



·  
· **UNIVERSIDADE DE ÉVORA**

· **ESCOLA DE CIÊNCIAS E TECNOLOGIA**

· **DEPARTAMENTO DE INFORMÁTICA**

· **VISUAL SMART Mail**

· uma aplicação para visualização exploratória de  
· *email*

· **Gil Palma Ndecky**

· Orientação: Prof. Teresa Cristina de Freitas Gonçalves

· **Mestrado em Engenharia Informática**

· **Dissertação**

· **Évora, 2017**



**Mestrado em Engenharia Informática**

**VISUAL SMART Mail**  
**uma aplicação para visualização exploratória de *email***

Gil Palma Ndecky

**Orientador**

Teresa Cristina de Freitas Gonçalves





# Sumário

Utilizadores de *email* podem encontrar dificuldades em lidar com todas as mensagens que recebem com eficácia e eficiência, especialmente quando trabalham num ambiente empresarial. Saber como priorizar respostas e onde concentrar esforços é uma tarefa contínua e árdua sem uma solução óbvia.

O *SMART Mail* tem como objetivo unir a exploração gráfica de dados à análise em profundidade de *emails* trocados e contactos relevantes num ambiente empresarial de forma acessível para eliminar obstáculos e atrasos em cada utilização. A investigação documentada nesta dissertação foi iniciada com o intuito de descobrir as plataformas e ambientes de desenvolvimento mais apropriados para o projeto *SMART Mail*. Pretende-se com o presente documento apresentar os principais resultados e conclusões obtidos durante a investigação, determinar quais as melhores soluções relevantes disponíveis de momento para maximizar a qualidade do produto desenvolvido e, finalmente, discutir o que foi realizado e o trabalho necessário no futuro.

**Palavras-chave:** *email*, eficiência, ambiente empresarial, exploração gráfica de dados, análise em profundidade.



## *VISUAL SMART Mail*

# Abstract

*Email* users may have difficulty managing all incoming messages with efficacy and efficiency, especially when working in a business environment. Knowing how to prioritise responses or where to focus one's effort is a continuous and arduous task without an obvious solution.

SMART Mail's objective is to unify graphical data exploration with accessible, in depth analysis of exchanged emails and relevant contact parsing in business environments in order to reduce obstacles and delays with every single use. The research presented in this dissertation began with the intent of discovering the most well suited platforms and development environments for the SMART Mail project. This document contains the main findings and conclusions obtained during the investigation, the best available relevant solutions at this time to later maximize the quality of the developed product and, finally, a review of all the developed work up to this point and the required work towards the future.

**Keywords:** *email*, efficiency, business environment, graphical data exploration, in depth analysis.



# Agradecimentos

Agradeço à minha orientadora, Teresa Gonçalves, pelo seu apoio, paciência e motivação.

Agradeço à Universidade de Évora pela oportunidade.

Agradeço à Viatecla e concretamente aos meus colegas Ruben Roques, Daniel Ramos, Pedro Jacob e, especialmente, ao Ricardo Raminhos, Pedro Seabra, Nuno Miranda, Vasco Carvalho e ao Filipe Vieira por terem contribuído para este projeto, direta ou indiretamente, com ajuda concetual ou mecânica.

Agradeço à minha mãe pela sua compreensão e apoio.



# Acrónimos

**CSS** Cascading Style Sheets

**HTML** Hyper Text Markup Language

**XML** Extensible Markup Language

**IDE** Integrated Development Environment

**C#** C Sharp

**MoSCoW** Must Should Could Would

**SVG** Scalable Vector Graphics

**SDK** Software Development Kit

**IE** Internet Explorer

**D3** Data-Driven Documents

**PGSDK** Presentation Graphics Software Development Kit

**API** Application Programming Interface

**D3** Data-Driven Documents





# Conteúdo

<b>Sumário</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Lista de Conteúdo</b>	<b>xi</b>
<b>Lista de Figuras</b>	<b>xvi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização do Trabalho . . . . .	2
1.2 Motivações . . . . .	2
1.3 Objetivo principal e Requisitos . . . . .	3
1.4 Principais Contribuições . . . . .	3
1.5 Organização da Dissertação . . . . .	4
<b>2 Estado da Arte</b>	<b>5</b>
2.1 Técnicas de interfaces visuais de apresentação e navegação de dados . . . . .	5
2.1.1 Soluções Gerais . . . . .	5
2.1.2 Soluções de gestão de <i>Email</i> . . . . .	11
2.1.3 Soluções de <i>Email Marketing</i> . . . . .	15
2.2 Potenciais ambientes alvo e respetivas soluções gráficas . . . . .	18
2.2.1 iOS . . . . .	19
2.2.2 Android . . . . .	23
2.2.3 Windows Phone . . . . .	30
2.2.4 HTML5 . . . . .	34
2.2.5 Plugin Outlook . . . . .	42

2.3	Desenvolvimento multi-plataforma móvel . . . . .	44
2.3.1	PhoneGap . . . . .	44
2.3.2	Appcelerator Titanium . . . . .	46
2.3.3	Xamarin . . . . .	50
<b>3</b>	<b>Protótipos</b>	<b>53</b>
3.1	Protótipo Android . . . . .	53
3.2	Protótipo <i>Windows Phone 8</i> . . . . .	54
3.2.1	Sparrow Toolkit . . . . .	55
3.2.2	Silverlight Toolkit . . . . .	55
3.2.3	Visifire . . . . .	55
3.3	<i>Plugin Outlook</i> . . . . .	55
3.3.1	Gráficos Nativos do <i>Outlook</i> . . . . .	55
3.3.2	Gráficos <i>Web</i> . . . . .	57
3.4	Protótipos Multiplataforma . . . . .	57
3.4.1	Protótipo <i>PhoneGap</i> . . . . .	58
3.4.2	Protótipo <i>Appcelerator Titanium</i> . . . . .	59
3.5	Resumo das experiências . . . . .	59
3.5.1	Aplicações para dispositivos móveis . . . . .	60
3.5.2	Outlook . . . . .	63
3.6	Conclusões Finais das Experiências . . . . .	64
<b>4</b>	<b>Aplicação Desenvolvida</b>	<b>67</b>
4.1	Conceitos . . . . .	67
4.1.1	Conceitos base . . . . .	67
4.1.2	<i>Webserver</i> portátil . . . . .	68
4.1.3	Tecnologias <i>Web</i> . . . . .	69
4.1.4	Tecnologias <i>Microsoft</i> . . . . .	69
4.1.5	Termos auxiliares . . . . .	70
4.2	Requisitos do produto . . . . .	71
4.2.1	Pilares Conceituais . . . . .	71
4.2.2	Requisitos Gerais . . . . .	72
4.3	Desenho da arquitetura do produto . . . . .	77
4.4	Trabalho Desenvolvido . . . . .	79
4.4.1	Infraestrutura . . . . .	79
4.4.2	<i>Sidebar</i> . . . . .	80

CONTEÚDO	xi
4.4.3 Utilização do <i>Plugin</i> . . . . .	81
<b>5 Planeamento de testes de usabilidade</b>	<b>85</b>
5.1 Grupo de teste . . . . .	85
5.2 Formato do teste . . . . .	86
5.2.1 Gestão do <i>plugin</i> . . . . .	86
5.2.2 Interação com componentes <i>web</i> . . . . .	86
5.3 Passos de Teste . . . . .	86
<b>6 Conclusões e trabalho futuro</b>	<b>89</b>
6.1 Trabalho Futuro . . . . .	89
6.1.1 Módulos adicionais . . . . .	89
6.1.2 Melhorias técnicas . . . . .	90
6.1.3 Sistema de pontuação de <i>emails</i> . . . . .	90
6.1.4 Desenvolvimento do <i>Dashboard</i> . . . . .	90
6.1.5 Futuro do <i>0Inbox</i> . . . . .	93
6.2 Conclusões . . . . .	93
<b>Referências</b>	<b>102</b>



# Lista de Figuras

2.1	<i>Wolfram</i> - Lista de números . . . . .	6
2.2	<i>Wolfram</i> - Floco de Neve . . . . .	6
2.3	<i>Datawatch</i> - Painel atualizado em tempo real . . . . .	7
2.4	<i>Cinemetrics</i> - Exemplo de um filme . . . . .	8
2.5	<i>Cinemetrics</i> - Comparação entre dois filmes (2001: Uma Odisseia no Espaço e Aliens) . . . . .	8
2.6	<i>Cinemetrics</i> - Episódios de três programas diferentes (Jacques Cousteau, jogo de Futebol, Simpsons) . . . . .	8
2.7	<i>Visual Action</i> - <i>Honeycomb iTunes</i> . . . . .	9
2.8	<i>Tableau</i> - Distribuição de visualizações . . . . .	10
2.9	<i>Gapminder</i> - Esperança Média de Vida - 1800 e 2012 . . . . .	10
2.10	<i>CARTO</i> - Todos os crimes . . . . .	11
2.11	<i>CARTO</i> - Roubos e vandalismo . . . . .	11
2.12	<i>Politemail</i> - Relatório com estatísticas . . . . .	12
2.13	<i>ClearContext</i> - Visão Geral . . . . .	12
2.14	<i>ClearContext</i> - Eficiência do utilizador . . . . .	12
2.15	<i>Bells &amp; Whistles</i> - Seleção de métricas em intervalo . . . . .	13
2.16	<i>Bells &amp; Whistles</i> - Tráfego diário . . . . .	13
2.17	<i>Gmail Meter</i> - Otimização . . . . .	14
2.18	<i>Gmail Meter</i> - Volume de mensagens trocadas . . . . .	14
2.19	<i>Xobni</i> - Tráfego de mensagens . . . . .	14
2.20	<i>GetResponse</i> - Estatísticas de uma <i>Newsletter</i> . . . . .	15
2.21	<i>SumAll</i> - Exemplo de aglomeração de dados . . . . .	16
2.22	<i>Silverpop Engage</i> - Campanha de marketing Adobe Analytics . . . . .	17

2.23	<i>MailChimp</i> - Métricas da campanha . . . . .	17
2.24	<i>Zoho Campaigns</i> - <i>Zoho Charts</i> . . . . .	18
2.25	<i>SendinBlue</i> - Gráfico de temperatura de <i>clicks</i> em <i>emails</i> . . . . .	19
2.26	<i>iOS</i> - Destaques da <i>Appstore</i> em <i>iPhone</i> . . . . .	20
2.27	<i>iOS</i> - <i>Appstore</i> em <i>iPad</i> . . . . .	20
2.28	<i>Core Plot</i> - Gráfico de setores inicial . . . . .	21
2.29	<i>Core Plot</i> - Painel <i>Coreplot</i> . . . . .	21
2.30	<i>Shinobi Controls</i> - Combinação de gráficos 2012 . . . . .	22
2.31	<i>JBChartView</i> - Precipitação mensal . . . . .	23
2.32	<i>JBChartView</i> - Precipitação anual . . . . .	23
2.33	<i>iOS: Chart</i> - Gráficos de área empilhados . . . . .	24
2.34	<i>iOS: Chart</i> - Gráfico 3D . . . . .	24
2.35	<i>Android</i> - Diferenças entre a primeira versão (esquerda), a versão <i>Jelly Bean</i> 4.1 (centro) e a <i>KitKat</i> 4.4 (direita) . . . . .	24
2.36	Vendas de <i>smartphones</i> . . . . .	25
2.37	<i>Android</i> - Distribuição de versões em Maio 2016 . . . . .	26
2.38	<i>AChartEngine</i> - Exemplo pouco polido . . . . .	27
2.39	<i>AChartEngine</i> - Exemplo mais polido . . . . .	27
2.40	<i>MPAndroidChart</i> - Gráfico de linhas simples . . . . .	28
2.41	<i>MPAndroidChart</i> - Gráfico de linhas estilizado . . . . .	28
2.42	<i>Skia</i> - <i>Composed Path</i> , <i>Sum Path</i> e <i>Shaders</i> . . . . .	28
2.43	<i>Androidplot</i> - Gráfico de barras . . . . .	29
2.44	<i>Androidplot</i> - Gráfico de barras com gradiente . . . . .	29
2.45	<i>Androidplot</i> - Gráfico de secções . . . . .	29
2.46	<i>GraphView</i> - Gráfico de linhas estilizado . . . . .	30
2.47	<i>Windows Phone</i> - Exemplos de personalização do menu inicial . . . . .	31
2.48	<i>Visifire</i> - Velocímetro . . . . .	32
2.49	<i>Visifire</i> - Indicadores para comparação de receitas . . . . .	32
2.50	<i>Visifire</i> - Combinação de gráficos de receitas e lucro . . . . .	32
2.51	<i>Visifire</i> - <i>Spline</i> complexo . . . . .	32
2.52	<i>Jack of Tools Pro (RadChart)</i> - Velocidade atual e histórico . . . . .	33
2.53	<i>HTML5</i> - “ <i>DeviantART muro</i> ” . . . . .	35
2.54	<i>HTML5</i> - <i>Fritzo Keys Classic</i> . . . . .	35
2.55	<i>HTML5</i> - <i>Fritzo Keys Flow</i> . . . . .	35

2.56	<i>HTML5</i> - SVG + D3JS . . . . .	36
2.57	<i>HTML5</i> - Barras empilhadas . . . . .	37
2.58	<i>HTML5</i> - Barras agrupadas . . . . .	37
2.59	<i>HTML5</i> - Países ordenados por Déficit . . . . .	38
2.60	<i>HTML5</i> - Dependências Flare . . . . .	39
2.61	<i>Highcharts</i> - Gráfico de área claro . . . . .	39
2.62	<i>Highcharts</i> - Gráfico de área escuro . . . . .	39
2.63	<i>Highcharts</i> - Gráfico de área com dados parciais . . . . .	40
2.64	<i>Highcharts</i> - Mapa de temperatura 2013 . . . . .	40
2.65	<i>CanvasJS</i> - Gráfico de área com coordenada selecionada . . . . .	41
2.66	<i>Google Charts</i> - Gráfico de secções . . . . .	42
2.67	<i>Google Charts</i> - Gráfico de Mapa . . . . .	42
2.68	<i>Google Charts</i> - <i>Treemap</i> . . . . .	42
2.69	<i>NUS extra</i> - Exemplo de mapa com informação sobre um marcador específico	46
2.70	<i>HealthTap</i> - Comunicação por texto ou por vídeo . . . . .	47
2.71	<i>BrowserQuest</i> - Exploração de uma praia . . . . .	47
2.72	<i>BrowserQuest</i> - Exploração de uma caverna . . . . .	48
2.73	<i>BrowserQuest</i> - Combate <i>online</i> com outro jogador . . . . .	48
2.74	<i>Appcelerator</i> - Gráfico de barras . . . . .	49
2.75	<i>Appcelerator</i> - Mistura de gráficos . . . . .	49
2.76	<i>Appcelerator</i> - Lista de parques (ParkYa) . . . . .	51
2.77	<i>Appcelerator</i> - Mapa de rua (ParkYa) . . . . .	51
2.78	<i>Appcelerator</i> - Detalhe de rua (ParkYa) . . . . .	51
2.79	<i>Appcelerator</i> - Pagamento (ParkYa) . . . . .	51
2.80	<i>Xamarin</i> - Exemplo de diferenças entre plataformas . . . . .	52
2.81	<i>Xamarin</i> - A mesma aplicação apresentada em cada uma das três platafor- mas alvo . . . . .	52
3.1	Protótipo <i>AChartEngine</i> - Painel de gráficos de tempo meteorológico . . . . .	54
3.2	Protótipo <i>AChartEngine</i> - Gráfico de secções interativo . . . . .	54
3.3	Protótipo <i>Visifire</i> - Gráfico <i>spline</i> com área do plano de fundo realçada . . . . .	56
3.4	Protótipo <i>Visifire</i> - Indicador circular . . . . .	56
3.5	Protótipo <i>Plugin Outlook</i> - Gráficos Nativos (Barras e Secções) . . . . .	56
3.6	Protótipo <i>Plugin Outlook</i> - Gráficos Nativos (Linhas, Pontos e Funil) . . . . .	56

3.7	Protótipo <i>Plugin Outlook</i> - Gráficos Nativos (Área e Teia) . . . . .	57
3.8	Protótipo <i>Plugin Outlook</i> - Apresentação de uma página <i>web</i> inteira . . . . .	58
3.9	Protótipo <i>PhoneGap</i> - Android . . . . .	59
3.10	Protótipo <i>PhoneGap</i> - Windows Phone . . . . .	59
3.11	Protótipo <i>Appcelerator</i> - <i>Android</i> . . . . .	60
3.12	Protótipo <i>Appcelerator</i> - <i>Windows Phone</i> . . . . .	60
4.1	<i>Treemap</i> - Milhões de <i>megawatt</i> por hora de energias renováveis por país em 2008 . . . . .	71
4.2	Desenho da estrutura do SMART Mail . . . . .	78
4.3	<i>0Inbox</i> - Splash . . . . .	82
4.4	<i>0Inbox</i> - Visão principal e Anexos . . . . .	82
4.5	<i>0Inbox</i> - Contacto . . . . .	83
4.6	<i>0Inbox</i> - Opções . . . . .	83
6.1	<i>0Inbox</i> - <i>Dashboard Mockup</i> . . . . .	91



# Capítulo 1

## Introdução

Apesar de a sua popularidade ter diminuído com o surgimento de imensos sistemas de comunicação alternativos, atualmente o *email* continua a ser um método de comunicação importante em muitas organizações. A permanência e assincronia das mensagens, combinadas com funcionalidades adicionais, tais como calendários partilhados entre diferentes utilizadores, complementam a caixa de correio de forma a que funcione também como uma base de dados, um fórum de conversações ou até como um assistente pessoal. Nenhum outro método de comunicação oferece o mesmo conjunto de funcionalidades com a mesma abertura, flexibilidade e familiaridade. Assim que tenham um endereço de *email*, utilizadores não precisam de mais registos ou definições para trocar mensagens com qualquer contacto.

No entanto, uma caixa de *email* sem ordem ou estrutura não é simples de navegar. Mesmo com alguma organização básica, tal como uma ordenação cronológica ou alfabética, conseguir encontrar os *emails* mais importantes diariamente não é uma tarefa fácil para quem os recebe em grandes quantidades. Ao tentar resolver estes problemas e outros, imensas soluções têm sido criadas nos últimos anos que tentam dar sentido a caixas de *email*. Uma das soluções mais antigas e bem sucedidas são as pastas e filtros personalizáveis que permitem a cada utilizador agrupar *emails* recebidos com base no endereço do emissor, título e corpo de texto do email. Um utilizador pode criar uma pasta dedicada a uma determinada organização ou indivíduo de alta importância de forma a nunca perder uma mensagem crucial para o seu negócio, ou uma pasta que receba tudo o que for enviado de um domínio pouco interessante de modo a que estas mensagens irrelevantes não encham a caixa de entrada. O exemplo mais conhecido de uma dessas combinações de pastas e filtros é a caixa de *spam*, onde todo o correio não desejado é automaticamente colocado. Os lembretes são outra solução pensada para facilitar a utilização de uma caixa de correio, oferecendo a

um utilizador a opção de revisitar *emails* importantes facilmente ao gerar eventos que os realçam numa data futura. Ambas estas soluções foram adotadas por muitas ferramentas de gestão de *email* e cada serviço ainda oferece as suas próprias soluções. Por exemplo, o *MailChimp* [1] fornece aos seus clientes ferramentas ligadas ao *marketing* e venda de produtos, enquanto que o *Xobni* [2][3] se integra, visual e funcionalmente, na plataforma de utilização de email *Office Outlook* [4] de modo a oferecer uma lista de contactos super poderosa sem perturbar o conforto do utilizador. Estas e outras soluções serão analisadas mais aprofundadamente no capítulo 2.

É com o objetivo de melhorar a utilização de *email* para utilizadores em ambiente empresarial que este trabalho foi iniciado.

## 1.1 Contextualização do Trabalho

O *SMART Mail* é um projeto da iniciativa da VIATECLA [5] com o objetivo de oferecer um complemento às capacidades do *Microsoft Outlook* e tornar-se uma ferramenta indispensável para empresários que assentam o seu negócio no *email*. A componente visual do *SMART Mail*, estudada neste documento, é complementada pelo conjunto de mecanismos lógicos que tratam de encontrar métricas compostas, tais como as que permitem pontuar e classificar *emails*, contactos e domínios.

## 1.2 Motivações

A natureza portátil dos *smartphones* e outros dispositivos móveis permite que utilizadores de *email* tenham acesso às suas caixas de correio em qualquer lado. O surto de popularidade deste tipo de dispositivos levou a que algumas organizações afiliadas a *email* se tenham concentrado no desenvolvimento de aplicações móveis que tirem o máximo proveito deste meio relativamente novo. Há que ter em conta, no entanto, que o aumento de popularidade de utilização de dispositivos móveis para ler *email* engloba todos os tipos de utilização e não só em ambientes empresariais.

Em ambientes empresariais, dezenas ou até mesmo centenas de *emails* são trocados todos os dias por cada utilizador e as soluções de *email* móveis podem não oferecer as funcionalidades e ferramentas necessárias para processar quantidades tão grandes de *emails* eficientemente. E mesmo que ofereçam, utilizadores frequentes de *email* procuram o conforto que vem de poder utilizar um rato e um teclado, além de potencialmente múltiplos monitores. Para este tipo de utilizador, apesar de a mobilidade por vezes ser importante, é crucial conseguir rapidamente escrever muito texto, completar mensagens com texto copiado de outra fonte ou gerir o calendário, anexos ou tarefas. Nesse caso, estes utilizadores poderão preferir um serviço *desktop*.

Além disso, as soluções de *email desktop* oferecem cada vez mais formas rápidas de vi-

sualizar métricas importantes. Seja através de gráficos ou de dados numéricos diretos, a oferta deste tipo de informação tem aumentado consideravelmente. Consequentemente, utilizadores habituais de *email* habituam-se a trabalhar com o auxílio de funcionalidades poderosas e fáceis de usar, levando a que desejem progressivamente mais funcionalidades avançadas de quaisquer plataformas de gestão de *email* com que trabalhem, especialmente de plataformas modernas. Apesar do aumento da procura e oferta deste tipo de funcionalidades, mesmo as soluções de topo atualmente são limitadas nos dados que apresentam e como o fazem. Cada solução proporciona certas funcionalidades mas nenhuma as oferece todas. Este vácuo foi a motivação principal para o desenvolvimento deste produto.

### 1.3 Objetivo principal e Requisitos

O objetivo principal do trabalho desenvolvido é oferecer formas poderosas de visualizar e explorar informação importante para utilizadores de *email*. Para atingir este objetivo, foram desenvolvidas as seguintes tarefas:

- Estudo e análise das soluções existentes mais promissoras de criação de gráficos;
- Estudo e análise das soluções existentes mais promissoras de gestão de caixas de *email*;
- Estudo e análise dos meios de desenvolvimento e plataformas mais adequadas para o produto;
- Desenvolvimento de protótipos que procurem testar a viabilidade de algumas combinações de plataformas e abordagens;
- Definição dos requisitos do produto final;

### 1.4 Principais Contribuições

Como um dos vários colaboradores deste projeto, não fui responsável pela totalidade do trabalho desenvolvido. Inicialmente, o meu plano de trabalho continha apenas a implementação dos aspetos visuais do produto, incluindo a implementação do estilo geral da interface e os gráficos numéricos. Devido ao meu nível crescente de conhecimento do estado do produto, proveniente do contacto necessário com a estrutura do código de forma a implementar as soluções gráficas, comecei a auxiliar outros colaboradores com tarefas incrementalmente mais estruturais até ao ponto em que me tornei uma das principais forças motrizes atrás da aplicação desenvolvida. As áreas por que fui responsável, parcial ou completamente, são:

- Estudo e escrita do estado da arte;

- Criação de protótipos;
- Desenvolvimento de parte da Infraestrutura;
- Implementação da leitura e processamento de meta dados de *emails* do *Outlook* e a passagem destes dados entre *C#* e *Javascript*;
- Planeamento de testes de usabilidade;
- Desenvolvimento da estrutura *HTML* para a *Sidebar*;
- Criação de versões executáveis para partilha e testes internos;

## 1.5 Organização da Dissertação

Este documento está dividido em quatro capítulos. O capítulo 2, Estado da Arte, apresenta os resultados da investigação feita com o intuito de descobrir todas as ferramentas que se alinhem com os requisitos do projeto e as plataformas mais utilizadas e interessantes onde o produto final poderá ser executado. De seguida, o capítulo 3, Protótipos, fala das aplicações de teste criadas para testar algumas das plataformas e ferramentas mais promissoras. O capítulo 4 apresenta o estado da Aplicação Desenvolvida. O capítulo 5 apresenta o Planeamento de testes de usabilidade e, finalmente, no capítulo 6 estão delineados os planos para trabalho no futuro e as conclusões formadas.

# Capítulo 2

## Estado da Arte

Neste capítulo vai ser introduzido o conjunto de dados recolhidos sobre as ferramentas e ambientes mais frequentemente associados ao *email* de modo a ficar a conhecer como melhor criar produtos, tanto poderosos como acessíveis, que preencham as lacunas existentes e não repitam erros que outros tenham cometido. Esta investigação foi iniciada com o intuito de descobrir as plataformas e os ambientes de desenvolvimento gráfico mais apropriados para o projeto *SMART Mail*. O capítulo está dividido em três secções: Técnicas de Interfaces Visuais, Potenciais Ambientes Alvo e Desenvolvimento Multi-plataforma.

### 2.1 Técnicas de interfaces visuais de apresentação e navegação de dados

Esta secção vai reunir e observar as melhores e mais interessantes ferramentas de visualização e análise de dados utilizadas hoje em dia, estejam estas relacionadas com *email* ou não, para que possam vir a ser consideradas no processo de conceção e desenvolvimento do *plugin*. Primeiro, na secção de Soluções Gerais são apresentadas várias ferramentas que utilizam gráficos de formas interessantes, não relacionadas com *email*, e posteriormente são mencionadas soluções que aliam gráficos a *email*, tanto para utilização pessoal como para análise de grandes volumes de mensagens.

#### 2.1.1 Soluções Gerais

O *Wolfram Alpha* [6] é um motor de conhecimento computacional com um grande número de possíveis aplicações. A caixa de pesquisa recebe tanto fórmulas matemáticas como

linguagem natural, especificamente em Inglês, e apresenta o conjunto de resultados ao utilizador. Esta ferramenta complementa toda a informação que disponibiliza com gráficos para visualizar os dados de várias formas, diretamente no *browser*. Na figura 2.1 podemos ver como o *Wolfram Alpha* interpreta uma lista de números, em particular quando coloca os números dessa lista num referencial para os visualizar. Como não foi realizado um pedido específico quanto à forma de processar e apresentar os dados o motor respondeu com as soluções mais prováveis. Os gráficos também podem ser específicos para um certo conceito como na figura 2.2, que representa o fractal “Floco de Neve” de Koch.

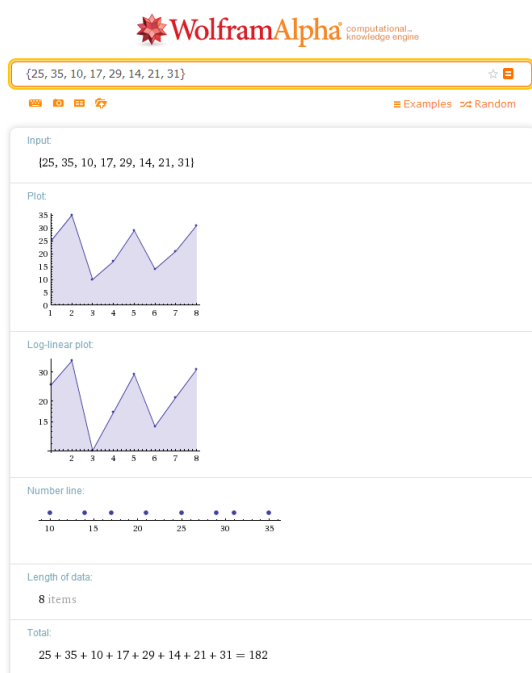


Figura 2.1: *Wolfram* - Lista de números

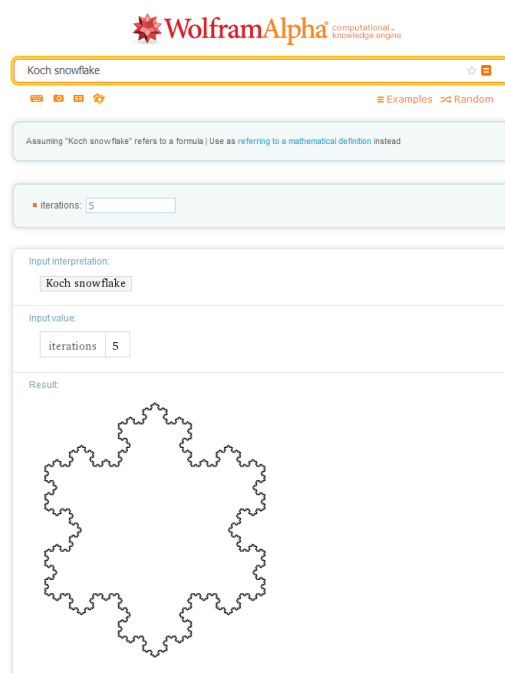


Figura 2.2: *Wolfram* - Floco de Neve

O *Datawatch* [7] oferece uma grande variedade de opções de visualização de qualquer tipo de dados com a possibilidade de o fazer em tempo real. A aplicação permite escolher a quantidade e tipo de gráficos e montar um *dashboard* personalizado e partilhável, podendo alternar entre uma visão geral de todas as variáveis inseridas e uma mais detalhada e concentrada. Esta solução foi desenhada para simplificar a criação de gráficos para qualquer utilizador, sendo mais acessível do que o *Wolfram Alpha* nesse sentido. No entanto, a oferta de gráficos está limitada aos géneros habituais, apesar de contar com uma variedade considerável, e não atinge o mesmo potencial no que toca a possibilidades de desenho. Esta diferença não é um problema mas, de facto, uma consequência de estas ferramentas terem públicos alvo e propósitos diferentes. Na figura 2.3 podemos ver um painel atualizado em tempo real que mostra a dispersão de várias transações, dependendo do seu intermediário e da moeda usada. Podem-se também criar painéis adaptáveis para qualquer combinação de *browser* e dispositivo, havendo diferenças no “*look and feel*” para ter o maior conforto em qualquer situação.



Figura 2.3: *Datawatch* - Painel atualizado em tempo real

*Cinemetrics* [8] é um projeto de Licenciatura de Frederic Brodbeck cujo propósito foi criar “impressões digitais” de filmes. Utilizando diversos dados sobre um filme, desde as cores principais de cada cena até ao movimento da câmara, este programa cria uma roda animada que sintetiza dados recolhidos de um filme numa imagem simples de interpretar, com a possibilidade de escolher cenas específicas para ter informação mais detalhada. Este sistema permite não só ter uma visão global de um filme instantaneamente como também facilita a comparação de filmes. As três componentes apresentáveis nesta forma de visualização são o tamanho, a irregularidade (ou movimento no caso da imagem ser animada) e as cores da roda, correspondendo à duração, ao movimento das imagens e às cores mais utilizadas, respetivamente. Contudo, esta ferramenta não funciona só com filmes, podendo ser aplicada a qualquer tipo de vídeo. Apesar de este projeto não ter aplicações científicas óbvias, esta forma de apresentar dados oferece um ponto de vista interessante fazendo surgir informação que não se obteria facilmente de outra forma. As figuras 2.4, 2.5, e 2.6 mostram os resultados obtidos com alguns filmes, um jogo de Futebol e um episódio dos *Simpsons*. Na figura 2.5 há uma diferença aparente na quantidade de movimento de cada filme. O '2001: A Space Odyssey' é um filme com muito pouco movimento, o que se torna ainda mais aparente quando em contraste com o 'Aliens', um filme cheio de ação e movimentos rápidos de câmara. A figura 2.6 mostra três programas televisivos lado a lado, incluindo um jogo de Futebol e um episódio dos *Simpsons*. Nesta imagem, a diferença de duração e cores entre diferentes fontes torna-se óbvia. Na página do projeto existe

um vídeo que mostra vários exemplos em movimento, permitindo ver como funciona a componente do movimento do filme melhor do que com as imagens estáticas apresentadas neste documento.

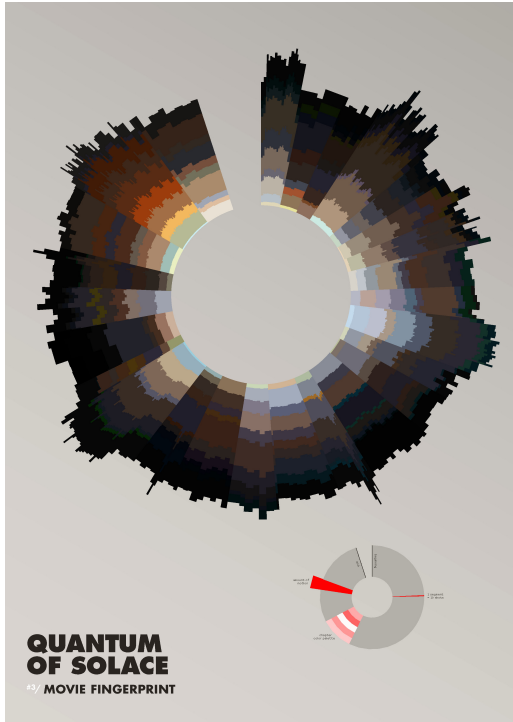


Figura 2.4: *Cinematics* - Exemplo de um filme

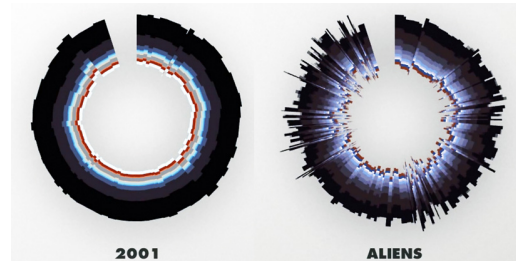


Figura 2.5: *Cinematics* - Comparação entre dois filmes (2001: Uma Odisseia no Espaço e Aliens)



Figura 2.6: *Cinematics* - Episódios de três programas diferentes (Jacques Cousteau, jogo de Futebol, Simpsons)

'The Hive Group' era uma empresa especializada em oferecer formas de criar *treemaps* para negócios. Tinham soluções mais robustas para ambientes de trabalho mais completos mas também ofereciam alternativas móveis e *web* para utilizadores sem acesso às ferramentas habituais. Durante o período de escrita desta dissertação, esta empresa fundiu-se com outra, *Visual Action* [9], e a nomenclatura dos produtos mudou um pouco.

O conjunto de soluções que oferecem conhecido anteriormente como *Honeycomb* agora é chamado *Flaremap*. Cada produto individual deste conjunto difere na forma de apresentação dos dados para clientes com diferentes necessidades ou limitações, sendo utilizável em várias combinações de plataformas ou dispositivos. Na figura 2.7 [10] podemos ver um *treemap* com as 100 músicas mais populares no *iTunes* em 2005 e pedir mais informação ao clicar em cada canção individual. Os exemplos interativos que a 'The Hive Group' disponibilizava no seu *site* já não se encontram disponíveis mas a *Visual Action* disponibiliza a potenciais clientes uma exploração ao vivo acompanhada por guias especializados [11].

O *Tableau* [12] é um conjunto de soluções, desenvolvido por uma empresa com o mesmo nome, cujo propósito é criar gráficos utilizando um conjunto variado e extenso de ferramentas com um foco em criação de mapas informativos e/ou interativos. Todas as soluções deste conjunto são usadas para os mesmos fins, divergindo apenas nos ambientes para que



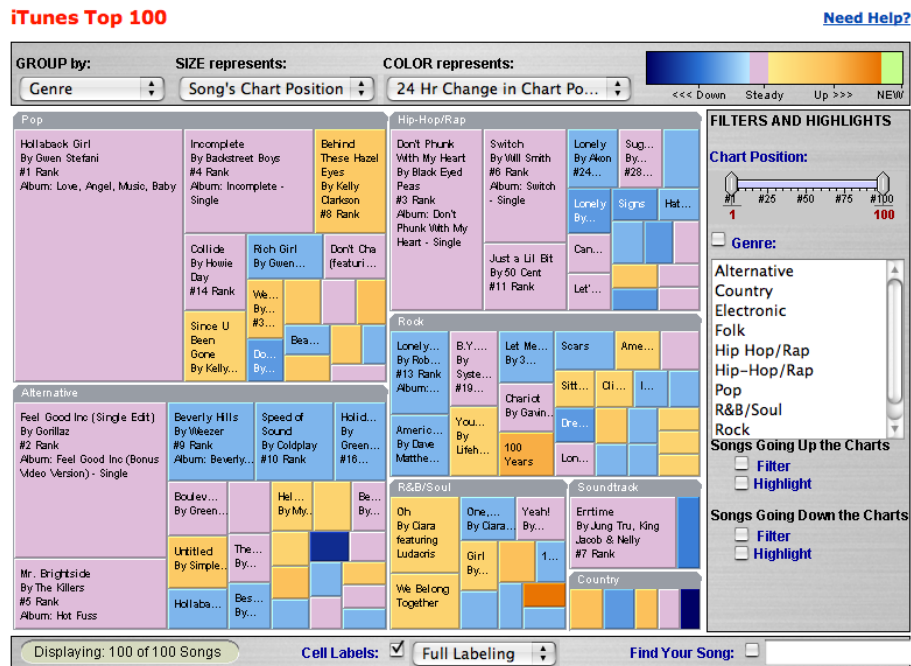


Figura 2.7: *Visual Action - Honeycomb iTunes*

cada uma foi desenhada. Tal como com o *Datawatch*, é possível criar painéis de controlo personalizados e partilhá-los.

A facilidade de adicionar mapas faz com que este conjunto de soluções se destaque quando o tema do negócio em questão afeta uma área extensa, permitindo conhecer as regiões geográficas que mais utilizaram cada parte de um negócio. O caso particular da figura 2.8 tem a distribuição de visualizações de uma página por secção do próprio e por país, seguido de um gráfico temporal. É possível ainda aceder a formas alternativas de ver os dados para ter uma visão pormenorizada. Existem demonstrações interativas no *website* [13], permitindo ver com detalhe os dados disponíveis.

A tecnologia por trás deste produto pertence à *RedPoint Global* [14], uma empresa com ferramentas desenhadas para oferecer tudo o que uma organização precisa para planear, criar e executar campanhas de *marketing*.

Hans Rosling [15] é um médico e estatístico que fundou o *Gapminder* [16], uma ferramenta de visualização de dados altamente personalizável e especializada em tornar evidente comportamentos e padrões globais. O *Gapminder* permite criar gráficos e animá-los de acordo com as alterações dos dados em cada ano, permitindo um tipo de exploração temporal dos dados não realizável com outras ferramentas. Na figura 2.9 está visível a relação entre o tempo médio de vida e o Produto Interno Bruto de cada país em 1800 e em 2012, respetivamente, com Portugal destacado. O tamanho de cada círculo representa a dimensão da população do país correspondente. Ao pressionar no botão “*Play*” no fundo da página, o *Gapminder* mostra o percurso de cada país desde o ano 1800 até ao ano 2012, evidenciando

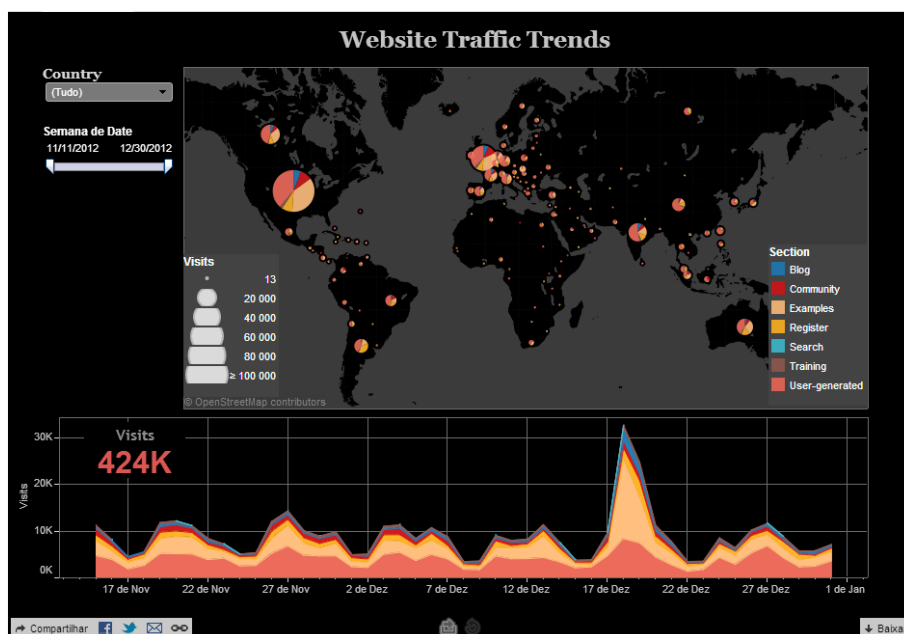


Figura 2.8: Tableau - Distribuição de visualizações

as diferenças entre os países neste estudo.

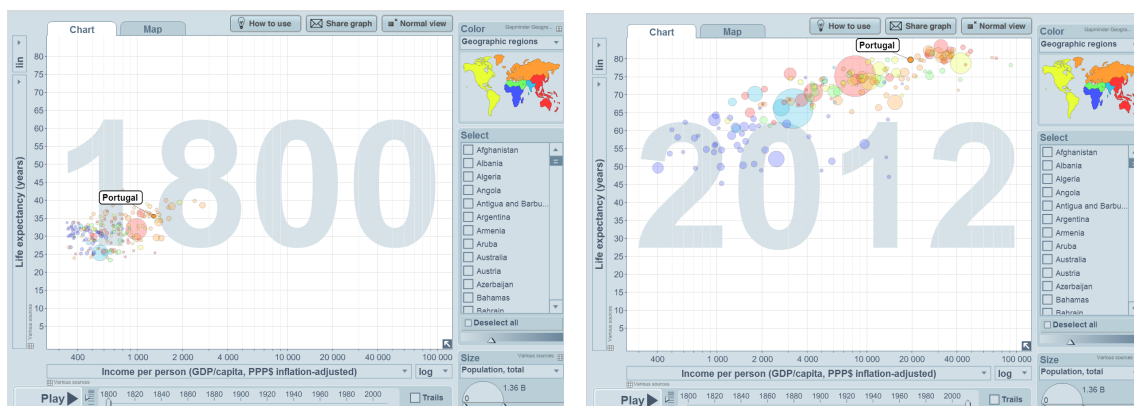
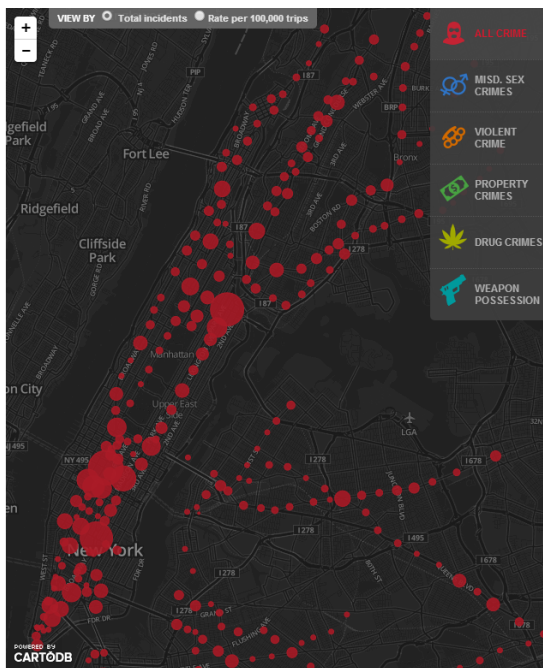
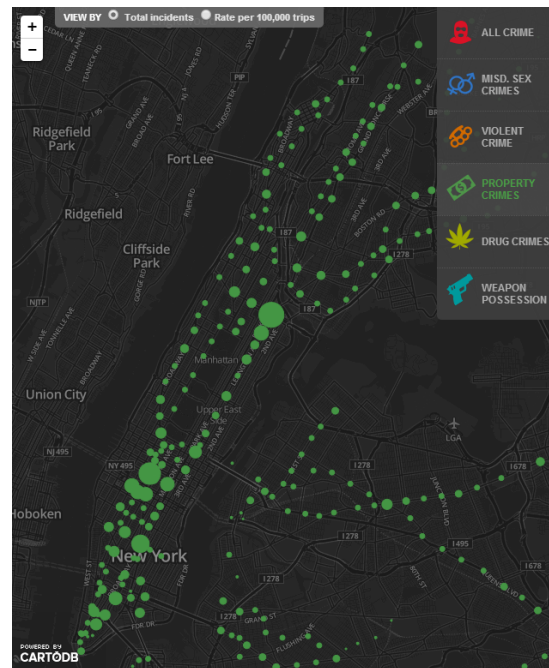


Figura 2.9: Gapminder - Esperança Média de Vida - 1800 e 2012

O CARTO [17] (com base no projeto *CartoDB* [18]) permite criar mapas personalizáveis e preenchê-los com dados. É também possível filtrar e explorar os dados inseridos para obter toda a informação disponível. O potencial desta ferramenta é grande, sendo útil para além da visualização de informação geográfica ou meteorológica, já que tudo o que tenha um impacto numa área mais extensa pode ser representado desta forma. As figuras 2.10 e 2.11 apresentam a atividade criminosa observada numa parte do metro de Nova Iorque, onde a primeira representa todos os crimes detetados e a segunda olha para apenas para roubos e atos de vandalismo.

Figura 2.10: *CARTO* - Todos os crimesFigura 2.11: *CARTO* - Roubos e vandalismo

### 2.1.2 Soluções de gestão de *Email*

O *Politemail* [19] é uma extensão do *Outlook* que oferece um conjunto de métricas e ferramentas para ajudar empresas com a gestão de *emails*. A visualização de dados não revoluciona a forma de olhar para as métricas de *emails* enviados mas é tudo feito de forma simples e integrada com o *Outlook*, sendo mais apelativo para quem não deseja ter de aprender a utilizar uma interface completamente nova.

Na figura 2.12 temos em forma de gráfico de barras a atividade relativa às mensagens de uma *newsletter*, tanto em termos de quantas pessoas receberam, abriram e responderam ao *email* como em termos de quantas mostraram interesse e o partilharam, para além de apresentar as métricas de forma numérica.

O *ClearContext* [20] é uma extensão para o *Outlook* que pretende ajudar a separar, organizar e classificar *emails*, tornando a vida dos seus utilizadores mais simples. É efetivamente uma melhoria direta ao *Outlook* base, agilizando a sua utilização. O *ClearContext*, no entanto, não adiciona novas funcionalidades para além da criação de gráficos, simplesmente reduz o esforço necessário do lado do utilizador para completar tarefas.

A figura 2.13 mostra uma visão geral da atividade da conta sem destacar nenhum elemento de informação. A capacidade de comparar os dados com todos os outros utilizadores do *ClearContext* globalmente, ou com subgrupos, apresentado na figura 2.14, fornece uma perspetiva diferente sobre a eficiência do utilizador, o que poderá ser útil para melhorar os seus métodos e hábitos, funcionalidade disponível só depois de criar uma conta nos *fóruns*

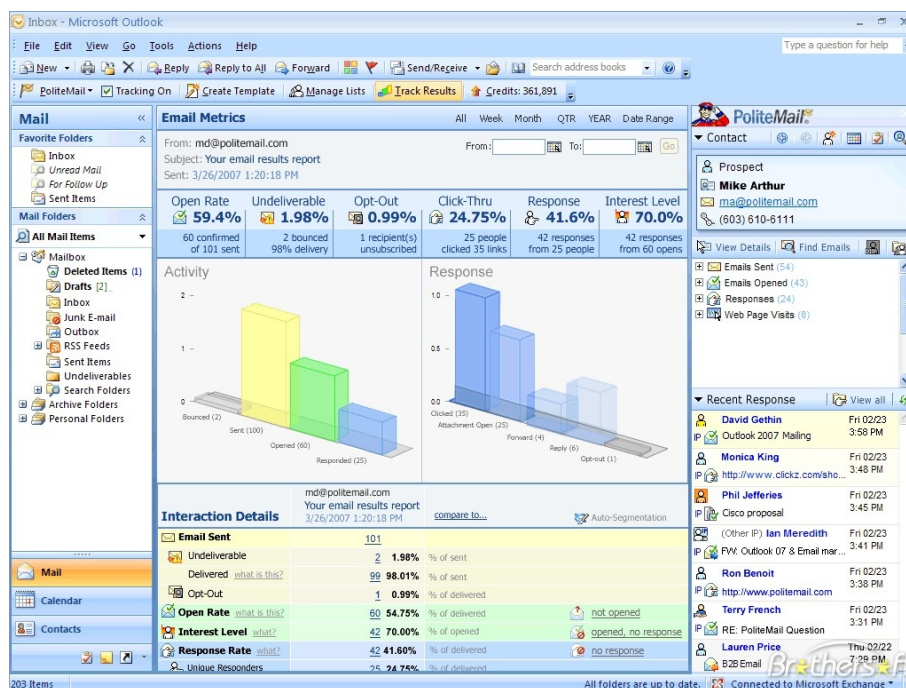


Figura 2.12: *Politemail* - Relatório com estatísticas

de apoio. O gráfico mais interessante é o que corresponde à *Effectiveness* que oferece uma visão instantânea do estado global da caixa de entrada e do desempenho do utilizador, baseado no tempo de resposta, quantidade de *emails* presentes na caixa de entrada e número de respostas enviadas.



Figura 2.13: *ClearContext* - Visão Geral

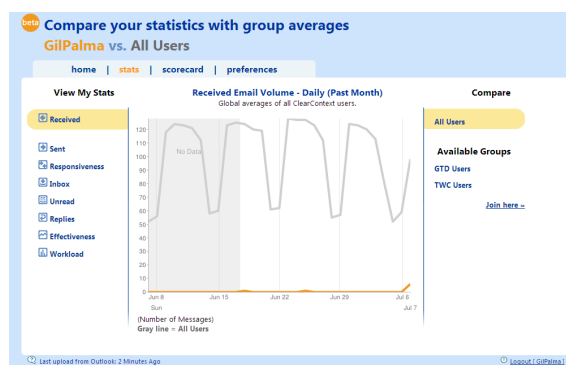


Figura 2.14: *ClearContext* - Eficiência do utilizador

O *Bells & Whistles* [21] é outra extensão para o *Outlook* que simplifica as funcionalidades existentes automatizando a inserção e adequação de normas, assinaturas, avisos legais e identificadores no “Assunto” de cada *email* para poupar tempo aos utilizadores na composição de mensagens. As métricas recolhidas são as mais comuns, não oferecendo inovações

nem nos dados nem na forma de os apresentar. No entanto, a possibilidade de guardar e partilhar os resultados permite criar um histórico detalhado com a informação disponível ou comparar os dados com outros utilizadores em particular. Para visualizar dados específicos, um utilizador pode seleccionar o intervalo de tempo e as métricas a incluir, dentro das possibilidades visíveis na figura 2.15. A figura 2.16 mostra um dos gráficos pedidos, o tráfego diário de *emails*, onde a linha vermelha representa os *emails* recebidos e a linha azul corresponde aos enviados.

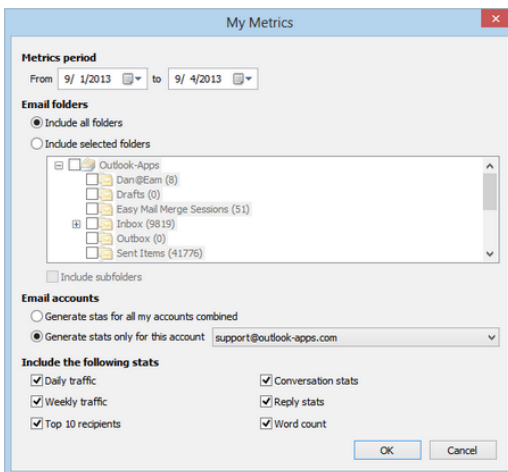


Figura 2.15: *Bells & Whistles* - Seleção de métricas em intervalo

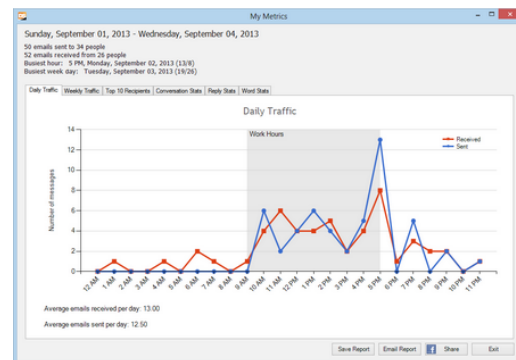


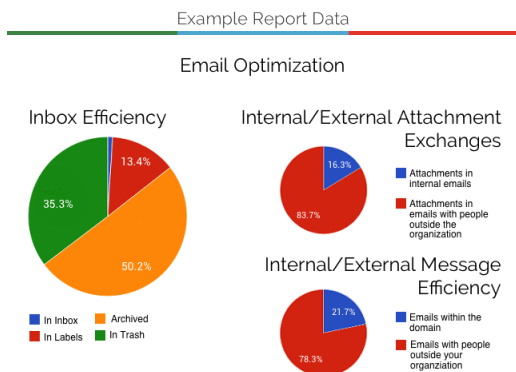
Figura 2.16: *Bells & Whistles* - Tráfego diário

O *Gmail Meter* [22] é uma aplicação instalada diretamente na conta *Google Drive* [23] de um utilizador. Deste modo, esta aplicação consegue aceder diretamente a todos os dados da conta de *Gmail*, executada a partir da própria *Drive*. Uma das suas principais vantagens é ter as métricas recolhidas e os gráficos relacionados disponíveis em qualquer dispositivo que tenha acesso à *Drive*, já que este pode ser utilizado através de aplicações dedicadas em dispositivos móveis ou fixos ou até a partir de navegadores de *Internet* como o *Google Chrome* e o *Microsoft Internet Explorer*. A *ShuttleCloud* [24], a empresa responsável pela criação e manutenção desta solução, não intervém durante a análise dos dados, sendo tudo calculado a partir da aplicação instalada na *Google Drive*.

As figuras 2.17 e 2.18 são exemplos de gráficos fornecidos pela *ShuttleCloud* e mostram a informação que é possível visualizar com esta solução; a figura 2.17 mostra a localização dos *emails* nas diversas pastas da conta e a proporção de *emails* trocados que contêm anexos, tanto para dentro da própria empresa como para outras; a figura 2.18 mostra o volume de *emails* trocados ao longo do dia, da semana e do mês.

O *Xobni* [3] (agora pertencente à *Yahoo* [25][2]) oferece as métricas mais comuns, semelhante às soluções anteriores, mas visualização de dados não é o ponto central da aplicação, apesar de esta estar aliada à visualização dos dados. Sendo uma extensão para o *Outlook*, esta pode aceder a meta dados de *emails* recebidos e conhecer pessoas mesmo que não exista um contacto formal guardado no cliente de *email*. A sua função chave beneficia





h

Figura 2.17: Gmail Meter - Otimização

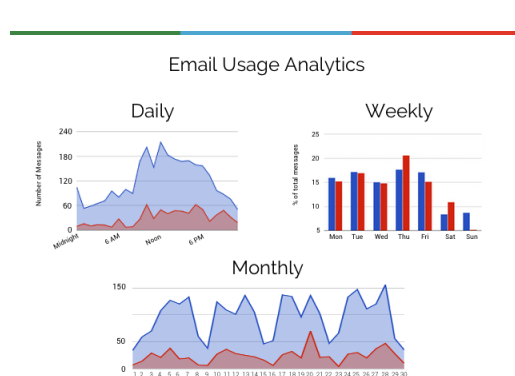


Figura 2.18: Gmail Meter - Volume de mensagens trocadas

deste acesso a meta dados e permite a pesquisa rápida de contactos, mesmo que não se encontrem na Lista de Contactos do Outlook. O aspeto visual desta aplicação permite conhecer detalhadamente a origem de cada email e ver nas medidas apresentadas os seus autores, independentemente de já terem sido formalmente registados ou não. A figura 2.19 apresenta os contactos com quem foram trocados mais emails e a quantidade de emails enviados e recebidos nas últimas 24 horas.

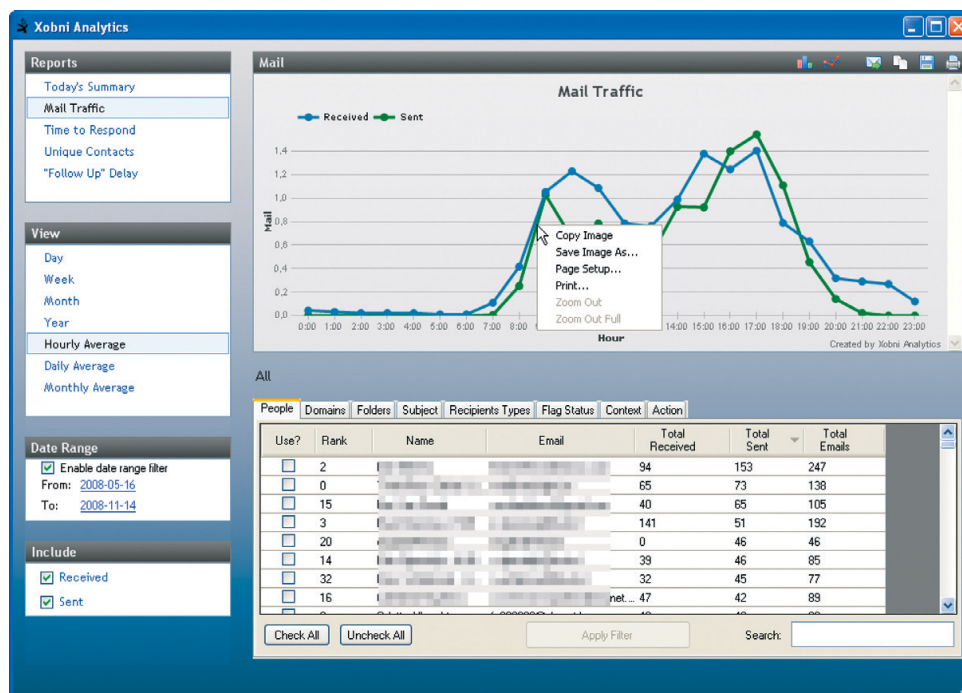


Figura 2.19: Xobni - Tráfego de mensagens

### 2.1.3 Soluções de *Email Marketing*

O *GetResponse* [26] oferece a capacidade de, não só armazenar e apresentar as métricas dos *emails* enviados, mas também sugerir e/ou aplicar alterações à forma atual de enviar mensagens para maximizar produtividade e eficácia. É dado como exemplo na página do produto o envio automático de *emails* a diferentes horas para diferentes grupos de subscritores. Diferentes indivíduos estão mais recetivos a certas alturas do dia e a solução oferecida permite programar envios personalizados de forma a maximizar receções positivas, ou seja, instâncias em que um recetor abre, lê e interage com a mensagem que recebeu. Outras funcionalidades do *GetResponse* incluem a capacidade de mostrar resultados globais lado a lado (presente na figura 2.20) e monitorizar os dispositivos mais usados pelos clientes e as redes sociais onde os *emails* são partilhados mais frequentemente com outros. Isto, combinado com a liberdade de pedir a exploração em profundidade dos dados, facilita a tomada de decisões e garante que os *emails* enviados estão otimizados para cada cliente. A visualização de dados apesar de usar gráficos tradicionais é mais intuitiva do que outras.

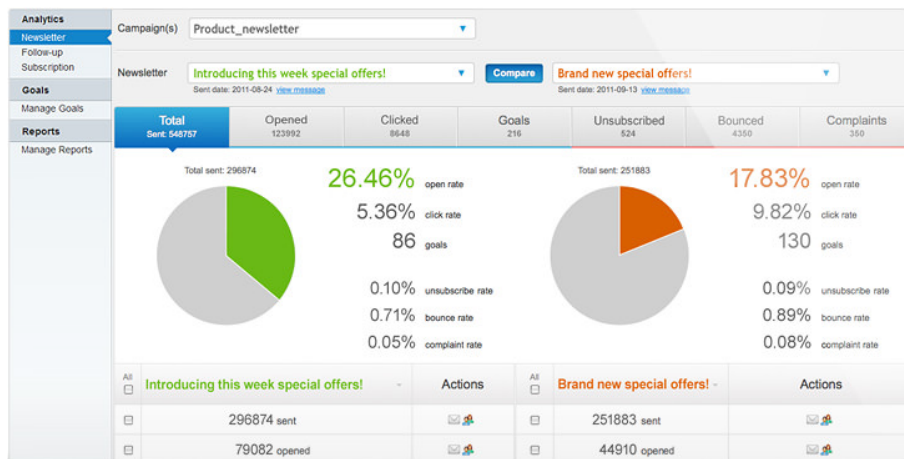


Figura 2.20: *GetResponse* - Estatísticas de uma *Newsletter*

Oferecendo análise de redes sociais, *emails* e mais, o *SumAll* [27] funciona como um aglomerador de dados de muitas fontes. Atualizados em tempo real, os gráficos apresentados nesta ferramenta permitem ver dados simples ou cruzados, até mesmo entre serviços diferentes, tal como a relação entre emails abertos e transferências feitas com *PayPal* a cada altura do dia. O *SumAll* agrega e processa dados de outras ferramentas de análise de métricas, tal como o *Google Analytics*, e de redes sociais, como o *Facebook* e *Twitter*. O objetivo é dar toda a informação possível aos seus utilizadores, permitindo-lhes escolher o melhor caminho a seguir em cada situação sem nunca perder a visão na globalidade, ainda com a possibilidade de unir dados semelhantes de fontes diferentes. Tudo isto está disponível a partir da aplicação *web* em qualquer computador, enquanto a versão para dispositivos móveis só permite analisar os dados de forma superficial. A figura 2.21 mostra os dados recolhidos do *Google Analytics*, do *Facebook Ads* e do *BigCommerce*.

Semelhante ao *SumAll*, o *Silverpop Engage*, produto desenvolvido pela *Silverpop* [28], une

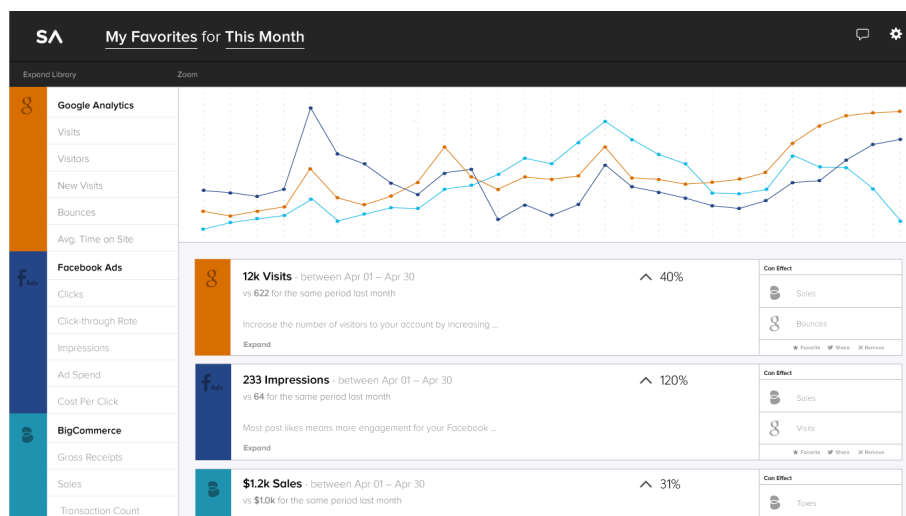


Figura 2.21: *SumAll* - Exemplo de aglomeração de dados

dados de redes sociais e outras fontes, permitindo chegar a informação que não está diretamente disponível. Em particular, o *Silverpop Email Insights* [29], uma extensão ao *Silverpop Engage*, reúne métricas de *emails* e permite visualizar os dados recolhidos com os tipos de gráficos mais comuns. Esta solução conta também com mais 23 extensões opcionais [30], tais como o *Adobe Analytics*, *Silverpop Revenue Analytics*, *IBM Core Metrics*, entre outras, capazes de acrescentar imensa informação apoiada por gráficos especializados, melhorando drasticamente a forma de visualizar os dados de uma campanha de marketing. A figura 2.22 mostra os dados de uma campanha do *Adobe Analytics*, incluindo as compras feitas pelo público-alvo, com o objetivo de revelar onde há as maiores barreiras entre os recetores e a finalização das suas compras.

O *MailChimp* [1] oferece uma visão mais detalhada de cada recipiente do que as soluções anteriores. Com esta aplicação, torna-se possível para uma organização conhecer o histórico, condições e preferências de cada cliente subscrito e até associar notas e respostas automáticas a pessoas em específico. Estas funcionalidades permitem que os gestores de *marketing* conheçam e personalizem a experiência de cada cliente, especialmente dos seguidores mais dedicados que se destacam da maioria. Na figura 2.23 são visíveis as métricas gerais de uma campanha de *marketing*.

A solução de *email marketing* da *Zoho*, *Zoho Campaigns* [31], em primeira análise parece uma solução como muitas outras, sem nada que a faça destacar. Mas quando aliada a outras ferramentas da mesma empresa, ou extensões de associados, a *Zoho Campaigns* torna-se bastante robusta. Em particular, a *Zoho Reports* [32] oferece a capacidade de criar painéis personalizáveis com vários tipos de gráficos para visualizar toda a informação necessária num só lugar. A figura 2.24 ilustra alguns dos tipos de gráficos disponíveis, todos mostrando os lucros obtidos em relação ao tempo, cliente, região ou produto. Os gráficos de barras, funil e setores mostram os lucros totais por ano. O gráfico de áreas dispõe os dados por região e no espaço de três anos. O gráfico de barras agrupado e o



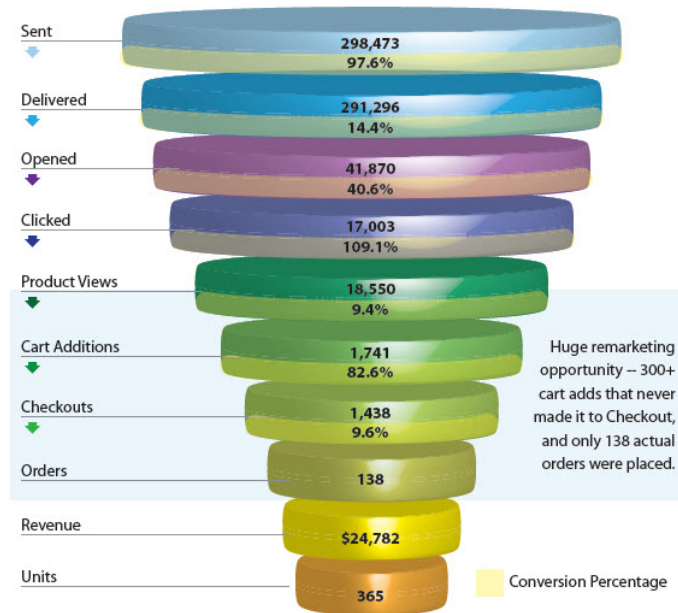


Figura 2.22: Silverpop Engage - Campanha de marketing Adobe Analytics

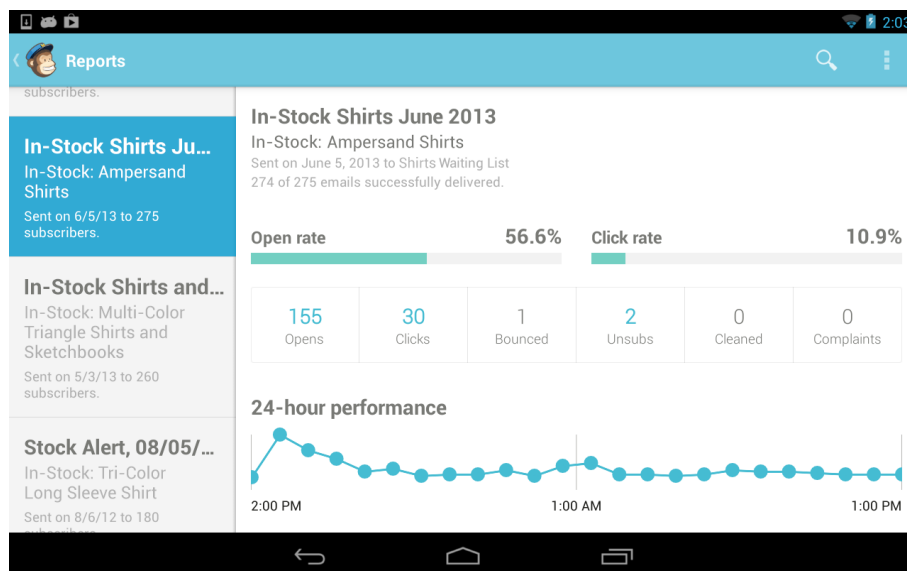


Figura 2.23: MailChimp - Métricas da campanha

gráfico de teia olham para a relação entre produtos e ano. O gráfico de dispersão apresenta os dados dispostos por produto e cliente.

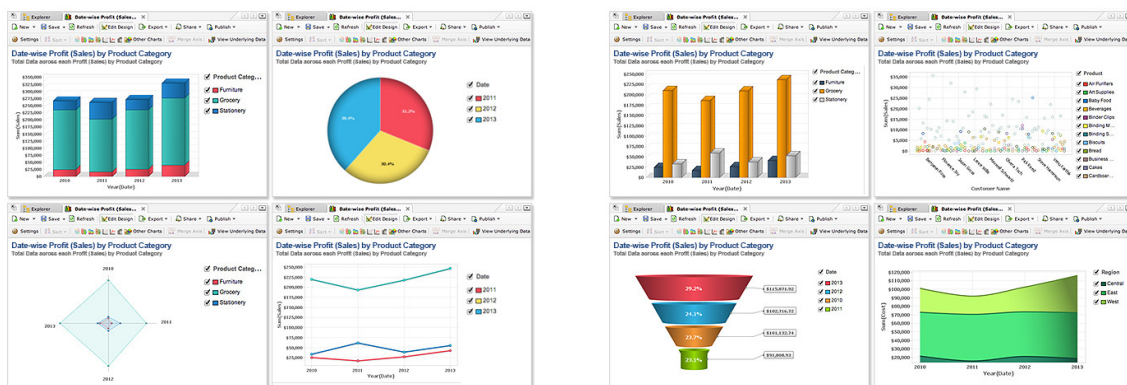


Figura 2.24: Zoho Campaigns - Zoho Charts

O *SendinBlue* [33] destaca-se das outras porque é uma das muito poucas que utiliza mapas de temperatura para mostrar o número de *clicks* em cada hiperligação de um determinado email, exibindo instantaneamente que zonas de um *email* geraram mais interesse. Essa forma de visualizar os dados não é nova mas é ainda pouco usada por serviços de *email marketing* e a sua utilidade neste meio é imediatamente aparente. Esta solução também oferece as funcionalidades mais comuns de um serviço de *email marketing*, sendo comparável a todos os outros mencionados anteriormente. A figura 2.25 exemplifica a aparência de um destes mapas de temperatura alternativos cujo comportamento é idêntico ao de um mapa de temperatura habitual. Apesar de ser funcionalmente equivalente, este estilo perde alguma facilidade de visualização ao escolher não obstruir o conteúdo de um *email* com cores e realces mais vívidos.

## 2.2 Potenciais ambientes alvo e respetivas soluções gráficas

A utilização de dispositivos táteis tem-se expandido a uma velocidade estonteante nos últimos anos. Entre *smartphones*, *tablets* e outros dispositivos semelhantes, tal como o *Microsoft Surface* [34] e o *iPad* [35], estes dispositivos têm-se inserido rapidamente nas vidas das pessoas, tanto para lazer como para trabalho. Mas, apesar de os dispositivos táteis terem começado a ganhar popularidade no início dos anos 2000, a maioria destes eram *smartphones* utilizados em ambientes profissionais. Foi em 2007 que a *Apple* [36] lançou o primeiro *iPhone* e começou a alcançar o público em geral, tornando a utilização de dispositivos táteis cada vez mais ubíqua. A partir daí muitos competidores surgiram e todo o mercado à volta deste tipo de tecnologia sofreu um crescimento explosivo, sem mostrar sinais de abrandar ainda hoje em dia.



Figura 2.25: *SendinBlue* - Gráfico de temperatura de *clicks* em *emails*

### 2.2.1 iOS

O *iOS* [37] é o nome dado ao conjunto de versões do sistema operativo utilizado nas diversas iterações de todos os dispositivos móveis da *Apple* [36], concretamente no *iPhone*, *iPad*, *iPad Mini*, *iPod Touch* e para todas as gerações da *Apple TV* a partir da segunda. Todos os produtos móveis da *Apple* utilizam simultaneamente a mesma versão do *iOS*, das quais já existem dez estáveis e uma próxima em desenvolvimento, permitindo uma comunicação excepcional entre todos eles e facilitando o desenvolvimento de aplicações, já que são feitas considerando principalmente a versão mais atual. Além disso, utilizadores de *iOS* habituados a um dispositivo móvel da *Apple* conseguem facilmente adaptar-se a outro, mesmo que nunca o tenham utilizado, devido às interfaces serem muito semelhantes em termos de aparência. As únicas diferenças superficiais estão no “*look and feel*” mas estas são introduzidas propositadamente para maximizar o conforto de utilização de cada tipo de dispositivo já que existem diferenças no tamanho do ecrã.

Além disso, os produtos da *Apple* têm acesso à *App Store*, um repositório de *software* específico para o *iOS* contendo mais de 1.2 milhões de aplicações. Isto permite a qualquer utilizador escolher e descarregar as aplicações que mais lhe agradam sem ter de se preocupar se cada uma é ou não compatível com o seu dispositivo ou quais os passos de instalação necessários. Na figura 2.26, a página inicial da *App Store* na versão 7 do *iOS*, apresentada no lado direito, realça algumas aplicações notáveis. O lado esquerdo da figura mostra apenas uma lista de aplicações gratuitas. A figura 2.27 também mostra a página inicial

da *App Store* mas vista a partir de um *iPad* com *iOS 7*. Como o *iPad* tem um ecrã substancialmente melhor e maior do que o ecrã do *iPhone*, a *App Store* e todas as outras aplicações disponíveis tiram proveito deste facto e oferecem uma experiência especializada de modo a maximizar conforto enquanto dão opções ao utilizador. Estes e outros fatores elevaram os dispositivos da *Apple* a um nível de sucesso estrondoso, sendo rapidamente adotados por um grande número de pessoas e estabelecidos como o padrão de qualidade para dispositivos táteis.



Figura 2.26: *iOS* - Destaques da *Appstore* em *iPhone*



Figura 2.27: *iOS* - *Appstore* em *iPad*

O *iOS* é uma ferramenta extremamente polida e robusta, mostrando as vantagens que um sistema unificado oferece. Simultaneamente, ao oferecer este nível de qualidade, o *iOS* é também um sistema fechado com uma série de restrições. Não é permitido qualquer tipo de alterações ao sistema operativo e todas as transferências de software têm de ser feitas a partir da *App Store*, impossibilitando a instalação de software que não tenha sido analisado e aceite pela *Apple*. O utilizador também não pode personalizar a disposição dos elementos de menus ou do ambiente principal, sendo forçado a utilizar a disposição de elementos que vem por omissão. Tudo isto, considerando ainda os preços elevados, afasta muitos potenciais clientes, levando a uma base de utilizadores dedicada mas relativamente pequena quando comparada com o mercado de todos os dispositivos deste tipo.

Com o intuito de descobrir o potencial de criação de gráficos na plataforma *iOS*, estes próximos parágrafos vão dispor os resultados das análises feitas às ferramentas existentes de criação de gráficos nesta plataforma e disponibilizar um panorama das capacidades de cada ferramenta ou biblioteca. Concretamente, vão ser analisadas principalmente as bibliotecas *Core Plot* [38] e *Shinobi Controls* [39]. Existem algumas outras, tais como o *JBChart-View* [40] e *iOS:Chart* [41] que apesar de serem geralmente inferiores às duas bibliotecas mencionadas anteriormente têm características fortes que também serão analisadas.

## Core Plot

Disponível no *GitHub* [42], a biblioteca *open source Core Plot* [38] oferece uma quantidade substancial de diferentes tipos de gráficos, um alto nível de personalização e documentação substancial. A qualidade desta biblioteca é equiparável a soluções pagas apesar de não ter o financiamento que essas têm. Permite alguma interação com os gráficos, tal como pressionar, aproximar ou movimentar determinados elementos para os realçar ou obter mais dados relevantes. A figura 2.28 mostra um gráfico de setores simples. Ao pressionar num dos segmentos do gráfico, este é aleatoriamente alterado para ter valores diferentes. A figura 2.29 tem um painel com vários tipos de gráficos apresentados simultaneamente, uma função muito útil e poderosa por si só.

Como um pequeno extra, esta biblioteca providencia conjuntos de cores pré-definidos que permitem alterar as cores de qualquer gráfico em qualquer momento e ainda a possibilidade de acrescentar conjuntos adicionais.

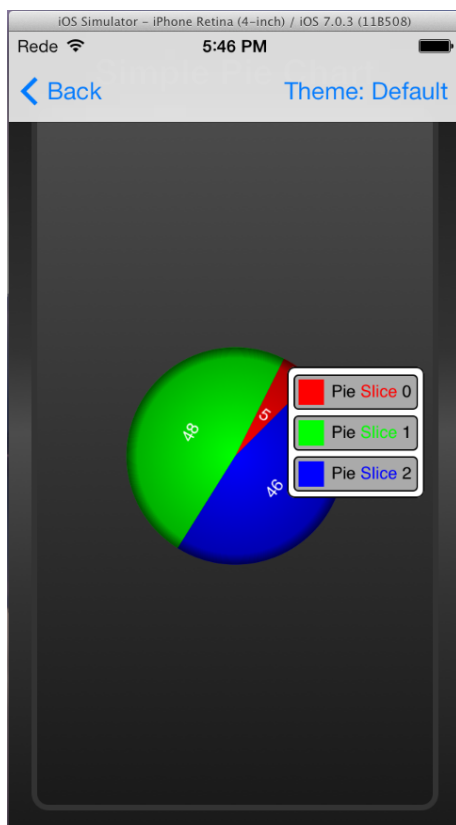


Figura 2.28: *Core Plot* - Gráfico de setores inicial

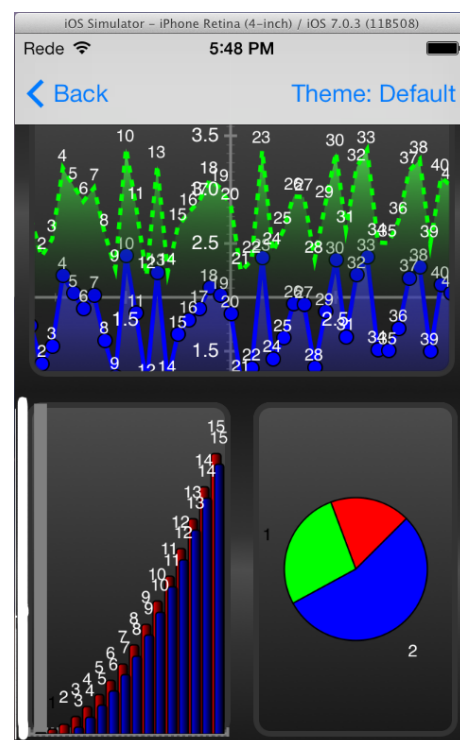


Figura 2.29: *Core Plot* - Painel *Co-replot*

## Shinobi Controls

Tendo em conta as funcionalidades necessárias para visualização profissional de dados, o *Shinobi Controls* [39] é a biblioteca de gráficos mais polida e poderosa atualmente para *iOS*. No entanto, esta biblioteca não é dedicada a *iOS*, estando disponível também para *Android*, nativamente nas duas plataformas. Esta oferece uma grande quantidade de diferentes tipos de gráficos, a possibilidade de manipulação e exploração dos dados em tempo real com representação de dados veloz e uma documentação extensa. As animações disponíveis são rápidas e fluidas, mais do que com o *Core Plot*. Sempre que um segmento de um gráfico de secções é pressionado, como o da figura 2.30, o gráfico reposiciona-se de forma a colocar o segmento em questão orientado para cima. Visualmente, esta solução oferece as melhores ferramentas, um facto especialmente aparente quando são visualizados dados em tempo real. Uma imagem estática não o permite ver mas a fluidez da atualização é notável e exemplifica as capacidades da biblioteca.

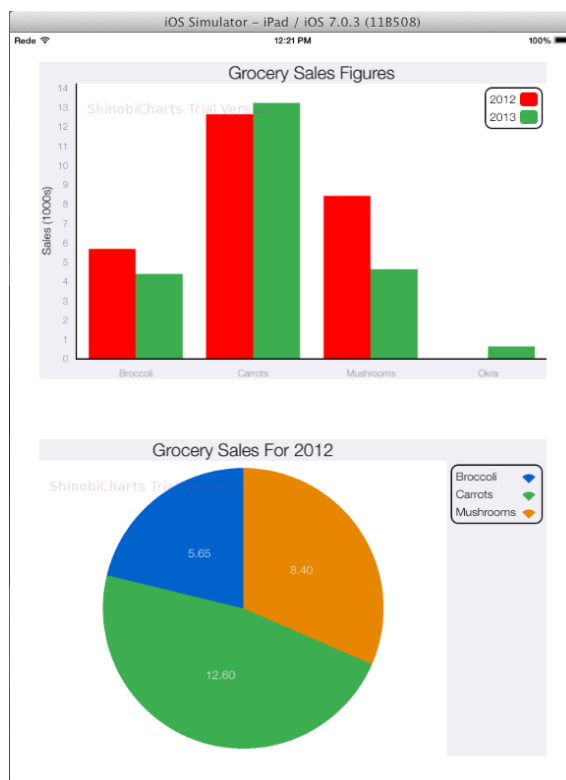


Figura 2.30: *Shinobi Controls* - Combinação de gráficos 2012

## JBChartView

Criada pela *Jawbone*, esta biblioteca [40] tem um aspeto notável. Muito polida e personalizável, a *JBChartView* é impressionante visualmente mas peca por só conseguir criar gráficos de barras e linhas. No entanto, com um elevado nível de personalização e um

arsenal considerável de animações, a *JBChartView* é uma biblioteca a ter em mente. As figuras 2.31 e 2.32 mostram visões mensais e anuais de precipitação numa dada área.

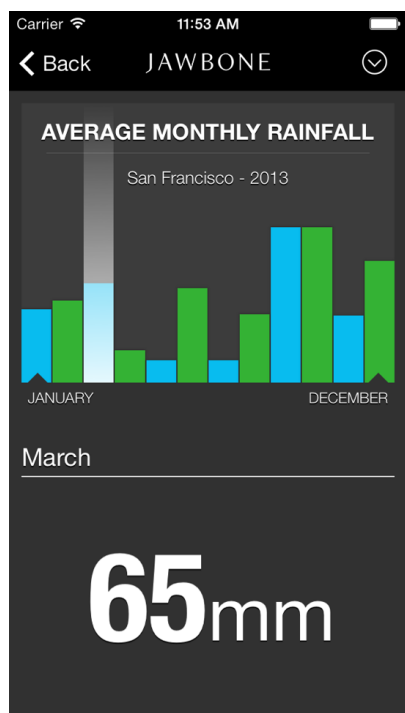


Figura 2.31: *JBChartView* - Precipitação mensal



Figura 2.32: *JBChartView* - Precipitação anual

### iOS:Chart

Pertencente à *ThreeDGraphics* [43], esta biblioteca [41] destaca-se por ser capaz de representar dados em 2D e em 3D, como visível nas figuras 2.33 e 2.34, e por permitir manipular os gráficos no espaço, aplicável tanto em *iOS* como em *macOS*, o sistema operativo da *Apple* para computadores de mesa. Esta biblioteca foi criada a partir da *PGSDK* [44], uma biblioteca de gráficos feita para sistemas operativos diferentes como o *Windows* e *Linux*. É semelhante a outras soluções, oferecendo interatividade, personalização e a capacidade de explorar os dados mas não tem o nível de personalização do *Core Plot*, devido a não ser *open source*, ou nem é tão eficiente ou polido como o *Shinobi Controls*.

### 2.2.2 Android

Essencialmente no lado oposto do espectro de sistemas operativos móveis, relativamente à filosofia do *iOS*, está o *Android* [45]. Um sistema operativo fundado por Andy Rubin e apoiado pela *Google*, o *Android* foi desenhado para oferecer liberdade de escolha e personalização ao nível do *software*, mas também ao nível do *hardware*, abdicando de alguma



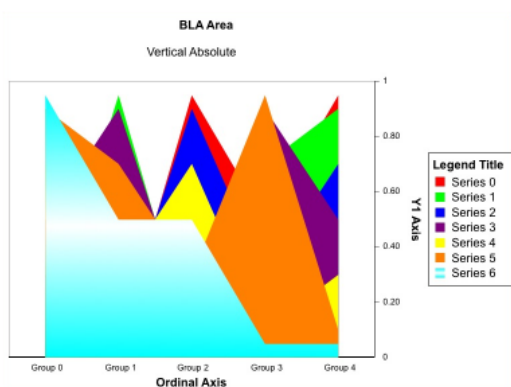


Figura 2.33: *iOS: Chart* - Gráficos de área empilhados

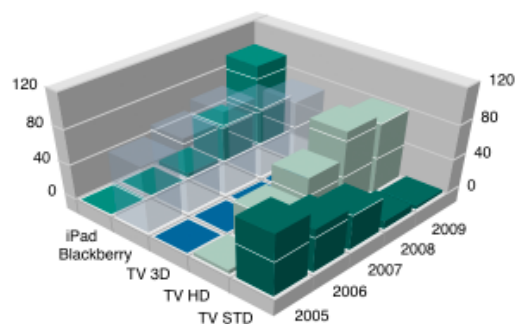


Figura 2.34: *iOS: Chart* - Gráfico 3D

da conveniência e utilidade que viria com a uniformidade. Ao contrário das dez versões do *iOS*, o *Android* já conta com catorze versões principais diferentes, com uma adicional para dispositivos *wearable*, e não existe nenhum controlo sobre qual está instalada em cada dispositivo. Isto introduz uma barreira, potencialmente pequena para *developers*, que não existe no *iOS*, já que aplicações têm de ser desenvolvidas com várias versões de *Android* e vários tamanhos de ecrã em mente. A figura 2.35 mostra as diferenças de aspeto do ecrã base entre a primeira versão, a versão *Jelly Bean* 4.1 e a *KitKat* 4.4, respetivamente. As diferenças entre as versões mais recentes são poucas mas fáceis de identificar. No entanto, é imediatamente aparente como as alterações foram muitas desde as fases iniciais do sistema operativo que tinham um aspeto significativamente diferente.



Figura 2.35: *Android* - Diferenças entre a primeira versão (esquerda), a versão *Jelly Bean* 4.1 (centro) e a *KitKat* 4.4 (direita)



No entanto, as desvantagens provenientes do “caos” introduzido pela ausência de controlo são equilibradas pelo facto de existirem diferentes experiências disponíveis, sendo possível encontrar um dispositivo *Android* para qualquer pessoa, já que é possível escolher combinações de *hardware* e versão do sistema operativo, priorizando a acessibilidade económica ou a qualidade tecnológica. A grande variedade de opções facilita a adoção de dispositivos *Android* para muitas pessoas já que podem adquirir os dispositivos que melhor se adaptam aos seus limites e necessidades. Há que notar que existem sempre certas semelhanças entre versões, garantindo que um utilizador não se sente perdido quando é deparado com uma versão diferente daquela a que está acostumado.

Graças aos seus pontos fortes, o *Android*, marcado a azul na figura 2.36, domina a competição em termos de unidades vendidas no total nos últimos anos [46]. O facto de existirem tantas alternativas a diferentes preços faz com que seja muito mais acessível adquirir um dispositivo *Android* do que um *iOS*.

A distribuição de versões por dispositivos dos últimos seis anos está apresentada na figura 2.37 [47]. Como se pode ver, na data registada mais recentemente nesse gráfico, aproximadamente 10% dos utilizadores de dispositivos *Android* utilizam a versão *Marshmallow* 6.0 e algumas pessoas ainda utilizam versões anteriores até à *Gingerbread* 2.3. Na página oficial de informação da distribuição de versões [48] temos a distribuição atualizada periodicamente em períodos de sete dias onde podemos ver quanto mudaram as proporções desde Maio de 2016. Durante a escrita deste documento, a utilização da versão *Marshmallow* tem subido gradualmente e atingiu 31.2% de todos os dispositivos *Android* presentemente.

**Global smartphone sales to end users from 1st quarter 2009 to 4th quarter 2013, by operating system (in million units)**

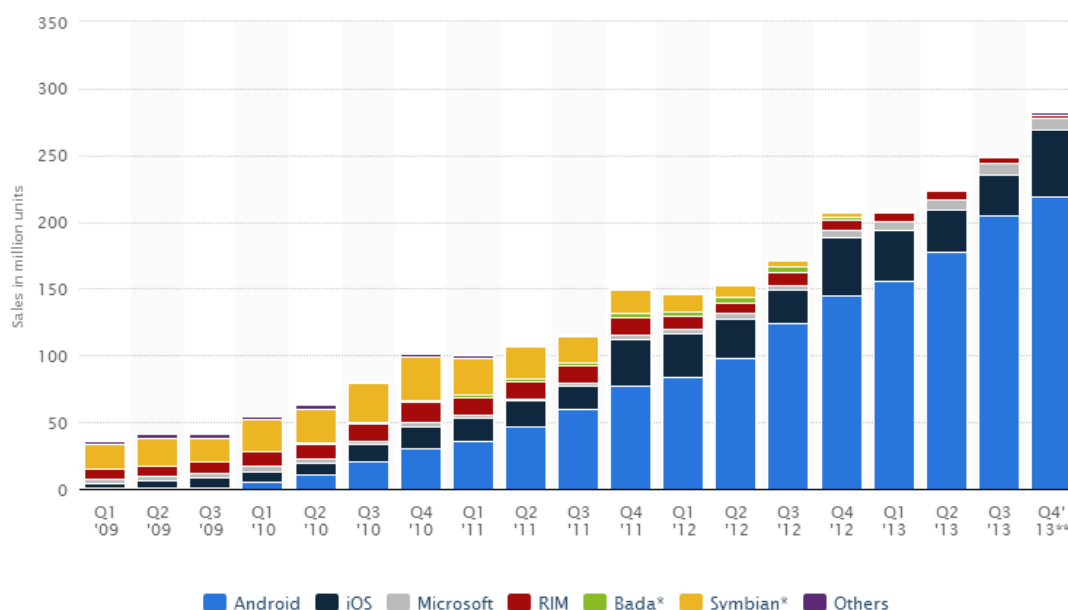


Figura 2.36: Vendas de *smartphones*

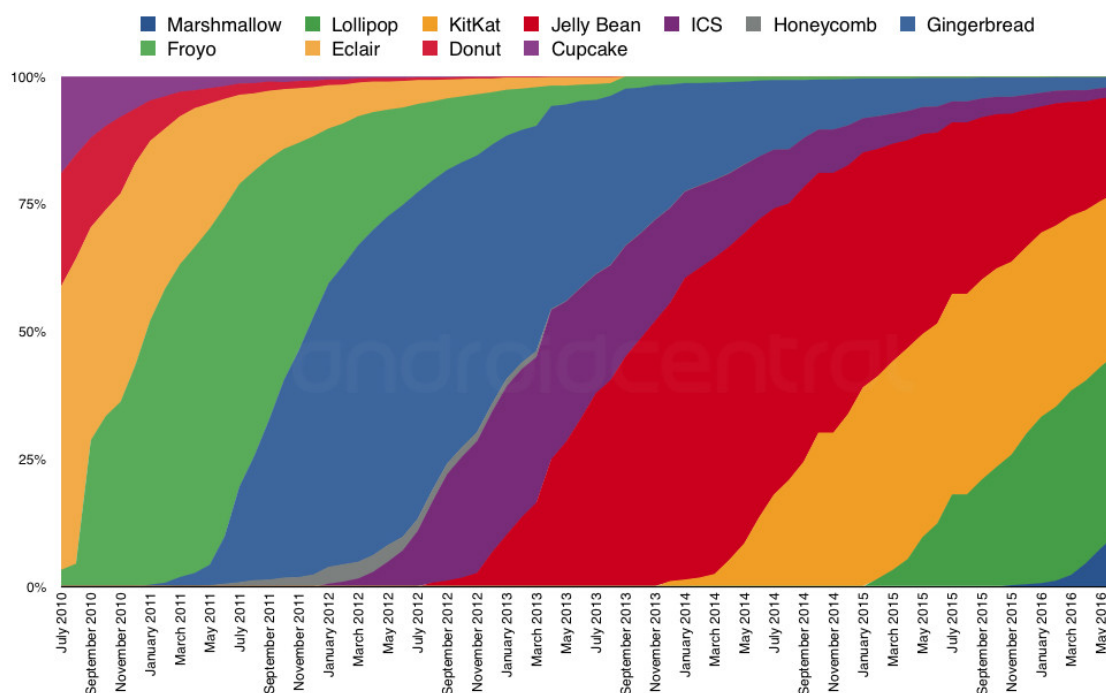


Figura 2.37: *Android* - Distribuição de versões em Maio 2016

A abertura do sistema operativo também permite que este seja usado em mais do que *smartphones* e *tablets*, podendo ser usado em muitíssimos dispositivos, tais como consolas de videojogos, câmaras digitais e alguns aparelhos desenvolvidos pela própria empresa *Android Inc.* tais como carros, televisões e *smartwatches*. Existe também o equivalente à *App Store* para dispositivos *Android* chamado *Google Play*. No entanto, este é apenas uma alternativa e existem vários repositórios disponíveis de onde é possível descarregar aplicações [49]. Por exemplo, o *F-Droid* [50] é um catálogo de aplicações gratuitas e *open source*, sendo ele próprio livre de custos mas financiado por doações. A *Amazon Underground App for Android* [51] é utilizada por dispositivos da linha de *tablets Kindle* e é gerida pela *Amazon*, garantindo que as aplicações disponíveis desse modo estão polidas e adaptadas para os dispositivos em questão. Nesta secção vão ser observadas as bibliotecas *AChartEngine*, *MPAndroidChart*, *Skia*, *Androidplot* e *GraphView*.

### **AChartEngine**

A biblioteca *AChartEngine* [52] conta com uma *API* robusta, documentação extensa e permite criar pelo menos dez tipos de gráficos base, entre gráficos de barras, gráficos de secções e gráficos de velocímetro, e ainda cruzá-los para ter visualizações personalizáveis. Dados podem ser inseridos de várias formas, sendo estabelecidos anteriormente no código ou inseridos em tempo real, e todos os elementos podem ser alterados e tornados interativos. Os gráficos gerados tendem a ser esteticamente simples e pouco polidos, não sendo muito

apelativos quando se pretende impressionar potenciais clientes com um novo produto. A figura 2.38 mostra um exemplo de um gráfico simples criado com esta biblioteca. Como se pode ver, existe algum *aliasing* e as cores são simples. No entanto, com as definições certas é possível desenhar gráficos mais cativantes, como ilustrado na figura 2.39. Mas, apesar da simplicidade gráfica, a sua versatilidade tem valor e é suficiente para tornar o *AChartEngine* num candidato para a tecnologia escolhida para o produto.

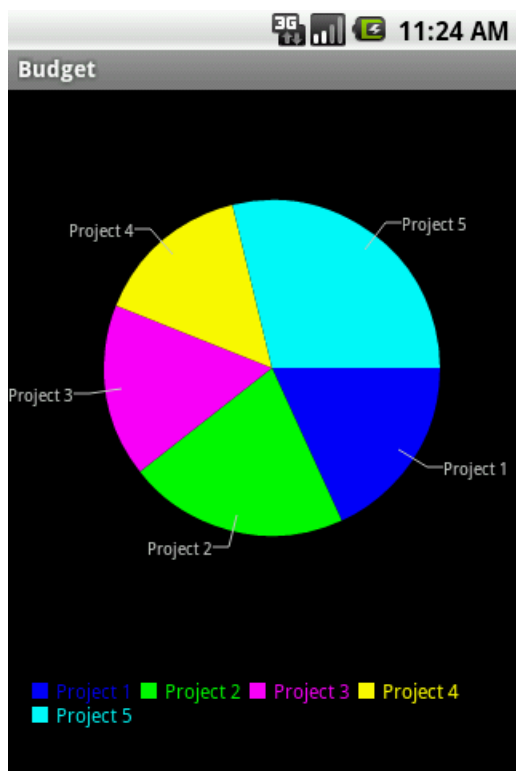


Figura 2.38: *AChartEngine* - Exemplo pouco polido

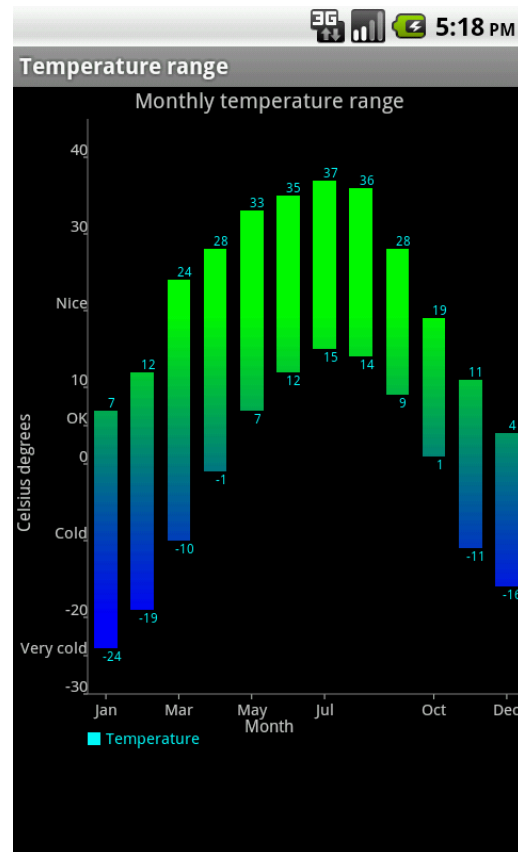


Figura 2.39: *AChartEngine* - Exemplo mais polido

## MPAndroidChart

Com algumas semelhanças à *AChartEngine*, a *MPAndroidChart* [53][54] permite desenhar gráficos de tipos pré-definidos. Mais polida e completa do que a biblioteca anterior, os gráficos contam com cores vibrantes e a possibilidade de terem estilos personalizáveis, tanto ao nível do tipo de gráfico como ao nível do estilo dos elementos individuais. As figuras 2.40 e 2.41 mostram dois estilos de gráficos de linhas.

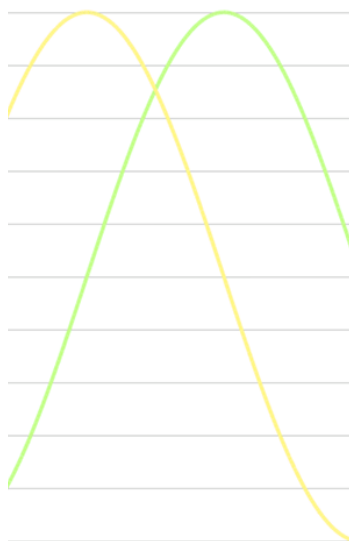


Figura 2.40: *MPAndroidChart* - Gráfico de linhas simples

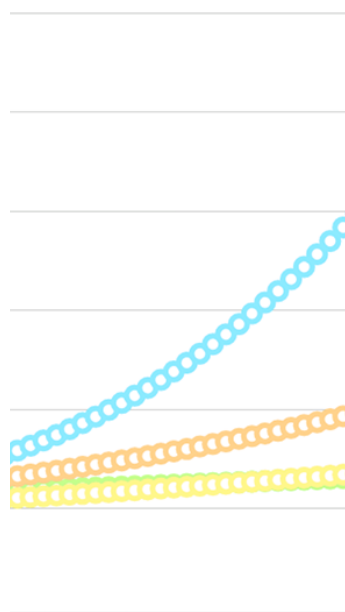


Figura 2.41: *MPAndroidChart* - Gráfico de linhas estilizado

## Skia

*Skia* [55] é uma biblioteca *open source*, da autoria da *Google* [56], que permite desenhar gráficos 2D de qualquer tipo através de combinações de formas, linhas e transformações. Sendo uma biblioteca com muito maior amplitude de capacidades do que outras, permitindo mais do que a criação de apresentações visuais de dados numéricos, a criação de gráficos não é tão simples inicialmente mas o potencial é enorme. A figura 2.42 mostra variações da mesma apresentação com diferentes efeitos.

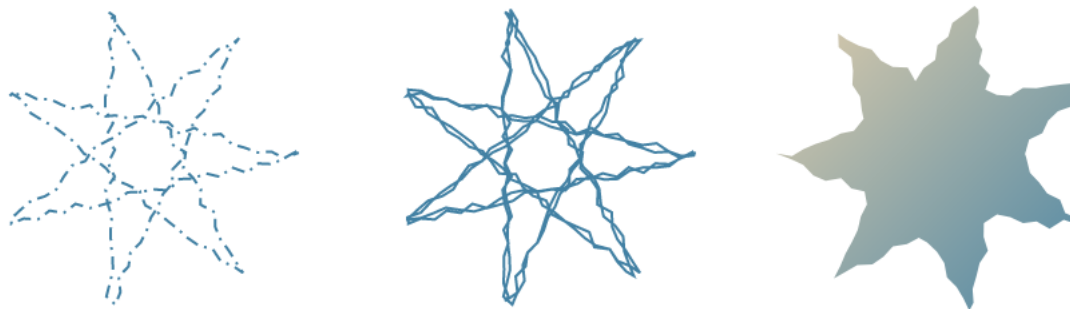


Figura 2.42: *Skia* - *Composed Path*, *Sum Path* e *Shaders*

## Androidplot

Esta biblioteca [57] também é dedicada a gráficos numéricos, como a *AChartEngine* [52]. Comparado com as duas bibliotecas anteriores, esta oferece a simplicidade de uso da primeira com a qualidade visual da segunda, sendo uma boa escolha caso o potencial máximo da Skia [55] não seja interessante em última análise. As figuras 2.43, 2.44 e 2.45 ilustram algumas das possibilidades. As capacidades da biblioteca podem ser testadas na aplicação *Android*, chamada *Androidplot Demos* [58], disponibilizada na *Google Play* [59].

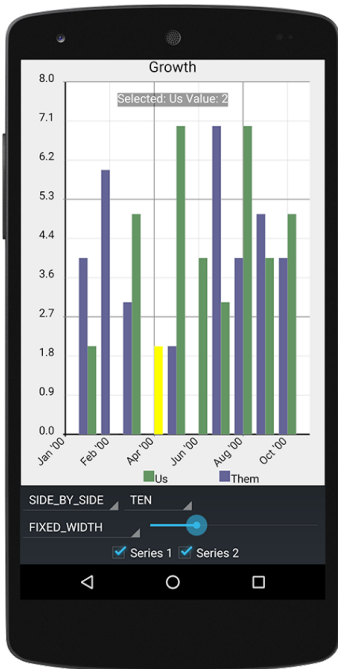


Figura 2.43: *Androidplot* - Gráfico de barras

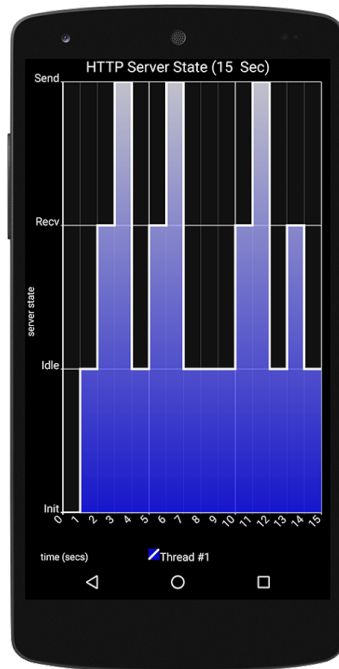


Figura 2.44: *Androidplot* - Gráfico de barras com gradiente

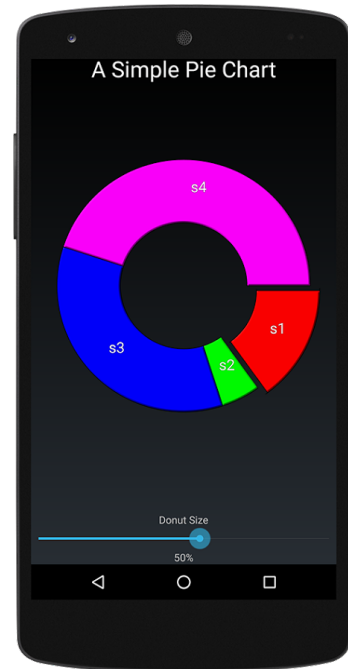


Figura 2.45: *Androidplot* - Gráfico de secções

## GraphView

*GraphView* [60] é mais uma biblioteca de gráficos como a *AChartEngine* [52] e a *Androidplot* [57] com algumas diferenças. É importante realçar que esta tem um número reduzido de possíveis gráficos (linhas, barras e pontos) mas permite ter combinações de gráficos simples para criar apresentações mais complexas. A possibilidade de trabalhar com conjuntos de dados incompletos e com dados inseridos em tempo real faz com que esta seja uma alternativa interessante. Tal como com a solução anterior, existe uma demo descarregável que dá a possibilidade de ver alguns exemplos completos a funcionar num dispositivo *Android* [61]. É possível criar apresentações de dados surpreendentemente densas, em termos de informação apresentada, com combinações das três variedades simples, como é visível na figura 2.46.



Figura 2.46: *GraphView* - Gráfico de linhas estilizado

### 2.2.3 Windows Phone

O *Windows Phone* é o sucessor do *Windows Mobile*, o sistema operativo móvel mais usado nos EUA em 2007 [62] antes do surgimento do *iPhone*, mas não é compatível com o seu antecessor. Apesar de não ter o sucesso que o *iPhone* e o *Android* tiveram, o *Windows Phone* conseguiu influenciar o mercado durante algum tempo. No terceiro trimestre de 2013, 24 países registaram mais vendas de *smartphones* com *Windows Phone* do que *iPhones*. No mesmo ano, em França, na Grã-Bretanha e nos EUA, *smartphones* com *Windows Phone* chegaram a controlar 12.5%, 11.9% e 6.3% dos mercados respetivos [63]. Mais tarde, em 2014, os três países onde o *Windows Phone* teve mais sucesso foram os EUA, a Índia e o Brasil com 11%, 7.5% e 6.9% do mercado, respetivamente [64]. Há que notar que estes dados devem ser interpretados como uma “fotografia” do estado destes mercados e não representa a globalidade das vendas. O mercado respondeu bem ao *Windows Phone* durante algum tempo e notou-se algum aumento nas vendas. No entanto, os valores aqui apresentados mostram aproximadamente o sucesso máximo da plataforma em termos de taxas de adoção. Apesar dos esforços por parte da *Microsoft*, hoje em dia dispositivos com *Windows Phone* vendem muito menos do que anteriormente [65][66][67][68].

Independentemente do sucesso da plataforma, o *Windows Phone* é um sistema operativo com alguns fatores interessantes. Com o objetivo de fornecer uma experiência coesa através de todos os dispositivos *Windows*, o *Windows Phone 8*, o *Windows 8*, a *Xbox 360* e a *Xbox One* partilham uma série de funcionalidades e semelhanças na interface. Isto

permite que um utilizador se sinta confortável a alternar entre o seu dispositivo móvel e fixo, semelhante à experiência que a *Apple* oferece, permitindo alguma personalização no *software*, alguma escolha no *hardware* e preços competitivos. A figura 2.47 mostra como os utilizadores podem personalizar a sua experiência com *Windows Phone* ao escolher não só o aspeto geral do menu inicial como a posição e proeminência das aplicações que mais lhe interessam. Além disso, este sistema operativo conta com o poder do *Microsoft Office* e dá a possibilidade de utilizar versões adaptadas a *smartphones* de ferramentas como o *Word*, *PowerPoint* e *Excel*. Como o *Windows Phone* provou alcançar um número significativo de utilizadores foi realizado o estudo com o intuito de averiguar as capacidades de criação de gráficos neste ambiente. Algumas ferramentas foram analisadas para se obter uma visão geral do potencial disponível, nomeadamente, o *Sparrow Toolkit*, o *Silverlight Toolkit*, o *Visifire* e o *RadChart*. O *Silverlight Toolkit* é especialmente relevante por ser uma solução oficial da *Microsoft*.



Figura 2.47: *Windows Phone* - Exemplos de personalização do menu inicial

### Sparrow Toolkit

O *Sparrow Toolkit* [69] é um conjunto de ferramentas de construção de gráficos para *Windows Phone* e *Silverlight* construído para ser flexível, rápido e eficiente. O projeto nunca saiu da fase de desenvolvimento e, durante a escrita deste documento, foi descontinuado. Por isso, algumas das funcionalidades planeadas não foram implementadas mas ainda é possível obter o código e criar alguns dos gráficos mais básicos, adicionar legendas, ampliar e movimentar os seus elementos. Existe muito pouca documentação, dificultando a aprendizagem, e a sua utilização podia ser mais intuitiva. Já não é uma alternativa relevante mas é importante saber que existe.



## Silverlight Toolkit

Este *toolkit* [70] oferece a possibilidade de criar aplicações com acesso a funcionalidades nativas do *Windows Phone*. Esta solução permite também criar gráficos simples mais polidos e sem muito código. No entanto, este *toolkit* não tem sido atualizado recentemente e não existe muita documentação, o que dificulta a sua utilização.

## Visifire

O *Visifire* [71] oferece gráficos de alta qualidade para diferentes plataformas, incluindo *Windows Phone* [72] e *Windows 8*. A solução permite criar 21 gráficos animados diferentes, com a possibilidade de escolher variantes 2D ou 3D, personalizá-los, combiná-los e usar o ecrã multi-toque para os manipular. Existem demonstrações descarregáveis do site do *Visifire* [71] que demonstram as suas capacidades [73].

Alguns exemplos notáveis da galeria na página do produto estão apresentados nas figuras abaixo. As figuras 2.48 e 2.49 mostram indicadores coloridos, respetivamente em forma de velocímetro e indicadores de nível que, apesar de poderem ter aplicações muito interessantes, são bastantes simples. Ilustrado na figura 2.50 está uma combinação de gráficos de barras e linha. Mais interessante ainda é a possibilidade de apresentar conjuntos de dados incompletos através de gráficos complexos como é visível na figura 2.51.



Figura 2.48: *Visifire* - Velocímetro



Figura 2.49: *Visifire* - Indicadores para comparação de receitas



Figura 2.50: *Visifire* - Combinação de gráficos de receitas e lucro

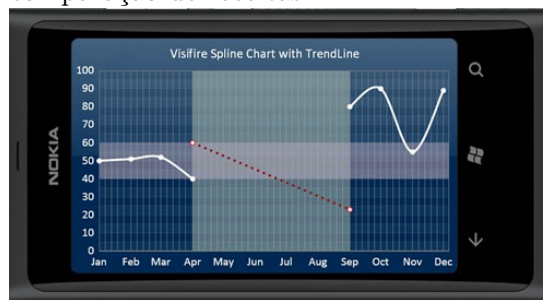


Figura 2.51: *Visifire* - *Spline* complexo



## RadChart

O *RadChart* [74] é uma biblioteca pertencente à *Telerik* [75] desenhada especificamente para desenvolver gráficos para aplicações *Windows Phone* em ambientes empresariais. Esta biblioteca oferece vinte diferentes tipos de gráficos e a possibilidade de os combinar e juntar para obter composições gráficas mais adequadas. Na demo disponível a partir do *website* [76] é possível explorar as possibilidades da solução. Tal como com o *Visifire*, os gráficos criados com *RadChart* são de alta qualidade e há um elevado nível de personalização e controlo. Gráficos criados podem ser tornados interativos, sendo possível realçar informação ao pressionar, movimentar ou ampliar os seus componentes.

O *Jack of Tools Pro* [77] é uma aplicação apresentada pela *Telerik* que utiliza *RadCharts*. Esta funciona como uma caixa de ferramentas virtual, fornecendo informação sobre a posição, orientação e movimento do utilizador, entre outras funcionalidades, através de diversos gráficos. A figura 2.52 mostra os gráficos relacionados com a velocidade do utilizador ao longo de um período de tempo. A linha vermelha no referencial representa a velocidade máxima do utilizador, enquanto que a linha azul representa a velocidade média e a área verde mostra a velocidade atual; o velocímetro mostra a mesma informação mas de uma forma mais simplificada.

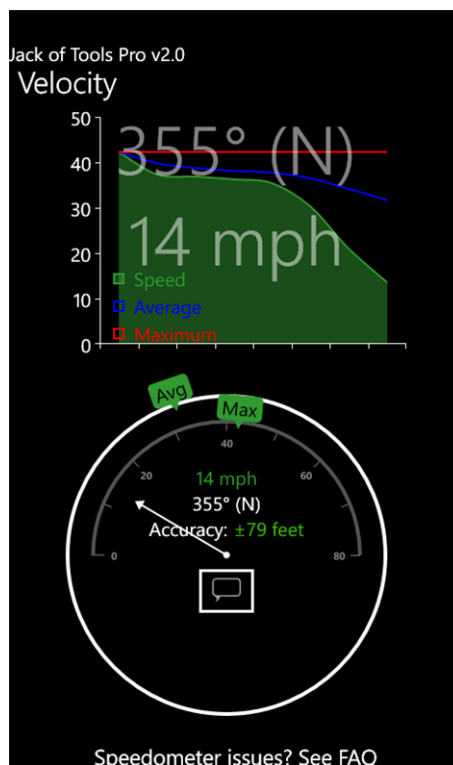


Figura 2.52: *Jack of Tools Pro* (*RadChart*) - Velocidade atual e histórico

### 2.2.4 HTML5

Por oposição aos temas observados nas secções anteriores, o **HTML5** [78] não é um sistema operativo mas sim uma *markup language* que permite disponibilizar *media* tal como vídeos, música e aplicações *web* complexas sem necessitar de quaisquer acrescentos. No entanto, devido às suas capacidades, é possível desenvolver aplicações com **HTML5** que podem ser utilizadas em qualquer tipo de dispositivo que tenha acesso a um navegador de *Internet*. Isto significa que, na prática, um navegador de *Internet* poderá tornar-se equivalente a um sistema operativo para certas aplicações. Um criador de *software*, quando deparado com a escolha de desenvolver uma aplicação para *iOS*, *Android* ou *Windows Phone*, pode simplesmente escolher criar uma aplicação *web* que funcione em todos estes e outros sistemas operativos, em qualquer computador portátil, computador fixo ou outro dispositivo compatível com **HTML5**. Esta versatilidade tem como custo alguma perda de conforto da parte do utilizador já que estas aplicações não serão executadas nativamente no sistema operativo do seu dispositivo, o que empobrece o “*look and feel*” devido às limitações técnicas das linguagens *web*.

Estas limitações têm vindo a tornar-se cada vez menos restritivas devido a constantes avanços tecnológicos, com o potencial de um dia deixarem de existir, o que pode aliciar criadores de *software* a escolher esta via. Como exemplo de esforços feitos para maximizar o conforto do utilizador, hoje em dia é possível ajustar uma aplicação *web* a cada sistema operativo de forma a que o código se adapte ao seu ambiente, escolhendo a existência e posição de elementos de forma a simular o aspeto de uma aplicação nativa.

O *responsive design*, uma filosofia de criação de *websites*, trata exatamente disso, promovendo aplicações *web* cuja aparência é transformada consoante o dispositivo utilizado. Seguindo esta ideologia, uma aplicação pode transformar a disposição do seu conteúdo para se adaptar a diferentes dimensões e resoluções de ecrãs. Mais do que introduzir adaptabilidade, o *responsive design* também protege uma aplicação de potenciais problemas futuros se for necessário atingir mais e novas plataformas, já que *websites* construídos com esta filosofia são mais flexíveis. O **HTML5** ainda está em fase de desenvolvimento mas já mostra imenso potencial e já começa a ser incorporado como alternativa ao antigo **HTML 4.1** em alguns *websites* e na maior parte dos navegadores de *Internet*. Por exemplo, o **HTML5** é usado no *Youtube*, sempre que a combinação de navegador e vídeo o permita, e na plataforma de *streaming* de vídeo *Twitch* [79]. Ao adotar esta tecnologia, ambos estes *websites* começaram a beneficiar de melhor desempenho e deixaram de depender de *software* externo que cada utilizador tem de manter atualizado.

O *DeviantART*, um site de partilha de trabalhos originais de artistas, contém o “*DeviantART muro*” onde é possível criar desenhos com ferramentas avançadas utilizando somente **HTML5**, como apresentado na figura 2.53. Numa experiência com o propósito de descobrir até onde é possível ir com os sons do **HTML5**, Fritz Obermeyer criou o *Fritzo Keys* [80], um “piano” fluido em que as teclas se influenciam entre si, tanto em termos visuais como auditivos. A aplicação até oferece quatro formas diferentes de visualizar as teclas, nome-

adamente, em forma de um piano mais clássico, ilustrado na figura 2.54, ou em forma de uma das alternativas: *wedges*, *thermal* e *flow*, esta ultima representada na figura 2.55.

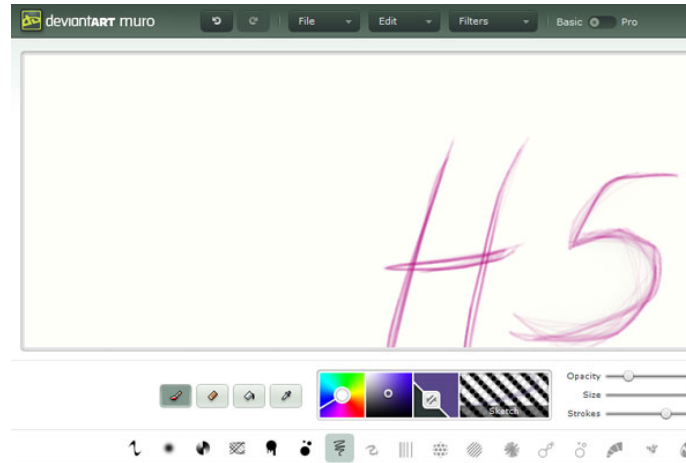


Figura 2.53: *HTML5* - “*DeviantART muro*”

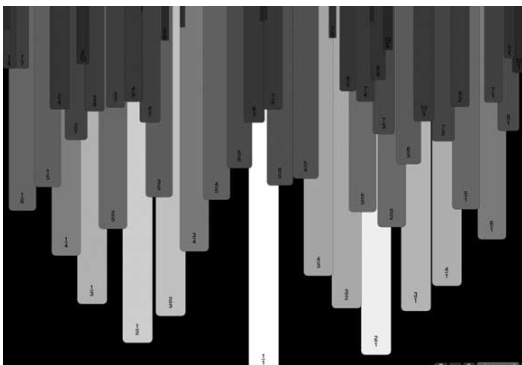


Figura 2.54: *HTML5* - Fritz Keys Classic

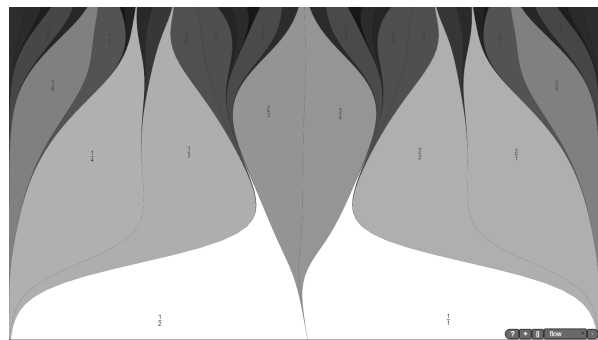


Figura 2.55: *HTML5* - Fritz Keys Flow

### D3

**D3** [81] é uma biblioteca *Javascript open source* para manipulação de documentos e visualização de informação que utiliza **HTML5**, **CSS** e **SVG**, um formato de imagens vetoriais com base em **XML**, que permite criar gráficos personalizáveis a partir de tabelas de dados. Esta biblioteca permite criar efetivamente uma quantidade infinita de formas de visualizar os dados, dependendo somente das capacidades e experiência dos utilizadores e dos gráficos que estes criarem [82]. A curva de aprendizagem desta biblioteca é significativamente maior do que a de outras já que é necessário trabalhar em baixo nível para desenvolver novos gráficos. Outra desvantagem desta biblioteca associada à dificuldade de aprendizagem está relacionada com a qualidade dos gráficos básicos. Sem algum trabalho investido na estética de um gráfico, o resultado inicial vai ter um aspeto muito simples e rudimentar. No entanto, é possível reutilizar gráficos já criados por outros utilizadores para diminuir o

volume de trabalho, dada a sua abrangência de desenvolvimento. Devido a isto, a totalidade de ferramentas já criadas é impressionante e continua a crescer, contando com novas formas inovadoras de apresentar e explorar dados, especialmente considerando que não são impostos limites nos tipos de gráficos que podem ser feitos.

No entanto, esta possibilidade de criar gráficos ilimitadamente complexos não proíbe versões mais simples de serem criadas. A figura 2.56 mostra alguns gráficos e *layouts* pensados para apresentar a atividade de utilizadores em cinco redes sociais hipotéticas ao longo de três semanas. Os primeiros dois tipos de gráficos, os de círculos e os de barras, mostram a mesma informação de duas formas semelhantes. Estes ilustram a quantidade de visualizações distintas de uma determinada página, representados pelos círculos exteriores, e a quantidade de interações nas páginas respetivas, ilustrados pelos círculos interiores. A representação é equivalente nos gráficos de barras, existindo uma correspondência entre as barras mais escuras e os círculos exteriores. O terceiro tipo de gráficos mostra o fluxo de utilizadores entre as cinco redes sociais. Cada arco exterior representa uma rede social e cada faixa colorida em contacto com um dos arcos representa quantos utilizadores mudaram de rede social. Exemplificando, relativamente à rede social azul na primeira semana, muitos dos seus utilizadores mudaram para as redes verdes e amarelas, entraram quase metade dos utilizadores da rede social castanha e uma pequena quantidade manteve-se sem alterações.

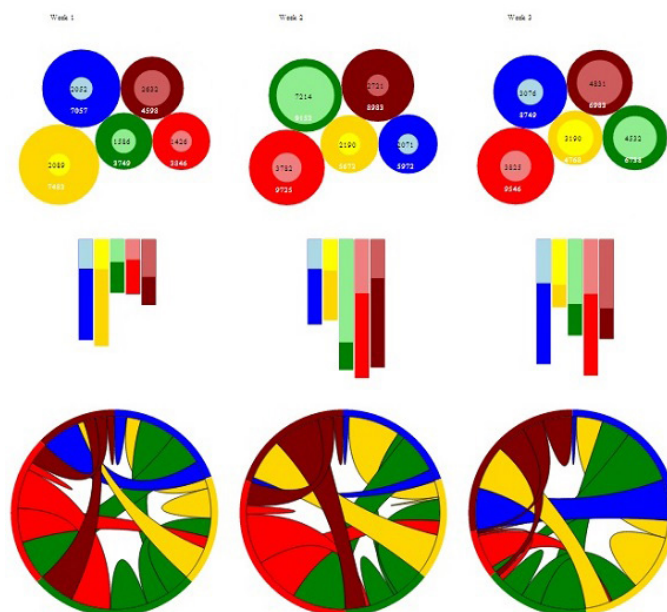


Figura 2.56: *HTML5* - SVG + D3JS

A figura 2.57 mostra um gráfico de barras empilhadas com quatro valores por barra, representados pelas quatro cores. Os mesmos dados podem ser visualizados com um gráfico de barras tradicional, como se pode ver na figura 2.58. A transição entre os dois tipos de gráficos é feita no momento e com uma animação rápida e fluida, demonstrando algumas

das capacidades de animação da biblioteca. A figura 2.59 mostra o déficit acumulado dos Estados Unidos da América entre 2001 a 2013. No centro do gráfico está apresentado o déficit dos EUA e os vinte países com a maior influência nesse valor, tanto do lado dos valores a receber como do lado dos valores a pagar, estão apresentados por cima e por baixo, respetivamente. A cada segundo que passa o eixo temporal de referência avança um mês e cada país muda a sua posição, cor e forma de modo a refletir a sua influência. Tudo isto acontece com animações fluidas e ilustra excecionalmente bem como animações podem ser criadas para qualquer gráfico mas só é possível vê-lo em ação no *website* [83]. Existem muitos exemplos de gráficos gerados com estas ferramentas em várias galerias de exemplos interativos [84][85], a destacar o *Morphogenesis Simulation* [86] que permite gerar um diagrama de Voronoi [87] onde as células se comportam como um fluido.

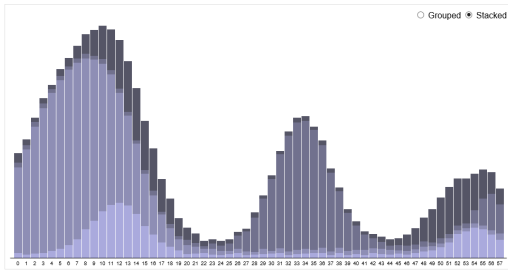


Figura 2.57: *HTML5* - Barras empilhadas

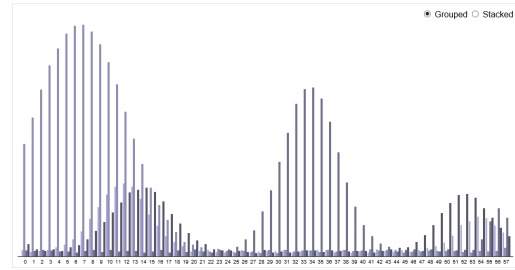


Figura 2.58: *HTML5* - Barras agrupadas

A figura 2.60 mostra as dependências das classes da linguagem de programação *Flare*. Cada círculo representa uma classe e cada linha mostra a relação de cada classe com as relacionadas. Além disso, a posição de cada ramo indica uma de três situações: se as classes aí apresentadas dependem somente de outras sem que outras dependam de si, se acontece o caso contrário ou uma combinação dos dois casos. É assim alcançada uma visão global do estado da linguagem. Com algumas adições torna-se possível escolher um conjunto de dados e aprofundar para obter informação mais específica.

## Highcharts

O *Highcharts* [88] é uma biblioteca *Javascript*, complementada por *HTML5* e *SVG*, que oferece um vasto potencial de criação e manipulação de gráficos. O *Highcharts* também é *open source*, apesar de não ser gratuito como o *D3*, mas conta com mais e melhor documentação. Aliado a este facto está a facilidade de programação acrescida proveniente de algumas ferramentas pré-existentes, tornando a aprendizagem mais acessível e simples. A complexidade de programação mais reduzida resulta num nível de personalização menor, mas a biblioteca já conta com uma grande variedade de tipo de gráficos.

Mesmo os gráficos mais básicos são muito polidos e beneficiam das funcionalidades do *Highcharts*. A figura 2.62 ilustra um esquema de cores alternativo ao da figura 2.61 e é apenas um dos esquemas pré-definidos que podem ser aplicados em qualquer momento com

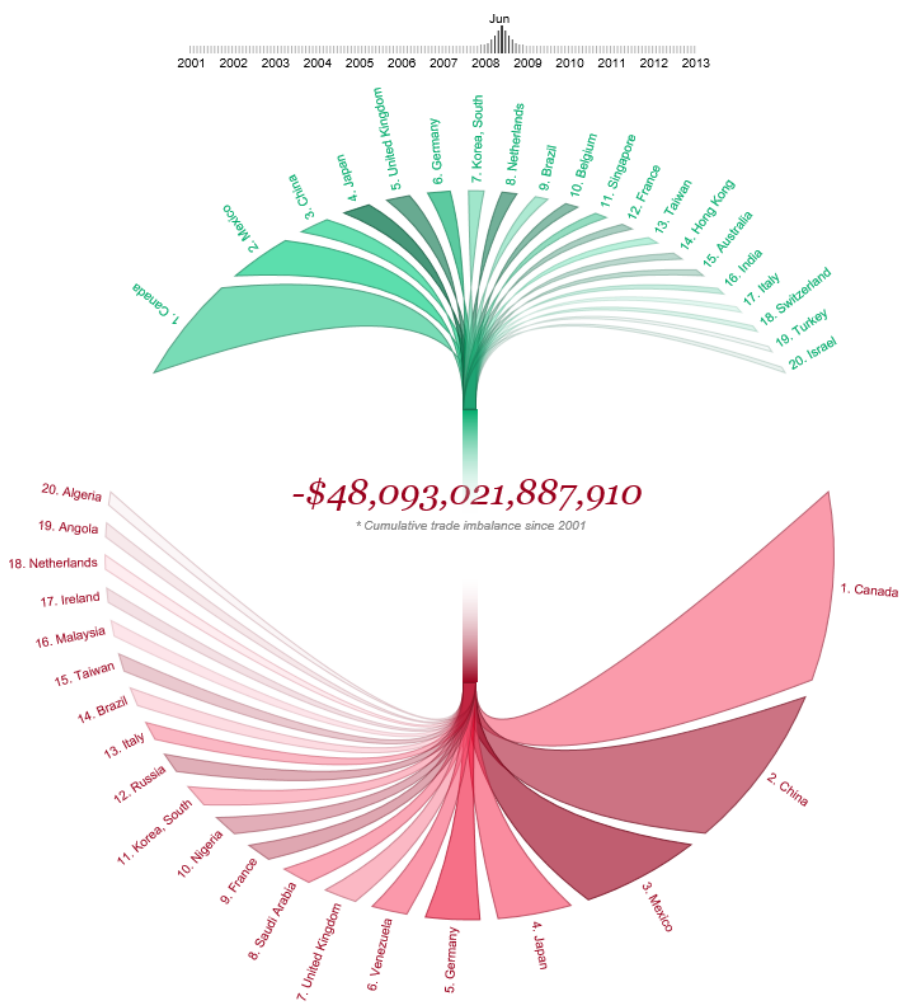


Figura 2.59: HTML5 - Países ordenados por Déficit

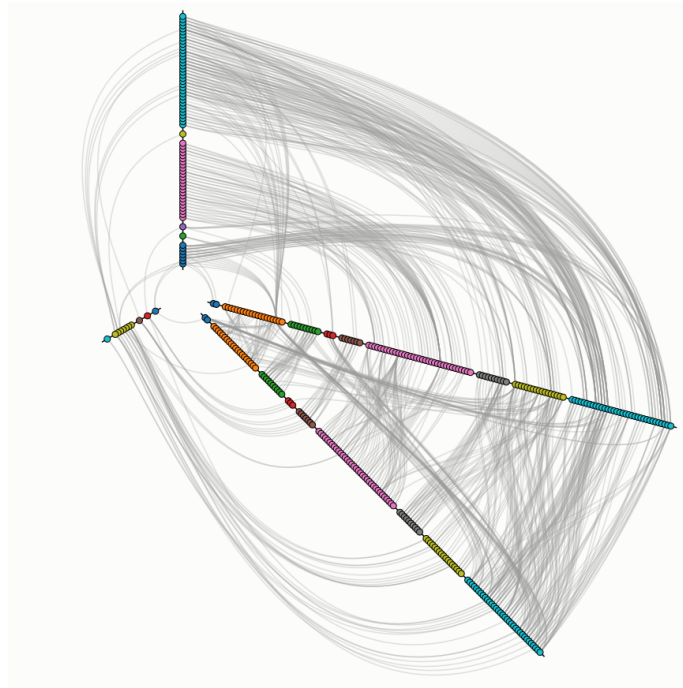


Figura 2.60: *HTML5* - Dependências Flare

facilidade. O *Highcharts* mostra a sua capacidade de criar gráficos com dados incompletos na figura 2.63, especificamente num gráfico de área com uma falha no quarto ponto. Para mostrar um pouco mais da variedade de tipos de gráficos é apresentado também um mapa da temperatura atmosférica ao longo do ano de 2013 e em função da hora do dia, visível na figura 2.64.

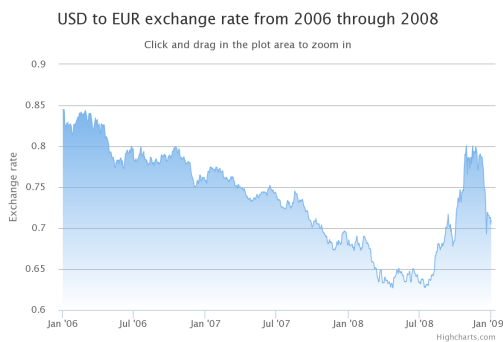


Figura 2.61: *Highcharts* - Gráfico de área claro

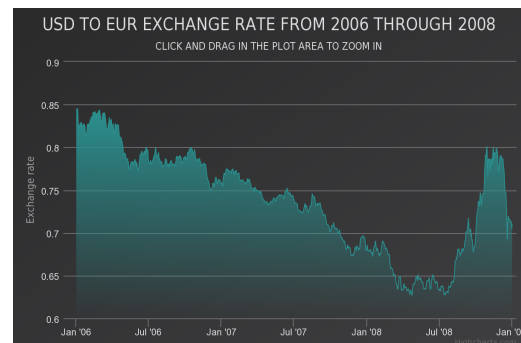


Figura 2.62: *Highcharts* - Gráfico de área escuro

### CanvasJS

Ao contrário das duas bibliotecas analisadas anteriormente, o *CanvasJS* [89] não utiliza *SVG* para desenhar os gráficos mas sim o *Canvas* de *HTML* [90]. A principal diferença

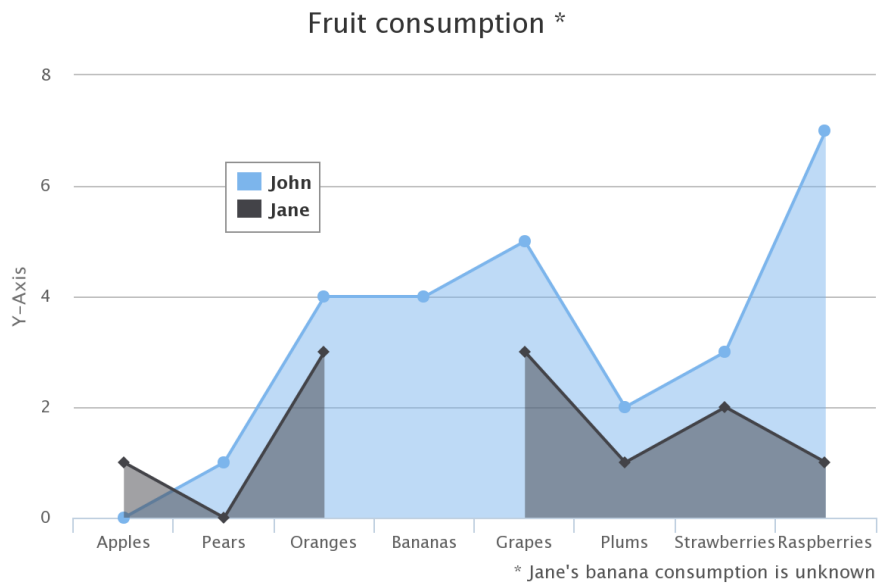


Figura 2.63: *Highcharts* - Gráfico de área com dados parciais

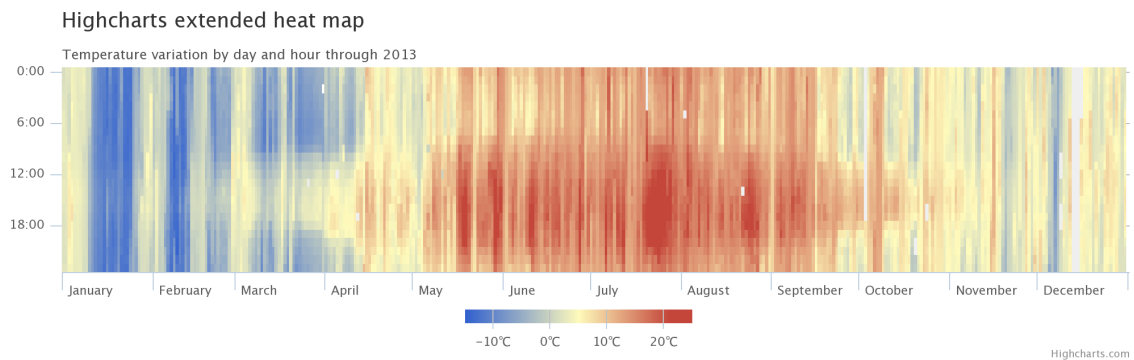


Figura 2.64: *Highcharts* - Mapa de temperatura 2013



entre as duas tecnologias é a utilização de vetores ou pixeis, respetivamente pertencentes aos *SVG* e *Canvas* [91]. Resumidamente, o *Canvas* é mais apropriado para desenhos estáticos velozes, mesmo que envolvam animações, enquanto que os *SVG* estão mais bem adaptados para criar gráficos com uma maior interação e capacidade de adaptação com o custo de uma maior carga de processamento.

Esta característica faz com que *CanvasJS* por si só não seja a melhor escolha para desenhar gráficos simultaneamente para várias plataformas já que estes têm de ser desenhados individualmente para cada uma de modo a garantir que a qualidade visual se mantém. No entanto, a sua facilidade de uso, nível de qualidade e extrema baixa carga de processamento tornam esta biblioteca uma solução apelativa em certas situações. Os tipos de gráficos mais comuns estão disponíveis e têm uma excelente qualidade. Como ilustrado na figura 2.65, passar o ponteiro do rato por cima de um gráfico dá informação mais detalhada sobre ele, especificamente sobre a secção ou nó selecionado.

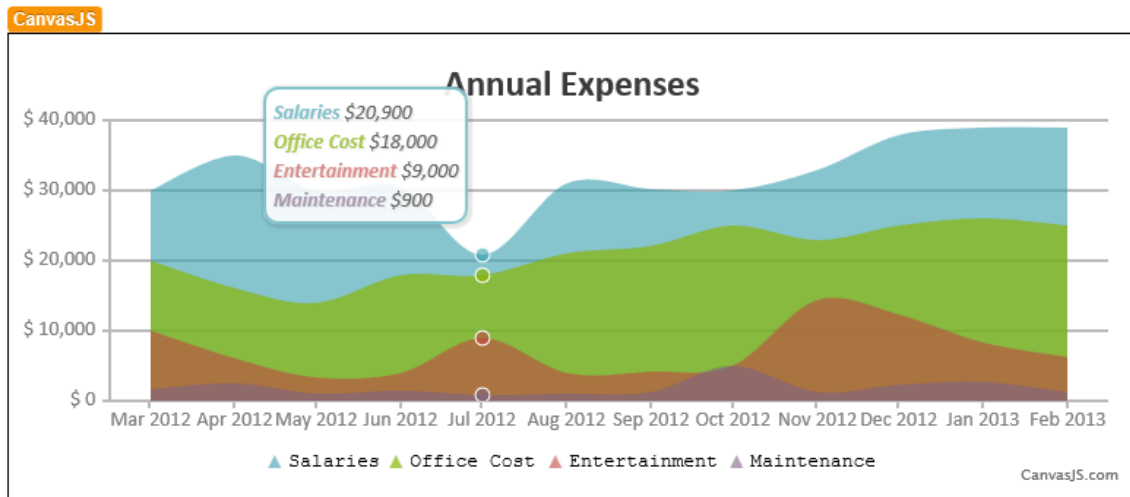


Figura 2.65: *CanvasJS* - Gráfico de área com coordenada selecionada

## Google Charts

A *Google* oferece uma biblioteca gratuita de criação de gráficos [92] de utilização direta que funciona com *Javascript*. Existem 28 tipos distintos de gráficos que podem ser personalizados para se adaptar às necessidades de cada utilizador e todos eles são interativos. Depois de importar pelo menos um dos muitos tipos de gráficos, basta inicializar uma nova instância de cada gráfico desejado, inserir os dados e carregar qualquer explorador de *Internet* com código *HTML* para obter um gráfico completo. A figura 2.66 contém o gráfico disponível no guia inicial [93] com um dos seus elementos selecionados. Ao pressionar tanto numa das secções como na etiqueta correspondente na legenda, essa secção fica realçada no gráfico. Funcionalidades como esta e outras funcionam sem qualquer tipo de trabalho por parte de cada utilizador e funcionam para qualquer gráfico do conjunto.

A simplicidade de uso eleva esta opção a um nível incrivelmente competitivo quando comparado com as outras soluções aqui analisadas. Existe uma galeria com exemplos de cada tipo de gráfico disponível de onde foram copiados um gráfico de mapa [94] e um *treemap* [95] que estão apresentados nas figuras 2.67 e 2.68, respetivamente. Alternativas como o D3 continuam a ter muito mais flexibilidade mas há vantagens na rapidez e facilidade de desenvolvimento do *Google Charts*.

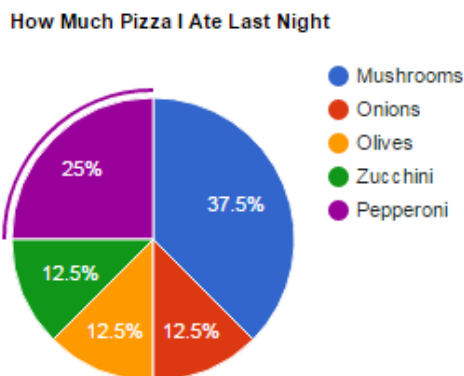


Figura 2.66: *Google Charts* - Gráfico de secções

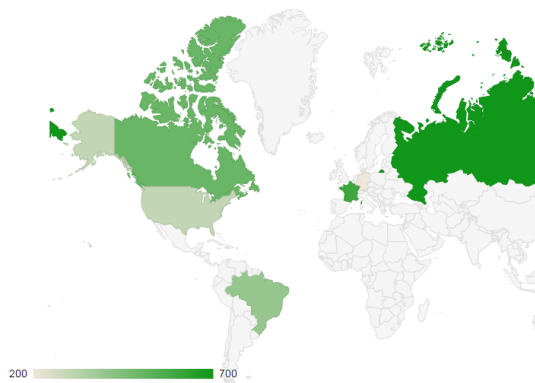


Figura 2.67: *Google Charts* - Gráfico de Mapa

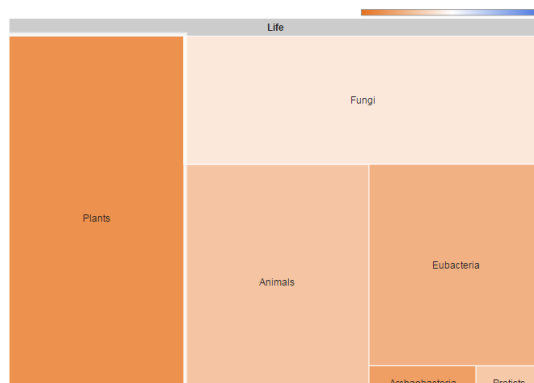


Figura 2.68: *Google Charts* - *Treemap*

### 2.2.5 Plugin Outlook

O *Outlook* permite a criação de extensões que acrescentam funcionalidades e propriedades. Esta possibilidade é muito interessante para um produto que tem em mira alcançar aproximadamente o mesmo público alvo que o *Outlook* alcança, especialmente quando já foram feitas outras extensões com objetivos semelhantes. Adicionalmente, existe uma ferramenta chamada *Add-In Express* [96] que expande e melhora as capacidades de criar extensões para o *Outlook*. A utilização desta incorre em custos adicionais e estes têm de ser pesados

com as suas funcionalidades para determinar se são significativamente importantes para alcançar os objetivos do produto mas é algo a ter em conta.

Existem duas principais formas de apresentar gráficos diretamente na interface do *Outlook*: com os próprios gráficos do *Visual Studio* ou com gráficos *web* num navegador de *Internet* embebido. Comum a ambas abordagens é a utilização de *Custom Task Panes*, painéis separados do resto da interface do *Outlook* com conteúdos ajustáveis. Deste modo torna-se possível adicionar variadíssimos elementos, não só botões e caixas de texto mas também gráficos de muitos tipos, caixas de notificações e temporizadores. Cada alternativa tem um conjunto de vantagens e desvantagens que serão analisadas nesta secção. A maioria da informação aqui incluída vem de experiências práticas em vez de aglomeração de exemplos obtidos de páginas de *Internet* já que é relativamente simples fazer experiências práticas e não há informação extensa sobre estas abordagens que se concentre numa única versão do *Outlook*.

### Gráficos Visual Studio

Na *Toolbox* do *Microsoft Visual Studio* está disponível a opção de introduzir um gráfico vazio numa *Custom Task Pane* associada a um *plugin*. Um gráfico introduzido deste modo, sem qualquer outro tipo de alteração, não é inicialmente visível já que não tem dados. O *Visual Studio* fornece uma pré-visualização de um gráfico genérico nessa situação para ser possível escolher o tipo, dimensões e posição mesmo antes de serem inseridos dados. Os dados podem ser inseridos ponto a ponto ou através da importação de ficheiros externos. Os gráficos podem ser personalizados não só em termos visuais mas também ao nível do comportamento. Alterações visuais incluem alternar entre duas e três dimensões, adicionar etiquetas e dicas de ferramenta, aplicar rotações e escolher as cores usadas. Gráficos criados deste modo podem responder a diferentes estímulos, tal como mostrar dicas quando o ponteiro do rato é posicionado em determinados elementos ou aceitar inserções de dados já depois de um gráfico ter sido desenhado.

### Gráficos em Web Browser embebido

Ao adicionar um elemento *Web Browser* a uma *Custom Task Pane* é possível aceder a uma página *web* diretamente no *Outlook*. Sendo essencialmente uma janela do *Internet Explorer*, este explorador está sujeito às mesmas regras e limitações a que está o original. A flexibilidade que provém de poder usar tecnologia *web* leva à possibilidade de desenhar não só gráficos mas também ecrãs equivalentes a páginas de *Internet* que contenham e complementem quaisquer gráficos informativos que sejam criados. O módulo *Web Browser* usa a versão atual do *Internet Explorer* com algumas alterações ao nível das definições, tal como a ativação obrigatória do modo de compatibilidade. Isto significa que grande parte das páginas *web* já existentes é no mínimo parcialmente compatível com este explorador mas certos *frameworks* ou bibliotecas potencialmente poderão não funcionar dado que

a tecnologia fundamental do explorador não está de acordo com as normas e *standards* modernos.

## 2.3 Desenvolvimento multi-plataforma móvel

O custo de desenvolver aplicações nativas para várias plataformas tem sido sempre elevado, até mesmo hoje em dia. Especialmente para pequenas organizações, este custo é frequentemente demasiado elevado para ser uma escolha viável. Com a divulgação do conceito de aplicações híbridas e com a ubiquidade crescente de dispositivos interativos, a procura de uma forma acessível para aplicações multi-plataforma de alta qualidade tem vindo a aumentar. *Frameworks* como o *PhoneGap* [97], o *Xamarin* [98] e o *Appcelerator Titanium* [99] foram criados para tentar satisfazer essa procura, prometendo ser capazes de dar aos criadores de *software* a hipótese de gerir apenas um projeto com a possibilidade de alcançar várias plataformas, resultando em trabalho simplificado e custos reduzidos a médio e longo prazo. Esta secção apresenta as descobertas feitas sobre os *frameworks* e as bibliotecas mais promissoras.

### 2.3.1 PhoneGap

O *PhoneGap* [97] é um *SDK open source* desenhado para criar aplicações móveis multi-plataforma com tecnologias *web*. Estas são apresentadas como se corressem nativamente apesar de serem efetivamente aplicações *web*. No entanto, aplicações feitas com este kit de desenvolvimento conseguem utilizar *plugins* que dão acesso a funções nativas [100]. Como a utilização de funções nativas é feita através de *plugins*, tarefas sem um *plugin* correspondente não são possíveis, invalidando a sua utilização.

Antes de explorar o *PhoneGap* com mais profundidade é importante saber que é uma distribuição do *Apache Cordova* [101]. Em Outubro de 2011, a *Adobe* [102] adquiriu a empresa *Nitobi* [103] e, conseqüentemente, os direitos do *PhoneGap*. Para assegurar que a natureza *open source* do projeto é preservada, em Outubro de 2012 o código do *PhoneGap* foi doado à *Apache Software Foundation* [104] com o nome *Cordova*. O *PhoneGap* e o *Cordova* existem simultaneamente, não existindo uma diferença significativa entre elas atualmente. Apesar de ambas terem a mesma base, poderão no futuro divergir consoante as alterações dos requisitos da *Apache* e da *Adobe*.

O *PhoneGap* utiliza exclusivamente *HTML*, *CSS* e *Javascript* e a sua biblioteca é continuamente expandida pelos seus utilizadores graças a ser *open source*. Adicionalmente, este *SDK* tem acesso a bibliotecas adicionais que expandem os seus limites. Devido à utilização de tecnologias *web*, bibliotecas que também utilizem estas linguagens podem complementar o *SDK*. Por exemplo, utilizando as bibliotecas *Flot* [105], *jqPlot* [106] ou *D3* [81] é possível apresentar gráficos para visualização e exploração de dados. Pressionar num dos nós do gráfico de linhas altera o gráfico de secções adjacente para refletir os dados relativos

ao ano do nó selecionado [107]. Já existem milhares de aplicações criadas até hoje com o *PhoneGap* tal como o *NUS extra - Student Discount Companion App*, o *HealthTap* e o *BrowserQuest*, entre outras [108].

Há que adicionar que o *framework Ionic* [109] pode ser acrescentada a um projeto *PhoneGap*, dado que esta utiliza tecnologias *web*, para melhorar a criação de aplicações. Com *Ionic*, um projeto para uma aplicação “vazia” ou seja, imediatamente após ter sido criado, tem uma série de configurações definidas à partida que diminuem a carga de trabalho dos programadores. A adição de *frameworks* como este tornam o *PhoneGap* numa solução ainda mais forte.

### **NUS extra**

O *NUS extra - Student Discount Companion App* [110][111] é uma aplicação que complementa os serviços disponibilizados pela NUS, ou *National Union of Students*. Os detentores do cartão de estudante da NUS podem aceder à página principal para obter informação geral sobre todos os temas relacionados com a vida de estudante e ainda juntar cupões virtuais que lhes interessem.

Ao passar para a aplicação *mobile*, os estudantes podem ver a localização dos locais mais próximos onde os cupões acumulados são aplicáveis e onde os melhores descontos podem ser encontrados, podendo contar com a ajuda de filtros, alertas e listas personalizadas para acelerar a pesquisa dos melhores negócios. A figura 2.69 contém um exemplo onde está destacada uma promoção num restaurante a um quilómetro de distância.

### **HealthTap**

*HealthTap* [112][113] está disponível tanto em dispositivos *iOS* como *Android* e permite que os seus utilizadores beneficiem da ajuda de médicos especializados, em forma de texto ou vídeo (visível na figura 2.70), em qualquer área sobre qualquer problema que tenham.

A capacidade de apresentar vídeo com *HTML5* combate alguns dos preconceitos antiquados em relação a tecnologias *web*. Adicionalmente, a aplicação tem uma classificação muito alta por parte dos utilizadores que já a utilizaram em ambas as plataformas onde está disponível, o que significa que o *PhoneGap* pode levar a experiências de alta qualidade em várias plataformas.

### **BrowserQuest**

O *BrowserQuest* [114] é um pequeno jogo multi-jogador disponível em *iOS*, *Windows* e *Windows Phone*. As mecânicas e a apresentação gráfica do jogo são bastante simples mas este destaca-se por permitir que um utilizador jogue *online* com outros, independentemente



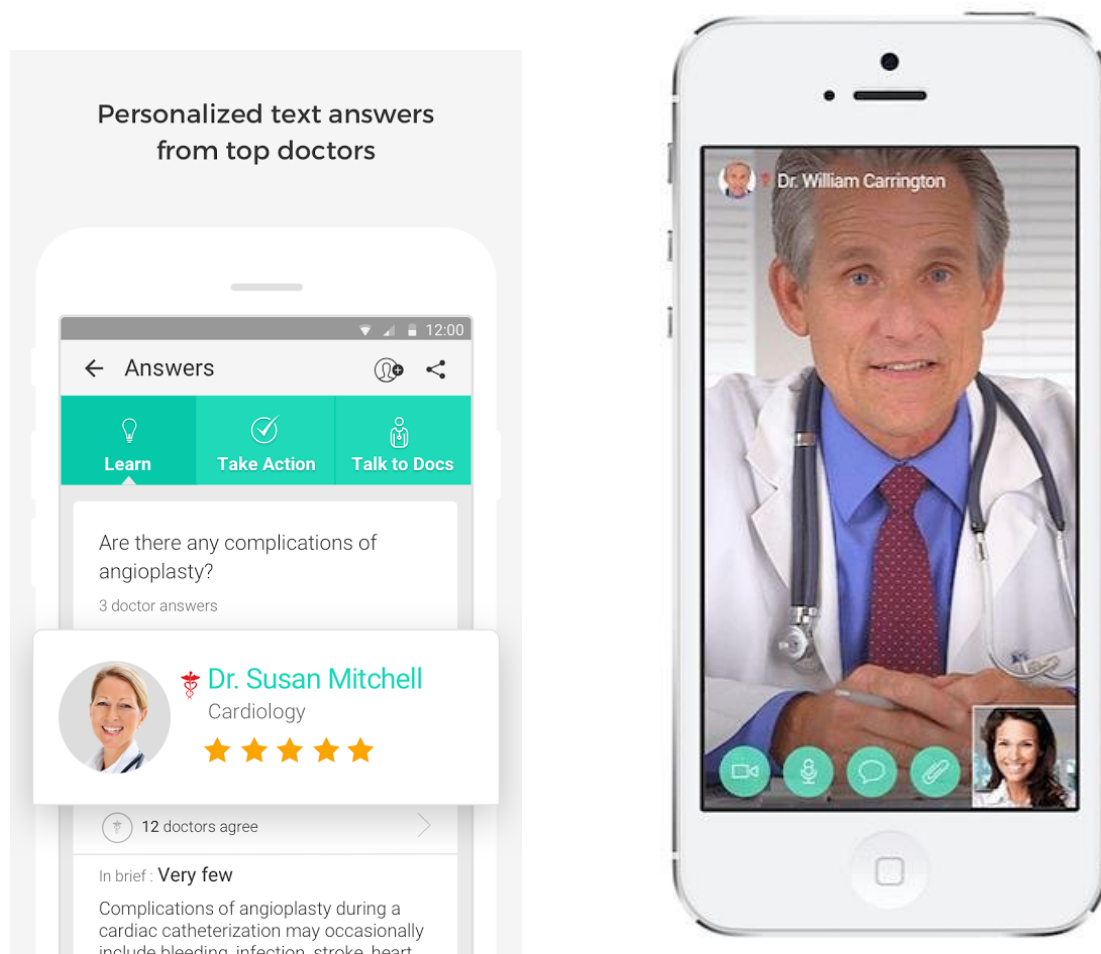
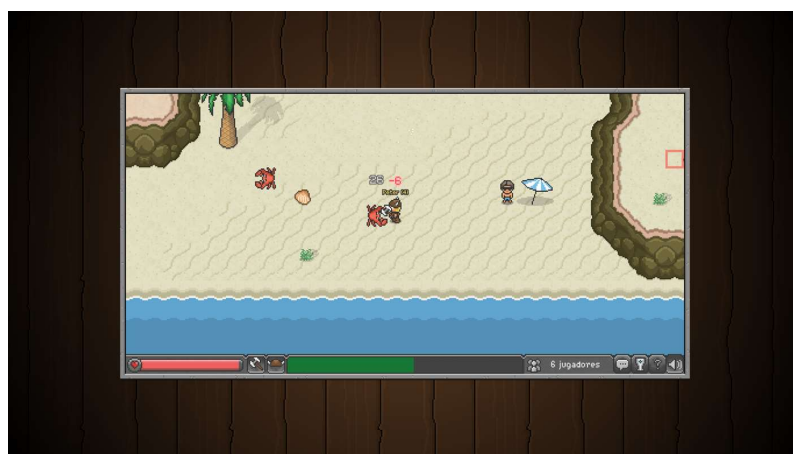
Figura 2.69: *NUS extra* - Exemplo de mapa com informação sobre um marcador específico

da plataforma que estejam a utilizar a qualquer momento. Isto dá a possibilidade de um utilizador interromper a sua sessão de jogo no seu explorador de *Internet* e continuar num dispositivo móvel *iOS* ou *Windows Phone*. As figuras 2.71, 2.72 e 2.73 mostram vários ambientes do jogo mas a terceira figura deste conjunto mostra um combate entre duas personagens controladas por utilizadores.

### 2.3.2 Appcelerator Titanium

A *Appcelerator* [99] é uma empresa de tecnologias *mobile* que criou e disponibilizou o *Titanium*, um *SDK open source* cujo propósito é fornecer uma série de ferramentas adaptadas para *iOS*, *Android*, *Windows* ou *Mobile Web* [115]. A escrita do código é feita em *Javascript* e posteriormente complementada pelo *Titanium API*. Esta abordagem dá acesso a funcionalidades nativas de cada plataforma mas a interpretação do código *Javascript* é feita no momento de execução, o que pode levar a pior desempenho. Isto significa que é possível criar aplicações para várias plataformas móveis sem ter de aprender a programar para cada sistema operativo individualmente mas, devido à forma como esta plataforma funciona, algum código tem de ser específico de cada plataforma alvo. Isto leva a ter de repetir algum trabalho por cada plataforma.

Tal como com o *PhoneGap*, o *Titanium* também tem acesso a bibliotecas de criação de gráficos. A própria biblioteca *D3* também é compatível com este *SDK* mas não é a única. As bibliotecas *CanvasJS* [89], *Chart.js* [116] e *Highcharts* [88] utilizam *Javascript* e *HTML5*

Figura 2.70: *HealthTap* - Comunicação por texto ou por vídeoFigura 2.71: *BrowserQuest* - Exploração de uma praia



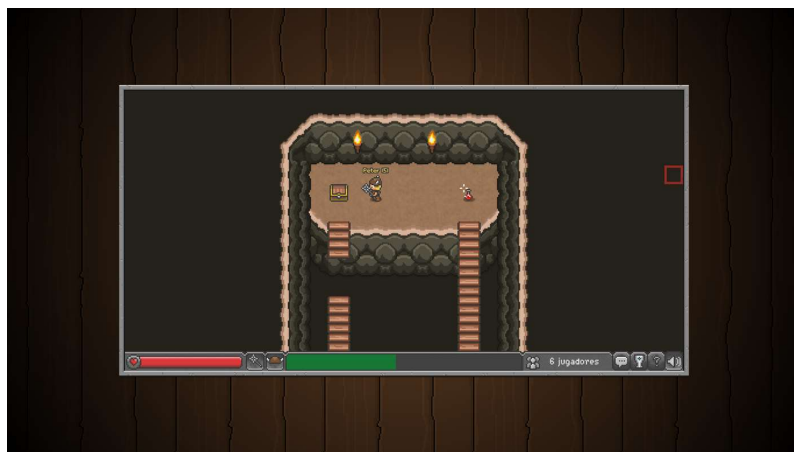


Figura 2.72: *BrowserQuest* - Exploração de uma caverna

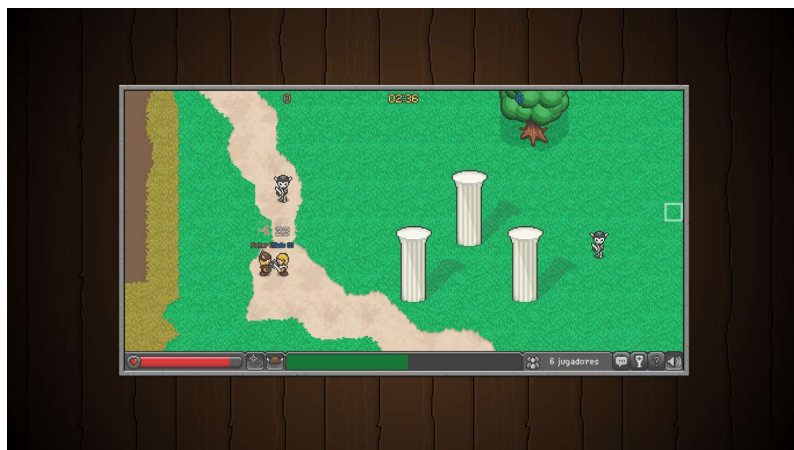


Figura 2.73: *BrowserQuest* - Combate *online* com outro jogador



para criar gráficos, estáticos ou interativos, para qualquer plataforma moderna [117]. A figura 2.74 mostra um gráfico de barras em *iPhone* e *Android*, respectivamente à esquerda e à direita. A figura 2.75 tem dados financeiros, incluindo um gráfico de linha, também em *iPhone* e *Android*. Os gráficos de ambas estas figuras foram feitas com a biblioteca *Highcharts* no ambiente *Titanium* e podem ser consultados no *blog* da *Appcelerator* [118].

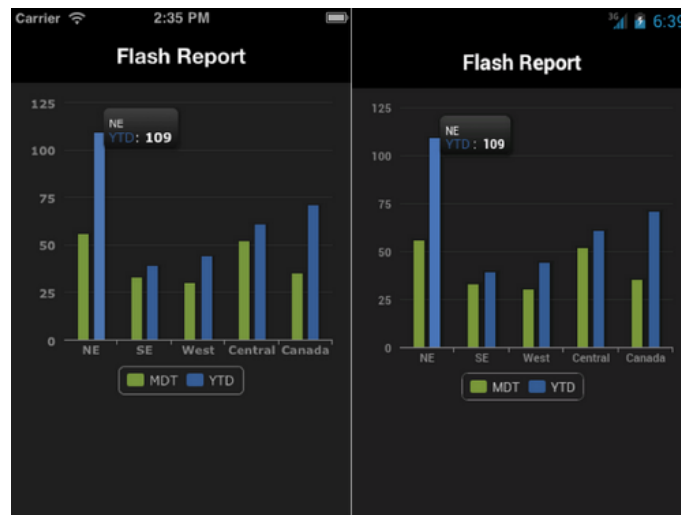


Figura 2.74: *Appcelerator* - Gráfico de barras

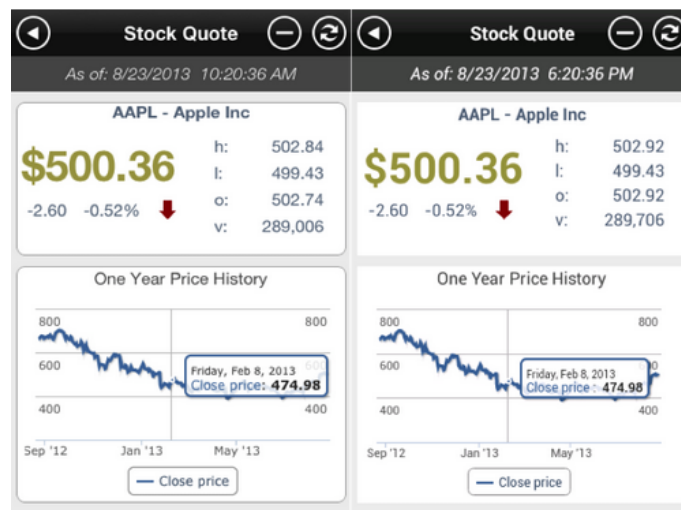


Figura 2.75: *Appcelerator* - Mistura de gráficos

## ParkYa

Um exemplo de uma aplicação original criada com o *Titanium* é o *ParkYa* [119], uma aplicação desenhada para facilitar o estacionamento de automóveis. A aplicação mostra onde existem lugares de estacionamento, quantos espaços estão disponíveis e o pagamento

necessário em cada um. Em caso de ser necessário pagar, a aplicação também permite pagar diretamente pelo *smartphone*. A aplicação também fornece as indicações para chegar ao lugar pretendido, tanto com um mapa como com instruções passo a passo. Ao lado das funcionalidades centrais, são também oferecidas soluções adicionais. A aplicação pode ser adaptada para uso individual ou para grupos de pessoas. Também facilita o processamento de carros mal estacionados, permitindo introduzir matrículas de automóveis e avisar as autoridades. Da figura 2.76 até à figura 2.79 podemos ver o progresso do utilizador, começando por ver uma lista de parques de estacionamento próximos na figura 2.76, o parque de *Cathedral Street* realçado na figura 2.77, mais detalhes revelados na figura 2.78 e, por fim, a escolha de quanto tempo ele deseja estacionar o carro e o preço associado a esse período na figura 2.79.

### 2.3.3 Xamarin

Com o *framework Xamarin* [98] é possível criar aplicações nativas para *iOS*, *Android* e *Windows Phone* com a mesma base de código *C#*. A versão da aplicação para cada uma das plataformas alvo requer código específico, o que significa que o código produzido desta forma, comparado com o do *PhoneGap*, não é tão flexível e simples de gerir, requerendo cuidado adicional. No entanto, este *framework* não tem a falta de acesso a funções nativas do *PhoneGap* nem os problemas de desempenho do *Appcelerator Titanium*, sendo uma alternativa muito interessante para algumas aplicações.

Além destas vantagens, aplicações criadas com *Xamarin* podem ser criadas de forma a que o seu “*Look and Feel*” se adapte à plataforma e ofereça uma experiência conforme as expectativas do utilizador. Aplicações criadas com outros *frameworks* tipicamente têm de oferecer um único estilo de apresentação que sirva para todas as plataformas alvo. Isto resulta em aplicações cujo estilo é muito semelhante ao estilo típico de uma plataforma, ignorando o estilo de todas as outras, ou num estilo que não se alinha a qualquer estilo conhecido. Ambas estas alternativas podem levar à alienação dos utilizadores, dependendo das plataformas escolhidas e do tipo da aplicação, e nesse caso o *Xamarin* tem mais uma vantagem sobre os outros dois *frameworks*.

As figuras 2.80 e 2.81 irão ajudar a ter uma melhor noção das capacidades do *Xamarin*, em particular a capacidade de adaptar o estilo e comportamento da aplicação automaticamente à plataforma.

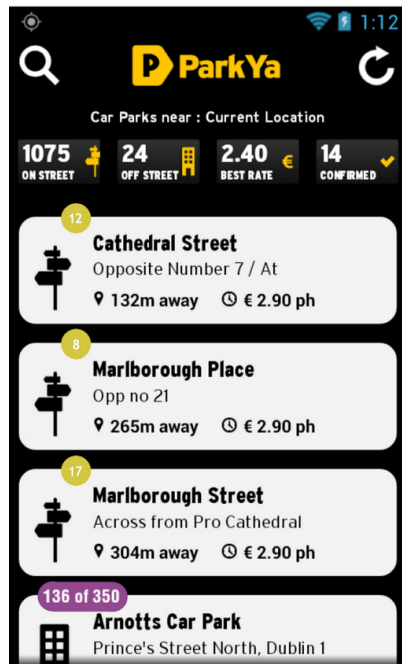


Figura 2.76: Appcelerator - Lista de parques (ParkYa)

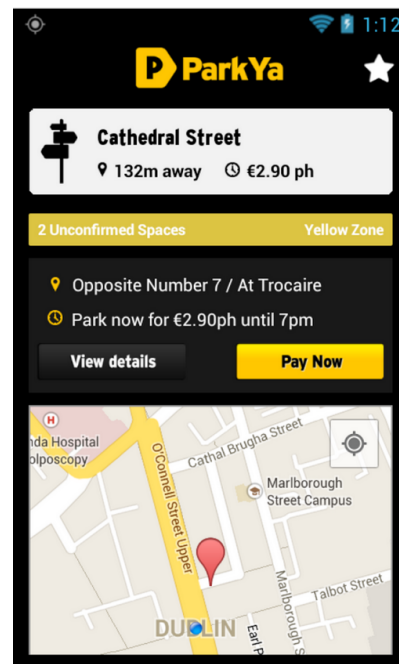


Figura 2.77: Appcelerator - Mapa de rua (ParkYa)

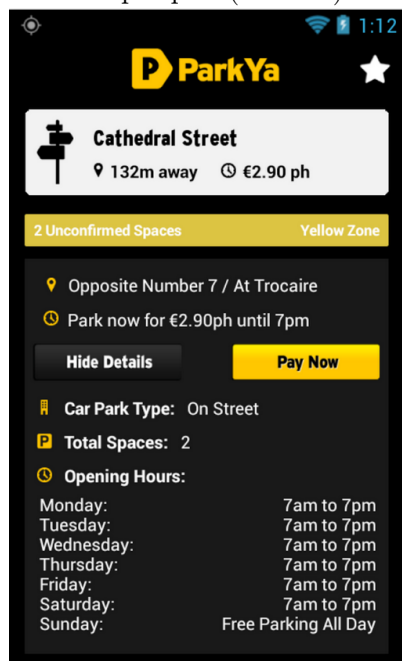


Figura 2.78: Appcelerator - Detalhe de rua (ParkYa)

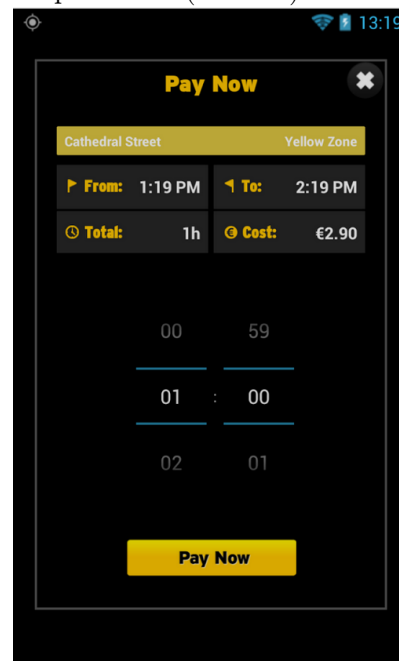


Figura 2.79: Appcelerator - Pagamento (ParkYa)

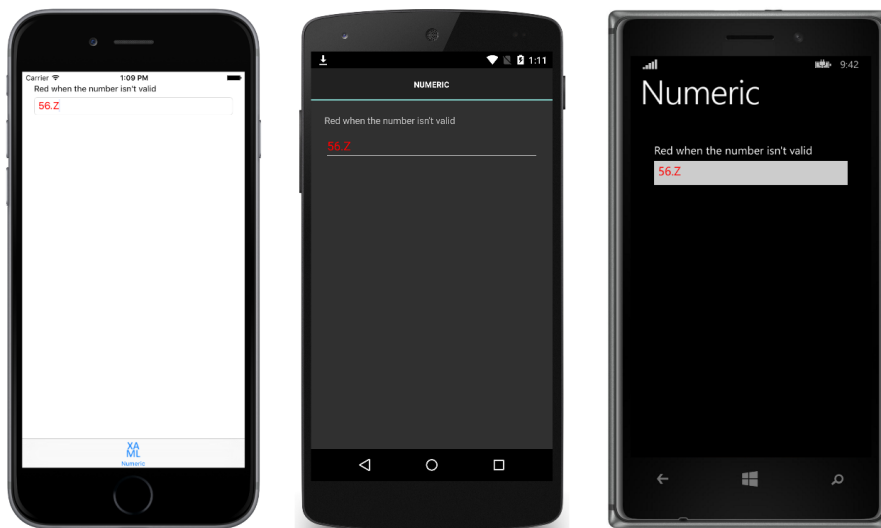


Figura 2.80: *Xamarin* - Exemplo de diferenças entre plataformas

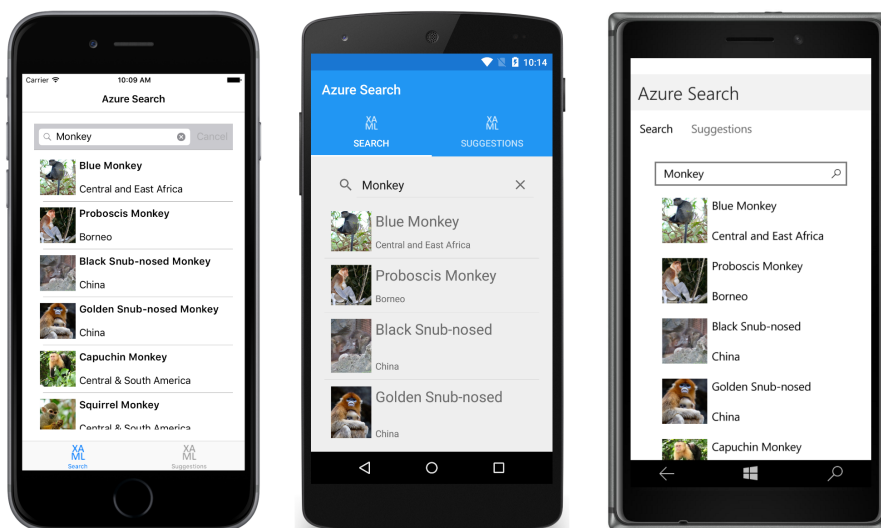


Figura 2.81: *Xamarin* - A mesma aplicação apresentada em cada uma das três plataformas alvo

# Capítulo 3

## Protótipos

Durante as fases iniciais do trabalho foram criados alguns protótipos para analisar o potencial das ferramentas mais acessíveis ou promissoras. Não foram feitos testes com todas as tecnologias desejadas devido à falta de recursos tecnológicos, experiência ou tempo. No entanto os testes feitos evidenciam os pontos fortes e fracos de suficientes alternativas tornando possível fazer uma escolha segura mas flexível para o produto. Neste capítulo estão apresentados os protótipos criados, observações relevantes e as conclusões obtidas do estudo feito, tanto ao nível dos protótipos como do estado da arte. Os critérios de seleção basearam-se na quantidade de documentação, na disponibilidade dos componentes requeridos e na capacidade de evolução futura. Concretamente, foram criados protótipos *Android*, *Windows Phone*, extensão *Outlook* e multi-plataforma móvel (*PhoneGap* e *Appcelerator Titanium*).

### 3.1 Protótipo Android

Em *Android* foi importado um exemplo funcional criado com a biblioteca *AChartEngine* [52]. Devido à falta de experiência e tempo disponível, não foi feito um exemplo de raiz nem foram incluídas muitas das funcionalidades mais interessantes, contendo apenas as mais básicas. No entanto, é simples importar um projeto mais detalhado que demonstre tudo o que a biblioteca permite fazer. Como foi descoberto com o exemplo testado, esta biblioteca permite arrastar e mover qualquer gráfico sem ter de adicionar código. Neste protótipo estão implementados cinco diferentes gráficos: gráficos de linhas, gráficos de barras, gráficos de secções, gráficos de dispersão e gráficos de bolhas. Para melhor conhecer as possibilidades da biblioteca foram feitas várias alterações, particularmente foi ativado em todos os gráficos presentes a capacidade de ampliar e diminuir os seus elementos com

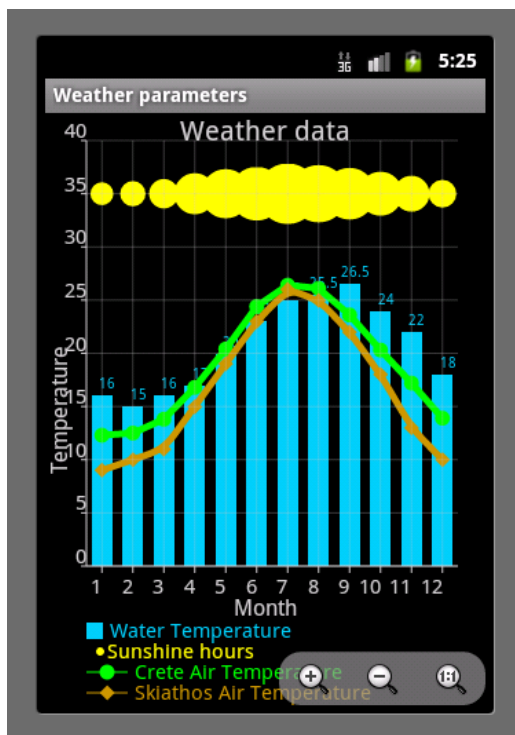


Figura 3.1: Protótipo *AChartEngine* - Painel de gráficos de tempo meteorológico

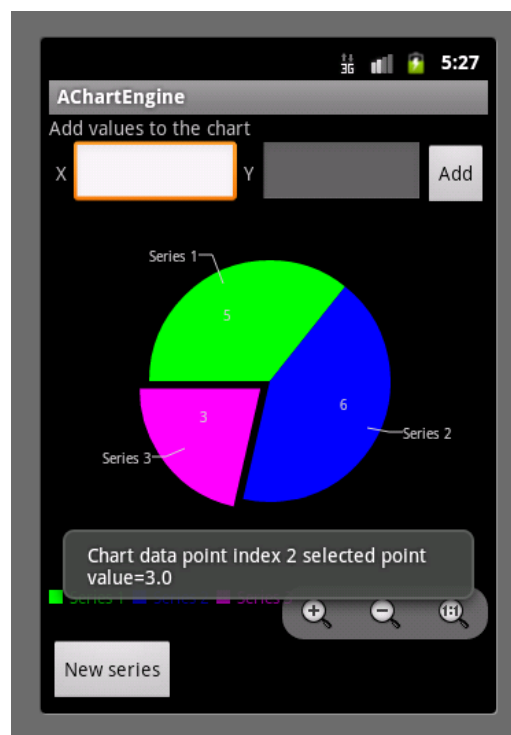


Figura 3.2: Protótipo *AChartEngine* - Gráfico de secções interativo

interações simples.

A figura 3.1 mostra como é possível adicionar múltiplos tipos de gráficos e múltiplas séries de dados numa única apresentação para obter um resultado o mais detalhado e informativo possível. Concretamente, neste ecrã estão combinados dois gráficos de linhas, um gráfico de barras e um gráfico de bolhas. A figura 3.2 mostra uma apresentação que começa vazia mas que passa a mostrar um gráfico de secções à medida que é preenchida com dados em tempo real. Também é possível pressionar em cada secção do gráfico para obter informação mais detalhada.

### 3.2 Protótipo *Windows Phone 8*

As quatro soluções apresentadas no estado da arte permitiram fazer testes simples num emulador de *Windows Phone*. Dependendo da biblioteca ou *framework*, foram feitos mais ou menos testes.

### 3.2.1 Sparrow Toolkit

Com dez tipos de gráficos tradicionais implementados, onze tipos adicionais em desenvolvimento e mais funcionalidades propostas, esta opção era promissora inicialmente. Durante a escrita desta dissertação, o desenvolvimento do *toolkit* foi cancelado e o interesse inicial diminuiu. Devido à dificuldade de acesso a documentação, o protótipo foi criado com apenas três gráficos simples sem nenhum tipo de interação.

### 3.2.2 Silverlight Toolkit

A criação de gráficos é apenas um dos aspetos deste *toolkit* e, conseqüentemente, não está tão polido como alternativas dedicadas a gráficos. Nesta experiência foram feitos uns testes simples mas há várias lacunas difíceis de ignorar: Pouca variedade de gráficos, falta de apoio da *Microsoft* e a exclusividade de utilização em dispositivos *Windows Phone*.

### 3.2.3 Visifire

Através da utilização da versão de avaliação do *Visifire* foi possível consultar e configurar uma série de gráficos pré-definidos. É de notar a possibilidade de personalizar o plano de fundo, de modo a realçar dados como se pode ver com a *trendline* do gráfico *spline* da figura 3.3. Esta funcionalidade é útil para auxiliar a representação de conjuntos de dados incompletos, o que poderá ser necessário no contexto da análise de *emails*. A figura 3.4 mostra um indicador animado, um bom exemplo do nível de variedade de gráficos disponíveis.

## 3.3 *Plugin Outlook*

Para testar a apresentação de gráficos no *Outlook* ambas as abordagens mencionadas no capítulo do Estado da Arte, gráficos nativos e gráficos *web*, vão utilizar *Custom Task Panes*. Apesar de o protótipo não se concentrar nestas ou nas suas capacidades e funcionalidades, é importante a sua menção já que são essenciais.

### 3.3.1 Gráficos Nativos do *Outlook*

Existe um elevado nível de personalização e é possível transformar gráficos previamente definidos em gráficos de tipos diferentes se os dados forem compatíveis. Por exemplo, os mesmos dados foram usados para gerar os gráficos das primeiras duas imagens (3.5 e 3.6). O gráfico de teia da imagem 3.7 não partilha dados com mais nenhum gráfico deste protótipo mas foi adicionado para mostrar que a variedade de modelos vai um pouco além do conjunto tradicional de gráficos que muitas ferramentas oferecem.

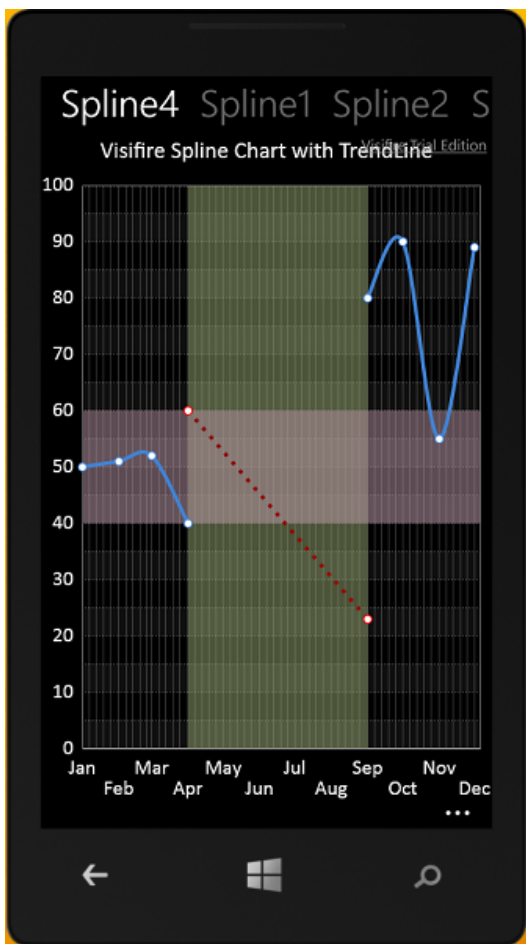


Figura 3.3: Protótipo *Visifire* - Gráfico *spline* com área do plano de fundo realçada



Figura 3.4: Protótipo *Visifire* - Indicador circular

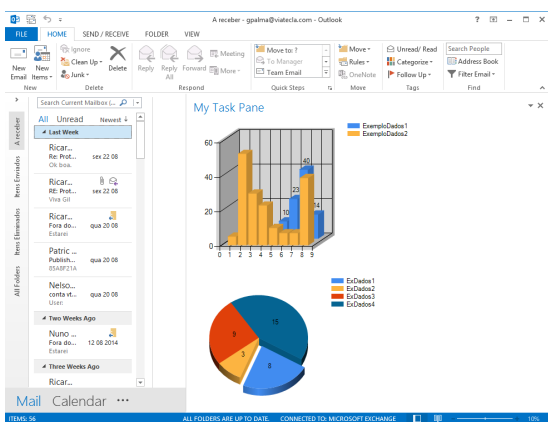


Figura 3.5: Protótipo *Plugin Outlook* - Gráficos Nativos (Barras e Seções)

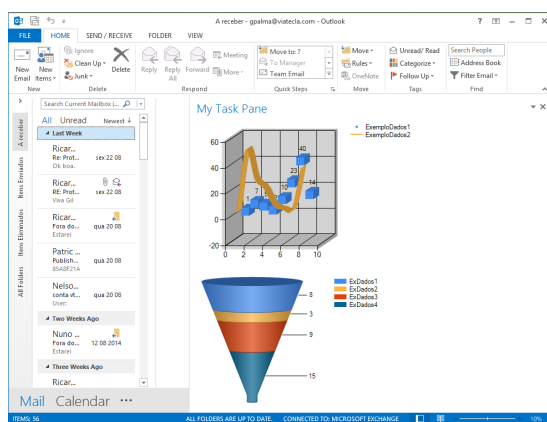


Figura 3.6: Protótipo *Plugin Outlook* - Gráficos Nativos (Linhas, Pontos e Funil)



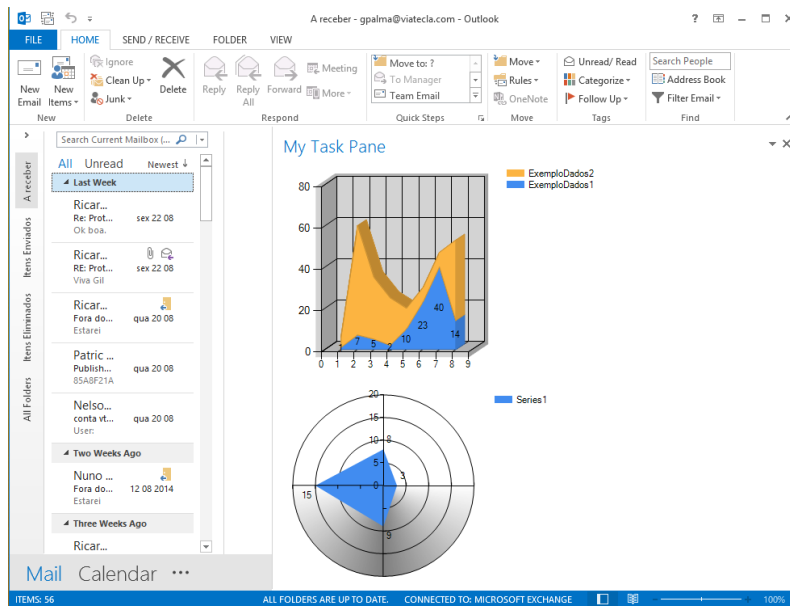


Figura 3.7: Protótipo *Plugin Outlook* - Gráficos Nativos (Área e Teia)

### 3.3.2 Gráficos Web

É possível apresentar elementos *HTML* complexos na janela do *Outlook*, como se pode ver na figura 3.8. Apesar de a página *web* apresentada não fazer parte do trabalho realizado, o método de a apresentar requereu alguma aprendizagem e trabalho. Esta abordagem não tem as limitações da alternativa nativa, o que torna possível não só ter muitos mais tipos de gráficos, incluindo tipos inovadores e especializados para o produto final, como também possibilita ambientes interativos e dinâmicos onde os gráficos possam ter utilidade e relevância máxima. No entanto, para que esta solução seja viável é necessário que exista uma conexão à *Internet* ou alguma outra forma de apresentar páginas *web*. De qualquer modo, devido aos limites dos *Custom Task Panes*, apenas páginas construídas propositadamente para este ambiente ou páginas com um desenho muito flexível não sofrerão ruturas na sua disposição. À medida que o *Web Browser* tenta adaptar o conteúdo ao espaço que lhe é dado, o gráfico resultante pode ficar deformado e difícil de analisar.

## 3.4 Protótipos Multiplataforma

Dado o estado de desenvolvimento das plataformas aquando da construção do protótipo, apenas o *PhoneGap* e o *Appcelerator Titanium* foram considerados. Na maioria dos testes feitos foi utilizado o mesmo gráfico de barras para mais facilmente descobrir as diferenças de comportamento e eficiência entre as duas plataformas selecionadas. Refira-se que esta escolha não representa de forma alguma a variedade de gráficos disponível e alguns exemplos adicionais são apresentados para dar a conhecer algumas das muitas alternativas.

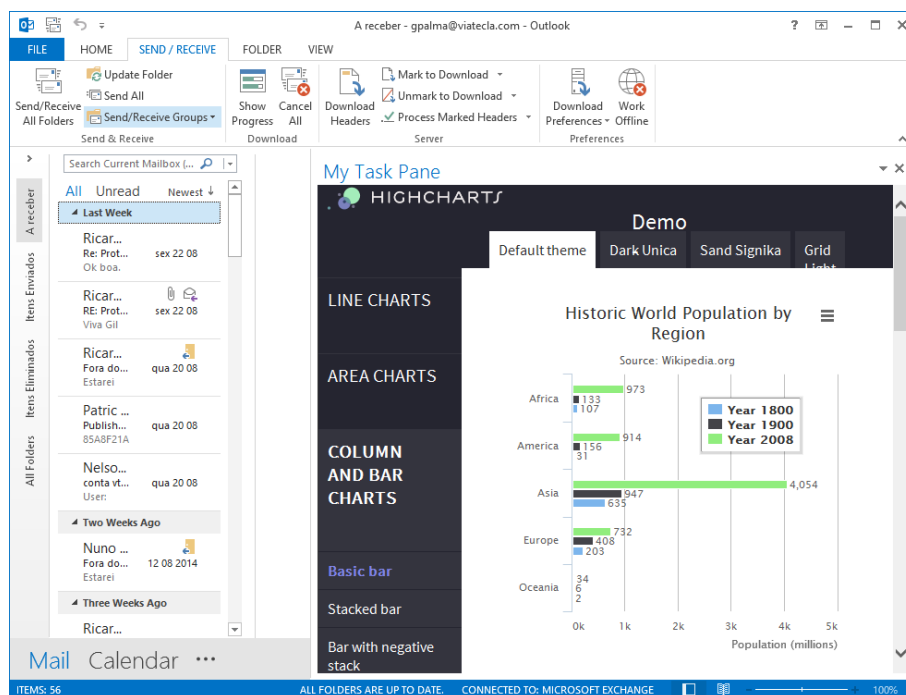


Figura 3.8: Protótipo *Plugin Outlook* - Apresentação de uma página *web* inteira

Cada protótipo foi testado em emuladores *Android* e *Windows Phone*. Apesar de não terem sido feitos testes em *iOS*, foi possível ficar a conhecer as diferenças principais entre os dois *frameworks*. As aplicações foram instaladas num dispositivo *Android* facilmente, no entanto a falta de um dispositivo *Windows Phone* permitiu apenas os testes no emulador e a falta de acesso consistente a um ambiente de desenvolvimento *macOS* levou à falta de testes em emuladores e dispositivos *iOS*.

### 3.4.1 Protótipo *PhoneGap*

Para testar as funcionalidades deste *framework* foi criada uma aplicação básica, semelhante a um “*Hello World*”, que pôde ser testada imediatamente nos diferentes emuladores ou dispositivos correspondentes às plataformas pretendidas. Foi posteriormente adicionado um gráfico exemplificativo grátis da *Highcharts* [88].

A figura 3.9 mostra o gráfico exemplificativo num emulador de *Android* versão 3.2 e a figura 3.10 mostra o mesmo gráfico num emulador de *Windows Phone* 8. Ambos foram criados a partir do mesmo código e ambos beneficiam de níveis de qualidade semelhantes em termos de animações e controlos. Ao rodar o *display* dos emuladores, equivalente a rodar um dispositivo fixo 90 graus, os gráficos também rodam. No emulador *Android*, ambas as dimensões do gráfico ajustam-se ao tamanho do ecrã. O mesmo não se verifica no emulador de *Windows Phone* onde apenas uma dimensão se adapta, resultando num gráfico mais difícil de ler.

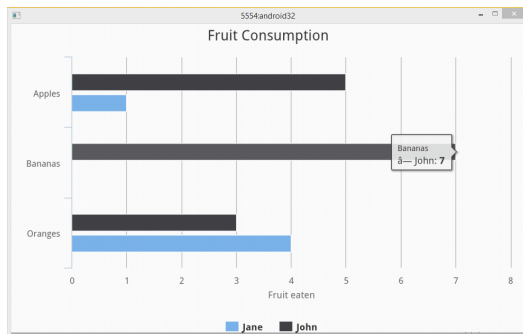


Figura 3.9: Protótipo *PhoneGap* - Android

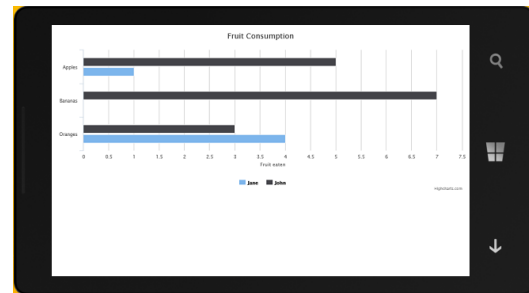


Figura 3.10: Protótipo *PhoneGap* - Windows Phone

### 3.4.2 Protótipo *Appcelerator Titanium*

A criação de projetos com *Titanium* é feita principalmente através do IDE da *Appcelerator*, o *Titanium Studio*. Através do *Studio* é possível gerir projetos, dispositivos e emuladores. Num ambiente *Windows*, as únicas alternativas para emuladores diretamente disponíveis são *Android* e *Blackberry*. Um emulador de *Windows Phone* também pode ser usado mas são necessários passos adicionais para que fique disponível. Apresentam-se de seguida os resultados obtidos a partir da execução do protótipo em *Android* e *Windows Phone*.

Foi desenvolvida uma aplicação muito simples com o mesmo gráfico usado com o *PhoneGap*. A figura 3.11 mostra como, apesar de a aplicação não se redesenhar automaticamente para melhor apresentar o gráfico, é possível ampliar ou reduzir o seu tamanho e reposicioná-lo para visualizar corretamente informação. A figura 3.12 mostra o resultado da execução da aplicação no emulador de *Windows Phone*. O conteúdo é o mesmo, automaticamente formatado para se adaptar ao dispositivo, mas parece faltar alguma flexibilidade já que esta não consegue rodar quando é simulada a rotação do dispositivo.

Os gráficos aqui apresentados foram desenhados em *webviews* por oposição a serem desenhados nativamente, o que subverte o propósito do *Titanium* já que é possível desenhar aplicações nativamente com a biblioteca *Titanium* para gráficos, chamada *charts* [118]. No entanto, a exclusividade desta biblioteca para *iOS* impossibilitou a criação de um exemplo de teste. Para tarefas como a inserção de elementos comuns, como botões e caixas de texto, esta plataforma permite programar em *Javascript* puro sem restrições de plataformas ou adição de módulos externos.

## 3.5 Resumo das experiências

Utilizadores de *email* contam com cada vez mais ferramentas para medir desempenhos e/ou para diminuir o esforço e tempo requeridos no processamento das suas mensagens. Um utilizador pode dar mais importância à eficiência e usar sempre o mesmo dispositivo ou alternar entre vários durante um único dia devido a requisitos de mobilidade. Depois



Figura 3.11: Protótipo *Appcelerator - Android*

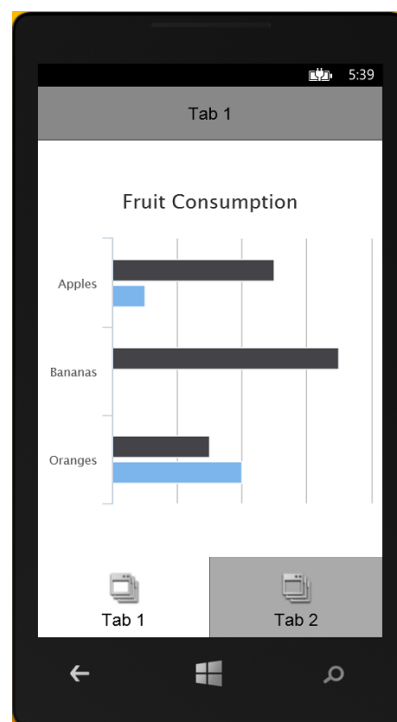


Figura 3.12: Protótipo *Appcelerator - Windows Phone*

de terminadas as experiências, reduziu-se o número de abordagens de desenvolvimento interessantes a apenas duas: aplicações para dispositivos móveis e extensões para o *Outlook*. Pretende-se com esta secção agrupar os principais resultados e conclusões obtidas durante as experiências e determinar quais as melhores soluções disponíveis de momento para desenvolver o *SMART Mail*.

### 3.5.1 Aplicações para dispositivos móveis

Não existe uma solução única para todas as situações e requisitos de desenvolvimento já que cada abordagem apresenta vantagens e desvantagens. Alta qualidade traz normalmente grandes custos e certas restrições. Sacrificando alguma dessa qualidade é possível diminuir os custos ou aumentar o número de dispositivos alcançados.

#### Aplicações Nativas

Num extremo do espectro de soluções estão as aplicações nativas. O desenvolvimento nativo para *iOS*, *Android* ou *Windows Phone* garante um melhor desempenho e qualidade de utilização quando comparado com outras abordagens mas requer o desenvolvimento de uma aplicação diferente, utilizando recursos específicos, para cada plataforma já que todas usam linguagens de programação diferentes. Atualmente, desenvolver nativamente para

cada plataforma continua a dar os melhores resultados mas torna-se dispendioso, tanto em termos de tempo como de dinheiro. Caso seja preferível desenvolver nativamente, a base de utilizadores de *Android* é significativamente maior do que a base de utilizadores de qualquer outra plataforma. No entanto, utilizadores com maior capacidade financeira, e consequentemente maior disponibilidade para gastar dinheiro em aplicações, tendem a preferir produtos da *Apple*, tal como *iPhones* e *iPads*. Proporcionalmente mais utilizadores de dispositivos *Apple* poderão preferir gerir e monitorizar os seus *emails* através dos seus dispositivos móveis do que utilizadores de dispositivos *Android* [120][121][122][123].

### Aplicações Web

Do outro lado do espectro, as aplicações *web* permitem desenvolver fácil e rapidamente uma única aplicação que pode ser utilizada em virtualmente qualquer dispositivo moderno, móvel ou fixo. Desta forma, uma aplicação perde alguma qualidade (já que vai executar a partir de um explorador de *Internet*, o que por si só já consome alguns recursos), não vai ter acesso a funcionalidades nativas e não vai oferecer uma experiência especializada para cada plataforma. O resultado deste tipo de desenvolvimento é uma aplicação de baixo custo ideal para oferecer serviços simples a grandes grupos de pessoas.

### Aplicações Híbridas

A ocupar o centro e grande parte do espectro de soluções estão as aplicações híbridas. Aplicações neste grupo contam com custos reduzidos de desenvolvimento, em comparação com o desenvolvimento nativo, e acesso a funcionalidades nativas, algo inacessível a aplicações *web*, beneficiando de vantagens de ambos os extremos do espectro.

Não existe uma única forma de criar aplicações híbridas já que o método de desenvolvimento e a proporção de código *web* e nativo podem variar. Quanto mais código nativo uma aplicação híbrida necessitar para uma plataforma, mais código terá de ser reescrito quando começar o desenvolvimento para outra plataforma. Quantas mais funções nativas forem incluídas no desenho, mais trabalho irá ser necessário para cada plataforma alvo. Quando possível podem-se substituir funções nativas por alternativas independentes de plataforma de modo a reduzir esforço no futuro, mas estas substituições têm de ser muito bem estudadas para não fazer sacrifícios excessivos de qualidade.

Apesar de todas as vantagens, esta não é a escolha necessariamente superior em qualquer situação. Devido à sua infância, existe relativamente pouca documentação e experiência por parte de programadores em geral, quando comparado com o volume de documentação e conhecimento que existe para aplicações *web* e nativas. A performance de aplicações híbridas pode ficar muito aquém do pretendido se contiverem processamento intensivo de gráficos e o código não for suficientemente eficiente. Além disso, todas as plataformas de desenvolvimento híbrido, por muito ou pouco versáteis que sejam, estão limitadas pelas restrições que as marcas *Windows* e a *Apple* impõem. Concretamente, é necessário um

ambiente *Windows* para criação de aplicações para *Windows Phone* e um ambiente *macOS* para o desenvolvimento de aplicações *iOS*. Apesar disto, o desenvolvimento híbrido tem imenso potencial, conta com incrementalmente mais casos de sucesso e poderá tornar-se o método de desenvolvimento mais comum no futuro se as linguagens associadas forem continuamente desenvolvidas, expandidas e otimizadas.

Atualmente, o *PhoneGap* [97], o *Titanium* [99] e o *Xamarin* [98] são alguns dos *frameworks* mais populares para desenvolvimento multi-plataforma. Cada um já permitiu criar milhares de aplicações até hoje, sendo até utilizados por grandes empresas como a *Logitech* [124] e a *RE/MAX* [125]. Ao utilizarem linguagens familiares, estes *frameworks* podem ser facilmente usados por equipas com experiência prévia em desenvolvimento de outros projetos ou suavizar a curva de aprendizagem para quem não tem muita experiência. A opção correta depende do projeto em questão e da experiência da equipa de desenvolvimento.

Cada *framework* tem os pontos fortes e fracos, evidenciados pelo estudo feito sobre estes. Uma análise comparativa é importante para determinar quais as situações onde é mais vantajoso utilizar cada um deles.

O *Titanium* e o *Xamarin* são mais apropriados para aplicações graficamente mais intensas, executando-as de um modo mais estável do que o *PhoneGap*. O interpretador *Javascript*, no caso do *Titanium*, apesar de tornar o início da execução mais vagaroso, garante uma melhor performance a médio e longo prazo já que o código nativo gerado vai executar melhor do que qualquer código *web* desenvolvido com *PhoneGap*. Mas estas escolhas poderão ser excessivas para aplicações mais simples, sem grandes requisitos gráficos ou computacionais, que beneficiariam de um desempenho igual ou até superior com o *PhoneGap*, devido à leveza deste *framework* relativamente a outros.

A instalação do *Appcelerator* inclui o *Titanium Studio*, um *IDE* baseado no *Eclipse*, para gerir todos os aspetos de um projeto [126]. O *Titanium Studio* instala automaticamente *SDK* necessários mas no caso de colisão de versões a configuração pode ser difícil. A partir daí o *framework* está pronto a ser utilizado. Depois de ter o ambiente de desenvolvimento pronto, a única barreira significativa, para além de aprender a linguagem caso seja necessário, é aprender a utilizar os componentes do *Titanium* necessários para ter aplicações a funcionar em cada uma das plataformas alvo.

O *PhoneGap* não é utilizável imediatamente depois de instalado já que precisa de *SDK* instalados manualmente para cada dispositivo alvo. Existe uma interface gráfica disponível na página do *framework* para quem a desejar [127] mas qualquer ferramenta de edição de texto é permitida e compatível. Testes de aplicações desenhadas deste modo têm a vantagem de poderem ser feitos diretamente no explorador de *Internet* além de em emuladores e dispositivos reais como com as outras duas abordagens.

O *Xamarin* pode ser utilizado através do seu *IDE* dedicado ou a partir do *Visual Studio* em forma de *plugin*, o que dá algum conforto a programadores que tenham experiência com estes elementos. A instalação dos *SDK*, ou de quaisquer outros componentes, está incluída no processo de instalação caso seja detetado que faltam elementos necessários para começar

o desenvolvimento.

O *Xamarin* oferece vários produtos mas aquele relevante para este projeto é o *Xamarin.Forms* [126] que permite criar aplicações para várias plataformas com o mesmo código base. Desta forma é possível alcançar dispositivos com *iOS*, *Android* e *Windows Phone*. O *Appcelerator* permite alcançar as mesmas plataformas que o *Xamarin* e ainda permite criar aplicações web que poderão ser usadas em qualquer dispositivo com um explorador de *Internet*. O *PhoneGap* é compatível com ainda mais plataformas do que as duas plataformas anteriores [128], permitindo criar aplicações também para plataformas não-móveis nativamente e para *Blackberry*. A compatibilidade com cada plataforma é limitada às funcionalidades nativas que cada *framework* disponibiliza, o que pode levar a que faltem funcionalidades críticas para um projeto. O tempo necessário para aumentar a compatibilidade com cada plataforma é algo elevado, o que limita o alcance e o potencial de aplicações criadas deste modo.

Cada *framework* oferece barreiras artificiais ao desenvolvimento, como falhas na documentação ou funcionalidades importantes ainda não desenvolvidas, que poderão ser eliminadas à medida que os correspondentes *frameworks* evoluem. Por enquanto, o *Titanium* e o *Xamarin* parecem ser as melhores escolhas para aplicações graficamente mais intensivas que necessitem de muito poder de processamento, sendo a principal diferença entre ambas as linguagens utilizadas. Contudo, quando também considerados os custos e compatibilidades de cada *framework*, o *PhoneGap* aparenta ser a solução que permite alcançar mais dispositivos com menos despesas. Há que considerar também que a maioria dos utilizadores de dispositivos móveis utiliza *iOS* e *Android*, logo este alcance adicional pode não fazer uma diferença significativa no número de utilizadores de uma aplicação.

### 3.5.2 Outlook

Apesar de cada vez mais utilizadores consultarem os seus *emails* em plataformas móveis, aplicações *desktop* de *email* continuam a ser usadas, sendo o *Outlook* um dos mais comuns [129]. Esta ferramenta oferece um alto nível de conforto e poder aos seus utilizadores, permitindo que estes completem tarefas mais complexas ou processem grandes volumes de mensagens.

### Extensões

Existem diferentes tipos de extensões, tal como o *PoliteMail* [19] que permite criar, gerir e monitorizar campanhas de *email marketing*, ou o *EZDetach* [130] que permite transferir anexos em grandes quantidades por oposição a ter de seleccionar cada um individualmente. Muitíssimos outros existem e a capacidade de os acrescentar e combinar no *Outlook* agrada a utilizadores suficientes para que esta ferramenta continue relevante. A criação de extensões é simples. Com o *Visual Studio* é possível criar uma extensão simples através de um misto de escrita de código e programação visual. Imediatamente após a inserção da ex-



tensão é disponibilizado um conjunto de elementos que podem ser adicionados para poder ter resultados depressa. Estes podem não ser os elementos finais desejados mas, dada a sua facilidade de configuração, estes elementos podem ser configurados mais tarde para satisfazer os requisitos.

### Compatibilidade

Ao utilizar a tecnologia de apresentação de gráficos do *Microsoft Office Word*, o *Outlook* tem alguns problemas em lidar com funcionalidades e técnicas modernas comuns com **HTML** e **CSS** [131]. Isto faz com que *emails* enviados para utilizadores de *Outlook* tenham de ser desenhados com essas limitações em mente. Sendo uma ferramenta popular já há bastante tempo, muitas organizações contam com a sua utilização (interna ou externamente) o que aumenta a viabilidade da utilização continuada.

### Observações gerais

O *Outlook* já está estabelecido há muitos anos e muitos utilizadores (especialmente o utilizador empresarial) preferem-no a alternativas, mesmo que tenham melhor performance ou mais funcionalidades. Apesar de a utilização de aplicações de *email* para *desktop* ter vindo a diminuir ao longo dos anos, o *Outlook* continua com uma base de utilizadores significativa. Este facto não pode ser ignorado mas não existe nenhuma garantia que os mesmos utilizadores continuem a preferir o *Outlook* quando existem cada vez mais opções interessantes.

## 3.6 Conclusões Finais das Experiências

O desenvolvimento do *SMART Mail* pode seguir diversos percursos relativamente à sua componente de interface. Aplicações móveis são responsáveis pela maioria dos *emails* abertos hoje em dia, levando à conclusão óbvia de disponibilizar o *SMART Mail* a dispositivos móveis. Escolher *Outlook* tem as suas vantagens também já que usar um dispositivo fixo permite trabalhar melhor com grandes quantidades de *emails*. No entanto, podem-se obter resultados semelhantes com uma página de *Internet* moderna, o que reduz os custos e ainda permite ser utilizado em qualquer dispositivo com um explorador de *Internet*, incluindo dispositivos móveis. Melhor ainda, uma aplicação híbrida consegue alcançar muitos dispositivos diferentes e ainda oferecer uma experiência algo adaptada a cada um deles. Escolher desenvolvimento nativo significa aumentar os custos, o que pode não ser facilmente justificado se a diferença de qualidade não for perceptível.

Com o avanço do desenvolvimento híbrido, este poder-se-á tornar o principal meio de desenvolvimento no futuro. A possibilidade de alcançar todos os utilizadores de plataformas móveis e fixas com aplicações de alta qualidade, desenvolvidas em relativamente pouco



tempo, vai continuar a impelir o estudo e evolução de *frameworks* de desenvolvimento multi-plataforma. Mas, de momento, aplicações nativas continuam a ser a melhor opção para criação de aplicações de alta qualidade e performance caso os custos não sejam proibitivos ou excessivos. Caso o foco da equipa de desenvolvimento não esteja nesses dois fatores, aplicações híbridas e *web* tornam-se extremamente aliciantes, dependendo das funcionalidades que se desejam implementar. Com *PhoneGap* ou *Appcelerator* é possível desenvolver para exploradores de *Internet* e dispositivos móveis simultaneamente.

A longo prazo, o desenvolvimento de aplicações móveis híbridas parece ser a solução com mais potencial. Contudo, esta abordagem oferece alguns riscos consideráveis provenientes da infância do meio, o que torna esta opção algo ideal para desenvolver no futuro. Uma extensão *Outlook* permite criar interfaces complexas e poderosas, usa tecnologia bem conhecida e conta com uma base de utilizadores que procura ferramentas *desktop* poderosas. Com estes factos, a escolha de uma extensão *Outlook* faz todo o sentido.



# Capítulo 4

## Aplicação Desenvolvida

Este capítulo apresenta a aplicação desenvolvida no âmbito desta dissertação. Contido neste capítulo está a apresentação dos conceitos, o desenho da arquitetura do produto, a análise de requisitos e a apresentação da aplicação.

### 4.1 Conceitos

A primeira, e atual, iteração do produto é um *plugin* para o *Microsoft Outlook*. A infraestrutura foi construída com **C#** mas toda a interface e componentes gráficos têm como base tecnologias *web*, nomeadamente **HTML**, *Javascript* e **CSS**. Neste capítulo são introduzidos com detalhe todos os conceitos e metodologias que estão na base para entender o percurso e os resultados do trabalho.

Os conceitos principais a conhecer são Infraestrutura, *Ribbon*, *Sidebar*, Contacto, Domínio, Organização, *Dashboard*, .NET, *Visual Studio*, **C#**, *webserver* portátil, **HTML**, **CSS** e *Javascript*. Adicionalmente, são dados a conhecer termos auxiliares necessários para entender estes mesmos conceitos centrais.

O termo Infraestrutura refere-se ao conjunto de conceitos base do *plugin* e *web server* portátil, dois componentes que utilizam a linguagem de programação **C#**.

#### 4.1.1 Conceitos base

*Ribbon* refere-se a uma área no topo do *Microsoft Outlook* e de outros produtos do *Office* onde se encontram muitos controlos nativos do programa, geralmente localizado no topo do

ecrã. Este elemento faz parte do *Outlook* e não foi criado com os outros conceitos base mas é importante mencioná-lo. Apesar de os controlos pré-existentes do *Outlook* não poderem ser manipulados, *plugins*, ou extensões, instalados poderão acrescentar novos elementos apresentados de forma semelhante aos pré-existentes.

Um Contacto, no contexto deste trabalho, tem como base a noção de contacto de *Outlook*, contendo no mínimo a mesma informação mas também quaisquer outros dados que possam ser relevantes para indivíduos num ambiente empresarial. O *Outlook* já associa alguma informação a cada endereço de *email*, tal como o nome, a organização em que trabalha e o seu cargo, números de telefone e morada. Um Contacto neste *plugin* tem toda esta informação e mais, tal como o seu estado de atividade, frequência de comunicação e importância para o utilizador.

Um Domínio é um grupo formado dinamicamente com Contactos cujos endereços de *email* partilham semelhanças, especificamente o domínio do endereço de *email*. À medida que *emails* são trocados com progressivamente mais Contactos, o *plugin* irá associá-los aos mesmos Domínios com base nos seus endereços. Contudo, um Domínio não é necessariamente uma organização, dado que uma organização pode corresponder a vários Domínios diferentes. Por exemplo, a *Microsoft*, sendo uma organização, é diferente das noções de *microsoft.pt* e *microsoft.com*, dois Domínios diferentes. Os endereços “joao@microsoft.com” e “maria@microsoft.com” fariam parte do mesmo domínio mas exemplos como “pedro@gmail.com” ou “susana@outlook.com” não estariam incluídos.

A *Sidebar* é o elemento visual primário da extensão. Uma vez instalada a extensão, uma barra vertical surge do lado direito do cliente *Outlook*. Nesta barra são apresentados controlos que maximizam o conforto e facilidade de utilização do *Outlook*, tal como ícones de Contactos e Domínios relevantes ao *email* selecionado.

O *Dashboard* é uma janela externa à interface principal acessível através de um botão na *Ribbon*. Nesta janela aparecem alguns controlos e gráficos mais complexos que destacam os *emails* mais importantes e permitem analisar em profundidade relações entre Contactos e Domínios. Apesar de não ser a componente mais imediatamente presente da extensão, é sem dúvida a mais poderosa para o utilizador.

#### 4.1.2 *Webserver* portátil

Para poder tirar proveito de todos os benefícios das tecnologias *web* sem uma conexão à *Internet*, unido ao produto está um *web server* portátil que permite obter o comportamento desejado das linguagens *web* em qualquer situação. Chamado *Griffin.Webserver* [132], coberto pela licença Apache 2.0 [133] e ainda na fase beta de desenvolvimento, este *web server* é uma solução simples e leve que permite hospedar as páginas *web* necessárias. Esta biblioteca não tem dependências a resolver e não requer qualquer ação da parte do utilizador final para configurar, sendo uma solução adequada para o produto.

### 4.1.3 Tecnologias *Web*

O conteúdo da *Sidebar* e do *Dashboard* foi desenvolvido com uma combinação de **HTML**, *Javascript* e **CSS**. É com estas tecnologias, e bibliotecas associadas, que páginas *web* são criadas. Tanto a *Sidebar* como o *Dashboard* são efetivamente páginas *web*, tornando a utilização destas tecnologias indispensável. Cada uma destas ferramentas tem a sua função distinta e todas são necessárias para que o resultado seja poderoso, apelativo e robusto.

Responsável pela estrutura base de uma página, o **HTML** permite definir o “esqueleto” do que será apresentado aos utilizadores. A quantidade de elementos, as suas posições gerais e os seus conteúdos são configurados com esta *markup language*. Somente com **HTML** é possível criar uma página simples e funcional, mas sem o auxílio de **CSS** e *Javascript* a página criada seria simplista e sem potencial para operações mais complexas.

O **CSS** por si só não permite criar conteúdo. Existe exclusivamente como complemento ao **HTML** e permite controlar a aparência da base desenvolvida. Atributos como tamanho e posição exata dos elementos **HTML** podem ser alterados, tal como as suas cores e ainda é possível acrescentar animações simples. Além disso, ao ter todas as definições visuais categorizadas e organizadas num único local, acrescentar e alterar parâmetros torna-se muito mais fácil à medida que o *plugin* cresce. Algumas das definições do domínio do **CSS** podem ser aplicadas diretamente com o **HTML** mas isto não é aconselhado na maior parte dos casos já que é contraditório às vantagens principais do **CSS**, levando a custos de manutenção muito mais elevados.

Com o *Javascript* torna-se possível criar comportamentos avançados para páginas *web*. Esta linguagem de programação, para além de contar com um enorme número de bibliotecas que expandem as suas capacidades, permite contornar os limites habituais do **HTML** e obter páginas *web* que se assemelham a aplicações instaláveis tradicionais. É uma linguagem com objetos diferente de muitas outras, especialmente devido à ausência de variáveis tipadas e à execução de código assíncrono.

### 4.1.4 Tecnologias *Microsoft*

O *framework* .NET oferece uma vasta e robusta coleção de classes, métodos e propriedades que simultaneamente permite desenvolver e expandir aplicações independentes além de criar extensões para outras previamente existentes. O próprio *framework* é extensível e conta com uma série de poderosas bibliotecas aliadas que preenchem algumas lacunas ou acrescentam ainda mais funcionalidades.

O *Visual Studio* é um IDE proprietário da *Microsoft* desenhado para trabalhar principalmente com a linguagem **C#** e o .NET *Framework*. No entanto, o *Visual Studio* é versátil e auxilia o desenvolvimento de projetos criados em diversas linguagens. Neste caso, o *Visual Studio* foi utilizado para criar a infraestrutura do programa e as páginas *web* hospedadas no *web server*.

C# é uma linguagem desenvolvida pela *Microsoft* e usada em conjunto com o *Framework .NET* para desenvolver projetos compatíveis com produtos *Microsoft*. Esta é uma linguagem poderosa, imperativa e orientada a objetos, com bastantes semelhanças ao *Java*.

Utilizando o *Visual Studio*, a linguagem de programação C# e o *Framework .NET* vão ser criados os dois elementos visuais principais do SMART Mail: a *Sidebar* e o *Dashboard*. Estes dois elementos são respetivamente uma *Custom Task Pane* e um *Windows Form*. Inserido em cada um destes estará um *webbrowser control* que acede ao *web server* portátil para poder utilizar todas as capacidades das tecnologias *web* sem necessitar de uma conexão à *Internet*. Este conjunto de elementos forma a base do *plugin*.

#### 4.1.5 Termos auxiliares

*Treemaps* são gráficos retangulares divididos em subáreas, também retangulares, em que cada uma representa uma porção dos dados relativamente à totalidade. Cada sub-retângulo tem as suas dimensões, incluindo a cor e os conteúdos, determinadas pelas propriedades da secção de dados respetiva. Os vários retângulos são depois ordenados pelas suas dimensões, tipicamente do canto superior esquerdo para o inferior direito. As principais vantagens deste tipo de gráficos são a sua natureza dinâmica, sendo muito fáceis de ler mesmo quando são atualizados frequentemente em tempo real, e a facilidade de criar *treemaps* secundários focados em subáreas de um gráfico base. Uma prática comum ao construir *treemaps* envolve permitir que utilizadores naveguem em profundidade pelos gráficos, tanto o principal como quaisquer outros secundários, ao selecionar sub-retângulos. Esta ação pode alterar o que é mostrado ao utilizador, nomeadamente, pode transformar o gráfico noutra diferente, apresentar dados em bruto (em formato de lista, por exemplo) ou criar um novo *treemap* apenas com o sub-conjunto de dados selecionados, permitindo uma exploração virtualmente infinita, apenas limitada pelos dados utilizados. Esta última opção é a mais interessante das três e uma das razões principais para utilizar *treemaps*. A figura 4.1 contem um ótimo exemplo da facilidade de leitura de um *treemap*.

No gráfico estão apresentados os dados relativos ao uso de energias renováveis de vários países em 2008 [134][135]. A área dedicada a cada país indica quantos milhões de *megawatt* por hora são gastos e a cor indica a diferença do valor atual em relação ao valor do ano passado, oferecendo uma noção imediata do estado e crescimento dos países apresentados. Países como a China, Estados Unidos da América e Canadá usam muitíssima energia renovável, comparativamente a outros países, e mostram um aumento relativamente ao passado. No entanto, países como a Rússia e a Índia têm vindo a usar menos energia renovável apesar de serem dos países que mais usam.

Existe uma série de métricas relativas a mensagens de correio eletrónico. Estas são o *Open Rate*, *Undeliverables*, *Opt-Out*, *Click-Through*, *Response Level* e *Interest Level*. *Open Rate* é a percentagem de mensagens abertas em relação a um conjunto de mensagens enviadas. *Undeliverables* é o termo usado para contabilizar as mensagens que nunca chegaram ao seu destino. A percentagem de pessoas que desistiu de um serviço corresponde à métrica

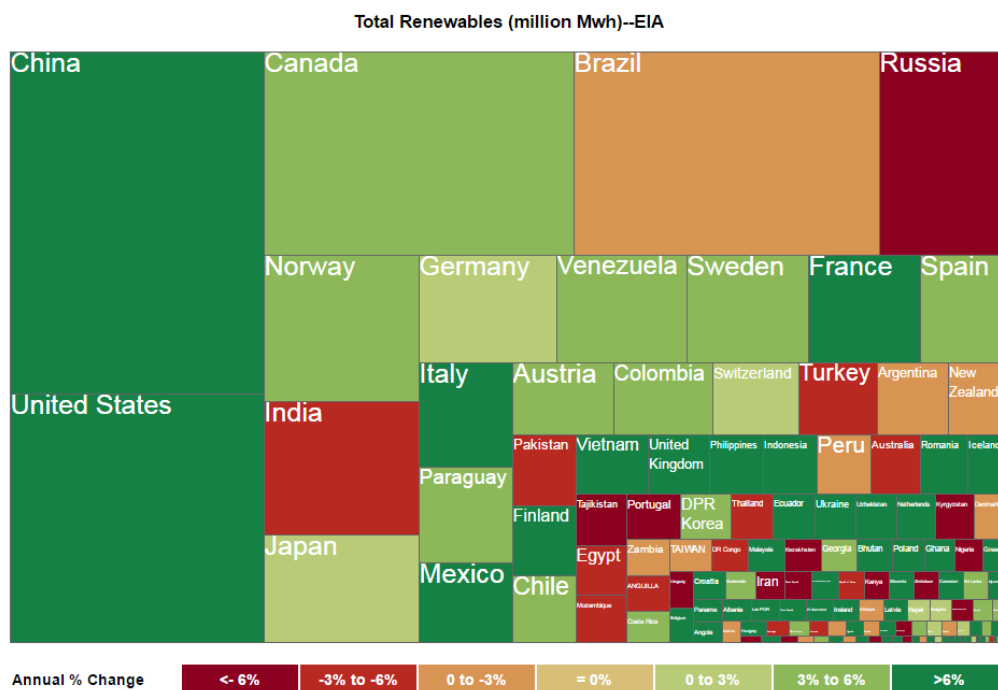


Figura 4.1: *Treemap* - Milhões de *megawatt* por hora de energias renováveis por país em 2008

*Opt-Out*. As estatísticas de *Click-Through*, *Response Level* e *Interest Level* correspondem, respetivamente, à quantidade de links pressionados numa mensagem, ao total de respostas enviadas por indivíduos inscritos num serviço e à percentagem de respostas relativas ao número de emails.

## 4.2 Requisitos do produto

Esta secção lista todos os requisitos definidos inicialmente, durante a fase de conceção do projeto. Estes foram criados antes das primeiras fases do desenvolvimento mas foram e serão atualizados conforme as lições aprendidas durante o período de trabalho com um esforço de nunca perder de vista os Pilares Concetuais. Cada requisito está classificado com o método *MoSCoW* e estes estão agrupados em duas secções: Requisitos Gerais e Requisitos Técnicos.

### 4.2.1 Pilares Concetuais

Para que um produto seja bem recebido, a equipa responsável tem de garantir que este seja relevante no meio onde se pretende que seja lançado, especialmente quando ainda se encontra nas fases de planeamento e conceção. Mas manter a relevância do produto depois da exposição inicial também é muito importante, já que o seu meio pode mudar,

o que significa que os seus requisitos podem mudar também para acomodar quaisquer alterações significativas. De forma a nunca perder de vista as ideias centrais e evitar perder propriedades ou funcionalidades centrais, equipas de desenvolvimento devem definir os objetivos gerais que o produto deve procurar atingir, independentemente de mudanças que sejam feitas ao longo do tempo. No caso do *SMART Mail*, estes pilares são os seguintes:

1. A componente principal da solução deve ser uma extensão do *Outlook*;
2. A solução nunca pode limitar ou bloquear quaisquer aspetos do *Outlook*, permitindo que o utilizador possa utilizar o *Outlook* como sempre utilizou;
3. Funcionalidades da solução que contenham funcionalidades do *Outlook* não podem pedir ao utilizador um maior esforço do que seria necessário sem a solução;
4. A solução deve considerar as necessidades de um utilizador corporativo antes das necessidades de qualquer outro tipo de utilizador.

#### 4.2.2 Requisitos Gerais

Os requisitos definidos nesta secção aplicam-se ao funcionamento e propriedades gerais do produto, sem contemplar detalhes concretos. Estes requisitos direcionam o desenvolvimento do produto de modo a que a sua versão final não seja diferente do que foi planeado inicialmente. Os requisitos definidos são os seguintes:

- A solução tem de melhorar a produtividade na utilização de *email* (Must);
- A solução tem de estar focada nas necessidades do utilizador corporativo (Must);
- A solução tem de assegurar a privacidade relativa a *emails* e contactos do utilizador (Must);
- A solução tem de manter toda a informação de *emails* e contactos no próprio cliente de *email* (Must);
- A solução não pode ser graficamente intrusiva (Must);
- A solução não pode incomodar o utilizador com processamento (Must);
- A solução tem de permitir unificar/agregar Contactos (Must);
  - A solução deve disponibilizar ferramentas gráficas que permitam a comparação, unificação e agregação de Contactos (Should);
- A solução tem de assegurar a persistência local de dados (originais e calculados) (Must);
- A solução deve assegurar a persistência em servidor de dados (calculados) para componentes exteriores a si própria (Must);



- Toda a informação relativa a Domínios é pública (Should):
  - utilizadores podem associar Contactos a Domínios existentes (Must);
  - utilizadores podem criar Domínios (com fluxo de aprovação) (Should);
  - utilizadores podem editar Domínios (com fluxo de aprovação) (Should);
  - a solução poderá obter os dados de Domínios através de serviços e/ou motores de busca, como o *LinkedIn* [136] e o *Google* [56] (Could);
- A solução deve minimizar os pedidos de confirmação/validação de informação (Should);
- A solução deve permitir agregar múltiplos Domínios (Should);
- A solução deve permitir atribuir múltiplos *emails* a um mesmo Contacto (Should);
- A solução deve permitir classificar Contactos como profissionais, pessoais (Should);
- A solução deve apresentar dados de natureza temporal sob a forma de linha cronológica (*timeline*) (Should);
- Gráficos interativos devem permitir escolher quais os dados a apresentar e como devem ser apresentados, oferecendo diferentes filtros e ordenações (Should);
- A solução deve ter suporte multilingue (Should);
- Deve ser criada uma plataforma *web* (*Web View*) que forneça toda a informação disponível a utilizadores, permitindo a consulta de dados mesmo em terminais que não tenham acesso à solução principal (Should);
- A solução deve dar a possibilidade de alterar a disposição dos elementos visuais, sobretudo no caso da *Sidebar*, com controlos de reposicionamento (por exemplo, com *drag & drop*) e/ou de visibilidade (com botões que ocultem ou revelem os vários elementos) (Could);
- A interface deverá ser modular e composta por blocos, para uma melhor evolução futura e melhor personalização (Could);

### Arquitetura e Tecnologia

- A solução tem de processar mensagens de *email* de dois modos (Must):
  - Em plano de fundo, mensagens são analisadas sequencialmente até toda a caixa de correio estar classificada;
  - Mensagens são classificadas individualmente a pedido do utilizador ou após certos eventos (tal como quando novas mensagens são recebidas);
- A solução não requer ligação à *Internet* (Must);
- A *Web View* requer ligação à *Internet* (Must);

- A solução deve associar a cada *email* um valor calculado representativo do seu valor para o utilizador (Must);
- A solução será materializada como uma extensão para o *Microsoft Outlook* (Must);
- A solução deve permitir a agregação de Contactos previamente existentes (Must):
  - Deve existir um controlo que invoque uma pesquisa de Contactos com dados semelhantes, sugerindo possíveis duplicados. Posteriormente deve surgir um painel para efetuar a união de Contactos (Should).
    - \* Deve existir uma ferramenta com opções avançadas que tornem a pesquisa de contactos conflituosos ou duplicados mais simples para o utilizador (Could);
  - Sugerir a adição de novos endereços importante aos contactos (Should);
  - Extrair dados, incluindo fotografia, de redes sociais (Should);
- O processamento de mensagens, tanto em plano de fundo como o processamento individual, não pode impedir a utilização do *Outlook* e mostrar o estado do progresso (Should);
- A extensão deverá ser desenvolvida primariamente para a versão 2013 do *Outlook* (Should);
- A *Web View* seguirá os princípios de *Responsive Design* (Should);
- A *Web View* será desenvolvida tendo em conta as plataformas seguintes (Should):
  - IE (versão 9) (Should);
  - Chrome (versão 35) (Should);
  - Firefox (versão 30) (Should);
  - Safari (versão 7) (Should);
- A capacidade de apresentar o texto da solução em várias línguas deve funcionar com dicionários (Should);
- Deverá ser possível obter dados adicionais de um contacto caso haja acesso a um servidor *Microsoft Exchange* (Should);
- Deverá ser possível extrair informação acerca do emissor de uma mensagem a partir da assinatura contida (caso exista) (Should):
  - Morada
  - Telefone/Extensão/Telemóvel
  - Endereço de páginas *web*
  - Nome
  - Cargo

- A visualização e manipulação de quaisquer dados adicionais de um Contacto deverão estar integrados no contexto da ficha do Contacto respetivo (Should);
- Associado a cada *email* deverá existir uma classificação qualitativa (Could);
- A solução pode obter informação sobre *emails* enviados, tal como o momento em que um *email* enviado é aberto, quantas pessoas o abrem de todos os recipientes (*open rate*), quantas pessoas seguem um *link* incluído no corpo da mensagem (*click rate*) (Could);

### Detalhes de Contacto

- A solução tem de apresentar a fotografia de Contactos seleccionados (Must);
- A solução tem de apresentar os logótipos dos Domínios associados a um Contacto (Must);
- A solução tem de apresentar graficamente as conversações em que o utilizador e os seus Contactos participem (Must);
- A solução tem de apresentar graficamente anexos trocados recentemente (Must);
- A solução tem de apresentar graficamente a data da primeira e última interação (Must);
- A solução tem de apresentar graficamente a rede de Contactos com que o utilizador troque *emails* geralmente ou a rede específica a cada mensagem;
- A solução deve ter painéis separados para detalhes de Contacto e de Domínio (Should);
- A solução deve apresentar graficamente métricas relativamente a Contactos individuais (Should):
  - Fluxo
  - Tempos médios de resposta
  - Gráfico de dispersão horária
  - *Timeline*

### *Dashboard*

- A solução tem de realçar *emails* prioritários. A prioridade de uma mensagem deve ser calculada a partir de vários fatores (Must):
  - Prioridade deve ser atribuída a mensagens marcadas como importantes no *Outlook* pelo utilizador (Should);
  - Prioridade deve ser atribuída com base no tempo de resposta de uma mensagem em comparação com o tempo médio de resposta, tanto em relação ao emissor da mensagem como em relação a todos os contactos registados (Should);
  - Prioridade deve ser atribuída com base na frequência de interação (Should);
  - Prioridade deve ser atribuída a mensagens recebidas em alturas do dia muito diferentes da média (Should);
  - Prioridade pode ser atribuída com base no corpo da mensagem (Could):
    - \* (Alternativa 1) Podem-se usar expressões regulares para identificar termos importantes;
    - \* (Alternativa 2) Pode-se usar a estatística TF-IDF para conhecer a frequência de termos do corpo do *email*, relativamente à própria mensagem e a todas as outras;
- A solução tem de apresentar graficamente os *emails* recebidos, especialmente os ainda não visualizados, agrupados e organizados pelos Domínios mais importantes (Must);
- A solução tem de apresentar graficamente Domínios ordenados com base na sua importância (Must):
  - Importância de Contactos dentro de um Domínio deve ser baseada no número de mensagens trocadas e tempos de resposta;
  - Importância dos Domínios deve ser baseada no número de mensagens trocadas e nos tempos de resposta;
  - Importância dos Domínios deve ser baseada no total de mensagens trocadas, detalhando mensagens recebidas e enviadas;
  - Importância dos Domínios deve ser baseada na média de tempo de resposta;
  - Importância dos Domínios deve ser baseada no número máximo de mensagens recebidas e enviadas num único dia;
  - Importância dos Domínios deve ser baseada no número de Contactos associados;
  - Importância dos Domínios deve ser baseada nas datas da primeira e última interação;
- A solução deve apresentar noções temporais, através de gráficos como histogramas, do início, fluxo ao longo do tempo e fim das interações com um Domínio, com ordenação por prioridade, ordenação alfabética ou ordenação temporal (Should);

- A solução deve utilizar gráficos *treemap* preenchidos com Domínios oferecendo a opção de obter sub gráficos com detalhes sobre os Contactos lá contidos (Should);
- A solução deve apresentar graficamente a importância de Domínios relativamente a outros através de gráficos de áreas, acompanhados de gráficos de secções para poder conhecer a importância de cada Contacto relativamente a outros no mesmo Domínio (Should);
- A solução deve apresentar métricas globais sobre a totalidade de Domínios e Contactos conhecidos (Should):
  - O total de mensagens, detalhando quantas são recebidas e enviadas;
  - O tempo médio de resposta;
  - A quantidade mínima e máxima de *emails* recebidos e enviados num único dia;
  - O total de Contactos conhecidos;
- A solução deve ser disponibilizada noutras plataformas, tal como num *website* (Should);

### 4.3 Desenho da arquitetura do produto

Os elementos principais da estrutura já foram mencionados por texto nesta secção mas a figura 4.2 apresenta o diagrama da arquitetura com toda a informação crítica. De seguida é apresentada a explicação dos vários elementos: caixas com cantos curvos representam eventos, tanto por parte do utilizador ou do *Outlook*; caixas com cantos angulares representam mecanismos lógicos; setas com pontas simples representam o envio de dados não discriminados; setas com pontas em losango representam ações resultantes de eventos ou mecanismos; setas picotadas representam dependências; elementos vermelhos estão ligados a ações do utilizador; elementos azul-claros fazem parte do *Outlook*; elementos amarelos correspondem ao *plugin*; elementos verdes correspondem ao *Griffin Webserver*; elementos azul-escuros correspondem à base de dados.

A aplicação está maioritariamente contida no *Outlook*, sendo a base de dados o único elemento da aplicação que não está. As duas formas de recolha de *emails* e o mecanismo de busca de dados do email estão contidos no código do *plugin* mas acedem à base de dados exterior. Os dois *webbrowsers* são *custom task panes* do *Outlook* inseridos através do *Visual Studio* para apresentar os dados relevantes. O seu funcionamento depende da execução do *Griffin Webserver*. A qualquer altura, o utilizador pode seleccionar um *email* existente para obter os dados relevantes, interagir com as componentes web para explorar os dados apresentados ou manipular os controlos do *plugin*, situados na *Ribbon*, para manipular as opções do produto.

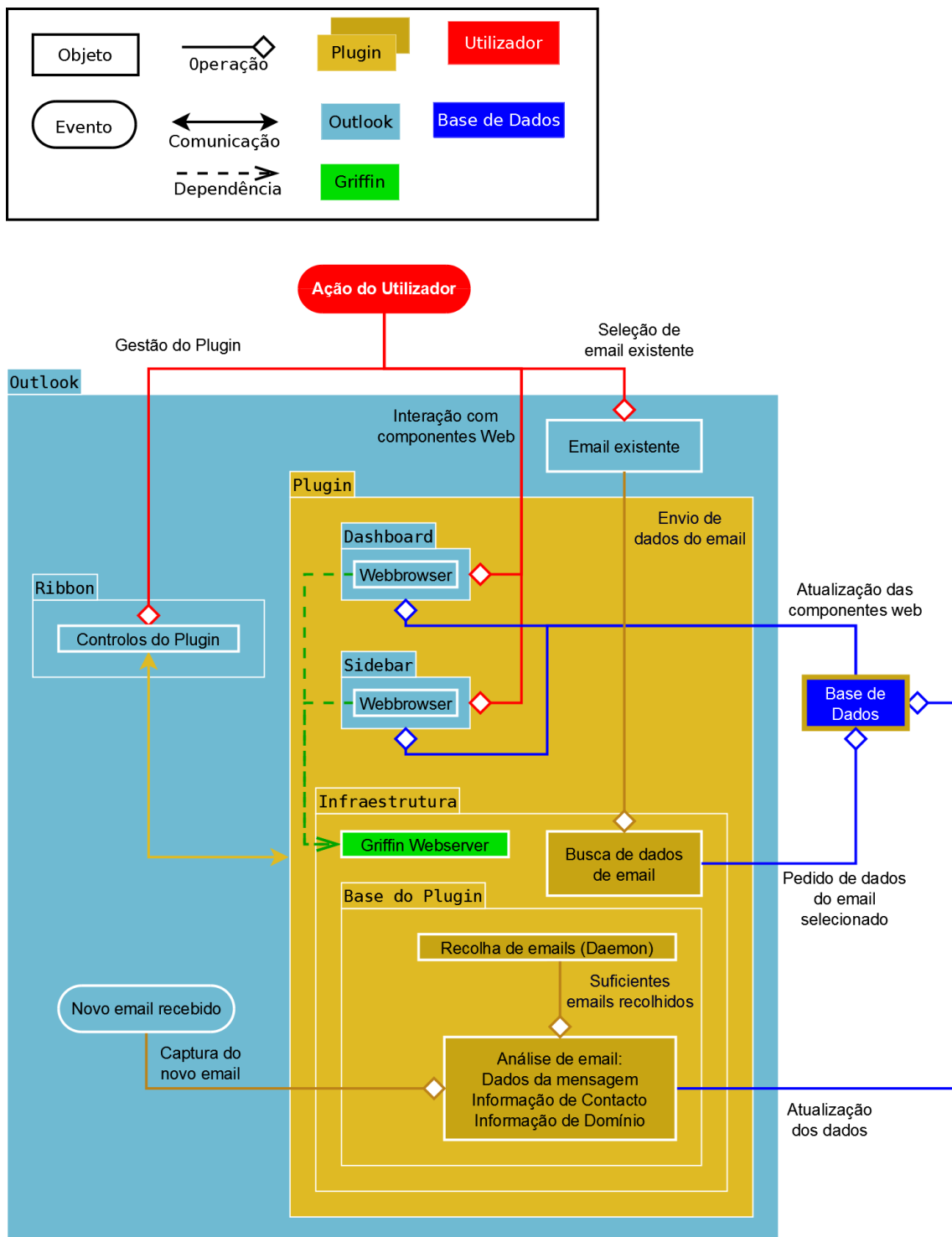


Figura 4.2: Desenho da estrutura do SMART Mail

## 4.4 Trabalho Desenvolvido

Devido ao tempo necessário para iterar as funcionalidades centrais, a Infraestrutura e a *Sidebar*, não foram desenvolvidos todos os aspetos planeados. Atualmente existe um produto desenvolvido, conhecido como *0Inbox*, e o estado deste está muito próximo do desenho original. Alguns dos requisitos originais do produto não foram concluídos e encontram-se listados como trabalho futuro. De seguida é apresentada a infraestrutura, a *Sidebar* e a utilização do *plugin*.

### 4.4.1 Infraestrutura

Na sua primeira versão, a *Infraestrutura* tem a capacidade de captar e analisar *emails* de duas formas. A primeira forma consiste em percorrer todas as mensagens na caixa de correio de um utilizador. Este processo é algo demorado, sendo necessárias várias horas para analisar uma caixa de correio com 300.000 *emails*. O tempo necessário melhorou ao longo do desenvolvimento e ainda pode ser melhorado através de otimização de tarefas, em particular a otimização do armazenamento da informação já que esta é a componente mais dispendiosa do processo. A segunda forma de captar *emails* funciona através do evento lançado pelo *Outlook* quando uma nova mensagem é recebida.

Independentemente da forma de captura, cada *email* captado é analisado para primeiro determinar se é válido ou não. Mensagens de erro e rascunhos gravados são exemplos de *emails* inválidos. *Emails* válidos são então examinados para obter informação de identificação única da mensagem, data e, o mais importante, contactos relacionados. Quando o utilizador pressiona num *email*, seleccionando-o, o programa toma conta e preenche a *Sidebar* com a informação relevante.

A comunicação entre a *Sidebar* (*HTML*) e a Infraestrutura (*C#*) é feita por camadas. Para que os dados usados na Infraestrutura sejam utilizáveis na *Sidebar*, e vice versa, algumas transformações são necessárias com certos tipos de dados. Números inteiros e *strings* podem ser transmitidos facilmente mas objetos e estruturas de dados requerem algumas adaptações. Camadas intermédias são mais genéricas e as camadas extremas são mais específicas, o que permite que cada camada se abstraia dos pormenores e possa ser menos complexa e mais fácil de manter.

Para a base de dados foi escolhido o *SQLite* [137] devido à sua simplicidade e leveza. Os tipos mais usados de bases de dados geralmente requerem um sistema de cliente e servidor, sendo mais apropriados para sistemas complexos em que a distribuição de dados por pedidos concorrentes é essencial. No caso deste *plugin* e de outras aplicações, em que pedidos concorrentes não vão ser um problema significativo, é mais vantajoso utilizar um sistema de base de dados que se concentre na portabilidade e simplicidade. Uma base de dados *SQLite* tem todas as capacidades necessárias para ter um armazenamento robusto, ágil e sem dependências, garantindo que funciona em qualquer situação e mesmo com poucos recursos.

Enquanto é armazenada informação de *emails* na base de dados, as componentes públicas dessa informação são transmitidas simultaneamente para serem armazenadas em servidores remotos. Estes vão armazenar toda a informação pública para que depois possa ser apresentada na *Web View*, intitulada como *Smart Catalog*, um *website* com um estilo de interface idêntico ao das componentes *web* do *plugin* de *Outlook*. Assim, um utilizador de *Outlook* pode aceder a certos dados mesmo que não tenha acesso a um dos seus dispositivos com o *Outlook* instalado.

#### 4.4.2 *Sidebar*

Utilizando **HTML** foi criada uma página *web*, com o conceito de *Responsive Design* em mente, adaptada ao formato retangular esperado para este módulo. Com o auxílio do *webserver* portátil, o *Outlook* passa a apresentar essa página. A Infraestrutura transfere a informação relevante para o código *Javascript* na página *web* alojada na *Custom Task Pane* que se manifesta numa alteração da *Sidebar* para representar toda essa informação.

Com a abertura do *Outlook*, a *Sidebar* apresenta a introdução, como visível na figura 4.3. A partir desta altura, a seleção de qualquer *email* por parte do utilizador substitui a introdução com a visão principal do *plugin*. A figura 4.4 exemplifica o que o utilizador vê depois de selecionar um *email*. Os primeiros dois painéis de informação apresentados são relativos ao emissor da mensagem e à mensagem propriamente dita. A relação do utilizador à mensagem, o tempo passado desde a sua receção e a quantidade de mensagens por ler provenientes do emissor em questão são apresentados no primeiro painel. No segundo painel está apresentado o sumário da informação do emissor, nomeadamente o seu nome, empresa, posição na empresa e estatísticas de comunicações anteriores entre este e o utilizador. Subsequentemente, o resto dos contactos envolvidos estão agrupados por Domínio (utilizando o domínio dos seus endereços de *email* respetivos), por relação à mensagem (dependendo se foram incluídos no campo de *To*, *CC* ou *BCC*) e finalmente por ordem alfabética. Desta forma, o utilizador fica imediatamente a saber quais as pessoas envolvidas e quais são as mais fortemente relacionadas.

Selecionar qualquer Contacto apresentado realça-o, permitindo ver e selecionar uma seta laranja que muda a visão do *plugin* para a visão detalhada de Contacto. Visível na figura 4.5 está esta visão preenchida com a informação de Contacto do emissor do *email* selecionado. É importante realçar que um Contacto, no contexto deste *plugin*, pode ter associados vários endereços de *email*, e todos os endereços do Contacto selecionado são aqui listados. Cada endereço pode ser classificado com “*tags*” de forma a ajudar o utilizador a distinguir os diferentes endereços. Toda esta classificação permite a um utilizador nunca perder a noção de com quem está a comunicar nem em que contexto o está a fazer, mesmo que participe em discussões muito diferentes com as mesmas pessoas. A partir da visão principal ou de Contacto, o utilizador pode selecionar o nome do Domínio para ser levado a uma visão semelhante à visão de Contacto com o logótipo deste (caso corresponda a uma organização que tenha logótipo) e todos os Contactos pertencentes a este Domínio.



Também visível na figura 4.4 estão dois módulos: o *Numbers* e o *Attachments*. Estes módulos são os primeiros dois elementos independentes desenvolvidos que visam expandir o conhecimento do utilizador relativamente à sua caixa de correio e especificamente ao Contacto realçado. O primeiro, *Numbers*, dá estatísticas relativas ao historial de comunicações entre o utilizador e o Contacto. O primeiro painel deste módulo dá ao utilizador a possibilidade de alterar o intervalo de tempo considerado no historial de forma a saber quantos *emails* foram trocados em diferentes períodos de tempo. O segundo módulo, *Attachments*, lista todos os anexos trocados entre o utilizador e o Contacto. Ordenados pela data do seu *email* correspondente, os anexos estão separados por quem os enviou. Selecionar um destes anexos abre um seletor de pastas para que o utilizador possa escolher um diretório onde criar uma cópia do mesmo.

#### 4.4.3 Utilização do *Plugin*

Quando o *Outlook* é aberto, o *plugin* inicia automaticamente a análise dos *emails*, como visível no lado direito da *Ribbon* da figura 4.3. No entanto, *emails* que já tenham sido analisados não o serão de novo graças à informação guardada na base de dados. Devido à forma como o *Outlook* pode ser configurado, algumas das pastas disponíveis podem estar armazenadas remotamente, o que requer uma verificação *online* com uma conta *Outlook* válida. De momento, se esta verificação falha, a análise dos *emails* não continua mas é algo que pode ser alterado.

A análise de *emails* não bloqueia a utilização do *Outlook* mas consome alguns recursos do computador. O equilíbrio entre o tempo necessário para analisar a caixa de correio e os recursos utilizados ainda não foi finalizado mas é algo que pode ser melhorado com o tempo de forma a nunca perturbar o utilizador. De qualquer modo, o *plugin* oferece um conjunto de opções ao utilizador para que este tenha sempre controlo sobre o estado do *Inbox*. Estas opções, visíveis na figura 4.6, permitem esconder a *Sidebar* (ou mostrá-la caso já tenha sido escondida), ativar ou desativar os processos em plano de fundo ou aceder a propriedades mais avançadas para configurar vários aspetos do *plugin*.

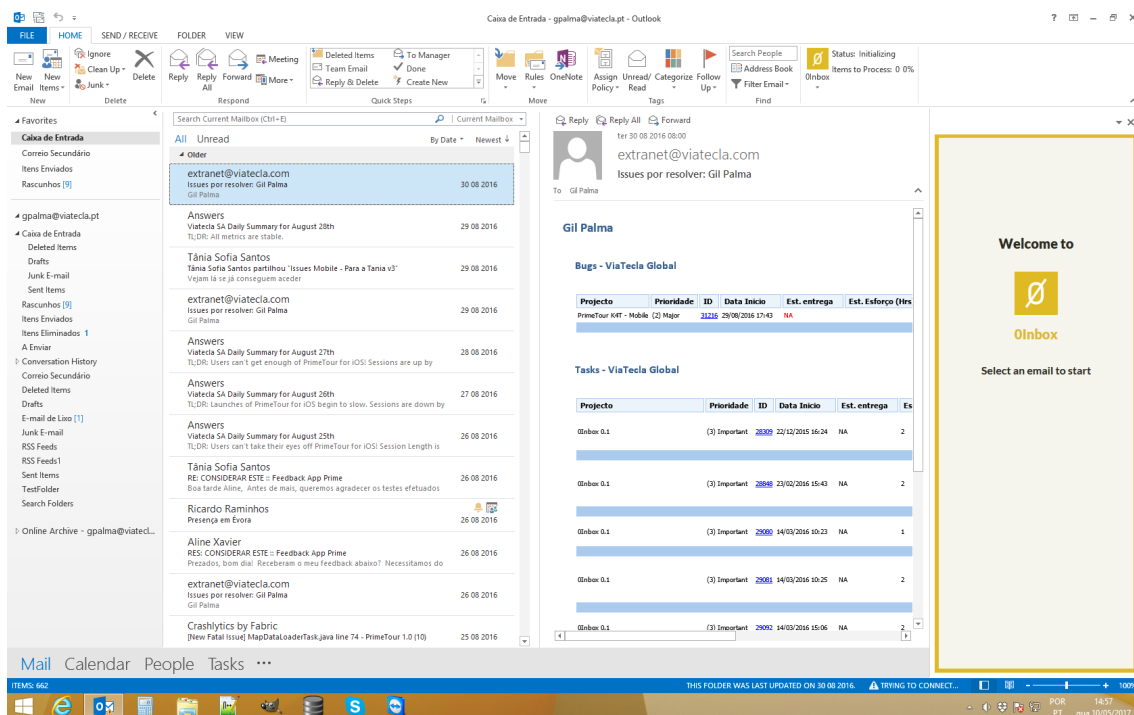


Figura 4.3: OInbox - Splash

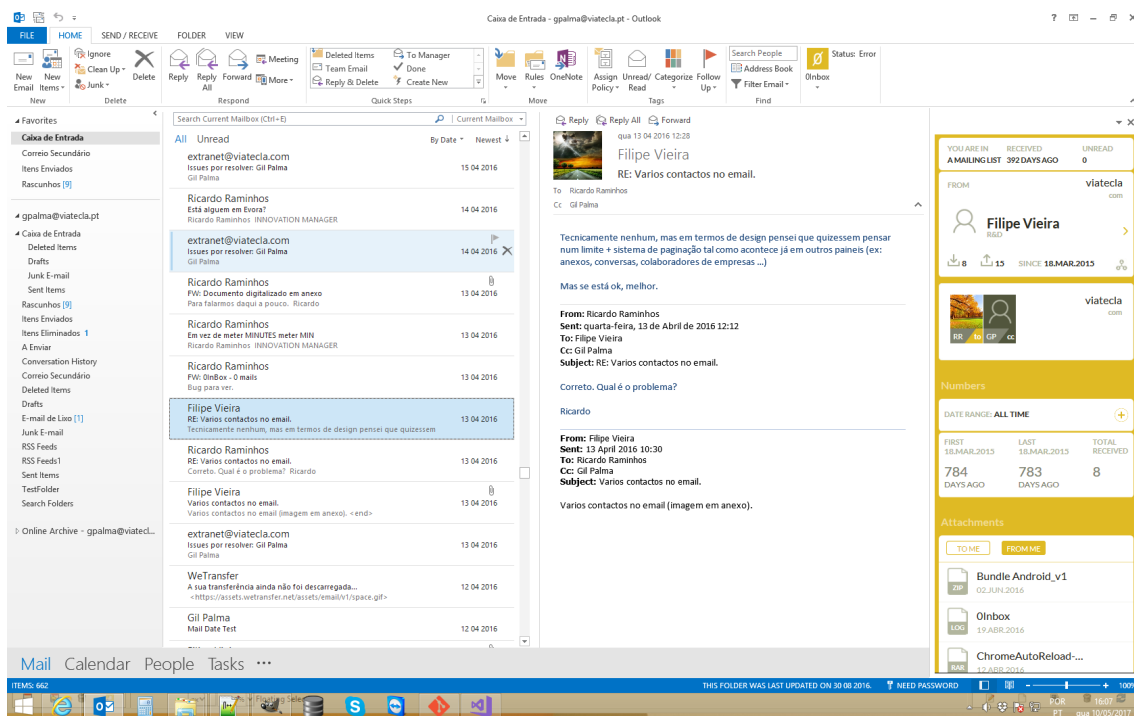


Figura 4.4: OInbox - Visão principal e Anexos

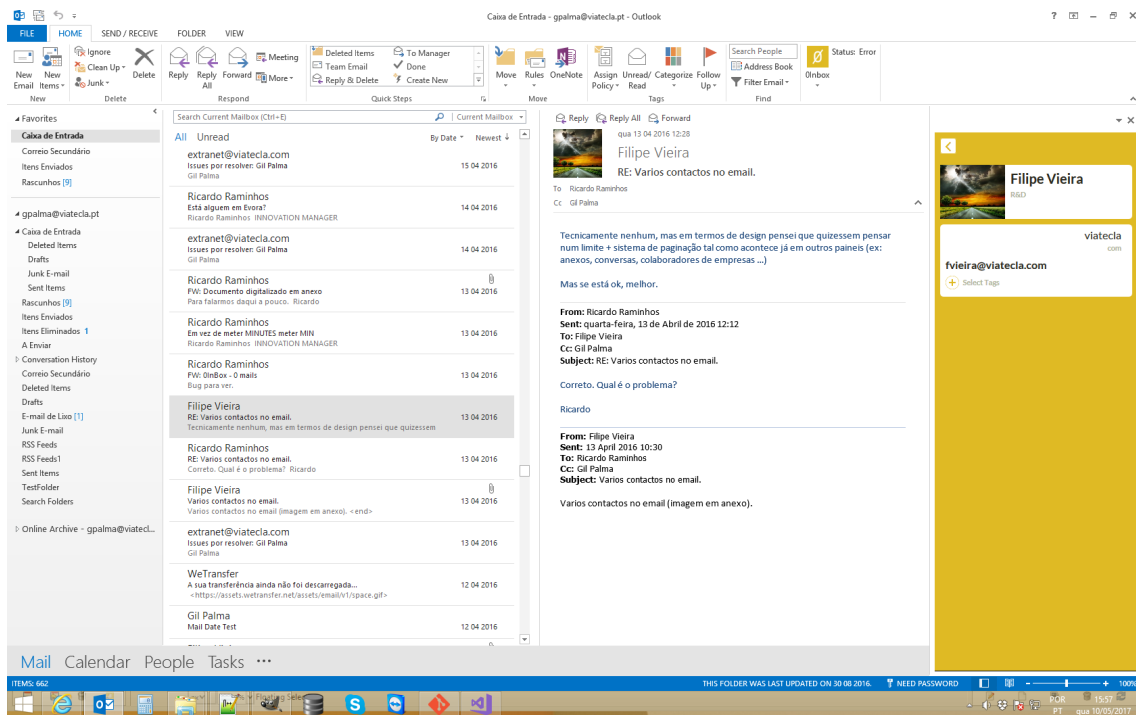


Figura 4.5: *0Inbox* - Contacto

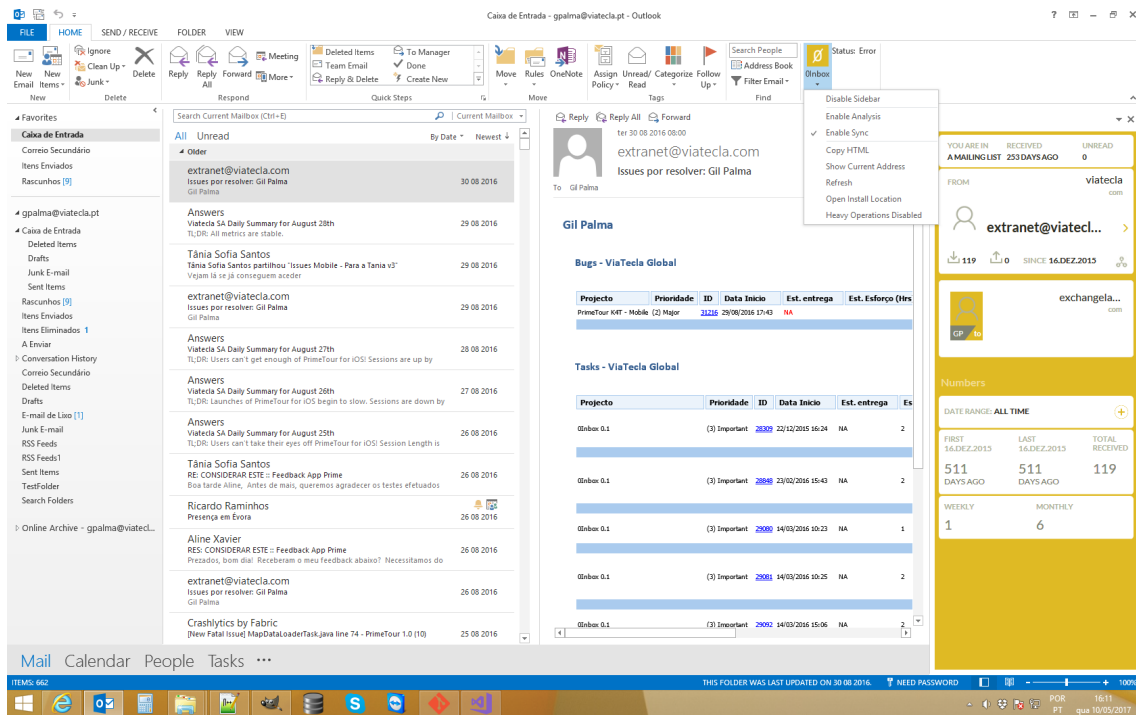


Figura 4.6: *0Inbox* - Opções



## Capítulo 5

# Planeamento de testes de usabilidade

Durante o decorrer do desenvolvimento do *plugin* não foram feitos testes de usabilidade. Quaisquer testes foram feitos com membros da Viatecla, principalmente pela equipa de desenvolvimento, mas estes testes não têm o mesmo valor que testes de usabilidade com potenciais utilizadores. Neste capítulo vão ser considerados testes de usabilidade a ser postos em prática no futuro para determinar o que é necessário para que o produto esteja pronto para ser disponibilizado comercialmente.

### 5.1 Grupo de teste

Considerando que este produto foi pensado para auxiliar utilizadores de *email* em ambientes empresariais, o grupo de teste terá de ser uma amostra desse público alvo. Para maximizar o ganho de informação é importante receber informação de utilizadores de *email* com diferentes necessidades e prioridades. Utilizadores que mais beneficiarão de ferramentas como o *0Inbox* são os que lidam com um grande volume de *emails* durante um dia de trabalho tal como gestores de projeto, de recursos humanos e de relações públicas. No entanto, esta ferramenta também deve ser capaz de ajudar outros utilizadores, mesmo que lidem com pequenas quantidades de *emails*.

Em síntese, é importante que o grupo de teste seja maioritariamente composto por quem lide diariamente com elevado tráfego de mensagens. O resto do grupo de teste deve ser composto por indivíduos que não utilizem *emails* frequentemente.

## 5.2 Formato do teste

Como descrito na secção do Desenho, existem três grupos de operações disponíveis para o utilizador. Um dos três grupos refere-se à seleção de um email existente, uma operação que não requer uma alteração do comportamento por parte do utilizador. O utilizador beneficiará do mecanismo de apresentação de informação relevante ao email selecionado sem ter de aprender como utilizar qualquer outro aspeto do *plugin*. Contudo, a gestão do *plugin* e a interação com as componentes *web* terão de ser testadas.

### 5.2.1 Gestão do *plugin*

A gestão do *plugin* é muito simples, consistindo apenas em algumas operações de ativar ou desativar elementos ou processos do produto. A partir do menu visível na figura 4.6, existem várias opções que serão testadas. O conjunto final de opções ainda não foi determinado mas as primeiras três irão estar presentes na versão final e serão feitos testes com estas.

### 5.2.2 Interação com componentes *web*

Os principais testes serão relacionados com os componentes *web*: a *Sidebar* e com o *Dashboard*. Apesar de o *Dashboard* não ter sido desenvolvido, as suas funcionalidades já foram planeadas e é possível conceptualizar os testes que serão necessários.

## 5.3 Passos de Teste

Estes são os passos necessários para testar o fluxo de utilização das funcionalidades da *Sidebar*:

- Selecionar um *email* para apresentar a sua informação na *Sidebar*;
- Selecionar cada um dos intervenientes do *email* na *Sidebar*;
- Navegar para a visão de Contacto do emissor do *email*;
- Navegar para a visão de Domínio a partir da visão de Contacto do emissor;
- Recuar para a visão principal do *email*;
- Navegar para a visão de Domínio do emissor a partir da visão principal;
- Recuar para a visão principal do *email*;
- Abrir a janela de *Company Network* a partir do ícone de rede no painel do emissor;
- Fechar a janela de *Company Network*;

- Alterar o intervalo de tempo do módulo *Numbers* para um intervalo arbitrário relevante aos *emails* presentes;
- Reverter o intervalo de tempo do módulo *Numbers* para os seus valores por defeito;
- Descarregar um anexo enviado e um anexo recebido a partir do módulo *Attachments*;

Os passos listados abaixo falam de funcionalidades não desenvolvidas mas incluídas no planeamento do *Dashboard*. Para ajudar a entender os passos de teste estão presentes na secção de Trabalho Futuro uma explicação mais detalhada e uma figura (6.1). Listados abaixo estão os passos necessários para testar o fluxo de utilização das funcionalidades da *Dashboard*:

- Abrir o *Dashboard* a partir do botão principal do *Inbox* na *Ribbon* do *Outlook*;
- Na secção de *Email Overview*, seleccionar o grupo de *emails* que requer mais atenção;
- Seleccionar o *email* que requer mais atenção para o abrir no *Outlook* e marcar como lido;
- Colocar o *Dashboard* em foco novamente;
- Retroceder para a secção de *Email Overview*;
- Navegar para a secção de *Contacts*;
- Seleccionar o primeiro Domínio listado para navegar para a sua página de detalhes expandida;
- Retroceder para a secção de *Contacts*;
- Seleccionar a opção de *New Entry* para adicionar um novo Domínio;
- Introduzir a informação do novo Domínio e concluir para o registar e retroceder para a secção de *Contacts*;
- Introduzir o nome do novo Domínio no campo de pesquisa de Organizações;
- Seleccionar o Domínio apresentado para navegar para a sua página de detalhes expandida;
- Retroceder para a secção de *Contacts*;
- Mudar a seleção desta secção de Domínios para *Contactos*;
- Seleccionar o primeiro Contacto listado para navegar para a sua página de detalhes expandida;
- Retroceder para a secção de *Contacts*;

- Selecionar a opção de *New Entry* para adicionar um novo Contacto;
- Introduzir a informação do novo Contacto e concluir para o registar e retroceder para a secção *Contacts*;
- Introduzir o nome do novo Contacto no campo de pesquisa de Contactos;
- Selecionar o Contacto apresentado para navegar para a sua página de detalhes expandida;
- Retroceder para a secção de *Contacts*;
- Navegar para a secção de *Alerts*;
- Selecionar a opção de investigar o primeiro *Follow-up Alert*;
- Selecionar a opção de ignorar o segundo *Follow-up Alert*;
- Selecionar a opção de contactar o primeiro *Loosing Contact Alert*;
- Selecionar a opção de ignorar o segundo *Loosing Contact Alert*;
- Navegar para a secção de *Global Organization Ranking*;
- Selecionar o Domínio mais relevante para o utilizador para navegar para a sua página de detalhes expandida;
- Retroceder para a secção de *Global Organization Ranking*;
- Navegar para a secção de *Smart Search*;
- Inserir termos de pesquisa para limitar o *Smart Search*;
- Selecionar o primeiro *email* apresentado para o abrir no *Outlook*;



# Capítulo 6

## Conclusões e trabalho futuro

Apesar de nem todos os requisitos terem sido cumpridos, na altura da escrita deste documento o estado do produto atingiu um bom nível de qualidade. A plataforma *web*, intitulada *Web View* na secção de 'Requisitos do produto' do capítulo 4, foi concluída mas não detalhada neste documento já que eu não participei no seu desenvolvimento. O *Dashboard* não foi desenvolvido já que a iteração dos aspetos críticos do produto impediu que tempo suficiente pudesse ser dedicado ao seu desenvolvimento. Contudo, como este faz parte do plano inicial do produto, o desenvolvimento deste componente terá de ser feito no futuro. Existe um planeamento das funcionalidades desejadas e uma visão do que se pretende obter finalmente e neste capítulo vai ser apresentada uma síntese desse planeamento e as conclusões tiradas durante o desenvolvimento do projeto.

### 6.1 Trabalho Futuro

Nesta secção vão ser apresentadas as várias tarefas planeadas para o futuro do desenvolvimento do *plugin*, particularmente a adição de novos módulos à *Sidebar*, melhorias técnicas importantes, o sistema de pontuação de *emails*, o desenvolvimento do *Dashboard* e o futuro do *Inbox*.

#### 6.1.1 Módulos adicionais

Um objetivo importante, mas não prioritário, é acrescentar mais módulos à *Sidebar* com funcionalidades importantes para utilizadores empresariais. Foi planeado um gráfico temporal que represente a distribuição de trocas de mensagens num período de tempo e outro

gráfico que disponibilize a dispersão de *emails* ao longo do dia. Com ferramentas como esta, a *Sidebar* capacita os seus utilizadores a saber como e quando enviar *emails* a cada interveniente de uma mensagem sem perder o contexto em que se encontra.

Os módulos existentes foram construídos de forma a serem independentes e auto-contidos para que possam ser reposicionados ou ocultados sem afetar o resto do conteúdo da *Sidebar*. Futuros módulos serão criados da mesma forma e no futuro será implementado um elemento visual para que o utilizador possa posicionar ou esconder módulos de forma simples.

### 6.1.2 Melhorias técnicas

Uma das tarefas a receber mais atenção antes da finalização do produto foi a otimização do desempenho. O equilíbrio entre velocidade de análise de *emails* e utilização de recursos do computador não foi terminado e essa deve ser a primeira tarefa a concluir. Existem algumas correções e alterações técnicas necessárias que, apesar de não afetarem a utilização normal do *plugin*, não seguem os requisitos estabelecidos ou podem causar problemas adicionais no futuro.

Concretamente, a análise de *emails* deve conseguir analisar todas as pastas que não dependam de uma conexão de *Internet* mesmo quando não é possível estabelecer uma conexão, o que não é o caso. Quando a função de análise não consegue validar se uma pasta requer uma conexão à *Internet* ou receber os dados de uma pasta remota, deve ser possível continuar com a análise até onde for possível, de acordo com o requisito que dita que o *plugin* tem de poder funcionar sem uma conexão à *Internet*. Outras correções, tanto ao nível de estabilidade como de funcionalidade, devem ser priorizadas antes de todas as outras.

### 6.1.3 Sistema de pontuação de *emails*

Como mencionado nos requisitos, quando *emails* são analisados estes devem ser pontuados com um valor calculado representativo do seu valor. Quando um *email* é recebido, o *plugin* deve aprender quão importante esse *email* é para o utilizador com base em vários fatores, tal como a rapidez e frequência de resposta. Depois de estabelecer uma hierarquia de importância entre os vários Contactos, e consequentemente entre os Domínios em que se incluem, uma pontuação pode ser dada a cada *email* com base nos dados recolhidos, tal como a hora de envio e intervenientes. Assim, *emails* podem ser classificados com base na sua pontuação de forma a prever a quais *emails* o utilizador dá mais valor e apresentá-los para reduzir o tempo necessário para lidar com todas as tarefas mais importantes.

### 6.1.4 Desenvolvimento do *Dashboard*

A tarefa mais importante depois das melhorias técnicas é a criação do *Dashboard*. Já foram criadas versões básicas deste componente com o propósito de compreender os seus limites

gerais mas nunca se chegou a uma ferramenta minimamente completa já que nunca foram adicionadas funcionalidades significativas.

Concettualmente, o *Dashboard* é um dos componentes mais importantes do *0Inbox* porque fornece ao utilizador um conjunto de ferramentas mais poderosas do que as encontradas na *Sidebar* sem nunca tirar o utilizador do contexto do *Outlook*. Concretamente é uma janela separada da janela principal do *Outlook* que mantém comunicação com este. Tal como a *Sidebar*, este é um componente semelhante com um *webbrowser* embebido que por sua vez apresenta páginas *web* servidas pelo *Griffin Webserver*. Na figura 6.1 é possível ver parte do *mockup* criado para guiar o desenvolvimento do *Dashboard*.

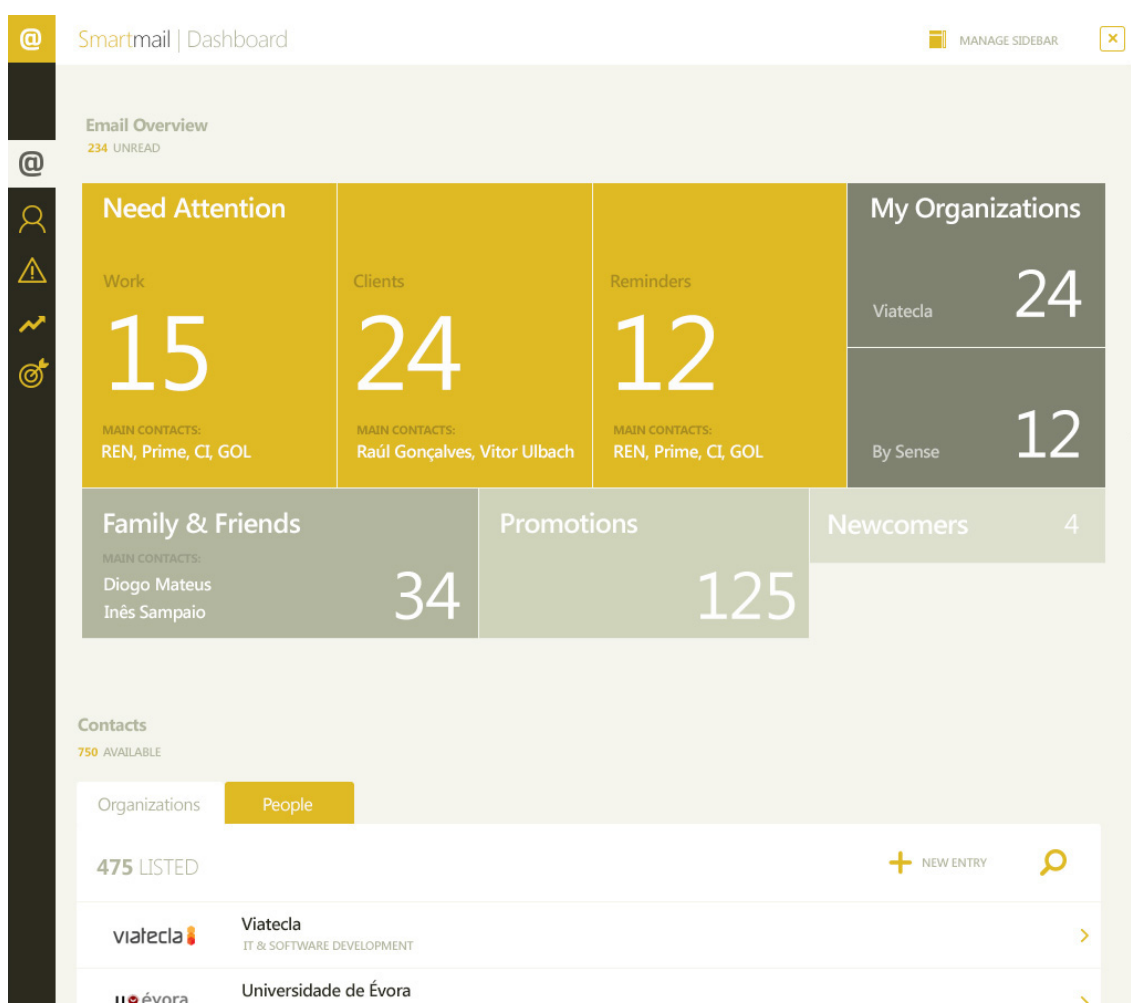


Figura 6.1: *0Inbox* - Dashboard Mockup

Este *mockup* foi feito ainda antes de o nome do produto ter sido mudado mas mantém-se relevante. Na borda superior da janela, além do símbolo e nome do *plugin*, existe um botão para gerir elementos da *Sidebar*. Esta gestão só fará sentido quando for implementado o mecanismo visual de esconder módulos específicos. Na borda esquerda encontram-se ícones que permitem navegar para as diferentes secções da *Dashboard* rapidamente. O resto da

imagem representa o conteúdo da janela onde a informação importante se encontra. Os ícones de navegação da borda limitam-se a deslocar a página verticalmente para colocar em foco a secção correspondente, muito como se veria num *website*.

Como mencionado anteriormente, o conteúdo do *Dashboard* está dividido em secções, especificamente cinco secções. Estas são a *Email Overview*, a *Contacts*, a *Alerts*, a *Global Organization Ranking* e a *Smart Search*. A primeira secção, como visível na figura 6.1, chama-se *Email Overview* e mostra um *treemap* com grupos de *emails*. Estes grupos são representados por retângulos cujo tamanho e intensidade de cor aumentam com a importância dos *emails* contidos. Grupos com *emails* considerados mais importantes recebem uma cor mais vibrante e maiores dimensões de forma a direccionar o utilizador aos *emails* que requerem atenção antes de todos os outros. Ao seleccionar um destes grupos, o utilizador é levado a uma nova visão onde os *emails* contidos são listados e ordenados por importância, sendo também oferecida a opção de saltar imediatamente para o *email* no *Outlook*. Caso faça sentido, um grupo deste *treemap* pode ser composto por subgrupos, representados também por *treemaps*.

A segunda secção do *Dashboard*, *Contacts*, lista todos os Domínios e Contactos registados no *plugin*. Ao seleccionar um registo, o utilizador é levado para páginas de informação detalhada do registo selecionado, também com a possibilidade de o remover ou alterar. É também oferecida uma opção de pesquisa, que devolve registos direta ou indiretamente relevantes, e uma opção para adição manual de novos registos, a partir de onde podem ser estabelecidas manualmente relações entre Domínios e Contactos.

A secção seguinte é a *Alerts*, responsável por apresentar ao utilizador dois tipos de alertas: tarefas com um prazo limite e contactos com quem o utilizador não tem comunicado. Apesar de oferecerem dados diferentes, ambos os tipos de alertas partilham o mesmo objetivo de ajudar o utilizador a manter-se atento à atividade da sua caixa de correio. A partir desta secção um utilizador pode escolher ignorar os alertas ou agir em conformidade com estes.

A quarta secção, *Global Organization Ranking*, volta a ter em conta os valores calculados para cada Contacto mas desta vez para dar uma noção visual, em forma de *treemap*, dos Domínios com quem o utilizador comunica com mais frequência e urgência. De forma semelhante ao *treemap* da primeira secção do *Dashboard*, cada retângulo representante de um Domínio tem uma cor e dimensões atribuídas proporcionalmente à sua importância e quando um Domínio é selecionado, o utilizador pode ser levado para a lista dos Contactos associados a esse Domínio ou para um subgrupo, tal como um grupo de Domínios cuja importância não é elevada o suficiente para ter um retângulo de proporções significativas no nível atual do gráfico. Também é apresentado um filtro cronológico para que o utilizador possa conhecer a importância dos Domínios num período específico.

Finalmente, na quinta secção, o *Smart Search* oferece a capacidade de listar *emails* com três filtros: nomes de Contactos, Domínios e linhas de conversação. Cada *email* listado é representado como uma entrada incluindo o seu assunto, intervenientes e data, a partir

de onde o utilizador pode abrir o *email* correspondente na janela principal do *Outlook*. Ao introduzir texto nos três filtros de pesquisa mencionados, esta lista atualiza-se para auxiliar o utilizador a encontrar *emails* de que precise. Esta pesquisa diferencia-se da pesquisa já incluída no *Outlook* porque utiliza as noções de Domínio e Contacto criadas para este *plugin*, permitindo que o utilizador possa traçar relações entre indivíduos e temas de conversa mesmo que os seus endereços de *email* mudem devido a uma mudança de emprego ou posição dentro de uma mesma organização.

Este é o plano atual para a criação do *Dashboard*. Depois de estas funcionalidades serem implementadas outras poderão ser adicionadas, mesmo que não contempladas neste planeamento, à medida que novas ideias surjam para transformar o *Dashboard*, e o *plugin* no geral, em ferramentas cada vez mais robustas e poderosas. É importante dedicar recursos a este componente do *0Inbox* já que inclui alguns dos elementos mais poderosos de todo o *plugin* e facilita a compreensão do potencial de utilidade deste produto a mais utilizadores de *email*, o que por sua vez facilita a aplicação de testes e posterior disseminação por potenciais clientes.

### 6.1.5 Futuro do *0Inbox*

O *plugin* de *Outlook* foi a plataforma escolhida para este produto mas há intenção de expandir o *0Inbox*. Os testes de usabilidade serão críticos no processo de compreensão das alterações necessárias para obter um produto completo. De seguida, é importante olhar para outras plataformas onde um serviço como este pode agilizar o fluxo de trabalho de utilizadores de *email*. Particularmente, com a abertura progressiva do serviço do *Microsoft Exchange*, ferramentas *mobile* ou puramente *web* podem ser os próximos passos lógicos do *0Inbox*. O desenvolvimento destas novas soluções não deve impedir o desenvolvimento das ferramentas principais, de forma a não acabar por ter várias soluções parciais, mas é importante manter este objetivo em mente para que o *0Inbox* se continue a adaptar às mudanças de paradigmas tecnológicos e sociais, continuando a ser uma ferramenta relevante e competitiva.

## 6.2 Conclusões

O *email* mantém-se como um dos melhores métodos de comunicação apesar da existência de muitas soluções alternativas e da sua relativa falta de funcionalidades modernas. O *0Inbox* propõe ajudar a trazer o *email* para a linha da frente, preenchendo várias lacunas das soluções existentes. Como mencionado no Estado da Arte, várias soluções existem mas nenhuma oferece o conjunto de ferramentas completa.

O seu desenvolvimento não foi tão veloz como desejado devido à iteração de várias funcionalidades até que chegassem à qualidade desejada mas o que foi desenvolvido é representativo do estado desejado para os elementos futuros do *0Inbox*. Com um investimento continu-

ado por parte da Viatecla, este produto poder-se-á tornar numa solução importante para organizações em todo o mundo, facilitando tanto a comunicação interna como externa.

Esta dissertação apresentou o contexto da motivação que provocou o desenvolvimento do *0Inbox*, toda a investigação realizada e o trabalho desenvolvido e futuro. Mas este trabalho não permitiu apenas desenvolver o *plugin* de *Outlook* mas também aprender mais sobre como organizações funcionam e se baseiam em comunicação para poder crescer. Mais do que nunca, a globalização crescente de organizações e países requer não só muita comunicação mas também toda a informação circundante a essa comunicação. Conhecer quando, como e exatamente com quem mensagens são trocadas é tão valioso como as mensagens em si. Independentemente do futuro do *0Inbox*, esta informação é valiosa noutros contextos e vai permitir que todos os elementos envolvidos neste projeto trabalhem com uma nova perspetiva e maior conhecimento do seu ambiente envolvente, tanto ao nível da organização como do mundo.

# Referências

- [1] “MailChimp.” <https://mailchimp.com/>. Acedido em: 19-12-2016.
- [2] “Xobni.” <http://address.yahoo.com/xobni>. Acedido em: 19-12-2016.
- [3] “Xobni for Outlook: Infinity.” <https://support.xobni.com/hc/en-us/articles/215189626-Download-Xobni-for-Outlook-Infinity-Edition>. Acedido em: 19-12-2016.
- [4] “Office Outlook.” <https://www.microsoftstore.com/store/msusa/pdp/Outlook-2016/productID.323024000>. Acedido em: 19-12-2016.
- [5] “Viatecla.” <http://www.viatecla.com/>. Acedido em: 19-12-2016.
- [6] “Wolfram Alpha Research - Computational Knowledge Engine.” <https://www.wolframalpha.com/>. Acedido em: 30-11-2014.
- [7] “Datawatch.” <http://www.datawatch.com/>. Acedido em: 27-12-2016.
- [8] F. Brodbeck, “Cinematics.” <http://cinematics.fredericbrodbeck.de/>. Acedido em: 2-1-2017.
- [9] “Visual Action.” <http://www.visualaction.com/>. Acedido em: 18-12-2016.
- [10] X. Liu and S. Fouché, “Viz4All.” [https://www.cs.umd.edu/class/spring2005/cmsc838s/viz4all/viz4all\\_a.html](https://www.cs.umd.edu/class/spring2005/cmsc838s/viz4all/viz4all_a.html), 2005. Acedido em: 18-12-2016.
- [11] “Visual Action Gallery.” <http://www.visualaction.com/gallery/>. Acedido em: 18-12-2016.
- [12] “Tableau.” <http://www.tableau.com/>. Acedido em: 3-1-2017.
- [13] “Tableau Software Gallery.” <http://www.tableausoftware.com/learn/gallery>. Acedido em: 3-1-2017.

- [14] “RedPoint.” <http://www.redpoint.net/>. Acedido em: 3-1-2017.
- [15] “Hans Rosling: Stats that reshape your world-view.” <https://www.youtube.com/watch?v=hVimVzgtD6w>. Acedido em: 18-12-2016.
- [16] “Gapminder.” <https://www.gapminder.org/>. Acedido em: 3-1-2017.
- [17] “CARTO.” <https://carto.com/>. Acedido em: 3-1-2017.
- [18] “CartoDB.” <https://github.com/CartoDB/cartodb>. Acedido em: 3-1-2017.
- [19] “PoliteMail.” <https://www.politemail.com/>. Acedido em: 3-1-2017.
- [20] “ClearContext.” <http://www.clearcontext.com/>. Acedido em: 3-1-2017.
- [21] “Bells & Whistles for Outlook.” <http://www.outlook-apps.com/product/bells-whistles-for-outlook/>. Acedido em: 3-1-2017.
- [22] “Gmail Meter.” <http://www.gmailmeter.com/>. Acedido em: 3-1-2017.
- [23] “Google Drive.” <https://www.google.com/drive/>. Acedido em: 3-1-2017.
- [24] “ShuttleCloud.” <https://www.shuttlecloud.com/>. Acedido em: 18-12-2016.
- [25] “Yahoo.” <https://www.yahoo.com/>. Acedido em: 3-1-2017.
- [26] “GetResponse.” <https://www.getresponse.com/>. Acedido em: 18-12-2016.
- [27] “SumAll.” <https://sumall.com/>. Acedido em: 19-12-2016.
- [28] “Siverpop Homepage.” <http://www.silverpop.com/>. Acedido em: 19-12-2016.
- [29] “Silverpop Email Insights.” <http://www.silverpop.com/Software/Mobile-Customer-Engagement/Email-Insights/>. Acedido em: 19-12-2016.
- [30] “Silverpop Engage Apps.” <http://www.silverpop.com/Software/Engage-Apps/>. Acedido em: 19-12-2016.
- [31] “Zoho Campaigns.” <https://www.zoho.com/campaigns/>. Acedido em: 16-1-2017.
- [32] “Zoho Reports.” <https://www.zoho.com/reports/>. Acedido em: 16-1-2017.
- [33] “SendinBlue.” <https://sendinblue.com/>. Acedido em: 16-1-2017.
- [34] “Microsoft Surface.” [https://www.microsoftstore.com/store/mseea/pt\\_PT/pdp/productID.326607500](https://www.microsoftstore.com/store/mseea/pt_PT/pdp/productID.326607500). Acedido em: 16-1-2017.
- [35] “iPad.” <http://www.apple.com/lae/ipad/>. Acedido em: 16-1-2017.
- [36] “Apple.” <http://www.apple.com/>. Acedido em: 16-1-2017.
- [37] “iOS.” <http://www.apple.com/ios>. Acedido em: 16-1-2017.



- [38] “Core Plot.” <https://github.com/core-plot/core-plot>. Acedido em: 16-1-2017.
- [39] “Shinobi Controls.” <https://www.shinobicontrols.com/>. Acedido em: 16-1-2017.
- [40] “JBChartView.” <https://github.com/Jawbone/JBChartView>. Acedido em: 16-1-2017.
- [41] “iOS: Chart.” <http://www.threedgraphics.com/tdg/products/tools/ioschart/>. Acedido em: 16-1-2017.
- [42] “GitHub.” <https://github.com/>. Acedido em: 16-1-2017.
- [43] “Three D Graphics.” <http://www.threedgraphics.com/tdg/>. Acedido em: 16-1-2017.
- [44] “PGSDK.” <https://www.threedgraphics.com/tdg/products/tools/pgsdk/pgsdk.php>. Acedido em: 16-1-2017.
- [45] “Android.” <https://www.android.com/>. Acedido em: 17-1-2017.
- [46] “Statista Smartphone Sales 2009 Q1.” <http://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system/>. Acedido em: 17-1-2017.
- [47] P. Nickinson, “Android Central - Android Versions.” <http://www.androidcentral.com/current-android-versions>, Junho 2016. Acedido em: 17-1-2017.
- [48] “Android Dashboards.” <https://developer.android.com/about/dashboards/index.html>. Acedido em: 17-1-2017.
- [49] S. Hill, “Alternativas ao *Google Play*.” <http://www.digitaltrends.com/mobile/android-app-stores/>, Setembro 2016. Acedido em: 17-1-2017.
- [50] “F-Droid.” <https://f-droid.org/>. Acedido em: 17-1-2017.
- [51] “Amazon Underground App for Android.” [https://www.amazon.com/gp/mas/get/android/ref=get\\_appstore?ie=UTF8&ref\\_=mas\\_rw\\_ldg](https://www.amazon.com/gp/mas/get/android/ref=get_appstore?ie=UTF8&ref_=mas_rw_ldg). Acedido em: 17-1-2017.
- [52] “AChartEngine.” <http://achartengine.org/>. Accessed: 17-1-2017.
- [53] “MPAndroidChart.” [https://github.com/PhilJay/MPAndroidChart?utm\\_source=android-arsenal.com&utm\\_medium=referral&utm\\_campaign=741](https://github.com/PhilJay/MPAndroidChart?utm_source=android-arsenal.com&utm_medium=referral&utm_campaign=741). Acedido em: 7-2-2017.
- [54] “MPAndroidChart - Android Arsenal.” <https://android-arsenal.com/details/1/741>. Acedido em: 7-2-2017.
- [55] “Skia.” <https://skia.org/>. Acedido em: 10-2-2017.
- [56] “Google.” <https://www.google.com/>. Acedido em: 21-12-2016.

- [57] “Androidplot.” <http://androidplot.com/>. Acedido em: 10-2-2017.
- [58] “Androidplot Google Play Demos.” <https://play.google.com/store/apps/details?id=com.androidplot.demos>. Acedido em: 10-2-2017.
- [59] “Google Play.” <https://play.google.com/store>. Acedido em: 10-2-2017.
- [60] “GraphView.” <http://www.android-graphview.org/>. Acedido em: 10-2-2017.
- [61] “GraphView Demo.” [https://play.google.com/store/apps/details?id=com.jjoe64.graphview\\_demos](https://play.google.com/store/apps/details?id=com.jjoe64.graphview_demos). Acedido em: 10-2-2017.
- [62] “Windows Mobile.” [http://en.wikipedia.org/wiki/Windows\\_Mobile](http://en.wikipedia.org/wiki/Windows_Mobile). Acedido em: 20-1-2017 (Wikipedia).
- [63] G. Kelly, “Windows Phone Market Share 2013,” Janeiro 2014. <http://www.forbes.com/sites/gordonkelly/2014/01/21/here-are-24-countries-where-windows-phone-outsells-the-iphone-and-why-it-does/#46d51cfc4675>. Acedido em: 20-1-2017.
- [64] P. Simões, “Windows Phone Market Share 2014,” Julho 2014. <http://pplware.sapo.pt/informacao/windows-phone-estados-unidos-india-e-brasil-dominam-o-mercado/>. Acedido em: 20-1-2017.
- [65] T. Warren, “Windows Mobile Sales 2016,” Janeiro 2016. <http://www.theverge.com/2016/1/28/10864034/windows-phone-is-dead>. Acedido em: 20-1-2017.
- [66] Surur, “Revealed: Only 1.2 million Windows Phones sold in Q2 2016.” <https://mspoweruser.com/revealed-1-2-million-windows-phones-sold-q2-2016/>, Julho 2016. Acedido em: 10-2-2017.
- [67] “IDC Smartphone OS Market Share, 2016 Q3.” <http://www.idc.com/promo/smartphone-market-share/os>. Acedido em: 10-2-2017.
- [68] C. Page, “Windows Phone claimed just 0.7 per cent of global smartphone sales in Q1.” <http://www.theinquirer.net/inquirer/news/2459088/windows-phone-claimed-just-07-per-cent-of-global-smartphone-sales-in-q1>, Maio 2016. Acedido em: 10-2-2017.
- [69] “Sparrow Toolkit.” <http://sparrowtoolkit.codeplex.com/>. Acedido em: 12-2-2017.
- [70] “Silverlight Toolkit.” <https://silverlight.codeplex.com/>. Acedido em: 12-2-2017.
- [71] “Visifire.” <http://www.visifire.com/>. Acedido em: 12-2-2017.
- [72] “Visifire Windows Phone.” [http://www.visifire.com/windows\\_phone\\_charts.php](http://www.visifire.com/windows_phone_charts.php). Acedido em: 12-2-2017.

- [73] “Visifire Windows Phone Chart Gallery.” [http://www.visifire.com/windows\\_phone\\_chart\\_gallery.php](http://www.visifire.com/windows_phone_chart_gallery.php). Acedido em: 12-2-2017.
- [74] “Telerik RadChart.” <http://www.telerik.com/products/windows-phone/overview/all-controls/chart.aspx>. Acedido em: 12-2-2017.
- [75] “Telerik.” <http://www.telerik.com/>. Acedido em: 12-2-2017.
- [76] “Telerik RadChart Demos.” <http://demos.telerik.com/aspnet-ajax/htmlchart/examples/overview/defaultcs.aspx>. Acedido em: 12-2-2017.
- [77] “Jack of Tools Pro.” <https://www.microsoft.com/en-GB/store/p/jack-of-tools-pro/9wzdnrcrfj3nb>. Acedido em: 12-2-2017.
- [78] “HTML5.” <http://en.wikipedia.org/wiki/HTML5>. Acedido em: 23-1-2017.
- [79] “Twitch.” <https://www.twitch.tv/>. Acedido em: 23-1-2017.
- [80] F. Obermeyer, “Fritzo Keys: The Rational Keyboard.” <http://fritzo.org/keys/>. Acedido em: 24-1-2017.
- [81] “Data-Driven Documents.” <https://d3js.org/>. Acedido em: 27-12-2016.
- [82] B. Siddiqui, “Data visualization, Part 1: Visualize browsing metrics with SVG and D3,” 2013. <http://www.ibm.com/developerworks/library/os-dataviz1/>. Acedido em: 27-12-2016.
- [83] “D3 USA Deficit.” <http://www.brightpointinc.com/interactive/ustrade/>. Acedido em: 24-1-2017.
- [84] “Bl.ocks.” <http://bl.ocks.org/>. Acedido em: 24-1-2017.
- [85] “D3.js Gallery.” <http://christopheviau.com/d3list/gallery.html>. Acedido em: 24-1-2017.
- [86] “Voronoi Animation.” <http://bl.ocks.org/MattWoelk/5687467>. Acedido em: 24-1-2017.
- [87] “Voronoi Diagram.” [http://en.wikipedia.org/wiki/Voronoi\\_diagram](http://en.wikipedia.org/wiki/Voronoi_diagram). Acedido em: 24-1-2017.
- [88] “Highcharts.” <http://www.highcharts.com/>. Acedido em: 24-1-2017.
- [89] “CanvasJS.” <http://canvasjs.com/>. Acedido em: 24-1-2017.
- [90] “Canvas Element.” [http://en.wikipedia.org/wiki/Canvas\\_element](http://en.wikipedia.org/wiki/Canvas_element). Acedido em: 24-1-2017.
- [91] “Canvas vs SVG.” <http://www.sitepoint.com/canvas-vs-svg-how-to-choose/>. Acedido em: 24-1-2017.

- [92] “Google Charts.” <https://developers.google.com/chart/>. Acedido em: 12-2-2017.
- [93] “Google Charts Quick Start.” [https://developers.google.com/chart/interactive/docs/quick\\_start](https://developers.google.com/chart/interactive/docs/quick_start). Acedido em: 12-2-2017.
- [94] “Google Charts Gallery: GeoChart.” <https://developers.google.com/chart/interactive/docs/gallery/geochart>. Acedido em: 12-2-2017.
- [95] “Google Charts Gallery: Treemap.” <https://developers.google.com/chart/interactive/docs/gallery/treemap>. Acedido em: 12-2-2017.
- [96] “Add-In Express.” <https://www.add-in-express.com/add-in-net/outlook.php>. Acedido em: 24-1-2017.
- [97] “PhoneGap.” <http://phonegap.com/>. Acedido em: 24-1-2017.
- [98] “Xamarin.” <https://www.xamarin.com/>. Acedido em: 24-1-2017.
- [99] “Appcelerator Titanium.” <http://www.appcelerator.com/>. Acedido em: 24-1-2017.
- [100] Guillermo Winkler, “Can you make a native app using phonegap?,” Fevereiro 2013. <http://blog.guillermowinkler.com/blog/2013/02/10/can-you-make-a-native-app-using-phonegap/>. Acedido em: 27-1-2017.
- [101] “Apache Cordova.” <https://cordova.apache.org/>. Acedido em: 27-1-2017.
- [102] “Adobe.” <http://www.adobe.com/>. Acedido em: 27-1-2017.
- [103] “Nitobi.” <https://www.crunchbase.com/organization/nitobi-software#/entity>. Acedido em: 27-1-2017.
- [104] “Apache.” <http://www.apache.org/>. Acedido em: 27-1-2017.
- [105] “Flot.” <http://www.flotcharts.org/>. Acedido em: 27-1-2017.
- [106] “jqPlot.” <http://www.jqplot.com/>. Acedido em: 27-1-2017.
- [107] C. Coenraets, “Building Interactive Mobile Dashboards with D3 and other Charting Libraries.” <http://coenraets.org/blog/2014/02/interactive-mobile-dashboard-d3-charts/>. Acedido em: 27-1-2017.
- [108] “PhoneGap App Gallery.” <http://phonegap.com/app/>. Acedido em: 27-1-2017.
- [109] “Ionic.” <http://ionicframework.com/>. Acedido em: 27-1-2017.
- [110] “NUS extra - Student Discounts.” <http://www.appbrain.com/app/nus-extra-student-discounts/org.nussl.NusExtraDiscountCompanionApp>. Acedido em: 27-1-2017.

- [111] C. C. Ltd, “Extra Companion App Launched By NUS.” <http://www.creativematch.com/news/extra-companion-app-launched-by/102082/>. Acedido em: 27-1-2017.
- [112] “HealthTap.” <http://phonegap.com/app/healthtap-find-doctors-and-free-answers/>, Agosto 2012. Acedido em: 10-2-2017.
- [113] “HealthTap HomePage.” <https://www.healthtap.com/>. Acedido em: 10-2-2017.
- [114] “BrowserQuest.” <http://phonegap.com/app/browserquest/>. Acedido em: 10-2-2017.
- [115] “Appcelerator Compatibility.” [http://docs.appcelerator.com/platform/latest/#!/guide/Titanium\\_Compatibility\\_Matrix-section-src-29004837\\_TitaniumCompatibilityMatrix-MobileDevelopment](http://docs.appcelerator.com/platform/latest/#!/guide/Titanium_Compatibility_Matrix-section-src-29004837_TitaniumCompatibilityMatrix-MobileDevelopment). Acedido em: 27-1-2017.
- [116] “Chart.js.” <http://www.chartjs.org/>. Acedido em: 27-1-2017.
- [117] L. Brenman, “Interactive Charts in Appcelerator Titanium.” <http://www.appcelerator.com/blog/2013/09/interactive-charts-in-appcelerator-titanium/>. Acedido em: 27-1-2017.
- [118] “Appcelerator Titanium Charts.” <http://gitt.io/component/ti.charts>. Acedido em: 4-4-2017.
- [119] “ParkYa.” <https://parkya.com/>. Acedido em: 27-1-2017.
- [120] A. Nanji, “How iPhone and Android Ownership Varies by Demographic.” <http://www.marketingprofs.com/charts/2013/10957/how-iphone-and-android-ownership-varies-by-demographic>, Junho 2013. Acedido em: 23-2-2017.
- [121] “Gmail, and Today’s Popular Email Clients.” <https://www.campaignmonitor.com/resources/guides/most-popular-email-clients/>. Acedido em: 23-2-2017.
- [122] B. Travis, “Android vs. iOS: User Differences Every Developer Should Know.” <http://www.comscore.com/Insights/Blog/Android-vs-iOS-User-Differences-Every-Developer-Should-Know>, Março 2013. Acedido em: 23-2-2017.
- [123] L. Bustos, “Android vs iOS: Demographics and Mobile Commerce Activity.” <http://www.getelastic.com/android-vs-ios-demographics-and-mobile-commerce-activity/>, Agosto 2013. Acedido em: 23-2-2017.
- [124] L. Inc., “Logitech Squeezebox™ Controller.” <http://phonegap.com/app/logitech-squeezebox%E2%84%A2-controller-2/>, Abril 2011. Acedido em: 13-2-2017.

- [125] “RE/MAX Real Estate Search.” <https://play.google.com/store/apps/details?id=com.remax.remamobile>. Acedido em: 13-2-2017.
- [126] “Xamarin Platform.” <https://www.xamarin.com/platform>. Acedido em: 14-2-2017.
- [127] “PhoneGap Desktop.” <https://cordova.apache.org/docs/en/latest/guide/support/>. Acedido em: 14-2-2017.
- [128] “Cordova Platform Support.” <http://docs.phonegap.com/getting-started/1-install-phonegap/desktop/>. Acedido em: 14-2-2017.
- [129] “Email Client Market Share.” <http://emailclientmarketshare.com/>. Acedido em: 15-2-2017.
- [130] “TechHit EZDetach.” [http://www.techhit.com/ezdetach/outlook\\_attachments.html](http://www.techhit.com/ezdetach/outlook_attachments.html). Acedido em: 15-2-2017.
- [131] “MailChimp Concepts: Email Clients.” <https://templates.mailchimp.com/concepts/email-clients/>. Acedido em: 15-2-2017.
- [132] “Griffin Webserver.” <https://github.com/jgauffin/Griffin.WebServer>. Acedido em: 21-12-2016.
- [133] “Apache 2.0 License.” <https://www.apache.org/licenses/LICENSE-2.0>. Acedido em: 21-12-2016.
- [134] “Using a treemap to analyze relative values of CO2 indicators.” <http://co2scorecard.org/home/researchitem/10>. Acedido em: 24-3-2017.
- [135] “CO2 Scorecard Treemaps.” <http://co2scorecard.org/countrydata/treemaps>. Acedido em: 24-3-2017.
- [136] “LinkedIn.” <https://www.linkedin.com/>. Acedido em: 21-12-2016.
- [137] “SQLite.” <https://www.sqlite.org/>. Acedido em: 24-3-2017.