElementById("LoadingText").innerHTML = "A Consultar
        Agendamentos...";
        $.ajax({
            type: "POST",
            url: "PaginaCliente.aspx/GetServiceSchedule",
            contentType: "application/json; charset=utf-8",
            data: "{NameOfService:'" + Service + "'",
            dataType: "json",
            async: true,
            success: async function (result) {
                try {
                    //Meter a data actual como input
                    //Acrescentar os pontos a Points
                    var AvailableSchedules = JSON.parse(result.d);
                    for (var i = 0; i < AvailableSchedules.length; i++) {
                        Points.Destination.Lat = AvailableSchedules[i].Latitude;
                        Points.Destination.Lon =
                        AvailableSchedules[i].Longitude;
                    }
                }
            }
        });
    }
}
```

Figura 2.3 - Chamada *AJAX Javascript*.

No contexto da página *web*, ao selecionar um dos botões dos serviços, o utilizador pretenderá tirar uma “*senha*” que reserve a sua vez nos serviços da câmara, assim como informar quando será atendido e que postos municipais (neste caso foram consideradas as juntas de freguesia existentes no concelho das Caldas da Rainha). É, portanto, necessário fazer uma chamada ao servidor da página que depois contacte o *CRM* da Câmara Municipal para fornecer essa informação. Do lado do servidor, ainda não estão disponibilizados quaisquer serviços, a instância a este ponto ainda não terá sido criada, mas já foi planeada. Supondo que agora recebemos informação sobre todos os agendamentos dos cidadãos no *CRM*, do lado do servidor será necessário desenvolver lógica que nos faculte quais as horas disponíveis dos balcões das freguesias, para que um cidadão seja informado do sítio onde houver o agendamento livre mais cedo. Tendo obtido esta informação, existe agora a questão do melhor caminho até ao balcão com maior disponibilidade. A *API* dos serviços de mapa *Bing Maps* contém também um *endpoint* onde são efetuadas chamadas *REST*. O objetivo nesta fase será obter direções até ao ponto onde está o balcão referido. Esta *API* retorna resultados incluindo vários meios de transporte, como transportes públicos, a pé, mas para a situação vigente apenas fora equacionado o método de transporte de uma viatura própria. Temos também um crucial aspeto no *Bing Maps* que é o facto de obter resultados consoante o trânsito até ao destino existente. É aqui que é demonstrada uma das mais interessantes funcionalidades geográficas neste trabalho, pois contendo os postos com as referidas disponibilidades e o tempo de viagem até cada um, é possível equacionar a soma de tempo até eles. ***Isto significa que embora um balcão se situe mais longe, poder-se-á ser atendido em menos tempo, na medida que os balcões mais próximos têm maior engarrafamento nas vias de transporte.***

Para efetuar a chamada ***REST*** à *API* do *Bing Maps*, são necessários os pontos de coordenadas do utilizador e os pontos do destino. Os pontos de coordenadas do utilizador foram obtidos através da

biblioteca nativa do *HTML5*, portanto ao aceder ao site, o utilizador terá de dar as permissões ao browser de localização do dispositivo em questão. As coordenadas das juntas de freguesia onde se encontram os balcões obtiveram-se na chamada **REST**. Depois foi implementado um ciclo *for* para que cada junta de freguesia seja retornado o caminho desde o ponto do utilizador até à freguesia em questão. Ou seja, uma chamada *GET* à API do *Bing Maps* para as coordenadas de cada junta de freguesia. Sabendo que esta chamada nos dá a rota e o tempo até ao destino basta agora implementar alguma lógica através de *javascript* que nos forneça o caminho *menos moroso*. Obtendo este resultado, é agora necessário *renderizá-lo* ao utilizador para que possa haver visualização. A *renderização* está presente no capítulo 3 dos resultados.

Por convenção, foram disponibilizados dois destinos à escolha do utilizador. O Destino *mais rápido* e o *destino mais próximo*. Um desenvolvimento que poderia ser estendido no futuro seria a capacidade de representar todos os destinos por tempo de viagem ou distância, pois poderá ser vantajoso ao utilizador saber para além do destino ao balcão, dado que poderá ter outros destinos à parte próximos desse balcão. Outro desenvolvimento seria a capacidade de o utilizador marcar a senha *para o tempo e lugar em que deseja começar o trajeto*. Isto é possível pois a API do *Bing Maps* tem a capacidade de elaborar uma previsão do trânsito num tempo posterior ao presente. Por último, poderia convir ao utilizador saber se quiser ser atendido a uma determinada hora, ser-lhe dado o tempo e lugar a que deveria sair para conceder essa possibilidade. Isto não esquecendo que o agendamento que pretender poderá já estar preenchido previamente por outro cidadão, e sendo assim, retornar as possibilidades que se aproximam do tempo de atendimento pretendido.

Foi também disponibilizado o *tempo médio por senha*. No presente caso, esta métrica é meramente estática dado que foi estabelecido que cada senha terá 15 minutos de atendimento. Caso se pretendam que os atendimentos sejam feitos por ordem de chegada e concluídos mal os cidadãos estejam servidos, esta métrica fará mais sentido, podendo ser calculada através de por exemplo histórico dos tempos de atendimento, ou então dada uma distribuição de probabilidade que calculasse os tempos médios para cada hora de atendimento.

Do lado do utilizador (*front-end*), até agora apenas foi escolhido um dos serviços e obtidos os tempos de atendimento assim como as rotas até aos respetivos balcões. De seguida, segue o *pedido de agendamento*. Como dito antes, existem dois botões, um com o destino mais rápido e outro com o mais próximo. Ao selecionar um botão uma vez, é dada a rota respetiva até ao destino em concreto. Por defeito, mal o utilizador clique no serviço que pretende fica automaticamente selecionado o destino mais rápido. Caso se clique no destino selecionado, é então agendada a senha, obtendo um descritivo do agendamento. Não esquecendo que o pedido de agendamento constitui uma chamada *AJAX* ao servidor, que de seguida contacta o CRM para registar o agendamento. Resumindo, existem 3 chamadas *AJAX* descritas na tabela 2.4.

Tabela 2.4 - Chamadas *AJAX* existentes do lado do site (cliente).

| Chamada | Descrição |
|---|---|
| <i>PaginaCliente.aspx/GetServiceSchedule</i> | Devolve todos os agendamentos do CRM mediante o input do nome do serviço da câmara pretendido |
| <i>PaginaCliente.aspx/CreateSchedule</i> | Cria um agendamento no CRM com base em parametros predefinidos JSON. |
| <i>http://dev.virtualearth.net/REST/V1/Routes?wp.0=(Latitude Origem),(Longitude Origem)&wp.1=(Latitude Destino),(Longitude Destino)&optmz=timeWithTraffic&key=(Chave Bing Maps)</i> | Devolve o caminho e o tempo de viagem de veículo próprio entre dois pontos mediante a colocação das suas coordenadas no URL. Este método tem em conta o trânsito em tempo real. |

De notar que para o acesso à API do Bing Maps, é necessária uma chave. Para isso, foi criada uma conta *Bing* que, para efeitos de desenvolvimento, não são cobradas taxas de utilização da API.

2.5.2 Lado do Servidor

O Servidor faz ligação entre o cliente e a instância de *CRM* da Câmara Municipal. Aqui, os desenvolvimentos são feitos na linguagem **C#**. Como irá existir a ligação ao *CRM*, são necessárias bibliotecas que nos facilitam o código no contexto do *Dynamics*. Então para isso, foram transferidos os pacotes **NuGet** necessários para o trabalho. Estes pacotes consistem em código compilados (bibliotecas - *DLLs*) que são partilhados entre a comunidade de programadores. O *Visual Studio* fornece-nos toda uma ferramenta que facilita a transferência destes ficheiros e os adiciona ao projeto final com apenas alguns cliques. As bibliotecas denominam-se como **Microsoft.Xrm.Client**, **Microsoft.Xrm.Sdk** e **Microsoft.Xrm.Sdk.Workflow**.

Para a comunicação entre o *CRM* e o servidor, existe um único ficheiro destinado a estabelecer denominado por **GetCRMService.cs**. Não esquecer que a instância de *CRM* ainda não foi criada. Neste ficheiro irão existir as variáveis do *username* e *password* do utilizador para o acesso ao *CRM*, assim como o seu *endpoint*. Irão existir pelo menos 2 métodos principais que respondem aos pedidos da tabela 2.5.

Tabela 2.5 - Métodos principais do lado do Servidor.

| Método | Descrição |
|---|--|
| <i>GetServiceSchedule(string NameOfService)</i> | Devolve todos os agendamentos do <i>CRM</i> mediante o input do nome do serviço da câmara pretendido |
| <i>CreateSchedule(string ScheduleInfoJSON)</i> | Cria um agendamento no <i>CRM</i> . |

O Primeiro método faz chamadas com a linguagem *FetchXML* para retornar todos os agendamentos do *CRM*, sendo a partir daqui que se determina o tempo livre de cada balcão que depois é fornecido ao utilizador. A partir daqui já teve de ser alguma estrutura *CRM* montada para se poderem testar estas chamadas. Esse ponto será abordado mais em diante neste trabalho. Destas chamadas, são parametrizadas as entidades necessitadas assim como os filtros para apenas retornar informação pertinente. A figura 2.4 mostra uma chamada enviada ao *CRM*.

```
#region fetchXMLGetAvailableServices
string fetchXMLGetAvailableServices = @"<fetch version=""1.0"" output-format=""xml-platform"" mapping=""logical"" distinct=""true"">,
  <entity name=""bookableresource"">,
    <attribute name=""name"" />,
    <attribute name=""createdon"" />,
    <attribute name=""resourcecategory"" />,
    <attribute name=""bookableresourceid"" />,
    <order attribute=""name"" descending=""false"" />,
    <link-entity name=""bookableresourcecategory"" from=""resource"" to=""bookableresourceid"" link-type=""inner"" alias=""af"">,
<link-entity name=""bookableresourcecategory"" from=""bookableresourcecategoryid"" to=""resourcecategory"" link-type=""inner"" alias=""ag"">,
  <filter type=""and"">,
    <condition attribute=""bookableresourcecategoryid"" operator=""eq"" uiname="" + NameOfService + @""
uitype=""bookableresourcecategory"" value="" + GuidOfRequestedService + @"" />,
  </filter>,
  </link-entity>,
  </link-entity>,
  <link-entity name=""msdyn_organizationalunit"" from=""msdyn_organizationalunitid"" to=""msdyn_organizationalunit"" visible=""false""
link-type=""outer"" alias=""Freguesia"">,
  <attribute name=""msdyn_longitude"" />,
  <attribute name=""msdyn_latitude"" />,
  </link-entity>
</entity>,"
```

Figura 2.4 - Expressão *FetchXML* que pesquisa informação sobre os balcões.

A chamada evidenciada tem como objetivo o retorno de entidades do tipo “*bookableresource*” (entidade nativa do *CRM* usada para representar os balcões – um recurso que pode ser agendado). As *tags* com o nome *attribute* representam os campos de *CRM* a retornar. As *link-entity* denominam as relações, semelhante ao *JOIN* no âmbito *SQL*. A *Filter* e *condition* representam a forma de filtrar registos na linguagem *fetchXML*. Neste caso, temos o nome do serviço assim como o seu *Id (Guid)* no *CRM*. Portanto, o objetivo desta *query* é devolver todos os registos de balcões que tenham um determinado nome de serviço (Exemplo: Certidões).

O segundo método cria um registo no *CRM* depois do utilizador o ter agendado no lado do cliente; mais especificamente um registo de senha e um registo de agendamento no respetivo balcão. Estes dois registos ficam associados para se saber qual a senha que corresponde ao respetivo agendamento. Em termos de *CRM*, o agendamento irá corresponder à entidade Reserva de Recurso Reservável, que corresponde ao registo de tempo em que um balcão está com um cidadão a ser atendido.

Os resultados destes métodos são depois enviados para o lado do cliente em formato *JSON*, que depois são tratados por *Javascript* e *renderizados* ao utilizador.

2.6 CRM – Dynamics 365

Estando agora em contexto de *CRM*, os desenvolvimentos foram elaborados primeiramente numa instância *trial* e só depois de concluídos foram migrados para a instância de demonstrações aos clientes da *Unipartner*. Por norma e boas práticas, todos os desenvolvimentos em *Dynamics 365* são encapsulados no que se designam de “soluções”, que no fundo são pacotes que contêm desenvolvimentos à medida. A solução criada para este projeto é denominada como “*UFieldService*”. As primeiras intervenções da parte de configuração e desenvolvimentos consistiram na criação de entidades e os respetivos atributos.

A antiga versão do *Dynamics (Microsoft Dynamics CRM)* era baseada em servidores no local do cliente, sendo que a nova versão (365) disponibiliza os serviços e toda a infraestrutura na *cloud*. É acessível através de um browser e não requer qualquer instalação da parte do utilizador. O licenciamento da aplicação no caso deste projeto está a cargo da empresa onde é elaborado o projeto, *Unipartner IT Services*.

2.7 Definição de Entidade

Uma entidade em contexto de *Dynamics* denomina-se por algo da qual se possa moldar e guardar registos com dados. Cada entidade possui um conjunto de atributos e cada atributo representa um item de dados de um tipo particular. Um exemplo em concreto será a entidade “Conta” (de um cliente). Esta entidade possui vários campos (Ex. Nome do Cliente, nº de telefone, Morada, Cartões de Crédito, informações sobre estes) da qual um utilizador possa inserir dados sobre um determinado cidadão. Um caso do *CRM* deste projeto será a entidade “**Ocorrência**”, que servirá para registar ocorrências no território do concelho de Caldas da Rainha. Muitas entidades do *CRM* vêm por defeito e não serão utilizadas. A tabela 2.6 demonstra um exemplo de uma entidade com a descrição de todos os seus atributos.

Tabela 2.6 – Descrição dos atributos da entidade Ocorrência

| Nome | Descrição |
|--------------|--|
| Nome | Nome da Ocorrência |
| Latitude | Latitude da ocorrência |
| Longitude | Longitude da ocorrência |
| Território | Território da freguesia onde a ocorrência se situa |
| Tipo | Tipo da ocorrência (Lazer, Trabalhos, Pedido, Participação) |
| Descrição | Descritivo da ocorrência |
| Proprietário | Utilizador do CRM que detém a ocorrência. Este campo é necessário por convenção. |

2.8 Definição de Dashboards

Os *dashboards* são painéis que mostram métricas e indicadores relevantes dos registos do CRM de uma forma visual que facilita a compreensão dos dados existentes. Estas visualizações poderão ser tabelas ou gráficos. Os *dashboards* são personalizáveis, e para este trabalho irão ser construídos quatro para o projeto. O *Dynamics 365* possui funcionalidades nativas para os construir de forma rápida e intuitiva, bastando escolher as entidades requeridas e as referidas visualizações calculadas com os registos armazenados no CRM.

Tabela 2.7 - Dashboards a serem elaborados neste projeto. Os primeiros três dashboards foram construídos de raiz, sendo que o último é uma personalização dum dos dashboards por defeito, não sendo construído de raiz.

| Dashboard | Descrição |
|---------------------|--|
| Aderência | Gráficos com métricas sobre a aderência aos serviços da câmara por junta de freguesia. |
| Atendimento | Dashboard usado para o atendimento ao balcão dos cidadãos do concelho. |
| Mapa de Ocorrências | Dashboard que permite um gestor de recursos no terreno ter visibilidade sobre as ocorrências no concelho e assim facilitar a mobilidade destes assim como a sua distribuição na resolução das ocorrências. Com este dashboard o gestor tem também a visibilidade das posições atuais dos colaboradores que estão no terreno, assim como o seu histórico de posições. |
| Quadro da Agenda | Dashboard que permite a visualização do tempo de ocupação de cada cidadão nos balcões de cada junta de freguesia. |

2.9 CRM – Criação de Entidades

Na tabela 2.8, segue um resumo de todas as entidades criadas e respetivos campos, assim como campos já existentes e modificados, e as suas relações. Todas as entidades possuem campos de sistema que são gerados e preenchidos automaticamente na criação de registos associados a essa entidade. Campos esse como “*createdon*”, “*createdby*”, “*processid*” que são necessários ao funcionamento e registo de dados base, e não serão incluídos aqui. Foram também usadas entidades nativas do sistema para diminuir esforço. De realçar que existem 2 tipos de atributo conhecidos como “*Pesquisa*” (chaves estrangeiras), que no fundo representam uma relação com outra entidade. Assim, poderá haver um atributo que caso seja de pesquisa apontando para a entidade “*x*”, poderá estar preenchido com qualquer registo dessa mesma entidade. O tipo Conjunto de opções trata que um campo apenas pode aceitar uma opção de um conjunto de opções pré-determinado.

Tabela 2.8 - Entidades criadas de início para este Trabalho.

| Entidade | Nome | Tipo |
|---|-----------------------|--------------------|
| Ocorrência (Entidade Espacial Pontual) | Descrição | Texto |
| | Latitude | Real |
| | Longitude | Real |
| | Nome | Texto |
| | Técnicos responsáveis | Pesquisa |
| | Território | Pesquisa |
| | Tipo | Texto |
| Senha | Descrição | Texto |
| | Nome da Freguesia | Pesquisa |
| | Nome | Texto |
| | Número | Inteiro |
| | Serviço Associado | Pesquisa |
| | Tipo | Conjunto de Opções |
| | Situação | Conjunto de Opções |

Tabela 2.9 - Entidades pré-definidas do CRM aproveitadas para este trabalho.

| Entidade | Nome | Tipo |
|--|------------------------|----------|
| Categoria de Recurso Reservável | Nome | Texto |
| Recurso Reservável | Nome | Texto |
| | Unidade Organizacional | Pesquisa |
| Reserva de Recurso Reservável | Nome | Texto |
| | Senha | Pesquisa |
| | Recurso | Pesquisa |
| Território | Nome | Texto |
| Unidade Organizacional (Entidade Geoespacial Pontual) | Nome | Texto |
| | Morada | Texto |
| | Longitude | Texto |
| | Latitude | Texto |
| | Território | Pesquisa |
| Utilizador | Nome | Texto |

Descrevem-se agora as entidades pré-definidas usadas na tabela 2.9. De seguida é dado uma pequena descrição no papel de cada entidade.

Ocorrência – Entidade Espacial (geográfica) do tipo pontual que serve para registar geograficamente ocorrências ocorridas no concelho. Tais ocorrências podem tratar-se de árvores caídas, eventos de lazer no concelho e outros acontecimentos localizados que sejam da responsabilidade da Câmara Municipal. Todos os registos criados nesta entidade são fictícios.

Senha – Esta entidade tem o objetivo de registar todos os atributos relativos a senhas que os cidadãos obtêm para serem atendidos no balcão. Um dos atributos, o serviço associado, é referente ao agendamento do respetivo balcão a que um cidadão se terá de deslocar para ser atendido.

Categoria de Recurso Reservável – Nesta entidade foram consignados os registos dos tipos de serviço que a Câmara Municipal pode ter. Relembrando, são os serviços de “**Certificações**”, “**Emissão de Certidões**”, “**Informações**”, “**Registo e Licenças**”. De notar que esta entidade está ligada à entidade Recurso Reservável, e sendo esta uma ligação de *N:N*, existe uma entidade intermediária conhecida como “**Associações de Categorias de Recursos Reserváveis**” que serve para registar as relações entre as duas.

Recurso Reservável – Esta entidade foi usada para representar os balcões em cada Junta de Freguesia. Em âmbito de *Field Service*, esta entidade tem como papel representar algum recurso que possa ser agendando através da entidade “Reserva de Recurso Reservável” (RRR), que mostra que este recurso está ocupado durante o tempo estabelecido no registo da RRR associado. Esta entidade poderá representar um técnico de terreno, equipamento, ou algum recurso que seja essencial para a conclusão de uma determinada tarefa ou caso.

Reserva de Recurso Reservável – Entidade que tem uma relação com “**Recurso Reservável**”, representa o intervalo de tempo que este recurso fica indisponível de ser reservado para a conclusão de outra tarefa. No exemplo em concreto deste trabalho, um Balcão (Recurso reservável) que está em atendimento a um cidadão, fica indisponível para atendimento a outros cidadãos durante o intervalo de tempo de reserva (Reserva de Recurso Reservável) deste balcão. O tempo de reserva de recurso reservável neste trabalho foi estabelecido como sendo de 15 minutos por convenção.

Território – Entidade que representa territórios no *CRM*. Foram criados registos consoante a existência real de Juntas de Freguesia concelho de Caldas da Rainha, sendo eles: **A dos Francos, Alvorninha, Carvalhal Benfeito, Foz do Arelho, Landal, Nadadouro, Nossa Senhora do Pópulo, Coto, São Gregório, Salir de Matos, Santa Catarina, Santo Onofre, Serra do Bouro, Tornada, Salir do Porto e Vidais.**

Unidade Organizacional – Entidade Espacial (geográfica) do tipo pontual usada para representar as Juntas de Freguesia de cada Território.

Utilizador – Entidade que representa os utilizadores no *CRM*. Aqui, apenas foi utilizado um utilizador, com o nome do autor deste trabalho, que servirá de utilizador do *backoffice* e também como técnico do terreno. Como o objetivo deste trabalho é um *POC*, não justifica a criação de mais utilizadores para a instância de *CRM*.

Implementação de uma solução de sistemas de informação Geográfica aplicado ao *Customer Relationship Management (CRM)*

Por último na figura 2.5 é apresentado o diagrama de classes UML que representa as relações das entidades.

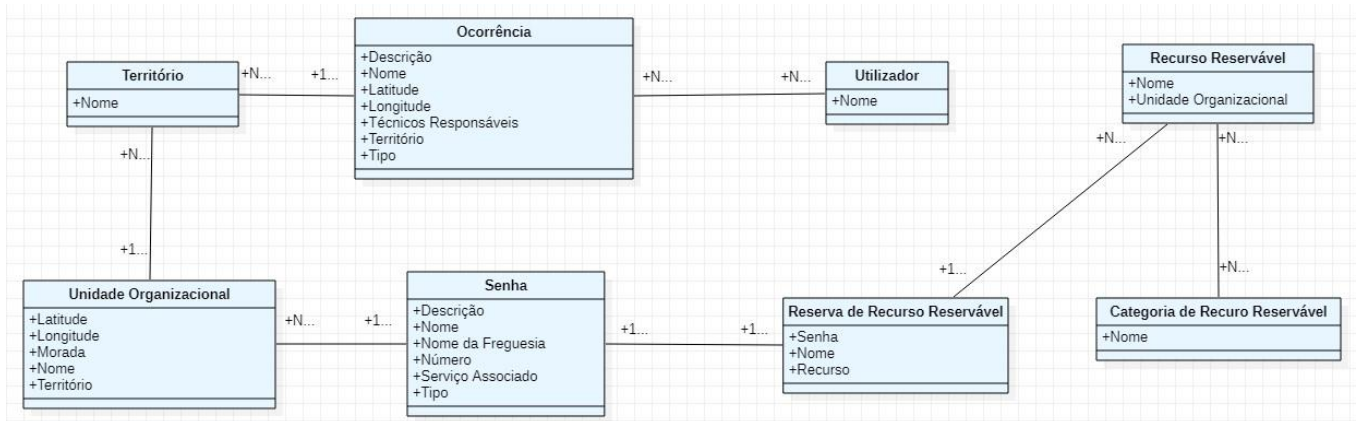


Figura 2.5 - Relações entre as entidades através dos campos de pesquisa. Cada ligação representa um atributo que estará a apontar para um registo em específico da entidade que aponta. (Campos de Pesquisa).

2.10 CRM – Configuração de Dashboards

Para configurar um *dashboard*, bastará num ambiente *Dynamics* ir a Definições Avançadas -> Soluções -> (Solução deste Projeto) -> *Dashboards*. Também podem tomar a forma de uma página web com código e acesso aos dados do CRM.

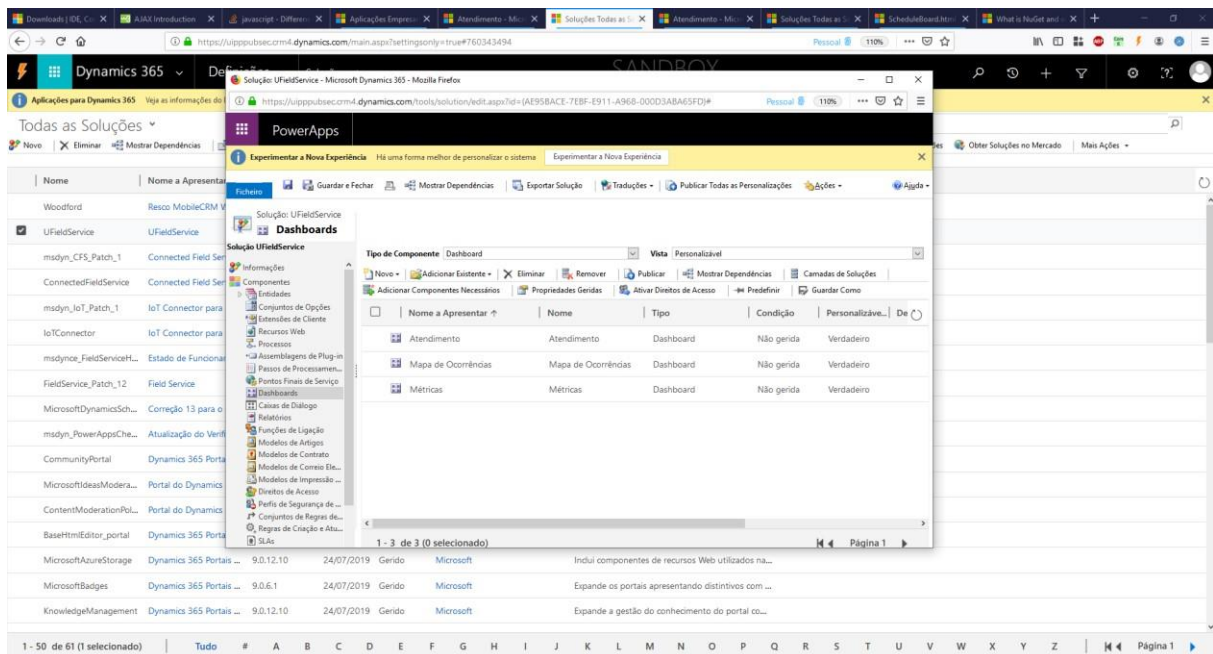


Figura 2.6 - Janela da Solução UFieldService onde estão os desenvolvimentos deste projeto. Na janela estão 3 dos 4 dashboards, pois o dashboard Quadro da Agenda é nativo e está à parte desta solução no CRM.

Implementação de uma solução de sistemas de informação Geográfica aplicado ao *Customer Relationship Management (CRM)*

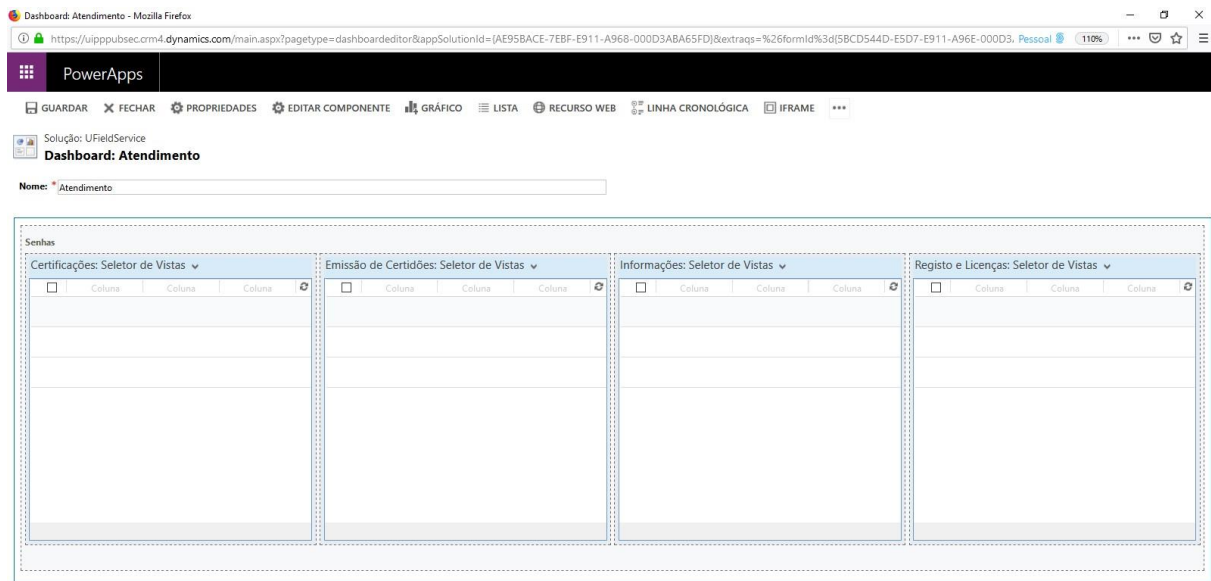


Figura 2.7 - Janela de personalização de um dashboard. São permitidos gráficos, listas (Ex: Senhas com a categoria de Certificação), Recursos Web entre outros. Este último é o mais interessante pois permite a introdução páginas web com código à medida.

Todos os desenvolvimentos presentes no CRM como recursos web estão descritos na tabela 2.10.

Tabela 2.10 - Ficheiros com os desenvolvimentos efetuados do lado do CRM.

| Ficheiro | Descrição |
|-----------------------------------|--|
| OcorrênciasDashboard.html | Página web que contém um mapa (Bing Maps) onde são posicionadas todas as ocorrências registadas. |
| OcorrênciasJavascript.html | Ficheiro javascript que renderiza numa tabela e mapeia os dados do CRM para o mapa. Utilizado também para a representação dos técnicos no Terreno assim como histórico destes. |
| MapaOcorrênciaForm.html | Página web que contém javascript e css embutido para representar uma ocorrência. Usado no formulário de uma ocorrência. |
| OcorrenciasStyles.css | Ficheiro que personaliza a página OcorrênciasDashboard.html |

Capítulo 3 – Resultados do Trabalho

Este capítulo irá demonstrar os resultados finais deste trabalho. Irá ser dada uma rápida abordagem de utilização do programa assim como visibilidade das funções base do CRM, mostrando ao mesmo tempo os desenvolvimentos elaborados.

3.1 CRM da Câmara Municipal

Ao entrar na aplicação *Field Service* (Mais concretamente “*Hub de Recursos no Terreno*”) somos confrontados com o ambiente apresentado na figura 3.1.

Aqui, estão destacadas as principais ligações por onde o Gestor de Serviços poderá começar o seu trabalho. Note-se que as seções da página são praticamente configuráveis à medida de um cliente final. De seguida estão discriminadas as ligações pelas quais um utilizador poderá explorar a aplicação.

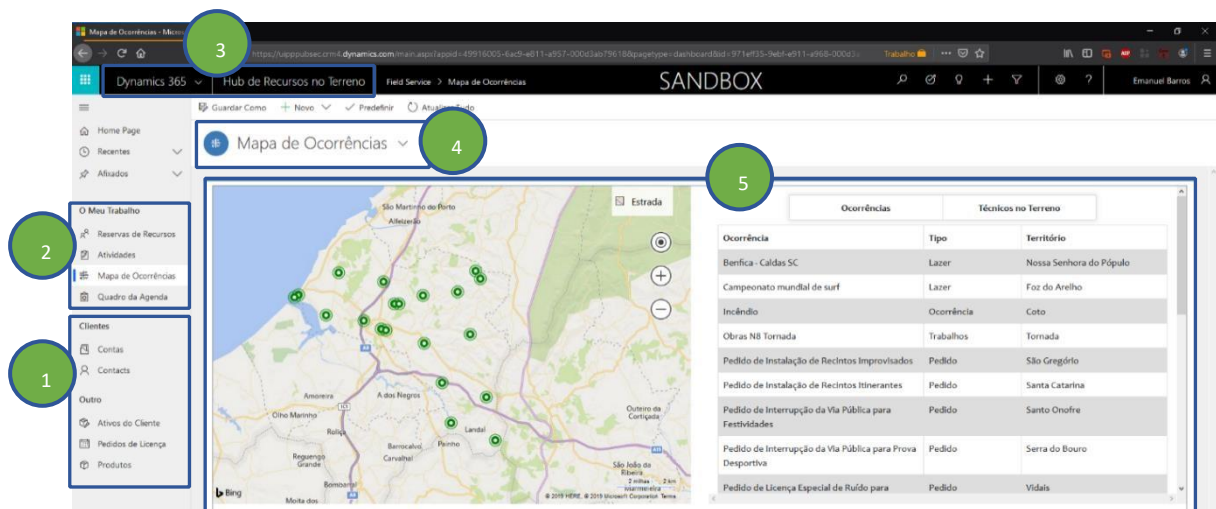


Figura 3.1 – Página Principal da aplicação *Field Service* do Dynamics 365. Tem como *dashboard* por defeito um dos desenvolvimentos feitos neste projeto.

De acordo com a figura 3.1:

1. Esta seção mostra ligações a registos a algumas das **entidades** previamente explicadas do CRM.
2. Secção que mostra os *dashboards* disponíveis ao utilizador. Na figura 3.1 está visível o *dashboard* “*Mapa de Ocorrências*”.
3. Seletor das aplicações existentes no CRM. Este poderá consistir em mais do que uma aplicação (Uma empresa poderá necessitar de gestão de vendas, marketing, serviço ao cliente), sendo que serão usadas por utilizadores de cada área. No nosso projeto, apenas será usada a aplicação de Recursos no Terreno (*Field Service*).
4. Seletor de *dashboards*.
5. Área que mostra o *dashboard* selecionado. Este *dashboard* consiste numa página *html* construída à medida desenvolvida neste projeto.

3.2 Visualização do Formulário da entidade Ocorrência

Cada entidade contém vários formulários que podem ser desenhados para acomodar a utilização de um gestor de CRM, e albergam os atributos que cada registo poderá ter. Por exemplo, ao clicar numa das ocorrências do dashboard da figura 3.1, o utilizador é redirecionado para o registo da ocorrência para que possa obter mais dados sobre ela assim como atualizá-los. A figura 3.2 apresenta o formulário configurado para a ocorrência. Estes registos são usados e demonstrados numa forma agregada nos dashboards. Existe a possibilidade de integrar nativamente a aplicação PowerBI com os dados desta instância de CRM, e obter visualizações que maximizem o Business Intelligence que auxiliem num próximo passo no desenvolvimento e manutenção da Câmara Municipal.

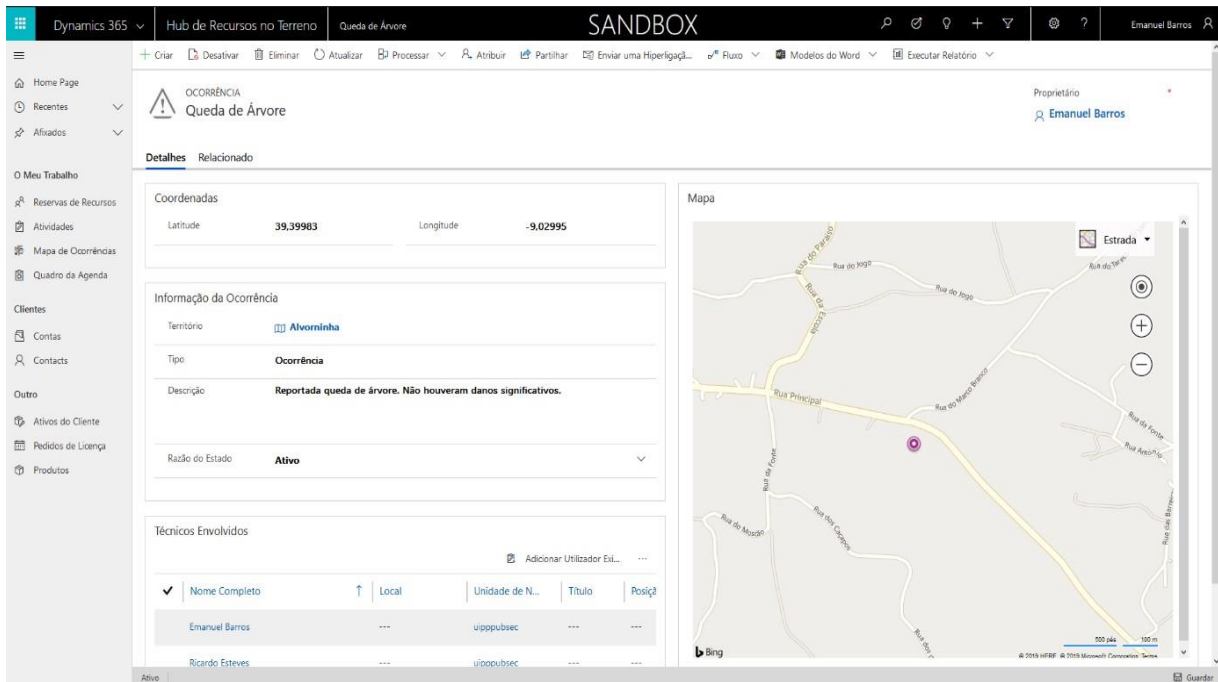


Figura 3.2 – Formulário da Entidade Ocorrência. Cada atributo está distribuído por secções. Existe a possibilidade de colocar páginas html nos formulários. Neste caso, a secção mapa contém uma página html que através do serviço Bing Maps, integra as coordenadas do formulário e representa-as no mapa.

3.3 Site da Câmara Municipal

Relembrando a primeira funcionalidade deste projeto, um utilizador poderá a partir da sua posição obter o caminho e o tempo médio da fila de espera para um determinado serviço da junta de freguesia mais próxima.

Isto é feito através da página web deste projeto na qual um cidadão poderá aceder (figura 3.3). Esta página comunica com o CRM da Câmara Municipal, e após uma pesquisa da disponibilidade dos balcões pertencentes às juntas de freguesia, é atribuída ao cidadão a senha do serviço da qual o cidadão pediu, assim como o caminho ótimo a percorrer (tendo em conta o fator trânsito) até à infraestrutura que tratará do seu pedido.

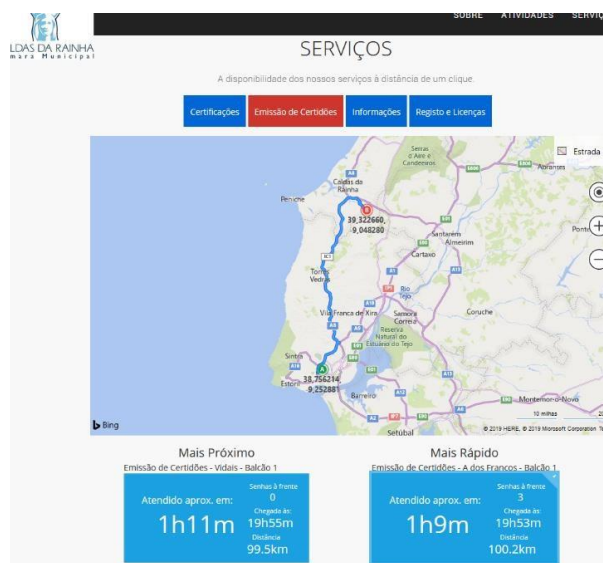


Figura 3.3 - Site dos serviços da Câmara Municipal. Aqui, o utilizador clica na área azul que contém o destino que deseja.

Em termos práticos, o utilizador seleciona no serviço pretendido e de seguida são despoletados o *destino mais próximo* (em relação à distância) e o *destino mais rápido* (poderá existir um destino mais longe mas com menos trânsito e fila de espera). Por defeito, está escolhido sempre o caminho mais rápido, pois é assumido que um utilizador queira ser servido o mais rapidamente possível.

Para “tirar” uma senha, basta agora selecionar novamente o botão do destino escolhido, e



Figura 3.4 – Resultado mais próximo do utilizador. Cada número representa uma métrica na tabela 3.6

será atribuída a senha correspondente ao serviço. Os quadrados informativos disponibilizam também o número de senhas à frente, a chegada ao destino (assumindo que o utilizador tenha um transporte próprio), e a distância até à chegada. Adicionalmente, em formato de *tooltip*, é dado “o tempo médio de espera”. Esta é uma funcionalidade fora o âmbito de SIG e aqui é assumido o tempo médio de cada atendimento dos cidadãos.



Figura 3.5 - Após clicar no seu destino, o site comunica com o CRM e este cria um agendamento para o utilizador.

Tabela 3.6 - Métricas informativas dadas ao utilizador na escolha de um dos destinos.

| Métrica | Descrição |
|---------|--|
| 1 | Soma do tempo de viagem e tempo da totalidade dos agendamentos. |
| 2 | Número de agendamentos (senhas) à frente. |
| 3 | Hora de chegada supondo que o utilizador parta agora ao destino. |
| 4 | Distância do trajeto traçado. |
| 5 | Tempo médio de espera por cada senha. |

3.4 Dashboard Atendimento

Do lado do *BackOffice (CRM)*, foi configurado um *dashboard* para que um trabalhador de um balcão dos serviços municipais possa atender os cidadãos que peçam senhas através do site. Estas são guardadas como registos *Senha* no CRM. As cores das senhas não possuem significado ao contexto.

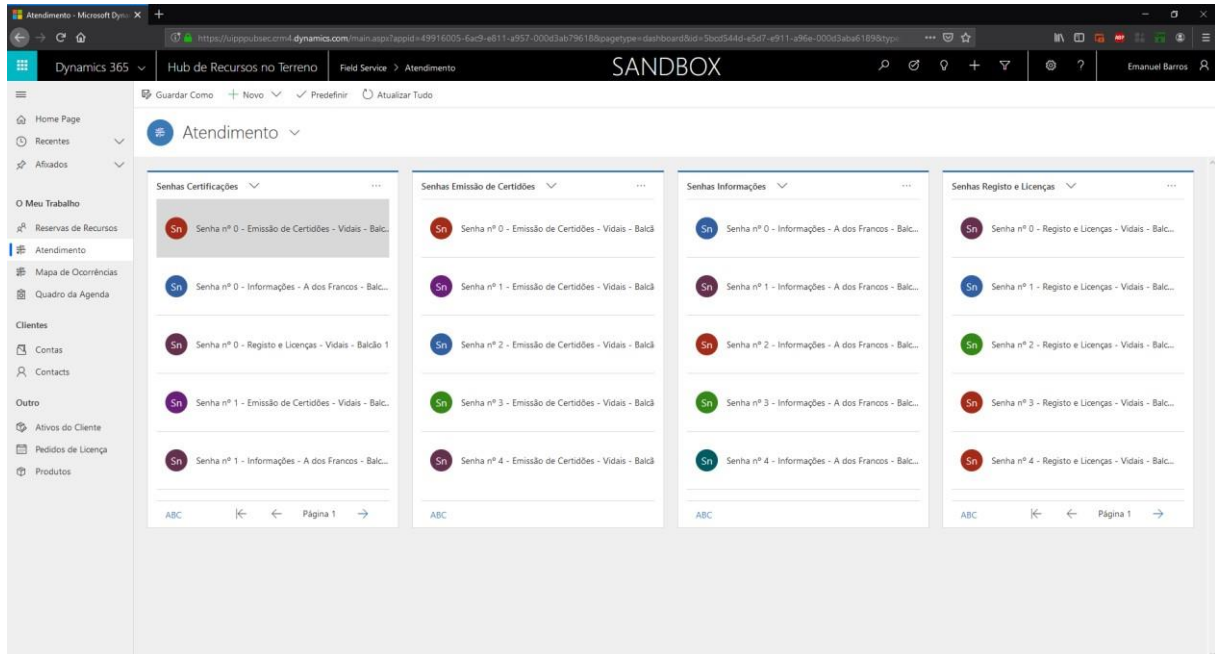


Figura 3. 6 - Dashboard do atendimento. Está dividido em 4 secções que despoletam as senhas de cada serviço.

Para tratar uma senha, basta o gestor clicar na senha que corresponde à vez do cidadão, e acederá à ficha (formulário) desta.

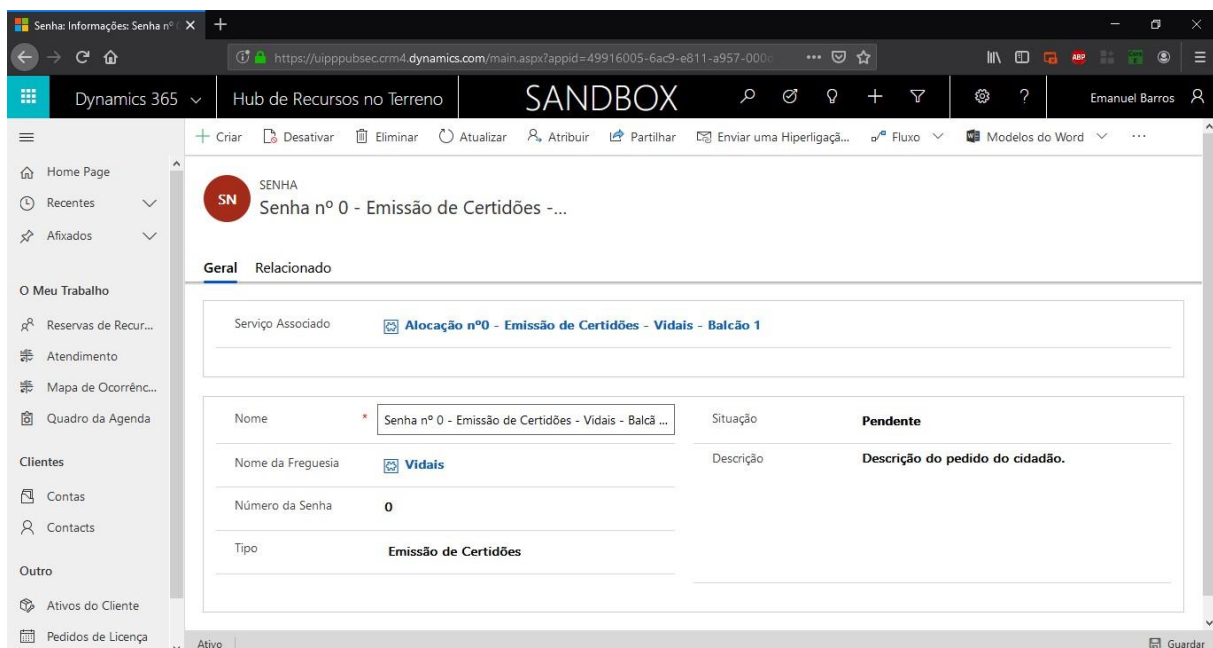


Figura 3.7 - Formulário de uma senha. Um formulário poderá ser depois customizado para acomodar as necessidades dos serviços da câmara.

Quando as senhas são criadas, têm por defeito o seu atributo “Situação” como pendente. Este atributo foi configurado para ter 3 valores possíveis, sendo os restantes “Concluído” e “Pendente”, por convenção demonstrativa. Um gestor poderá depois *tratar* a senha colocando um destes estados e clicando em “*Guardar*”. A senha depois de concluída será inativada e não constará depois na listagem prévia.

3.5 Dashboard Quadro da Agenda

Este *dashboard* permite a visualização do atendimento da Câmara Municipal num formato mais global. Aqui, é visível cada balcão de todas as freguesias com os respetivos agendamentos e a possibilidade da filtragem de balcões consoante o Território, Junta de Freguesia, Tipo de Serviço e a ordenação. À direita na figura 3.9, estão ilustrados a azul os agendamentos para todas as senhas que os utentes tiram no site da Câmara Municipal. Recordemos as duas formas sugeridas de agendamento das senhas:

- Poderá ser atribuído um tempo pré-definido que aparecerá neste *dashboard* e o gestor poderá saber o trabalho que se avizinha. Por exemplo, cada senha poderá alocar 15 minutos do tempo do gestor.**
- De outro modo, será feita a reserva sem um tempo previamente estimado, e seguidamente é contabilizado o tempo de atendimento.**

As duas abordagens têm vantagens e desvantagens cuja discussão transcende o âmbito deste projeto. Para efeitos demonstrativos, foi escolhida a primeira abordagem e assumiu-se que cada atendimento demora em média 15 minutos. Então, quando o cidadão tira uma senha, no CRM irá ser agendado os próximos 15 minutos disponíveis de um balcão.

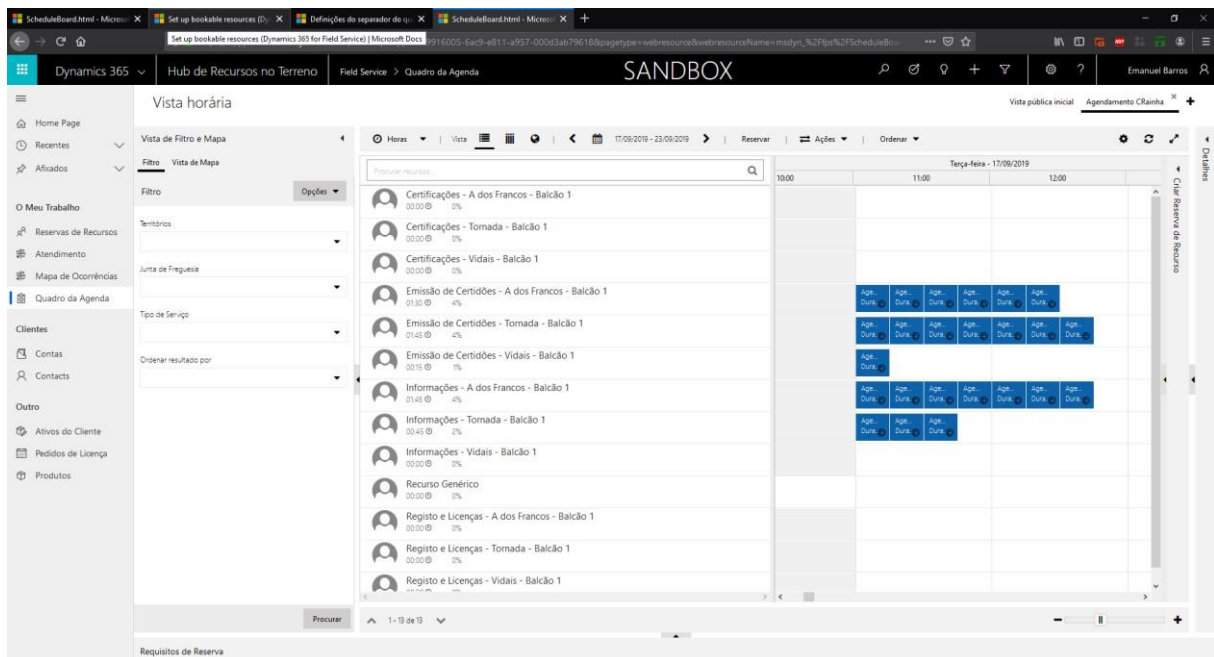


Figura 3.8 - Dashboard do Quadro de Agendamento.

Implementação de uma solução de sistemas de informação Geográfica aplicado ao *Customer Relationship Management (CRM)*

Este *dashboard* está presente por defeito na *app* de *Field Service* do *Dynamics 365*, o trabalho do projecto apenas contempla a sua configuração. A secção à esquerda permite a filtragem dos balcões por território, Junta de Freguesia, Tipo de Serviço e permite a ordenação. Esta filtragem irá corresponder aos balcões que aparecem na secção do meio. A secção mais à direita corresponde ao agendamento temporal de cada balcão. Na prática, cada senha obtida pelos cidadãos irá corresponder aos quadrados azuis à direita. Apenas necessitou de alguma configuração para acomodar a utilização do *CRM* no contexto de uma Câmara Municipal. Num contexto prático, este *dashboard* permite a visualização do que os gestores poderão ter em termos de trabalho.

Ao nível técnico, apenas se destaca a criação de um novo filtro para a visualização dos agendamentos por balcão. Este *dashboard* representa os registos da entidade “Recurso Reservável” e os agendamentos (*Reserva de recurso reservável*) numa forma gráfica, como já exemplificado. Foram utilizados os seguintes filtros para os agendamentos: **Territórios, Junta de Freguesia, Tipo de Serviço e Ordenação dos resultados**. Isto permite visualizar os balcões associados a essas condições, assim como os agendamentos a eles criados. Este *dashboard* foi desenvolvido pela *Microsoft* há menos de um ano a quando a escrita deste trabalho, e presentemente a única forma de parametrizar os filtros com as entidades desejadas é através de uma espécie de parametrização *xml*, como mostra a figura 3.10. Através do que já estava escrito por defeito, foi possível apenas mostrar os filtros relevantes, mediante alguma dificuldade pois ainda existe pouca documentação no que toca à personalização deste *dashboard*.

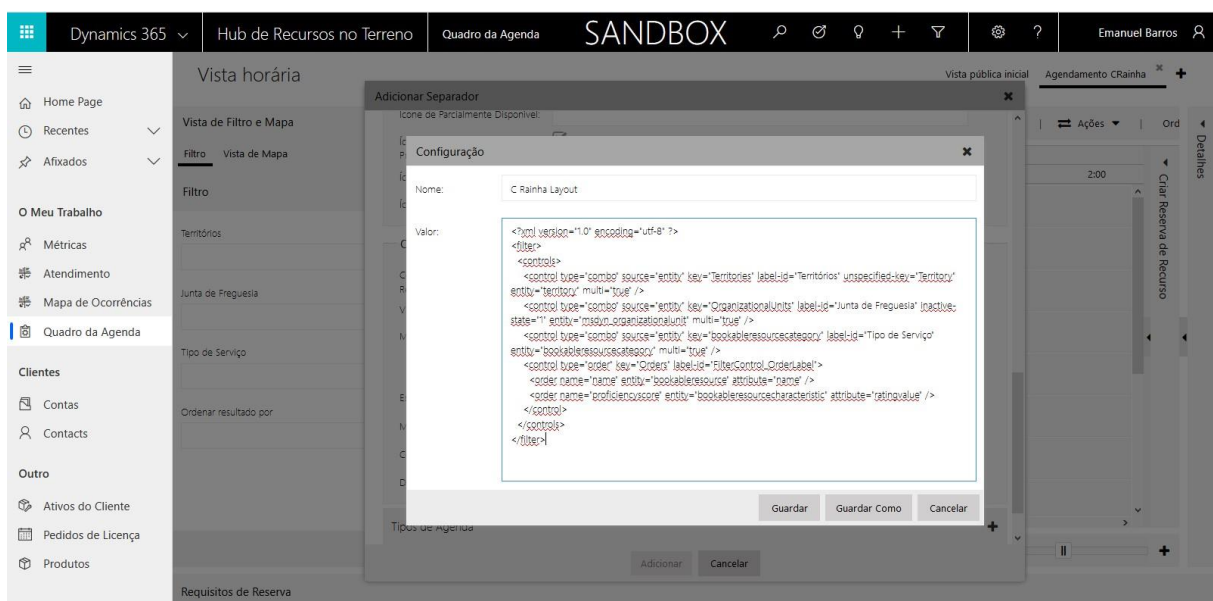


Figura 3.9 – Código do Filtro de balcões (Entidade Recursos Reserváveis).

3.6 Dashboard Métricas

Este *dashboard* mostra métricas das senhas tiradas consoante a localização do serviço das mesmas. Um gestor da câmara poderá tirar partido destes dados para poder canalizar esforços ou reforçar os balcões consoante a necessidade dos serviços.

Da esquerda para a direita e de cima para baixo, estão presentes os seguintes gráficos:

- Senhas por Junta de Freguesia
- Senhas por Tipo de Serviço
- Senhas por Tipo de Serviço de cada Junta de Freguesia

Os dados mostrados são puramente demonstrativos e servem apenas para apresentação, não se baseando em nenhuma situação real.

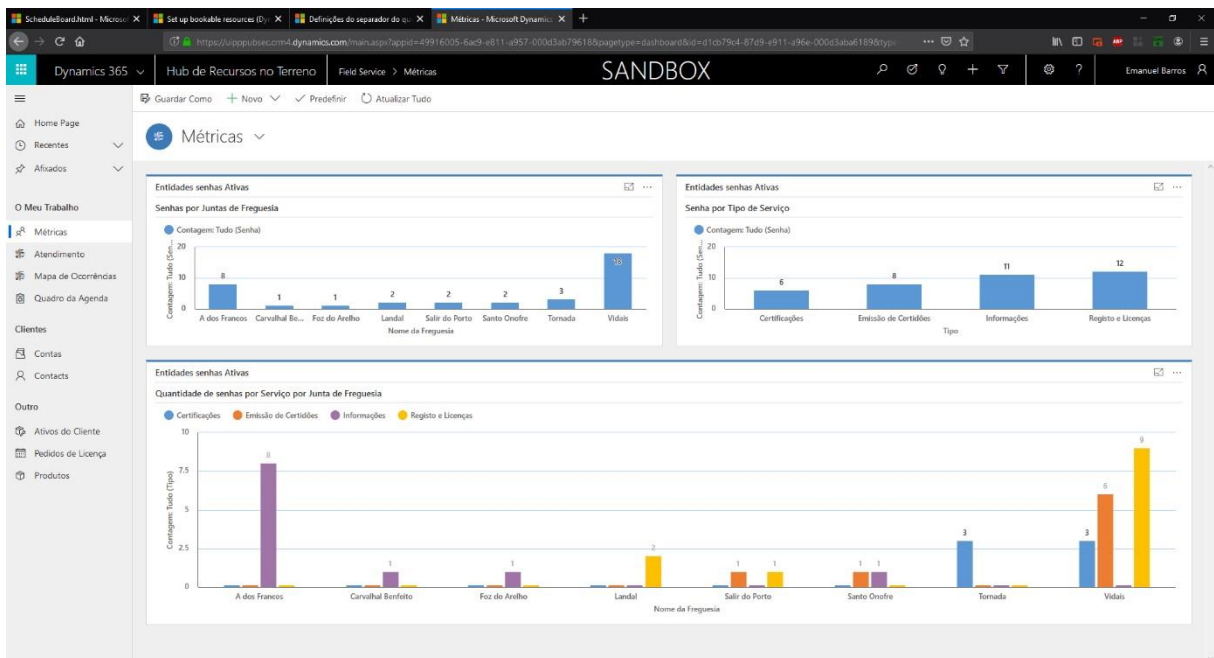


Figura 3.10 - Dashboard de métricas.

3.7 Georreferenciação de Incidentes

A segunda funcionalidade desenvolvida no CRM trata-se da **Georreferenciação de Incidentes**. Esta funcionalidade consiste em desenvolvimentos focalizados em diferentes áreas do *CRM*, sendo que estas serão explicadas diferenciadamente.

3.7.1 Ciclo da resolução de ocorrências

O ciclo proposto para este POC será como exemplificado na seguinte figura:

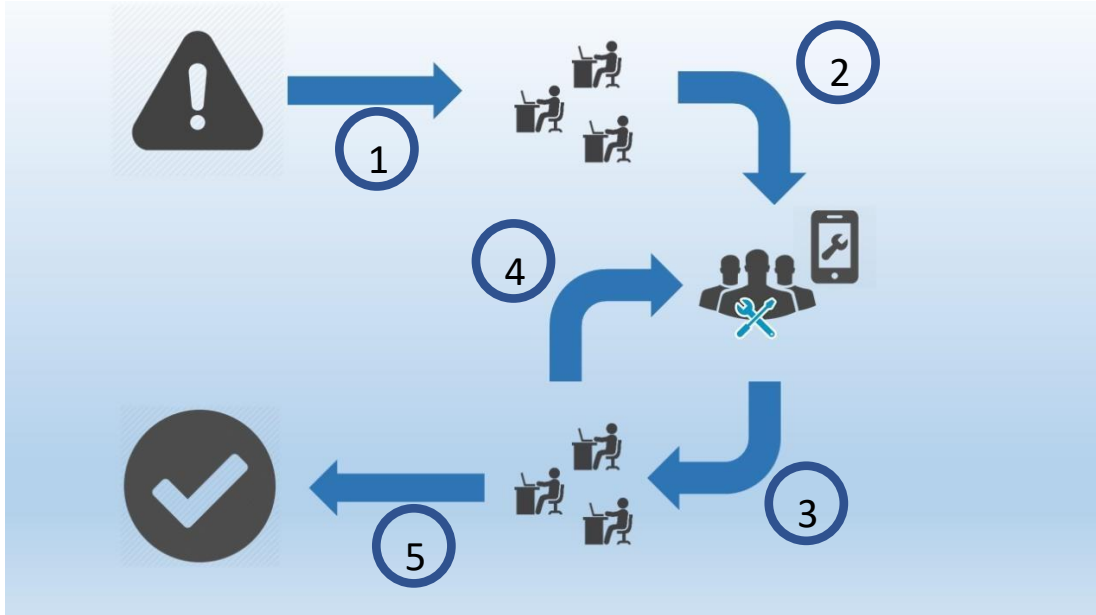


Figura 3.11 - Ciclo de resolução de incidentes proposto.

1. É reportado um incidente. Os gestores do *backoffice* na Câmara georreferenciam e guardam o registo no *CRM*.
2. Os gestores fazem a comunicação aos técnicos no terreno e estes são mobilizados para a localização do incidente.
3. Os técnicos atualizam a informação no *CRM* (através da aplicação *Mobile CRM* da *Resco*) com mais dados de interesse, com imagens e uma localização mais exata do registo.
4. Os Gestores do *backoffice* comunicam com os técnicos e atualizam a situação do incidente ao longo do tempo.
5. Os técnicos resolvem o incidente e poderão dar este como fechado, comunicando esta informação ao *backoffice* da câmara. Os gestores atualizam o incidente como concluído no *CRM*.

3.8 CRM – Dashboard de Ocorrências

O *dashboard* Mapa de Ocorrências apresenta todas as ocorrências no concelho. É composto por duas secções, um mapa com pins de ocorrências à esquerda e uma tabela de registos destas à direita (figura 3.13). Um utilizador poderá passar o rato pela ocorrência no mapa e é despoletado um *tooltip* com mais informação sobre o incidente em questão. Ao mesmo tempo é indicado na tabela qual o registo correspondente. Foi desenvolvido completamente à medida, sendo no fundo uma página web que comunica com o CRM através de chamadas *fetchxml* (como foi embebida no CRM, ela tem acesso aos seus dados), e usa os dados dos registos de ocorrências para mapear os incidentes no mapa e preencher a tabela ao lado através da API do Bing Maps. Tanto os registos na tabela como os pins são selecionáveis, redirecionando o utilizador ao formulário do registo correspondente. A partir daqui, sempre que um gestor obtém a informação de um técnico sobre a conclusão de uma ocorrência, este poderá fechá-la atualizando o atributo “Situação”. Como a *query fetchXML* contém o filtro de ocorrências ativas, as que foram concluídas deixam de aparecer no *dashboard*. Um futuro desenvolvimento poderia passar na coloração dos pins consoante a situação da ocorrência correspondente.

Ao selecionar um incidente, o utilizador é redirecionado ao registo correspondente para o poder tratar, ou seja, aceder a todas as informações adicionadas pelos técnicos no terreno assim como poder atualizá-las. As ocorrências são disponibilizadas na tabela tendo apenas os atributos Ocorrência (Nome), Tipo e o Território na qual está presente. Existem três ficheiros associados a este *Dashboard* (exceto o *MapaOcorrênciasForm.html*, que está presente no formulário das ocorrências) já descritos anteriormente e estão sediados no CRM através de *Recursos Web*.

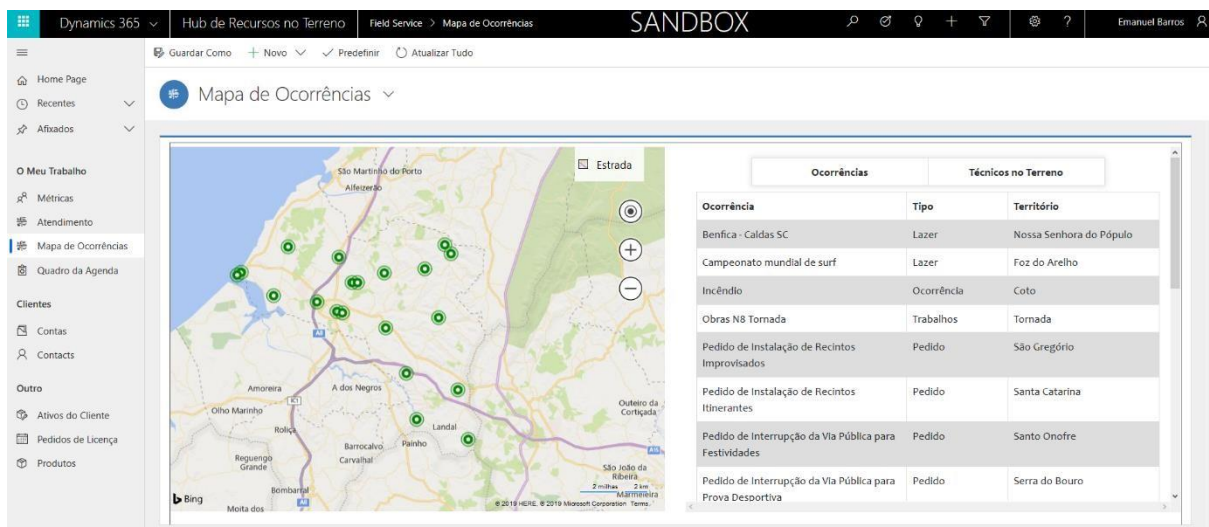


Figura 3.12 - Dashboard de Ocorrências. Este dashboard é na sua totalidade uma página web.

3.9 Funcionalidades no CRM – Localização dos Técnicos no terreno.

Os gestores poderão também visualizar as posições geográficas dos técnicos do terreno, guardadas também no CRM. Esta funcionalidade é acedível quando o utilizador seleciona “Técnicos no terreno”. Neste caso de teste, apenas se utilizou um utilizador, e no mapa é apresentada a última posição conhecida. A aplicação móvel está parametrizada para registar constantemente a posição atual do utilizador consoante o detalhe que um cliente pretenda (ex. de 1 em 1 minuto, ou caso o utilizador

Implementação de uma solução de sistemas de informação Geográfica aplicado ao *Customer Relationship Management (CRM)*

se desloque mais de 10 metros). No caso de se registarem muitas posições (maior detalhe), existe a desvantagem do CRM ficar muito carregado com registos de posição.

Para visualizar o histórico, bastará clicar com o lado direito do rato no utilizador, e clicar em histórico de posições.

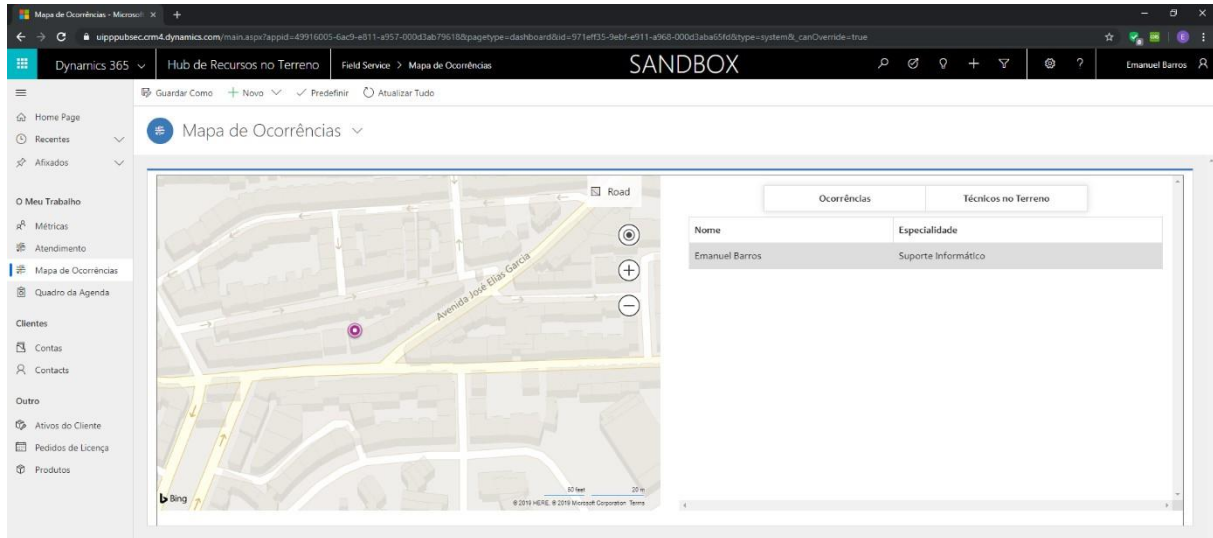


Figura 3.13 - Representação da última posição do técnico no terreno.

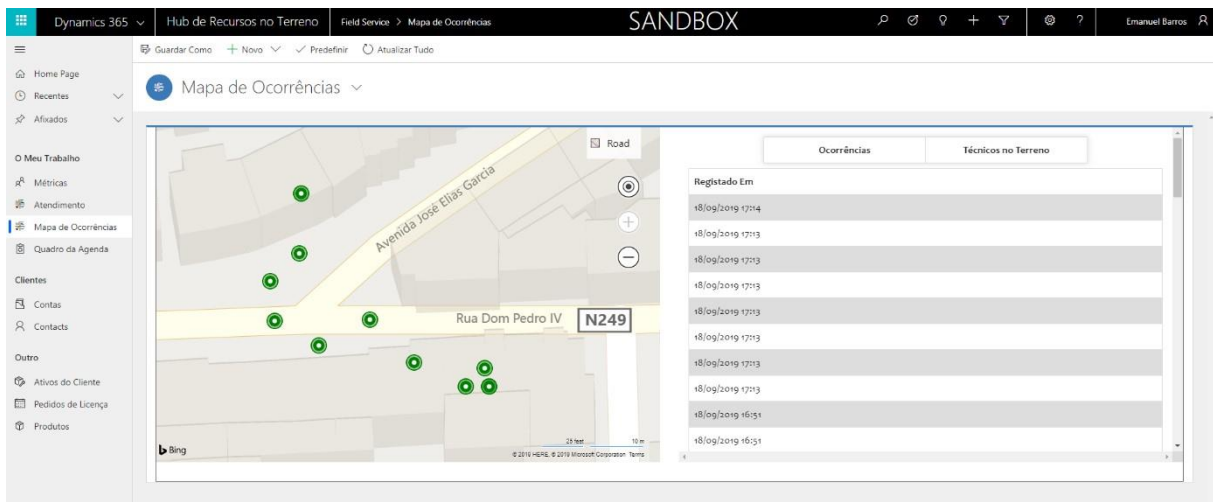


Figura 3.14 - Representação das últimas posições do utilizador "Emanuel Barros".

3.10 Técnicos no Terreno – Aplicação *Resco Mobile CRM*

Um técnico no terreno terá acesso à instância de *CRM* da Câmara Municipal através de um *smartphone*. A aplicação usada denomina-se *Resco Mobile CRM* e como referido, é um complemento do *Field Service* para o *CRM* da câmara. *Resco* é a entidade responsável pela aplicação, da qual faz parceria com a *Microsoft*.

Para personalizar o *CRM* que aparecerá nos telefones dos técnicos, foi necessária a instalação de uma solução denominada por *Woodford*, que permite a total configuração deste. A partir daqui a pessoa responsável pela manutenção do *CRM* poderá ditar quais os elementos que ficaram disponíveis aos técnicos, como *dashboards*, entidades, registos, entre todo o conteúdo do *CRM* da Câmara Municipal. A grande vantagem de usar esta aplicação é que esta tem a possibilidade de utilizar o *GPS* do *smartphone* para registar as posições dos técnicos. A solução *Woodford* vem também com entidades extra, destacando-se a entidade “*Mobile Audit*”. Esta entidade possui campos como a latitude e longitude, assim como o campo de pesquisa que aponta para o utilizador correspondente. Então, ao ligar a aplicação (É possível parametrizá-la para ser colocada em funcionamento *background*), a aplicação vai criando registos deste tipo através das informações dadas pelo *GPS* do telemóvel, e numa próxima oportunidade de sincronização com o *CRM* da câmara, transfere estes registos.

3.11 CRM Mobile

Os técnicos no terreno têm acesso ao *CRM* da câmara através da aplicação *Resco CRM Mobile* que vem acompanhada da instância *CRM*. Esta aplicação acede ao *CRM* e contém todas as funcionalidades deste, mediante uma vista adaptada ao *mobile*. Vem com todas as funcionalidades nativas de um *CRM*, mas apenas serão descritas as que fazem parte do foco deste trabalho. Depois de estar no local da ocorrência, o Técnico pode tirar uma foto e atualizar os registos da ocorrência, estes são depois sincronizados no *CRM* e serão visíveis para todos os operadores. Infelizmente e devido a esta aplicação ser paga, depois da licença ter expirado, não me foi possível retirar *printscreens* com a aplicação customizada para este contexto, sendo que na figura 3.16 são disponibilizadas as entidades por defeito do *CRM*.

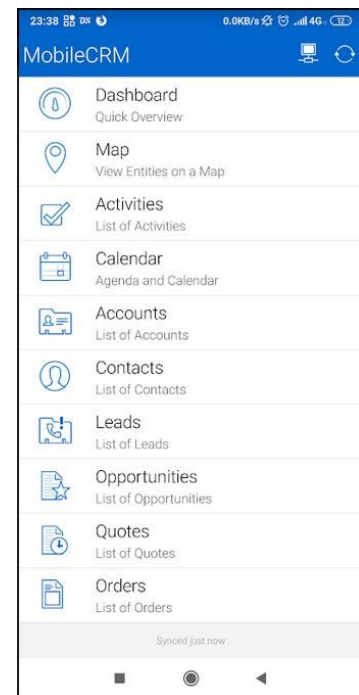


Figura 3.15 - Vista do *CRM Mobile* para um técnico no terreno.

Capítulo 4 – Demonstração do funcionamento do Projeto

Esta secção promove o passar por todas as funcionalidades do projeto de modo a que um utilizador final possa experimentar esta prova de conceito corretamente, passando por todos os passos de um agendamento, experimentando a aplicação *Resco CRM Mobile*, e interagindo com os *dashboards*. De modo a colmatar alguns erros nas funcionalidades, o *debugging* foi feito através do *DevTools* oferecido pelo próprio browser, assim como a capacidade do *Visual Studio* em fazer *debug* linha a linha do código. O Servidor foi testado na máquina localmente através da funcionalidade do *Visual Studio IIS Express*. Muitas outras funcionalidades, como a mobile, são possíveis, mas não foram desenvolvidas. Isto assenta no facto de este produto ter o objetivo de ser mostrado a um potencial cliente, e caso haja interesse da outra parte, existirão necessidades à medida para os vários clientes diferentes.

4.1 Funcionalidade 1

De modo a demonstrar que as funcionalidades do projeto funcionam, é primeiro testado o site da Câmara Municipal onde um utilizador escolhe o serviço pretendido. Ao escolher o Trajeto mais rápido, é apresentado o caminho entre o ponto onde o utilizador se encontra (Lisboa) e o ponto onde a junta de freguesia se encontra. Caso se seleccione o destino mais próximo, o mapa devolve o caminho até lá como demonstrado na figura 4.1.

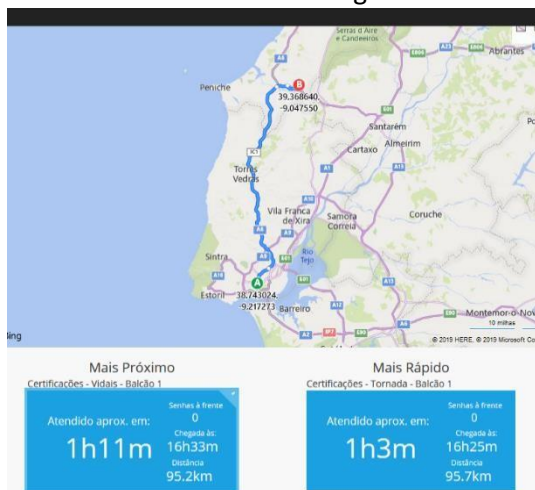


Figura 4.1 – Itinerário ao destino mais próximo.

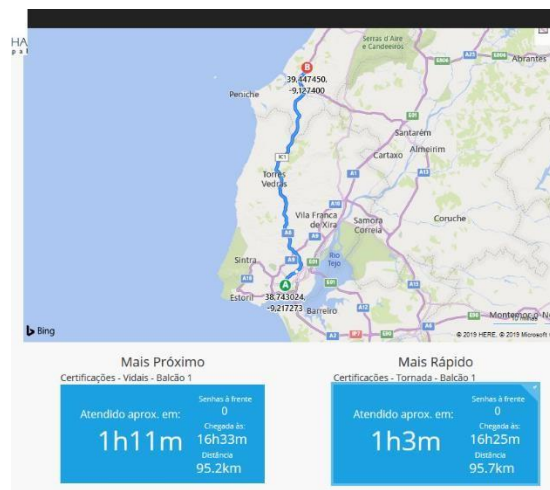


Figura 4.2 - Itinerário ao destino mais rápido.

Ao clicar então num dos botões depois deste estar selecionado, é dada ao utilizador a senha. Clicando mais vezes, este número irá incrementando.

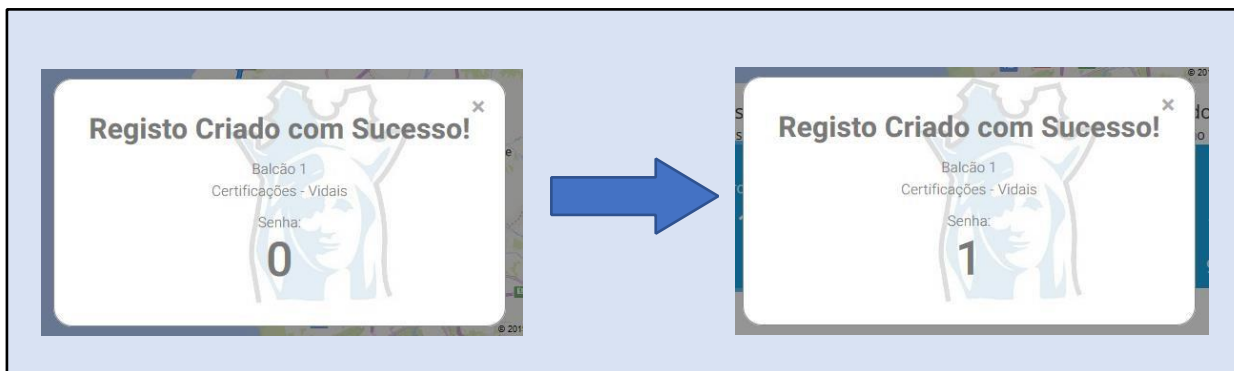


Figura 4.3 – Ao clicar mais vezes no destino, um utilizador está a tirar mais senhas

Implementação de uma solução de sistemas de informação Geográfica aplicado ao *Customer Relationship Management (CRM)*

No lado do *CRM* no *dashboard* do atendimento, são despoletadas as senhas criadas, bastando agora o utilizador clicar numa delas para começar o atendimento.

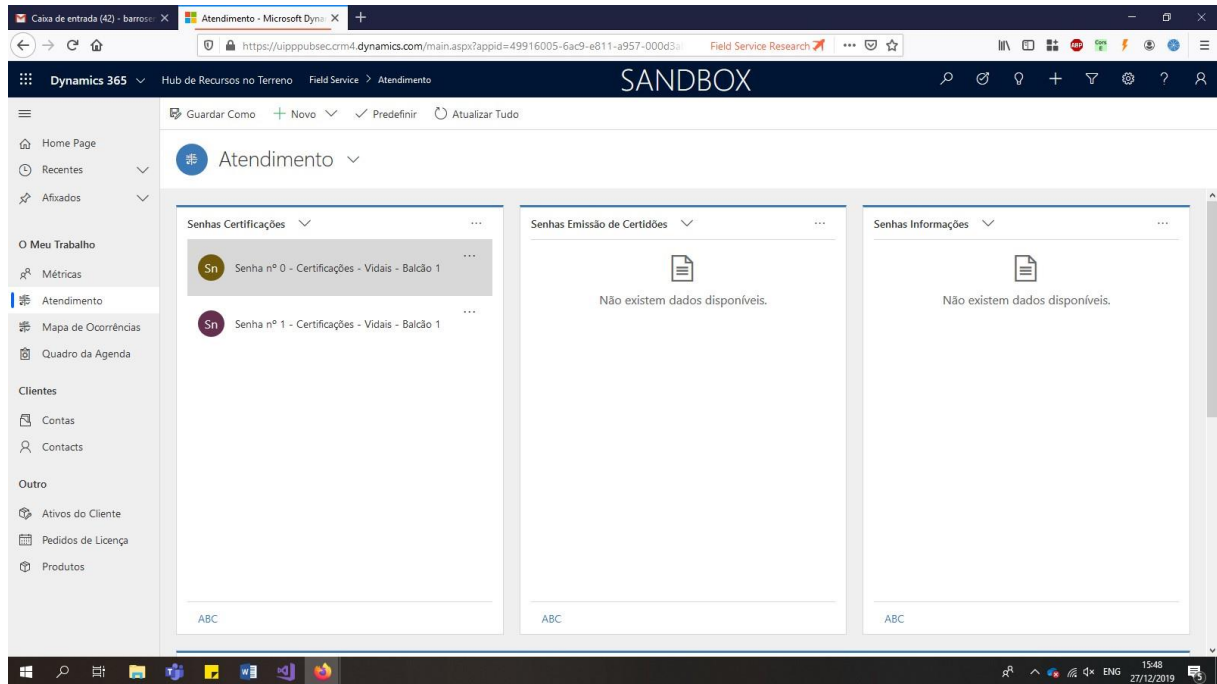


Figura 4.4 – As novas senhas aparecerão neste dashboard para o atendimento.

Ao seleccionar o registo, o utilizador é redireccionado para o formulário da senha onde poderá dá-la como concluída depois de atendido o cliente, assim como escrever algumas notas necessárias. O formulário poderá ser configurado à medida do cliente, sendo que neste caso apenas tem os dados base da Senha ilustrado na figura 4.5.

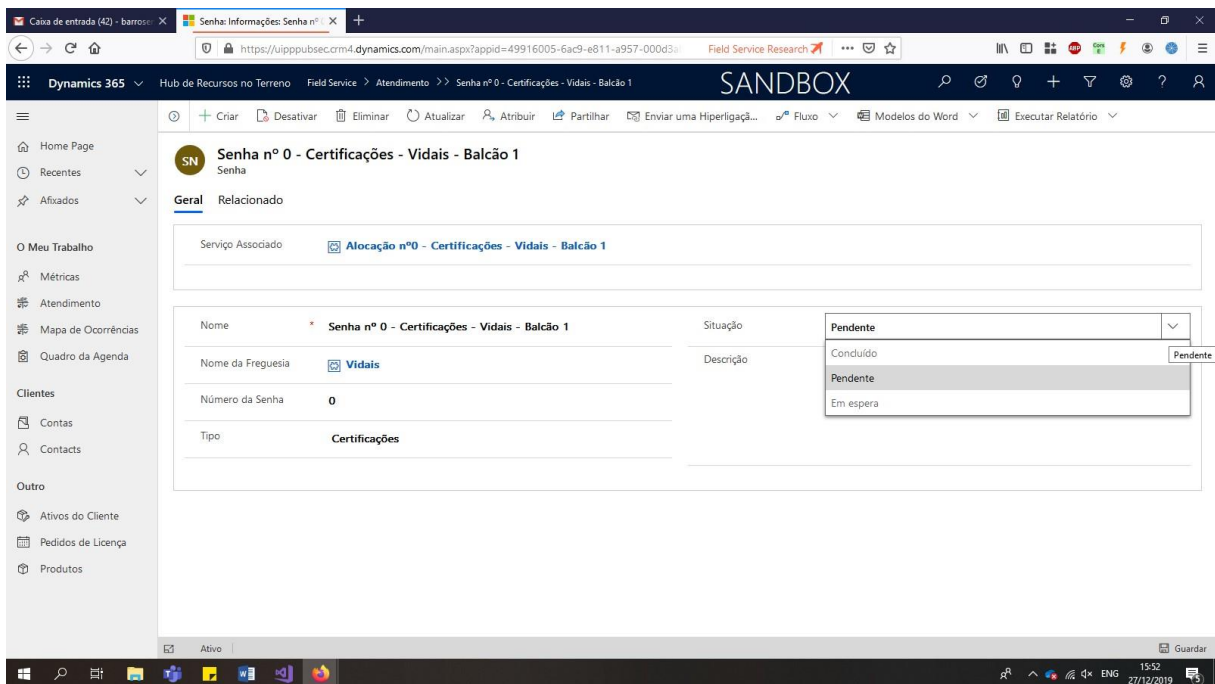


Figura 4.5 – Formulário de uma senha. Aqui o utilizador poderá marcar a senha como concluída depois do atendimento. Esta deixará de aparecer no dashboard anterior.

No *dashboard* do Quadro da Agenda, as senhas aparecerão no quadro como ilustra a figura seguinte:

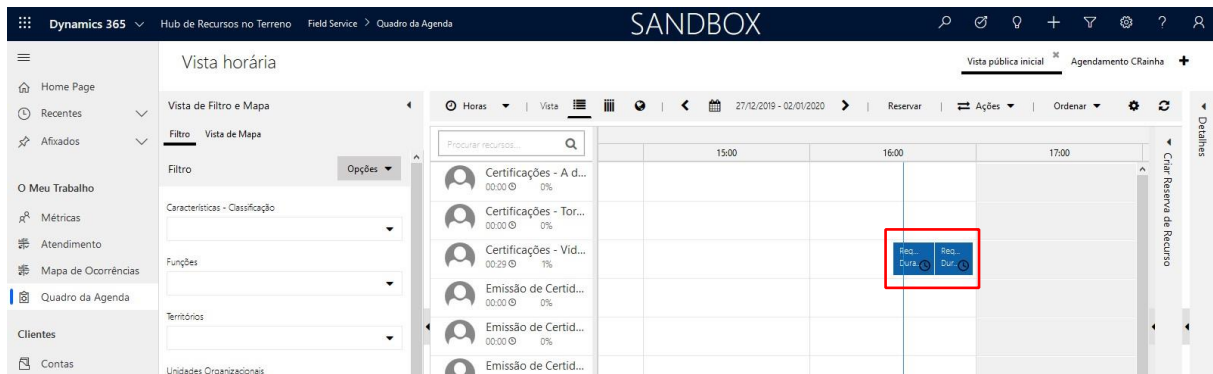


Figura 4.6 – As senhas como reservas de recurso reservável (neste caso o respectivo balcão).

4.2 Funcionalidade 2 – Georreferenciação de Incidentes

No que toca à georreferenciação, esta poderá ser feita de duas formas, ou através do BackOffice, ou por um técnico no terreno. Esta última apenas foi testada durante o desenvolvimento, sendo que devido à licença da aplicação ter expirado, não será incluída neste capítulo. Quando estamos no *dashboard* Mapa de Ocorrências, bastará clicar no símbolo '+' como demonstra a figura 4.7, e incluir os dados para a criação. A nova ocorrência é depois mapeada no mapa. Como as ocorrências são normalmente criadas pelos funcionários no terreno, a aplicação preenche automaticamente as coordenadas da ocorrência, no entanto no caso do *backoffice* estas têm de ser introduzidas manualmente. Existe a possibilidade de desenvolvimento delas serem automaticamente introduzidas consoante por exemplo o clique no mapa sobre a localização destas.

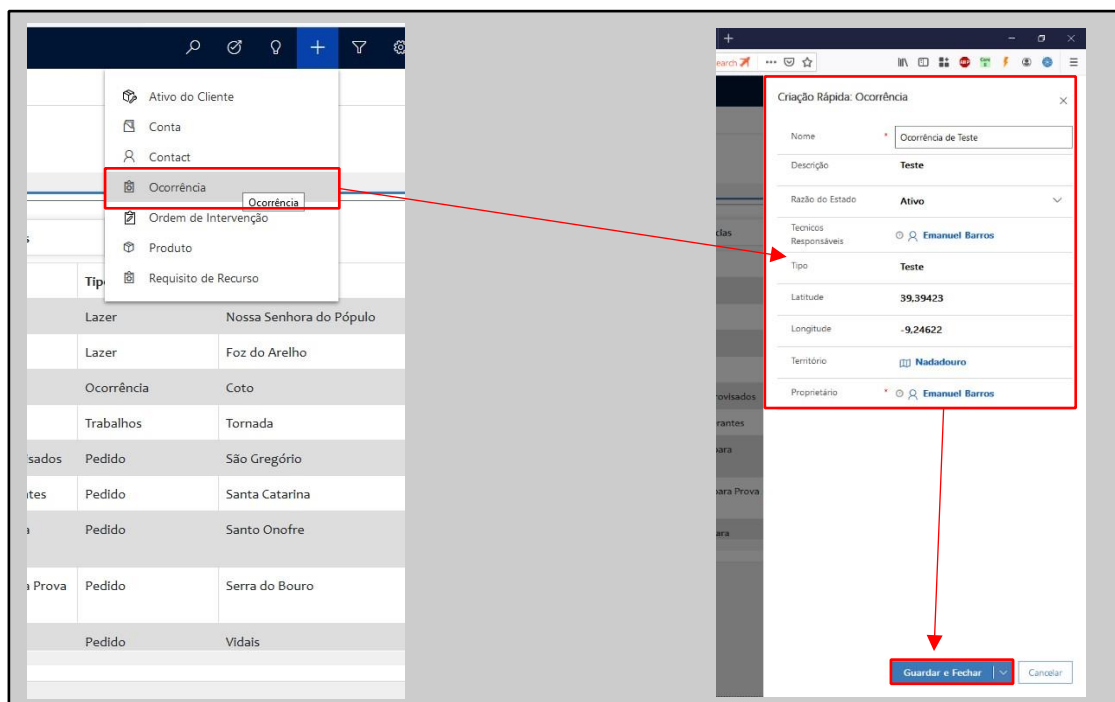


Figura 4.7 – Criação de uma nova ocorrência no lado do backoffice.

Implementação de uma solução de sistemas de informação Geográfica aplicado ao *Customer Relationship Management (CRM)*

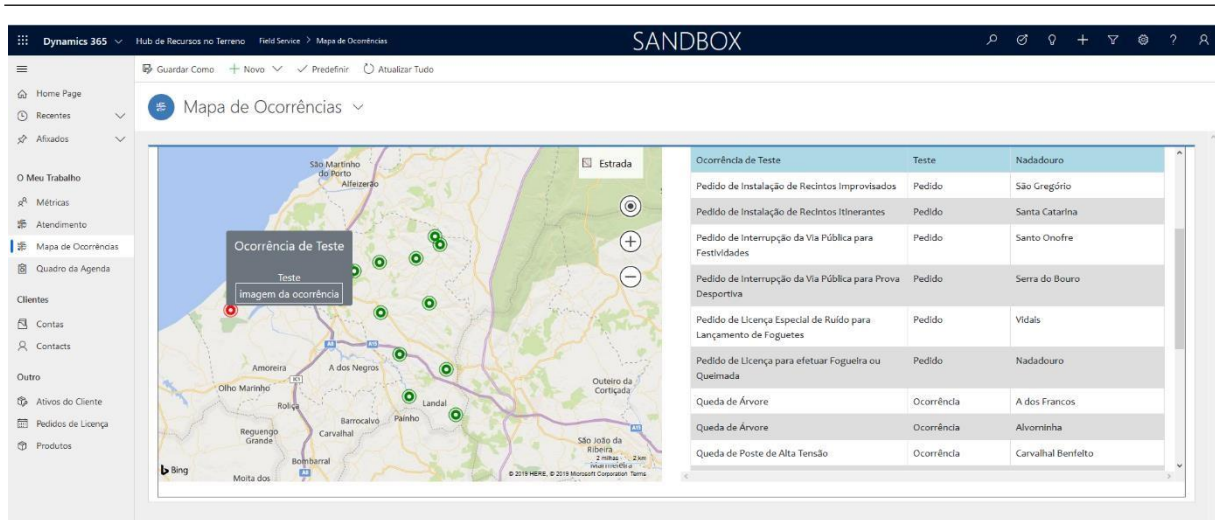


Figura 4.8 – A nova ocorrência do dashboard.

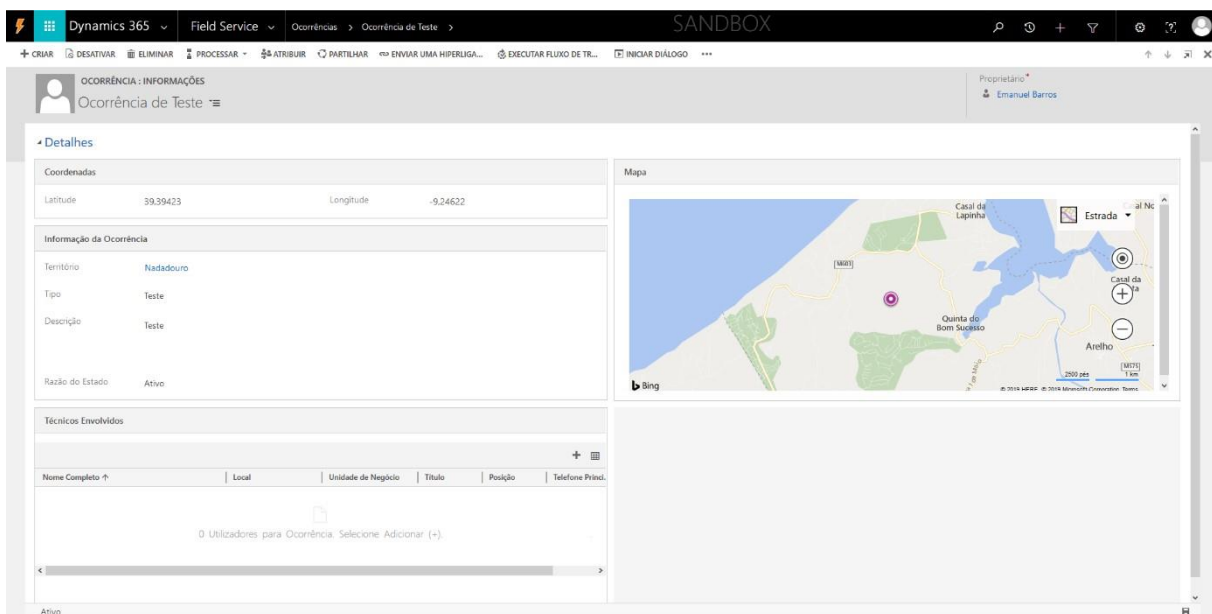


Figura 4.9 – Formulário da nova ocorrência.

Repare-se que na figura 4.9, onde está o formulário do novo registo criado, não está presente o técnico “Emanuel Barros” envolvido na ocorrência. Isto porque deu-se um erro na medida em que este técnico já está envolvido numa outra ocorrência no CRM. Este é o comportamento por defeito do Dynamics 365 no que toca a alocar os recursos reserváveis, podendo esta restrição ser retirada.

4.3 Funcionalidade 3 - Georreferenciação da última posição conhecida de um Técnico do terreno, assim como o histórico de posições desse mesmo técnico.

Como dito anteriormente, esta funcionalidade depende de uma licença da aplicação *RESCO*, na qual existem 30 dias para poder utilizar na sua totalidade. Depois deste tempo, a não ser que seja paga a licença, a aplicação fica indisponível. Durante a configuração desta aplicação e o teste da integração com a instância de Dynamics, o autor deixou a aplicação ligada no telemóvel da qual usou para este projeto enquanto fez uma viagem por Portugal, num período de férias. Aqui, ainda era usada a instância *trial*, antes desta ter sido migrada para a instância da empresa. Aí, outros 30 dias foram disponibilizados devido a esta instância ser diferente, no entanto à data destes testes, já este tempo teria passado novamente. A imagem 4.10 mostra o registo destas posições durante os dias de férias do autor, registados na instância de CRM. Estes registos não foram migrados para a nova instância, tendo sido depois perdidos quando a instância TRIAL expirou. É de notar as partes sem rasto. Constituem o facto de neste lado, o GPS não estar ligado. Hoje em dia, existem outras formas de conseguir as coordenadas aproximadas de um telemóvel sem recorrer ao GPS, e embora a aplicação não disponha de tais, poderá ser de interesse incluí-las no futuro para prevenir falta de posições quando o GPS está indisponível.

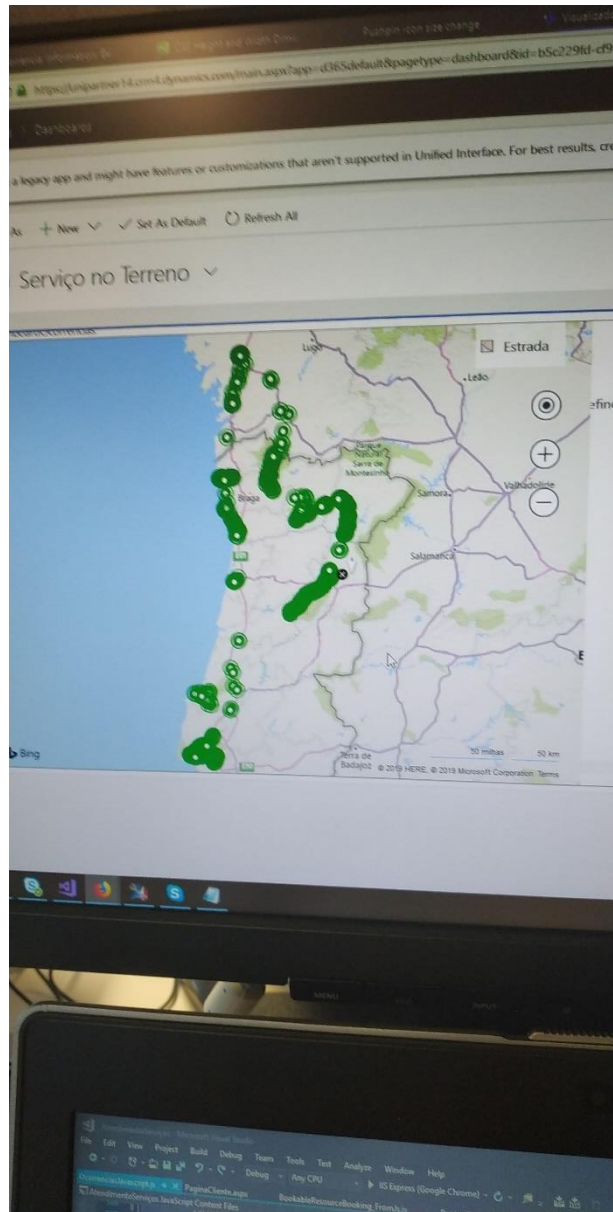


Figura 4.10 – Posições registadas durante a viagem do autor.

Capítulo 5 – Conclusão

O último capítulo deste projeto descreve os desenvolvimentos das funcionalidades de uma forma sucinta, as competências adquiridas assim como dificuldades encontradas e o trabalho futuro.

Neste trabalho, foi elaborado um sistema de informação geográfica tendo como base um CRM da *Microsoft* denominado por *Dynamics 365*. O objetivo do trabalho foi elaborar uma prova de conceito que permitisse a uma câmara municipal ter a gestão de técnicos no terreno e um sistema de agendamento de serviços que permite através do conhecimento geográfico da posição de um cidadão, o balcão na qual some o menor tempo de viagem e espera na fila deste. Isto permite a uma Câmara Municipal reduzir os tempos de espera dos atendimentos assim como gerir e canalizar os esforços nos balcões, pois com este sistema existe a visibilidade da aderência dos cidadãos aos balcões da câmara, sendo possível encaminhar esforços para os balcões com mais agendamentos.

A Centralização da informação no que toca a ocorrências e o posicionamento dos técnicos no terreno permite uma forma de tratar as ocorrências de forma mais simples e menos burocrática.

5.1 Competências Adquiridas

Este projeto permitiu ter visibilidade sobre as principais funcionalidades da *app Field Service* do *Dynamics 365*. Este conhecimento será fulcral caso exista algum cliente que pretenda consultadoria nesta área, pois este conhecimento ajudará a transmitir a melhor solução de serviço no terreno no âmbito de CRM a um cliente final. Aqui tornou-se benéfico a exploração do *Field Service*, pois é a área de CRM onde gostava de poder continuar como programador.

As competências adquiridas foram de facto mais técnicas. Com o desenvolvimento da tecnologia, pretende-se reduzir o desenvolvimento por código. Isto é devido à natureza *on-demand* da sociedade de hoje em dia, um cliente pretende um produto o mais rapidamente possível para poder ter ganho sobre ele o mais cedo possível. O conhecimento ao nível de linguagens web assim como linguagens *back-end* e desenvolvimento baixo-nível poderá não ter tanto impacto na área de CRM, visto que cada vez mais esta área exigem menos programação. Estão sendo desenvolvidas plataformas onde um utilizador poderá *configurar* e não *programar*. Plataformas essas como as *PowerApps*, similares a *OutSystems*. O facto de se poder configurar um CRM à medida do cliente evitando a programação minimiza as probabilidades de haver falhas nos sistemas e consequentes perdas para o cliente. Os conhecimentos adquiridos poderão apenas ser benéficos no caso de se ter de desenvolver mitigando todas as necessidades de um cliente futuro cirurgicamente.

A experiência *back-end* é benéfica nos próximos projetos, pois a probabilidade de haver problemas durante o desenvolvimento e situações de *trial and error* fica muito mais mitigada.

5.2 Dificuldades encontradas

A maior dificuldade encontrada foi de todo a gestão e desenvolvimento deste projeto com o pouco tempo disponível. O facto de trabalhar numa área diferente dos meus estudos, assim como ter ocupações que me ajudam em termos monetário tornou muito difícil o continuar deste trabalho ao longo do tempo, havendo fases de paragem do projeto. O facto de me ter de voltar a reinserir e a familiarizar-me novamente com toda a arquitetura contribuiu bastante para esta demora.

Outra dificuldade tratou-se de o autor ser o único da empresa explorar a área de serviços no terreno, não tendo feedback de ninguém caso tivesse alguma dificuldade, restando apenas a internet como fonte de resolução e familiarização dos problemas encontrados.

5.3 Trabalho futuro

Dada a natureza de constante desenvolvimento e novas ideias deste trabalho, e com o avançar da tecnologia, existirá sempre temas a desenvolver neste trabalho.

No produto final do projeto como um todo, os maiores desenvolvimentos são ao nível informático, de forma a que este trabalho passe de um POC a um sistema complementemente funcional à medida de um cliente final (Ex. uma Câmara Municipal).

Agora com a tecnologia *Portals do Dynamics 365*, não existe a necessidade de ter a página ao público num servidor à parte, visto que esta tecnologia oferece integração nativa com o *CRM*, assim como métricas bastante interessantes que otimizam o negócio, como saber onde um cidadão clicou no site, quanto tempo lá esteve, etc.

Devido à natureza da câmara e aos vários âmbitos que esta contém, um dos trabalhos sugeridos neste POC foi a otimização de rotas na recolha do lixo no território municipal. Ou uma aplicação ao público que permitisse um cidadão tirar uma foto de uma ocorrência e enviar para os serviços municipais de forma a analisar a situação. Estas não foram desenvolvidas devido ao mínimo tempo disponível para este trabalho.

Referências Bibliográficas

- Business to Community, *A Brief History of Customer Relationship Management*. Disponível em: <https://www.business2community.com/business-innovation/brief-history-customer-relationshipmanagement-01245936> [Online] [Acedido em Outubro de 2018]
- Code Project [2011]. *What is JQuery and How to Start using JQuery?*[Online] Disponível em: <https://www.codeproject.com/Articles/157446/What-is-jQuery-and-How-to-Start-using-jQuery> [Acedido em Junho de 2019]
- Extend, 2018. *What is REST – A Simple Explanation for Beginners, Part 1: Introducion*. [Online] Disponível em: <https://medium.com/extend/what-is-rest-a-simple-explanation-for-beginners-part-1introduction-b4a072f8740f> [Acedido em Julho de 2019]
- Forbes, 2018. *Council Post: Why is Customer Relationship Management so Important?* [Online] Disponível em: <https://www.forbes.com/sites/forbesagencycouncil/2017/10/24/why-is-customerrelationship-management-so-important/#36893a0b7dac> [Acedido em Outubro de 2018]
- MDN, 2018. *AJAX – Guias do Programador*. Disponível em: <https://developer.mozilla.org/ptPT/docs/Web/Guide/AJAX> [Acedido em Junho de 2019]
- METRO 4, 2019 – *Popular CSS, HTML and Javascript Library* [Online] Disponível em: <https://metroui.org.ua/intro.html> [Acedido em Julho de 2019]
- Microsoft Docs, 2018 – *Descrição Geral do Dynamics 365 for Field Service*. [Online] Disponível em: <https://docs.microsoft.com/pt-pt/dynamics365/customer-engagement/field-service/overview> [Acedido em Julho de 2019]
- Microsoft Docs, 2018. *Introduction to Entities (Developer Guide for Dynamics 365 for Customer Engagement apps)* [Online] Disponível em: <https://docs.microsoft.com/en-us/dynamics365/customer-engagement/developer/introduction-entities> [Acedido em Junho de 2019]
- Microsoft Docs, 2019. *Bing Maps REST Service*. [Online] Disponível em: <https://docs.microsoft.com/en-us/bingmaps/rest-services/index> [Acedido em Julho de 2019]
- Microsoft Docs, 2019. *Use FetchXML to construct a query*. [Online] Disponível em: <https://docs.microsoft.com/en-us/powerapps/developer/common-data-service/use-fetchxmlconstruct-query> [Acedido em Julho de 2019]
- Microsoft Docs, 2010. *IIS Express Overview*. [Online] Disponível em: <https://docs.microsoft.com/enus/iis/extensions/introduction-to-iis-express/iis-express-overview> [Acedido em Julho de 2019]
- SalesForce, 2018. Disponível em: <https://www.salesforce.com/eu/learning-centre/crm/what-is-crm/> [Online] [Acedido em Outubro de 2018].
- SearchWinDevelopment [2015]. *What is .NET Framework?* [Online] Disponível em: <https://searchwindevelopment.techtarget.com/definition/.NET> [Acedido em Junho de 2019]
- Top cloud providers 2019: AWS, Microsoft Azure, Google Cloud; IBM makes hybrid move; Salesforce dominates SaaS | ZDNet [Online] Disponível em: <https://www.zdnet.com/article/top-cloudproviders-2019-aws-microsoft-azure-google-cloud-ibm-makes-hybrid-move-salesforce-dominatessaas/> [Acedido em Julho de 2019]

What is Cloud Computing? A Beginner's Guide | Microsoft Azure [Online] Disponível em:
<https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/> [Acedido em Julho de 2019]