



Licenciatura em Sistemas e Tecnologias de Informação

**Automatização de processos de vendas com recurso a tecnologia móvel**

Projeto Final de Licenciatura

Elaborado por Luis Carlos Rodrigues Baptista Neves

Aluno nº 20091288

Orientador: Professora Filipa Taborda

Barcarena

Junho 2013

Universidade Atlântica

Licenciatura em Sistemas e Tecnologias de Informação

**Automatização de processos de vendas com recurso a tecnologia móvel**

Projeto Final de Licenciatura

Elaborado por Luis Carlos Rodrigues Baptista Neves

Aluno nº 20091288

Orientador: Professora Filipa Taborda

Barcarena

Junho 2013

O autor é o único responsável pelas ideias expressas neste relatório

## **Agradecimentos**

À professora Filipa Taborda, pela sua orientação, disponibilidade e motivação que me proporcionou durante a realização deste trabalho.

À Combined Insurance, pelo apoio e incentivo e ao meu colega Sérgio Guerra por me ter aturado durante este tempo.

Aos meus amigos e familiares por todo o apoio.

E por último e em especial à minha mulher e ao meu filho por tudo.

## **Resumo**

Muitas empresas à procura de uma maior eficiência de processos adotam tecnologias de informação e comunicação baseadas na mobilidade. Essa procura tem sido feita também na gestão das forças de vendas há vários anos, mas com a competitividade do mercado cada vez mais feroz, está-se a assistir à necessidade de cumprir objetivos e fazer crescer os lucros.

Este trabalho apresenta uma solução aplicacional de mobilidade, com base num caso real, de um sistema de automatização de força de vendas.

Um dos propósitos do trabalho foi o desenvolvimento de um protótipo que obteve como principal resultado, a possibilidade da força de vendas da Combined Insurance preencher e submeter as apólices vendidas para o sistema central através de uma aplicação para o sistema operativo Android.

## **Abstract**

Many companies looking for a greater efficiency of processes have adopted information and communication technologies based on mobility. That search has also been made in the management of sales forces for several years now, but with the market competitiveness getting increasingly fierce, the need to achieve goals and grow profits is becoming more and more necessary.

The present paper is about an application that demonstrates, based on a real case, the applicability of a sales force automation system.

One of the purposes of this paper was the development of a prototype that had as the main result, the possibility of the Combined Insurance sales force to fulfill and submit policies sold into a central host system through an application for the Android operative system.

## Índice

1.	Introdução .....	12
1.1.	Motivação e problema.....	12
1.2.	Questão de investigação e postura epistemológica .....	13
1.3.	Recolha de dados e universo .....	13
1.4.	Âmbito do projeto .....	14
1.5.	A Combined Insurance.....	14
1.6.	Objetivos .....	14
1.7.	Estrutura do documento .....	16
2.	Fundamentação teórica e metodológica .....	16
2.1.	Tecnologia Móvel .....	16
2.2.	Comunicação e Serviços Móveis nas Empresas .....	18
3.	Levantamento e caracterização da situação atual (AS-IS) .....	19
3.1.	Ciclo de Gestão Administrativa e Comercial .....	19
3.2.	Preenchimento das apólices .....	21
4.	Desenvolvimento e avaliação do protótipo .....	24
4.1.	Escolha do sistema .....	24
4.2.	Proposta de protótipo e arquitetura .....	25
4.3.	Desenvolvimento do Protótipo.....	28
4.3.1.	Descrição dos Casos de Uso e atividades.....	28
4.3.2.	Aspetos gráficos e interação com o sistema operativo.....	34
4.4.	Modelo de dados .....	37
4.5.	Avaliação e Resultados .....	37
5.	Tecnologia de desenvolvimento.....	40
5.1.	Android .....	40

5.2.	Eclipse e o Android SDK.....	42
5.3.	Técnicas de Programação Android .....	43
5.3.1.	<i>Conceitos gerais -Activities, Intents e AndroidManifest</i> .....	43
5.3.2.	AsyncTask.....	45
5.3.3.	Services .....	46
5.3.4.	<i>BroadCastReceivers</i> .....	46
5.3.5.	SQLite .....	46
6.	Conclusões .....	47
6.1.	Trabalho Futuro.....	47
7.	Bibliografia .....	48

## Tabela de Figuras

Fig.1 - Diagrama Atividades AS-IS .....	20
Fig.2 - Apólices Pendentes 2012.....	23
Fig.3 - Diagrama componentes - Arquitetura geral.....	27
Fig.4 - Diagrama Atividades To-Be.....	28
Fig.5 - Caso de Uso.....	29
Fig.6 - Diagrama Atividades Fluxo da Aplicação.....	32
Fig.7 - Diagrama de Sequência Produto Proposto .....	33
Fig.8 - Interação entre a aplicação e Sistema Operativo Android - Adaptado de (Gonçalo Lopes, 2010) .....	34
Fig.9 - Modelo de Dados Relacional.....	37
Fig. 10 - Exemplo de assinatura.....	38
Ilustração 11 - Ciclo de Vida de uma Activity.....	43

## **Lista de Abreviaturas e siglas**

ADB – Android Debug Bridge

ADT – Android Development Tools

ANR – Application Not Responding

API - Application Programming Interface

AVD – Android Virtual Device

CMS – Content Management System

GPRS - General Packet Radio Service

GPS – Global Position System

GSM - Global System for Mobile

IDE - Integrated Development Environment

JSON – JavaScript Object Notation

NIB – Número de Identificação Bancária

OHA – Open Handset Alliance

PNG - Portable Network Graphics

SDK – System Development Kit

SFA – Sales Force Automation

SQL – Structured Query Language

SGBD – Sistema de Gestão de Bases de Dados

TIC – Tecnologia de Informação e Comunicação

UI – User Interface

XML - eXtensible Markup Language

## 1. Introdução

Muitas empresas à procura de uma maior eficiência de processos adotam tecnologias de informação e comunicação baseadas na mobilidade. As tecnologias móveis de informações e comunicação incluem a infraestrutura tecnológica para a conectividade, tais como Bluetooth, 3G e GPRS, bem como os equipamentos móveis, tais como *smartphones*, telemóveis, *notebooks* e *tablets*.

As organizações ao equiparem os colaboradores com tecnologia móvel, possibilitam o acesso à informação a partir de qualquer lugar e a qualquer hora. As organizações que adotam estas tecnologias não fornecem apenas mobilidade aos seus trabalhadores, mas também reorganizam os seus processos de negócio, procedimentos operacionais, estrutura organizacional e sistemas de remuneração em torno das necessidades emergentes de novos modelos de gestão.

Essa reorganização tem sido também debatida nas gestão das forças de vendas há vários anos, mas com a competitividade do mercado cada vez mais feroz, está-se cada vez mais a assistir à necessidade de cumprir objetivos e fazer crescer os lucros.

### 1.1. Motivação e problema

Como motivação base para a proposta deste trabalho, está a necessidade de melhorar o sistema de vendas na Combined Insurance. Este trabalho sugere uma proposta de solução de um sistema de SFA suportado por sistemas móveis.

Numa visão mais abrangente o objetivo deste projeto é identificar as potenciais soluções a utilizar pela Combined Insurance, de modo a ajudar na definição do modo como a organização poderá operar, pode melhorar a oferta dos seus produtos e serviços, a conectividade entre locais remotos e o acesso a mais e melhor informação, dada a dispersão e mobilidade dos seus agentes comerciais.

Devido a fatores como a agressividade e competitividade dos mercados globais e uma necessidade de melhorar a qualidade do serviço e de acesso à informação por parte dos

seus agentes comerciais, a Combined Insurance está disposta a apostar na utilização de dispositivos móveis (*tablets*) por parte dos seus agentes comerciais.

Os *tablets* serão equipados com uma aplicação para efetuar a apresentação dos produtos/serviços ao cliente e respetiva recolha dos dados inerentes à sua aquisição. Neste ponto surge o conceito de SFA. Hunter & Pernault (2006) definem SFA como qualquer T.I. aplicada às vendas que melhore, facilite e permita uma tarefa de venda.

### **1.2. Questão de investigação e postura epistemológica**

A questão de investigação deste trabalho é: Qual a melhor forma de otimizar a comunicação/registo de apólices e SFA na realidade da Combined Insurance?

O método de investigação utilizado neste trabalho é a pesquisa-ação.

No que diz respeito à abordagem epistemológica, e tendo em conta a pergunta de partida, será seguida uma visão positivista.

### **1.3. Recolha de dados e universo**

Existem três métodos utilizados para a recolha de dados:

2. A entrevista;
3. A observação;
4. A análise documental

No âmbito deste trabalho, a recolha de dados é feita através da análise documental, da observação e de entrevistas com os diversos elementos que fazem parte do universo de estudo.

A seguinte tabela identifica os entrevistados:

Nome	Função
Pedro Fidalgo	Diretor de Formação
David Batista	Formador
Rui Enes	Regional Manager
Carlos Oliveira	Agente Comercial

#### **1.4. Âmbito do projeto**

A gestão de todo o ciclo de vida do processo de vendas requiere intervenção num vasto conjunto de aspetos que exigem aplicações abrangentes capazes de dar resposta às diversas fases que constituem o referido ciclo.

Este trabalho não tem o propósito de fazer a gestão de todo o ciclo do processo de vendas da Combined Insurance. Este trabalho incide essencialmente sobre as fases denominadas de preenchimento das apólices e envio para o escritório da Combined Insurance. Atualmente este é o processo interno da Combined Insurance que consome mais recursos na atividade de toda a empresa.

#### **1.5. A Combined Insurance**

A Combined Insurance é uma multinacional com mais de 3.5 milhões de clientes espalhados por todo o mundo e 85.000 em Portugal. Foi fundada em 1922 e está estabelecida em Portugal desde 1998. Está registada no Instituto de Seguros de Portugal e é membro da Associação Portuguesa de Seguradoras. Está localizada em Paço de Arcos e tem na sua equipa cerca de 100 profissionais da atividade seguradora e mais de 300 agentes comerciais, para prestar o melhor serviço aos clientes.

A Combined Insurance está integrada na ACE Group of Companies, fornecedor líder de seguros e resseguros de danos comerciais com ativos consolidados em mais de 2.2 mil milhões de euros.

#### **1.6. Objetivos**

Tendo em conta a finalidade deste projeto, desenvolvimento de uma aplicação móvel para melhorar o processo de vendas da Combined Insurance, podemos apresentar os objetivos a alcançar em cada fase, respetivas tarefas e resultados.

O primeiro objetivo consiste na revisão literária acerca da computação móvel e a sua aplicabilidade e impacto nas organizações.

O segundo objetivo consiste em enquadrar e clarificar os conceitos associados ao desenvolvimento de aplicações para Android.

O terceiro objetivo consiste na caracterização do processo atual e do processo otimizado.

O quarto objetivo consiste em desenvolver a arquitetura com base nos requisitos levantados no terceiro objetivo.

O quinto objetivo consiste no desenvolvimento de uma aplicação com base no modelo proposto.

O sexto objetivo consiste na validação e testes da aplicação desenvolvida.

O sétimo objetivo consiste na proposta de trabalho futuro.

Objetivo	Tarefa	Resultado
Revisão da literatura associada à computação móvel	Revisão teórica/bibliográfica	Enquadramento conceptual dos conceitos associados à computação móvel
Revisão dos conceitos associados ao desenvolvimento de aplicações em Android	Revisão teórica/bibliográfica	Enquadramento conceptual dos conceitos associados ao desenvolvimento de aplicações em Android
Conceção e desenho do modelo	Construção de um protótipo	Arquitetura do modelo
Implementação de uma aplicação com base no modelo proposto	Construção de um protótipo	Aplicação com base no modelo proposto
Validação do modelo	Formulação e proposta de trabalho futuro	Aplicação com base no modelo proposto
Evolução concetual		Projeto de trabalho futuro

### **1.7. Estrutura do documento**

Este documento encontra-se organizado da seguinte forma: no capítulo 2 são apresentados os principais conceitos para uma correta compreensão do objeto de estudo. O capítulo 3 descreve a análise efetuada aos processos e define os requisitos do sistema a implementar. O capítulo 4 trata da arquitetura do sistema a implementar e da estrutura de dados que suporta todo o sistema. No capítulo 5 são identificados aspetos importantes das tecnologias de desenvolvimento e são avaliados os resultados dos testes feitos tendo como termo de comparação os indicadores prévios relativos aos processos em estudo. Por último, o capítulo 6 apresenta as considerações finais e as propostas para trabalho futuro.

## **2. Fundamentação teórica e metodológica**

Este capítulo tem como objetivo o enquadramento concetual dos problemas associados à adoção de SFA e de tecnologia móvel por parte das empresas.

### **2.1. Tecnologia Móvel**

O primeiro telemóvel surgiu em 1973 e apesar de ter sido inventado nesse ano só começou a ser comercializado dez anos mais tarde. Inicialmente o sinal era analógico, designado como sendo da primeira geração, e é simbolizado com a sigla 1G. Os equipamentos pertencentes a esta geração só permitiam efetuar e receber chamadas.

No fim dos anos 80, começaram a ser comercializados equipamentos da segunda geração, já capacitados para enviar e receber mensagens, *roaming* e *voice mail*. Estes equipamentos já suportavam o GSM.

Os equipamentos da terceira geração (3G) surgiram já com funcionalidades de multimédia, como a capacidade de tirar fotografias, capturar imagem e som.

Desde a década de 90 até ao presente, temos vindo a vivenciar um grande e rápido desenvolvimento e proliferação de tecnologias para a comunicação entre dispositivos móveis. A popularização destas tecnologias e dispositivos tem permitido o acesso a informações remotas onde quer que se esteja fisicamente, disponibilizando assim uma grande variedade de serviços e aplicações aos utilizadores.

A necessidade de trocar informações certas no momento e local certo, impulsionou o desenvolvimento de tecnologias da informação e comunicação. As tecnologias digitais móveis intensificam a criação de “territórios informacionais” como novas formas de controlo, de convergência entre as mobilidades físicas e informacionais e dão um novo significado a espaço e lugar.

“Criada como um meio para a libertação, nos primeiros anos da sua existência global, a Internet parecia pressagiar uma nova era de libertação. Os governos podiam fazer muito pouco para controlar fluxos de comunicação capazes de transcender a geografia e, portanto, as fronteiras políticas (...). É um terreno controverso em que se está a disputar a nova e fundamental batalha a favor da liberdade na Era da Informação (...). Dos interesses partilhados pelo comércio e pelos governos surgiu uma variedade de tecnologias de controlo.” (Castells, 2001).

Fatores sociais, jurídicos, económicos, geográficos e políticos da sociedade são influenciados pela tecnologia que vem sendo desenvolvida à medida que a própria sociedade se capacita, não só para a utilizar mas também para a transformar. Neste contexto, as pessoas e as organizações são pressionadas e reagem de forma a adaptarem-se e a estarem ativas num mercado complexo, global e supercompetitivo.

Com o passar do tempo a mobilidade poderá passar a não se restringir apenas aos telemóveis, mas essencialmente a arquiteturas de comunicação que permitam maior segurança, flexibilidade e portabilidade de informação.

A plena ligação e comunicação caracterizou-se numa primeira fase, pelo surgimento da Internet através de sistemas distribuídos, e está a despontar para uma fase em que esta rede de computadores assumirá mobilidade.

A tecnologia computacional facilita a organização e administração de informações, contribuindo para aumentar a produtividade do serviço. O que se espera de um sistema computadorizado é que ele forneça o maior número de benefícios possíveis.

Segundo Rivera (2008), a tecnologia móvel melhora a precisão dos dados, são eficientes para a captura de dados e reduzem o tempo gasto para a digitação da informação.

Por outro lado, Curl e Robinson (2008) destacam algumas deficiências associadas a esta tecnologia, tais como perda de dados, duração da carga de bateria, algumas limitações de memória, tamanho do ecrã e dificuldades de utilização do equipamento.

## **2.2. Comunicação e Serviços Móveis nas Empresas**

De acordo com Heck (2004), permitir o acesso à informação, em qualquer altura e lugar, realçar a capacidade de decisão e criar um ambiente centrado no utilizador são exemplos da computação móvel nas empresas.

Segundo Gribbins (2003), as TIC móveis fazem com que processos específicos nas empresas sejam mais eficazes e eficientes, criam uma abordagem mais personalizada no acesso à informação, tomada de decisão e comunicação do que as tecnologias por cabo.

As TIC móveis, têm um número de características únicas, nas quais se incluem ubiquidade, conectividade, acessibilidade, acesso, portabilidade e localização (Junglas & Watson, 2003).

De acordo com Perry (2001) as características mais valorizadas pelos utilizadores de TIC móveis são a conveniência e eficiência. Kalakota & Robinson (2002) dão o exemplo de profissionais que viajam muito, e que podem utilizar o tempo morto em aeroportos e hotéis mais eficientemente com recurso às TIC móveis, consultando assim importante informação corporativa.

Em suma, as TIC móveis aplicadas às áreas funcionais certas e às pessoas certas conduzem as empresas para maior agilidade, adaptabilidade, tempo real e eficiência nos custos (Gribbins, 2003).

“As redes de comunicação sem fios estão a difundir-se em todo o mundo mais rapidamente que qualquer outra tecnologia de comunicação ao longo da história” (Castells, 2007).

### **3. Levantamento e caracterização da situação atual (AS-IS)**

Este capítulo foca-se no processo de vendas que caracteriza a situação atual da Combined Insurance, a caracterização é decomposta em dois pontos, o Ciclo de Gestão Administrativa e Comercial e o Preenchimento de Apólices. Tanto num caso como no outro, a implementação de um SFA trará melhorias significativas e aumento de eficiência dos processos.

As principais dificuldades encontradas no processo de vendas atual são o isolamento do atual sistema de gestão, o uso excessivo de papel e uma grande quantidade de tarefas repetitivas que consomem muito mais tempo do que deviam sem que adicionem valor algum.

O modo de funcionamento do sistema de vendas em vigor na Combined Insurance, é um sistema moroso e muito suscetível a enganos.

#### **3.1. Ciclo de Gestão Administrativa e Comercial**

Atualmente todo o material de trabalho disponibilizado aos Agentes Comerciais é de suporte físico, desde os *flyers* de vendas, às apólices e aos manuais de subscrição.

Toda a gestão e controlo destes materiais têm elevados custos em termos financeiros e recursos humanos alocados. Os comerciais estão dispersos por todo o Portugal Continental e ilhas e a partir do escritório todo o envio de material para os Agentes é feito por correio, correndo o risco de ser extraviado.

O mesmo acontece quando os Comerciais têm de enviar a produção semanal para o escritório, neste caso o risco aumenta consideravelmente para a Combined Insurance, visto que são contratos assinados e pagos pelo cliente.

Após a receção da produção, a equipa de *Production Processing* instalada no escritório, insere toda a produção no sistema de gestão central da companhia (AS400).

O quadro abaixo ilustra a morosidade de todo este processo.

Tarefas	Semana 1					Semana 2					Semana 3				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1. Realizar Venda	█	█	█	█	█										
2. Recolher Produção						█									
3. Enviar Produção							█								
4. Verificar Produção								█	█						
4. Inserir Produção									█	█					
5. Exportar para AS400										█					
6. Pagar Comissão											█	█			

Desde logo pode-se começar a identificar trabalho redundante, na medida em que os contratos são preenchidos pelo Agente comercial junto do cliente e depois a equipa de *Production Processing* volta a inserir essa mesma informação no sistema de gestão central da companhia.

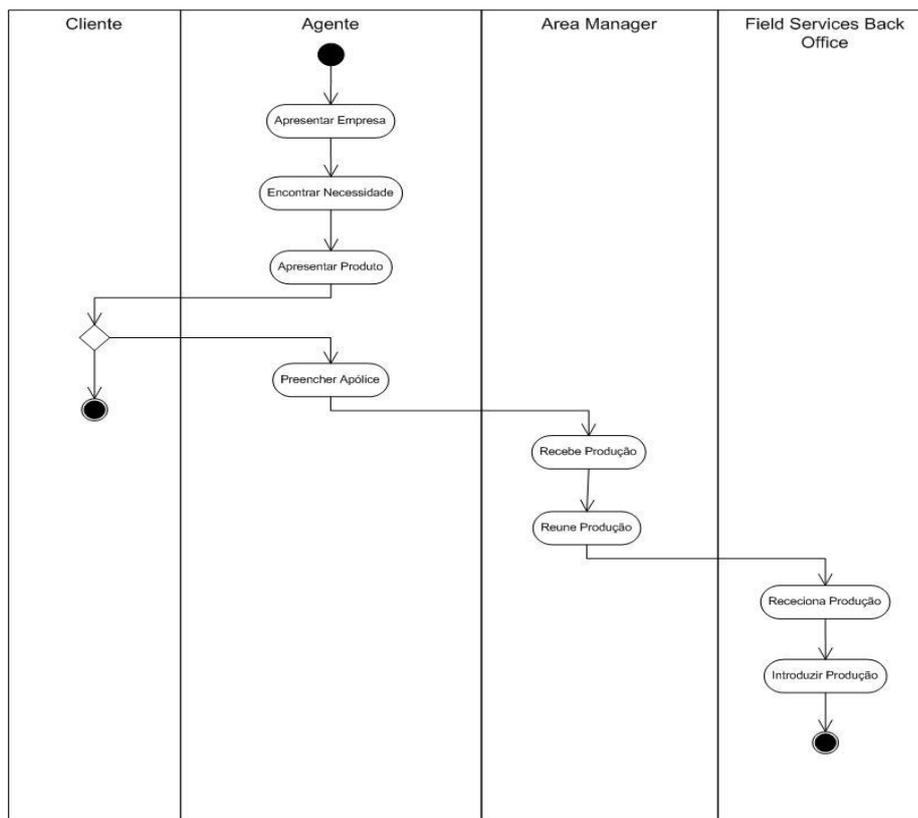


Fig.1 - Diagrama Atividades AS-IS

Durante o processo de observação, foi verificado que em média um colaborador da equipa de *Production Processing* demora cerca de três minutos e trinta segundos a inserir uma apólice no sistema.

Durante o ano de 2012, foram inseridas no sistema 42.874 apólices no AS400, o que dá em média 357 dias a inserir apólices no AS400, isto se fosse só um colaborador. A equipa de *Production Processing* tem atualmente 6 colaboradores a inserir apólices no sistema o que dá em média por colaborador 59,5 dias para a inserção de todas as apólices vendidas no ano de 2012.

### **3.2. Preenchimento das apólices**

Para a subscrição das apólices o Agente da Combined Insurance tem de preencher na apólice os dados pessoais do cliente e no caso das apólices de Saúde alguma informação sobre a sua condição médica.

Neste capítulo foca-se na caracterização do estado atual do preenchimento das apólices e identificam-se melhorias que poderiam ser feitas com a implementação de um SFA.

A Combined Insurance comercializa cinco produtos:

1. Plano de Hospitalização por Acidente (AHCP)
2. Plano de Incapacidade por Acidente (ADP)
3. Plano de Hospitalização por Doença (SHIP)
4. Plano de Incapacidade por Acidente (SIP)
5. Plano de Proteção no Cancro (CSP)

Os produtos têm informação comum mas também há informação específica para cada produto. O quadro abaixo representa resumidamente a informação necessária a cada produto:

Elementos	Produtos				
	AHCP	ADP	SHIP	SIP	CSP
Plano	x	x	x	x	x
Tomador do Seguro	x	x	x	x	x
Pessoa Segura	x	x	x	x	x
Modo de Pagamento	x	x	x	x	x
Débito Bancário	x	x	x	x	x
Questionário SaS SIP				x	
Questionário SaS CSP					x
Questionário SaS SHIP			x		
Beneficiários	x				

Como representado no quadro acima, grande parte da informação preenchida é comum a todas as apólices. Uma das grandes dificuldades dos Agentes durante o processo de preenchimento das apólices, é que caso o Cliente queira subscrever os cinco produtos, o Agente terá de preencher a mesma informação nas cinco apólices diferentes.

Outras das dificuldades encontradas é a não validação dos dados *on time*, que só é feita quando as apólices chegam ao escritório da Combined Insurance e são inseridas no sistema. Semanalmente em média ficam 14% das apólices pendentes. O gráfico abaixo mostra as razões pelas quais as apólices ficam pendentes.

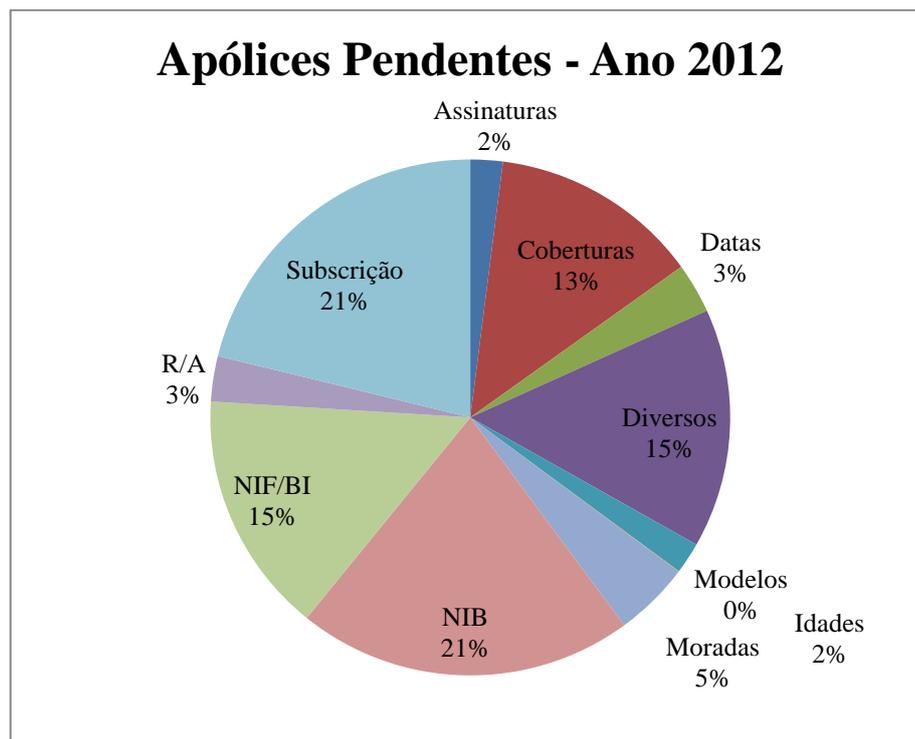


Fig.2 - Apólices Pendentes 2012

A redução das apólices que ficam pendentes por falta de dados e/ou dados incorretos é um dos grandes objetivos da implementação de um SFA. Uma apólice pendente gera várias desvantagens, nomeadamente, a desmotivação ao Agente por ficar com a comissão da apólice retida, o Agente é obrigado a gastar tempo que podia ser canalizado para mais visitas a clientes na resolução dessas apólices e o trabalho administrativo aumenta com a gestão de apólices pendentes.

#### **Problemas existentes no sistema atual:**

1. Muito moroso;
2. Visto que as apólices são preenchidas em papel, ocorrem enganos que só são verificados pela equipa de *Production Processing*. A consequência é que os comerciais ficam com a comissão referente a essa apólice retida, até que a situação seja resolvida;
3. Muito trabalho manual;
4. Grande distância temporal entre a venda e o pagamento da comissão;

5. Trabalho redundante;
6. Falhas nos serviços de correios, extravio.

**Foram identificadas as seguintes oportunidades de melhoria:**

1. Preenchimento de dados com validação *on time*
2. Guardar informação que pode ser pré preenchida
3. Guardar informação para posterior utilização
4. Apresentação ao cliente feita de acordo com as Boas Práticas

#### **4. Desenvolvimento e avaliação do protótipo**

Este capítulo foca-se na escolha do sistema que serviu de base ao desenvolvimento do protótipo e a sua fundamentação. São também abordados os aspetos relacionados com a modelação das bases de dados, com o desenvolvimento da aplicação e com a sua avaliação/validação.

##### **4.1. Escolha do sistema**

Inicialmente começou-se por explorar uma solução baseada numa plataforma *web*. Esta solução apresentava a vantagem de ser multiplataforma e se fosse baseado num CMS, possibilitaria, em trabalho futuro, a adoção de funcionalidades de maior interação com os utilizadores com relativa facilidade.

O que levou ao abandono desta solução, foi o facto de um dos requisitos ser a possibilidade de funcionamento também em modo *offline*. Como a força de vendas vai estar equipada com *tablets* e poderão não ter acesso à *internet* no sítio em que se encontrem, esta solução poderia não funcionar.

A escolha acabou por recair no desenvolvimento de uma aplicação nativa para Android. Ponderou-se também o desenvolvimento para iPad mas foi escolhido o sistema Android devido a:

- Ser uma plataforma aberta.
- Não ter custos com licenças.

- Maior leque de opções em termos de equipamentos.
- Possibilidade de compilar a aplicação em diversas plataformas, para iOS fica-se restringida ao Mac OSX.

#### **4.2. Proposta de protótipo e arquitetura**

A arquitetura tecnológica serve para definir os tipos de tecnologias utilizadas num determinado sistema.

A arquitetura do sistema proposto divide-se em três módulos.

O primeiro é a aplicação (). Esta utiliza o Android como sistema de operativo e Java como linguagem de programação, usando bibliotecas específicas Android, presentes no SDK do mesmo e fornecidas pela Google para controlo do dispositivo.

Os restantes módulos são soluções já desenvolvidas e mantidas pela Combined Insurance com as quais a aplicação comunica. O primeiro destes é o AS400, que é o sistema central da companhia. O outro módulo é um *web service* que irá funcionar como intermediário entre o AS400 e o protótipo. Este módulo tem como principal objetivo o tratamento dos dados recebidos através da aplicação e, após o correto tratamento destes, irá realizar o envio dos dados resultantes para o AS400.

Na comunicação entre a aplicação e o servidor da Combined Insurance são utilizados *web services*. *Web service* é uma solução utilizada para distribuição de serviços em que os seus componentes são independentes de plataforma e permitem a interoperabilidade entre aplicações (Booth, 2004)

Essa independência de plataforma deve-se ao *standard* (aberto) utilizado para a troca de mensagens, neste caso o JSON (Fig. 3). JSON consiste num formato de texto para a serialização de dados estruturados com o objetivo de ser simples e portátil. Fazendo uma comparação direta com o XML, o JSON tem vantagens em termos de tamanho e de *overhead* na estrutura de dados e também no facto de ser mais facilmente gerado e tratado pelas aplicações. Tal como o XML, é um formato baseado em texto.

Uma das dificuldades que os Agentes têm atualmente é a obrigatoriedade de preencher os dados dos clientes em todas as apólices.

Foi então criado um novo conceito, chamado de “Apresentação”. Cada visita a um cliente é uma apresentação. Todos os dados introduzidos referentes ao cliente são associados a uma apresentação para que seja possível a sua reutilização caso o cliente queira mais que um produto.

A possibilidade da aplicação trabalhar em modo *offline* permite aos comerciais realizarem o seu trabalho sem restrições. Esta dificuldade foi ultrapassada com a criação de um serviço transparente para o utilizador, que vai verificando a existência de apólices na aplicação que ainda não foram submetidas para o servidor, e se existirem tenta automaticamente submeter as mesmas. Este é um serviço transparente para o utilizador.

Esta funcionalidade levantou outra questão que foi a proteção dos dados. Guardar informação de cliente num equipamento móvel que se pode extraviar, criou a necessidade de implementação de uma biblioteca, o SQLCipher, que encripta tanto a estrutura de dados como os próprios dados guardados na base de dados SQLite.

A comunicação entre a aplicação e o servidor da Combined Insurance é feita através de um *web service* e a informação é enviada através de JSON tal como se pode ver na arquitetura.

A arquitetura tem o objetivo de apresentar uma abstração dos componentes da aplicação, o conjunto de funcionalidades que são encapsuladas e que têm algum tipo de responsabilidade dentro da aplicação e os conetores. Esses conetores são responsáveis pela coordenação e comunicação entre os componentes.

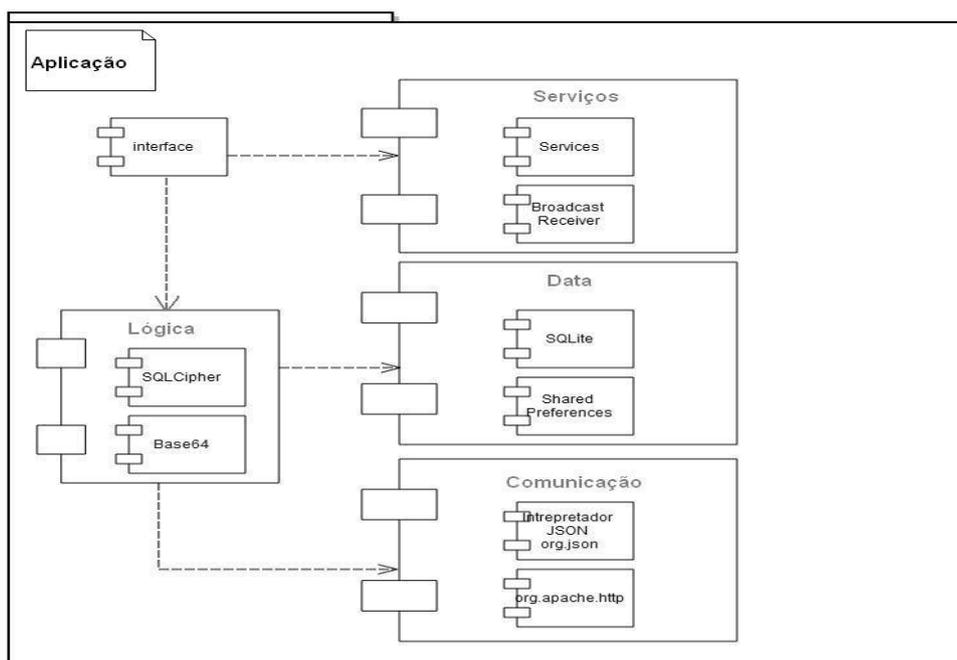


Fig.3 - Diagrama componentes - Arquitetura geral

Através da figura acima é possível visualizar os diversos componentes da aplicação e como comunicam entre si. São mostradas também as livrarias internas e externas que foram importantes para o desenvolvimento da aplicação. Para um maior detalhe são descritos cada um deles:

1. **Interface** – Responsável por agregar as classes responsáveis por controlar os ecrãs da aplicação, ou seja, as *Activities* que fazem a interação com o utilizador.
2. **Lógica** – Contém as classes responsáveis pelo acesso aos componentes de Data e de Comunicação.
3. **Data** – Este componente é responsável por agregar as classes responsáveis pelas operações no SQLite e SharedPreferences.
4. **Comunicação** – Realiza as funções de comunicação com o servidor e possui as classes que manipulam as chamadas e tratamento das respostas do servidor, ou seja, criação e manipulação dos objetos JSON criados para a comunicação entre a aplicação e servidor.

### 4.3. Desenvolvimento do Protótipo

Com as oportunidades de melhoria identificadas, foram modelados vários diagramas tais como, diagramas de componentes, diagramas de caso de uso, diagramas de atividades, diagramas de sequência e o modelo relacional da estrutura de dados que suporta toda a aplicação.

#### 4.3.1. Descrição dos Casos de Uso e atividades

Tendo em conta as oportunidades de melhoria identificadas foi modelado o diagrama de atividades e os diagramas de caso de uso.

O diagrama de atividades com o desenho do processo de acordo com as oportunidades de melhoria identificadas é o seguinte:

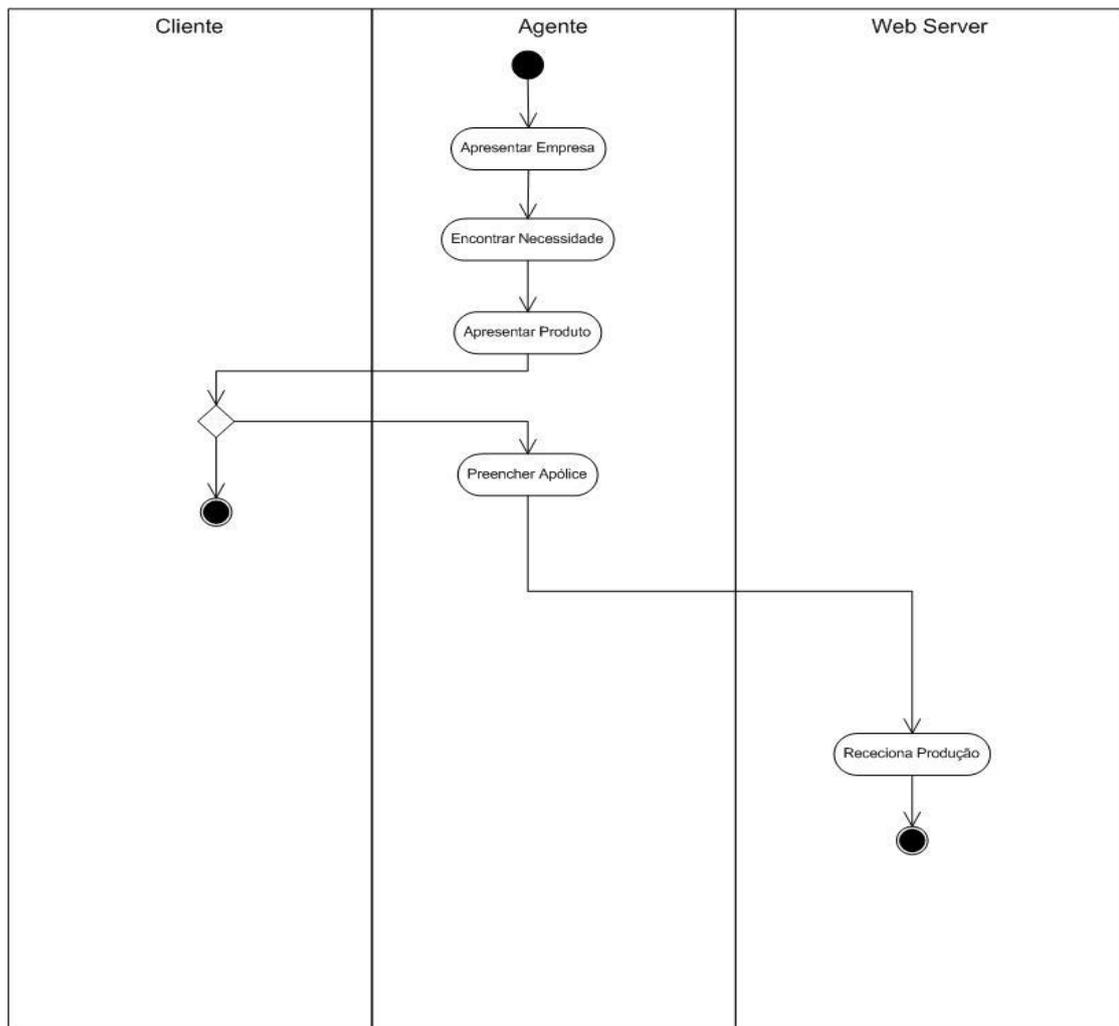


Fig.4 - Diagrama Atividades To-Be

O diagrama de caso de uso do sistema proposto tem um Ator denominado de “Utilizador”. O “Utilizador” poderá realizar quatro ações sobre o sistema.

1. Começar uma nova apresentação que implica sempre a criação na aplicação da mesma.
2. Consultar as apresentações salvas anteriormente e que estão guardadas na aplicação para posterior utilização.
3. Consultar as apólices que já foram submetidas através da aplicação.
4. Configurar as definições da aplicação.

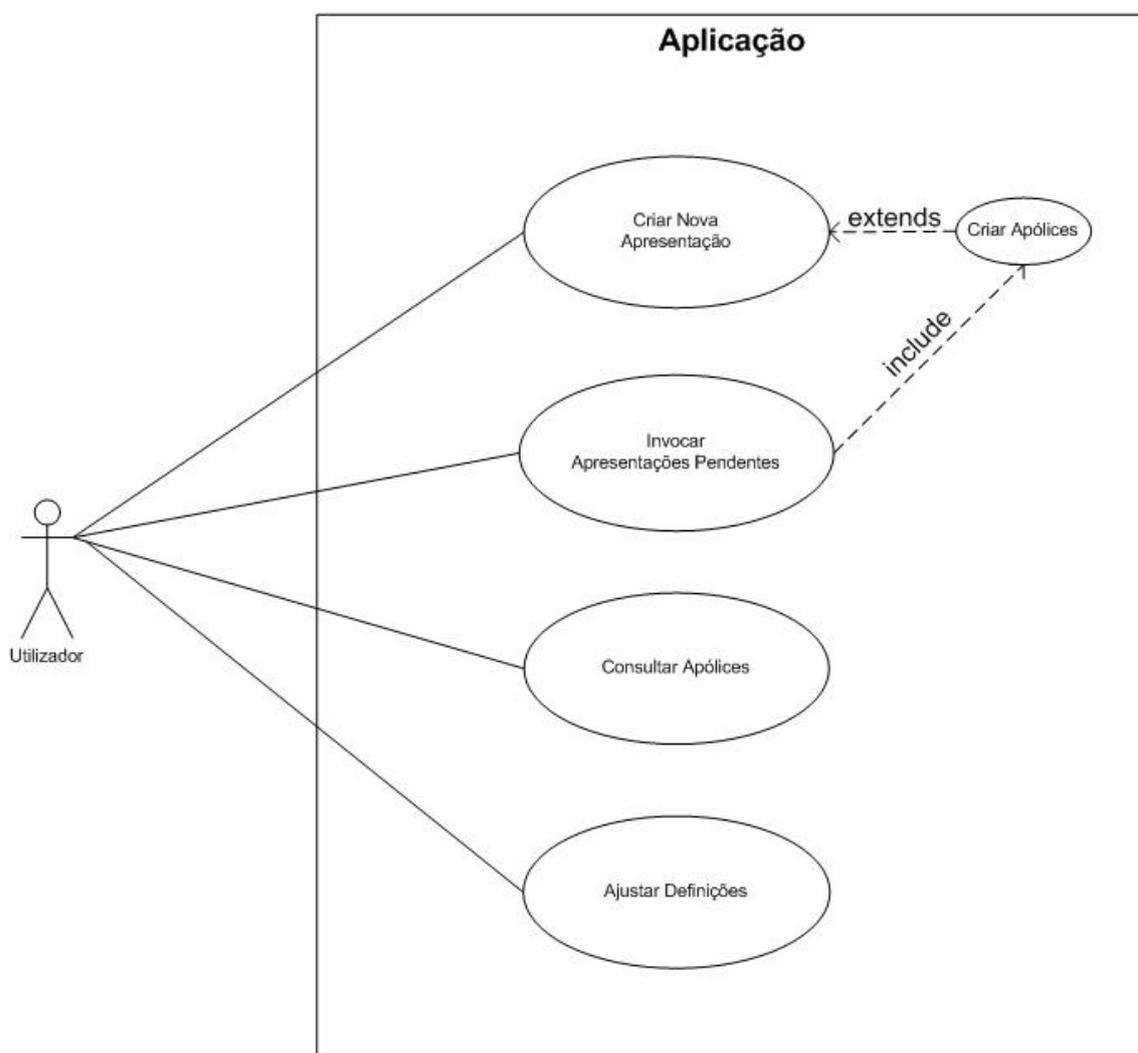


Fig.5 - Caso de Uso

Serão de seguida apresentadas as tabelas descritivas de cada um dos casos de uso apresentados na figura anterior.

<b>Nome</b>	<b>Criar Nova Apresentação</b>
Âmbito	Registo de apresentação a cliente
Finalidade	Permitir ao ator o registo de apresentação e apólices do cliente
Atores	Agente
Pré-Condições	Nada a apresentar
Pós-Condições	A aplicação regista a apresentação e todos os dados relativos a essa apresentação com ou sem apólices fechadas e submetidas para o servidor
Fluxo Típico de Eventos	<ol style="list-style-type: none"> <li>1. O utilizador coloca a aplicação a executar</li> <li>2. O sistema apresenta o ecrã que contém o botão de iniciar apresentação</li> <li>3. A aplicação cria um registo de apresentação</li> </ol>
Fluxo Alternativo	Nada a apresentar

<b>Nome</b>	<b>Invocar Apresentações Pendentes</b>
Âmbito	Consultar e aceder a apresentações previamente realizadas
Finalidade	Permitir ao ator o acesso às apresentações salvas na aplicação e o acesso a uma específica
Atores	Agente
Pré-Condições	Apresentação que o ator quer aceder tem que estar registada na aplicação
Pós-Condições	A aplicação regista a apresentação e todos os dados relativos a essa apresentação (com ou sem apólices fechadas e submetidas para o servidor)
Fluxo Típico de Eventos	<ol style="list-style-type: none"> <li>1. O ator coloca a aplicação a executar</li> <li>2. A aplicação apresenta o ecrã que contém o botão para consultar apresentações salvas</li> <li>3. O ator carrega no botão</li> <li>4. A aplicação apresenta o ecrã que contém a lista de apresentações salvas</li> <li>3. O ator escolhe a apresentação que quer da lista de apresentações salvas</li> </ol>
Fluxo Alternativo	Nada a apresentar

Nome	Consultar Apólices
Âmbito	Consultar as apólices que já foram submetidas para o servidor através da aplicação
Finalidade	Permitir ao ator o acesso às apólices submetidas para o servidor através da aplicação
Atores	Agente
Pré-Condições	Apólice que o ator quer consultar tem de ter sido submetida através da aplicação
Pós-Condições	Nada a apresentar
Fluxo Típico de Eventos	<ol style="list-style-type: none"> <li>1. O ator coloca a aplicação a executar</li> <li>2. A aplicação apresenta o ecrã que contém o botão para consultar apólices submetidas</li> <li>3. O ator carrega no botão</li> <li>4. A aplicação apresenta o ecrã que contém a lista de apólices submetidas</li> <li>3. O ator escolhe a apólice que quer da lista de apresentações salvas para ver os seus detalhes</li> </ol>
Fluxo Alternativo	Nada a apresentar

Nome	Ajustar Definições
Âmbito	Consultar e alterar as definições da aplicação, tais como dados pessoais do ator para ter informação pré preenchida aquando do preenchimento das apólices
Finalidade	Permitir ao ator o acesso às definições da aplicação
Atores	Agente
Pré-Condições	Nada a apresentar
Pós-Condições	Definições ajustadas
Fluxo Típico de Eventos	<ol style="list-style-type: none"> <li>1. O ator coloca a aplicação a executar</li> <li>2. A aplicação apresenta o ecrã que contém o botão para aceder às definições</li> <li>3. O ator carrega no botão</li> <li>4. A aplicação apresenta o ecrã que permite a inserção da informação</li> <li>3. O ator salvar as definições</li> </ol>
Fluxo Alternativo	Nada a apresentar

O diagrama de atividades detalhado abaixo, representa o fluxo normal da aplicação, tendo em conta as *use cases* e os requisitos referidos.

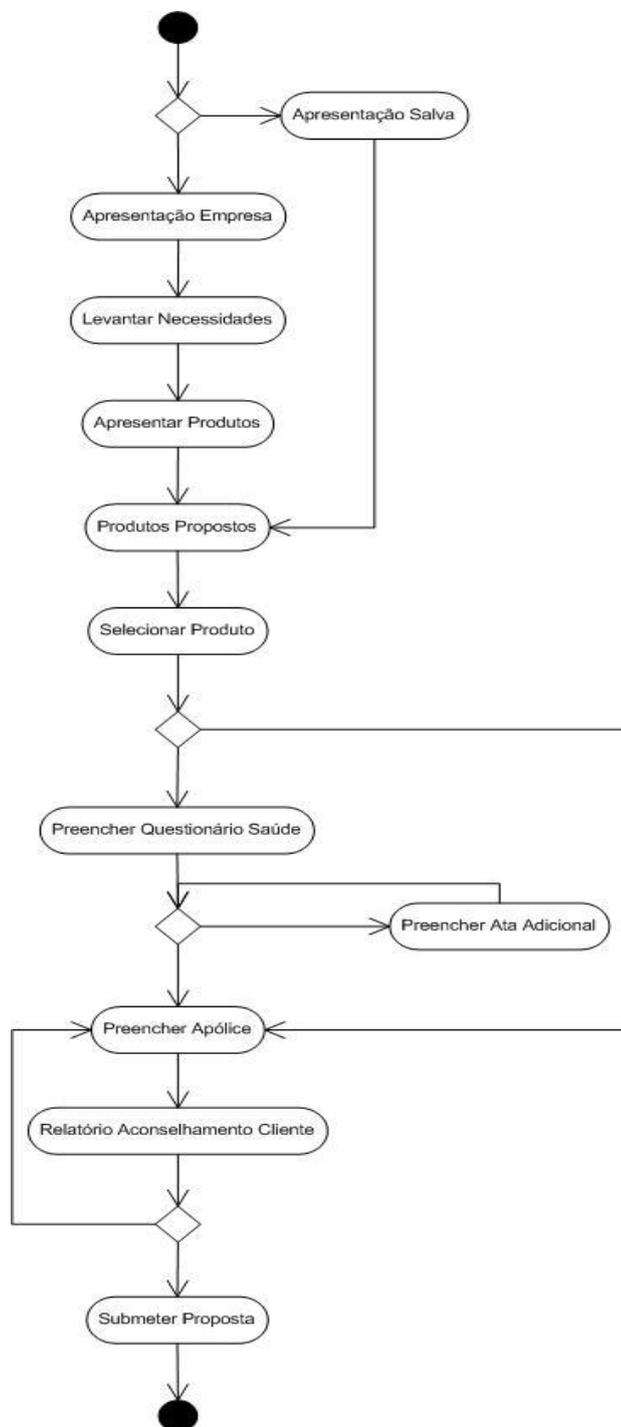


Fig.6 - Diagrama Atividades Fluxo da Aplicação

O seguinte diagrama de sequência mostra como os vários objetos interagem uns com os outros e as mensagens que trocam entre eles nas fases de preenchimento dos Questionários de Saúde, Preencher Apólices e Relatório de Aconselhamento ao Cliente.

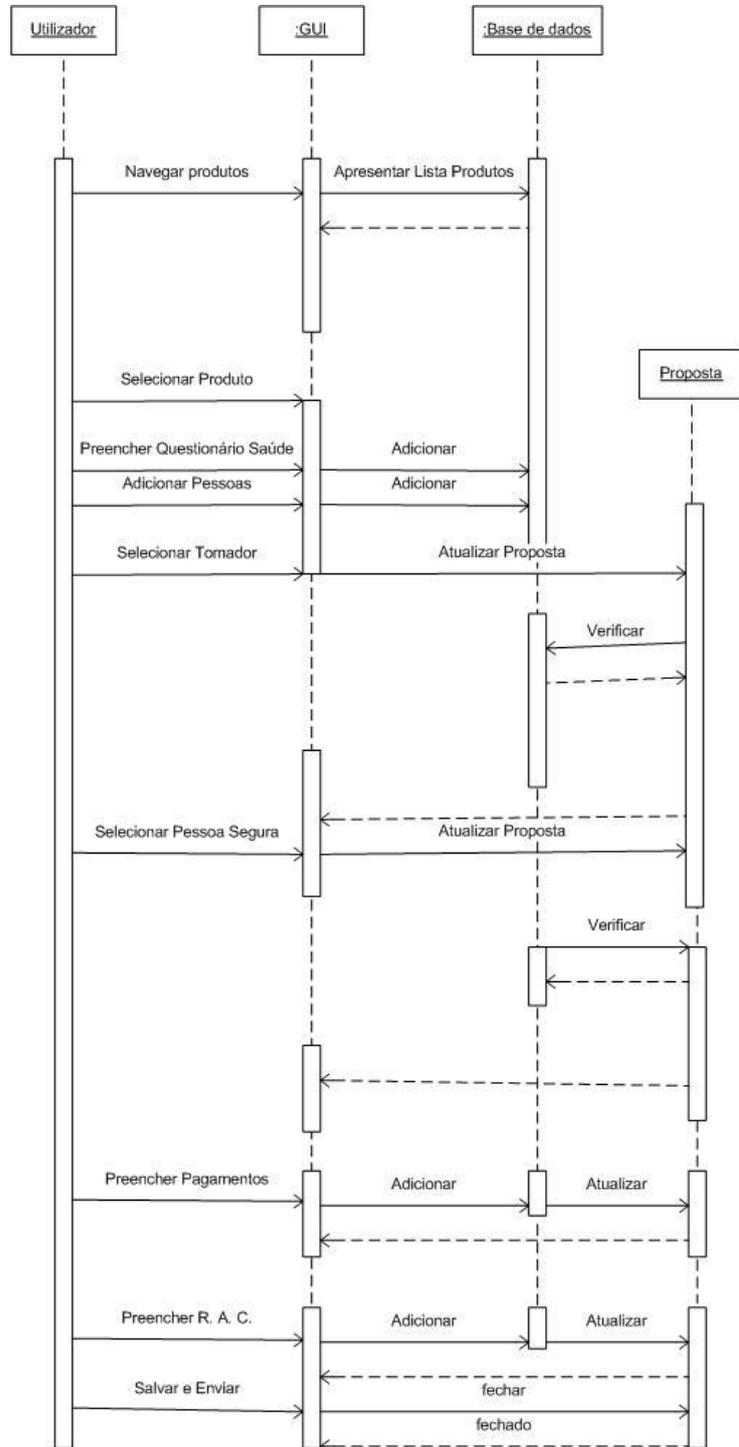


Fig.7 - Diagrama de Sequência Produto Proposto

#### 4.3.2. Aspetos gráficos e interação com o sistema operativo

Não sendo um dos requisitos, não houve grandes preocupações com o aspeto gráfico da aplicação. Os aspetos relacionados com a cor, o tipo e o tamanho de letra ou alguns componentes de visualização foram trabalhados de forma muito simples e de acordo com as normas da Combined Insurance pelo menos nas fases de preenchimento de dados.

A única preocupação em termos gráficos que houve foi o desenvolvimento das funcionalidades de mudar de página com o arrastar do dedo (*swipe*), *zoom-in* e *zoom-out* aquando da apresentação da Combined Insurance e de cada um dos produtos.

Um dos requisitos da aplicação é a possibilidade dos clientes poderem assinar as apólices para celebração do contrato. O desenvolvimento desta funcionalidade centra-se na interação entre a aplicação e o sistema operativo Android.

Na figura seguinte podemos observar este mecanismo, onde é possível verificar a interação entre vários componentes da aplicação e o sistema operativo.

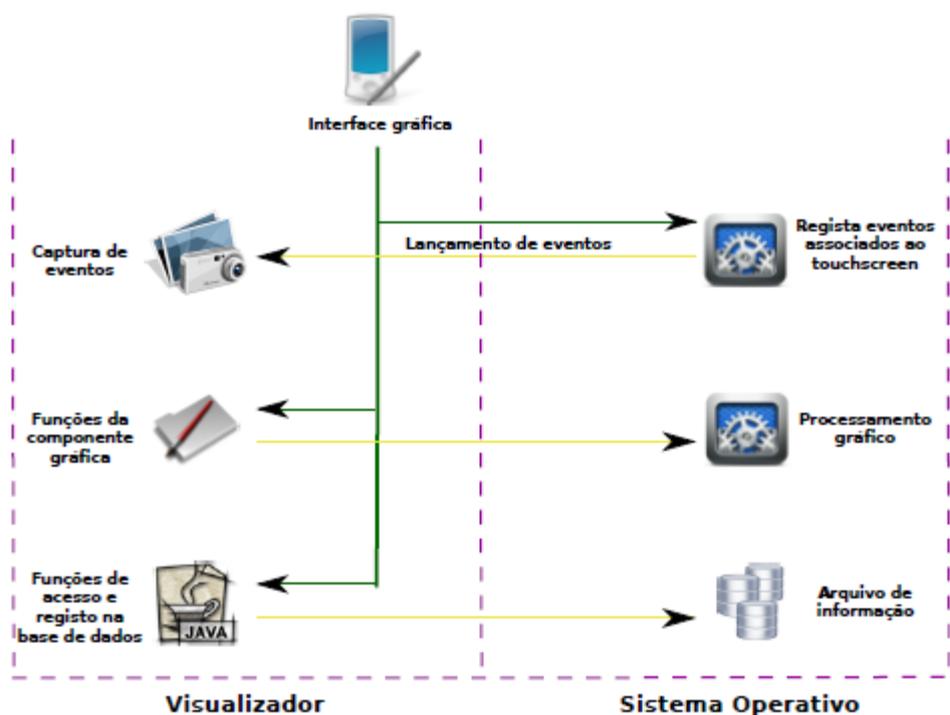


Fig.8 - Interação entre a aplicação e Sistema Operativo Android - Adaptado de (Gonçalo Lopes, 2010)

Na imagem acima a “Captura de eventos” encontra-se relacionado com o *touchscreen* e teclado do aparelho. É responsabilidade do sistema operativo a deteção da existência de um toque no ecrã, definindo a posição do mesmo e classificando-o consoante o tipo de evento.

Existem três eventos associados ao *touchscreen*, “ACTION\_DOWN”, “ACTION\_UP” e “ACTION\_MOVE”. Um simples toque no ecrã irá gerar um evento ACTION\_DOWN, seguido de um ACTION\_UP, enquanto um toque associado a um movimento (toque contínuo no ecrã) irá gerar para além dos eventos anteriormente referidos, vários eventos ACTION\_MOVE. O sistema operativo irá indicar a existência do evento à aplicação assim como o tipo de evento. Cabe então à aplicação utilizar as informações recebidas e dar o seguimento correto ao movimento detetado.

Em baixo exemplo de código exemplificativo de captura de eventos para a recolha da assinatura:

```
@Override
public Boolean onTouchEvent(MotionEvent event)
{
    float x = event.getX();
    float y = event.getY();

    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            touch_start(x, y);
            invalidate();
            break;
        case MotionEvent.ACTION_MOVE:
            touch_move(x, y);
            invalidate();
            break;
        case MotionEvent.ACTION_UP:
            touch_up();
            invalidate();
            break;
    }
    return true;
}
```

Após o cliente assinar é gerado um ficheiro PNG temporário, que é apenas convertido em texto através de Base64. O ficheiro PNG é convertido para ser possível que seja enviado através de um objeto JSON na comunicação entre a aplicação e o *webservice*.

As “Funções de componente gráfica” mostram a interação entre aplicação e o sistema operativo para a representação gráfica. Esta representação é assegurada por uma rotina que está em constante comunicação com a camada de processamento gráfico do sistema operativo.

As “Funções de acesso e registo na base de dados” estão diretamente ligadas às necessidades do programador porque o Android providência várias opções para registo de dados. Na aplicação foram utilizados duas soluções consoante as necessidades, uma passou pela utilização do SQLite e pela utilização de uma memória associativa via classe SharedPreferences, pois fornece uma *framework* genérica que permite gravar e recuperar pares de valores persistente.



Automatização de processos de vendas com recurso a tecnologia móvel – Sistemas e Tecnologias de Informação

Durante o desenvolvimento do protótipo houve um grande cuidado para que a aplicação tivesse uma pequena curva de aprendizagem e que não fugisse muito ao modelo em papel, tal como foi definido nos requisitos. A mais-valia referida por todos os participantes além da redução de tempo na inserção de dados foi a validação automática dos dados introduzidos, tais como campos de preenchimento obrigatório.

A única funcionalidade onde os utilizadores indicaram maiores dificuldades foi na assinatura digital com recurso a uma caneta para ecrãs capacitivos.

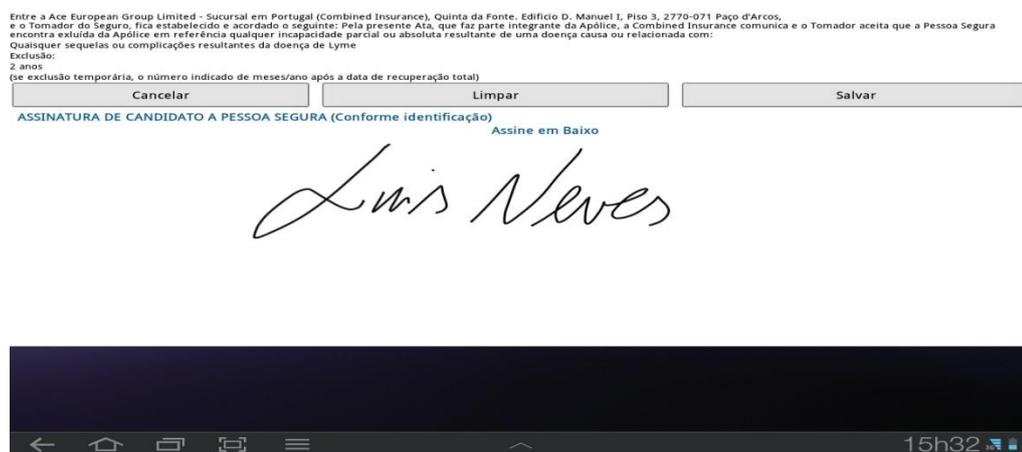


Fig. 10 - Exemplo de assinatura

A assinatura no papel é muito diferente de assinar num *tablet* como é o caso. Principalmente no ecrã capacitivo em que o único ponto de contato com o ecrã só pode ser a caneta porque a aplicação não consegue distinguir o que é o dedo do utilizador ou a caneta. Para um utilizador experiente torna-se mais fácil do que para alguém que não tem muito contato com tecnologia.

A seguinte tabela retrata em média o tempo despendido no preenchimento de apólices por produto:

Elementos	Produtos				
	AHCP	ADP	SHIP	SIP	CSP
Papel	18	17	17	30	27
Tablet	11	10	10	19	18

\* Valores em minutos

Os testes foram realizados com uma amostra de 100 apólices escolhidas aleatoriamente da produção de uma determinada semana.

Logo na primeira apólice, ou seja, sem reutilização da informação previamente inserida já há um ganho de produtividade. Este ganho é em parte justificado pelo preenchimento automático de campos como moradas e localidades através do código postal, banco e balcão através do NIB.

No teste seguinte foi simulado um cliente que queria os cinco produtos da Combined Insurance e os ganhos em termos de tempo foram ainda mais significativos. Como demonstra a tabela abaixo:

Elementos	Produtos				
	AHCP	ADP	SHIP	SIP	CSP
Papel	18	17	17	30	27
Tablet	6	5	4	10	7

\* Valores em minutos

Outro dos testes realizados foi a validação de campos no momento da venda para evitar que apólices fiquem pendentes. Para a realização dos testes foram utilizadas todas as apólices pendentes de uma semana e foram inseridas através da aplicação e chegou-se às seguintes conclusões:

1. Todas as apólices pendentes por falta de dados em campos obrigatórios foram detetadas;
2. Todas as apólices com datas, número de identificação fiscal, número de identificação bancária e código postais errados foram detetadas;

3. Todas as apólices com idades de pessoas seguras que não correspondessem ao produto também foram detetadas;
4. Não foram detetadas apólices em que as assinaturas estavam trocadas. Não foi criado nenhum mecanismo para provar a veracidade ou comparar assinaturas.

Adicionalmente a todos estes benefícios identificados para os Agentes comerciais no preenchimento das apólices, chegou-se à conclusão que praticamente todo o processo administrativo envolvente deixaria de existir, nomeadamente:

1. Envio de material para a força de vendas;
2. Receção e conferência de produção enviadas pelos Agentes;
3. Inserção manual da produção no AS400;
4. Gestão das apólices pendentes.

## 5. Tecnologia de desenvolvimento

Este capítulo introduz uma visão geral sobre a plataforma de desenvolvimento Android. O objetivo é apresentar algumas características como os componentes, bibliotecas e APIs que formam a plataforma.

### 5.1. Android

O Android é uma plataforma *opensource* desenvolvida e distribuída pela OHA e atualmente mantida pela Google. A OHA é um consórcio de diversas empresas, distribuídas pelos setores de telecomunicações, *software* e *hardware*, com o âmbito de desenvolver *open standards* para dispositivos móveis. Fornece suporte a inúmeras tecnologias essenciais num *smartphone*, tais como *touchscreen*, comunicação *bluetooth* e *wireless*, localização GPS.

A plataforma Android é mais do que um sistema operativo, constituindo um completo conjunto de *software* para dispositivos móveis que, para além do sistema de operativo, inclui também importantes aplicações.

O Android inclui diversas aplicações pré-instaladas, como por exemplo a calculadora, alarme, lista de contatos, entre outros.

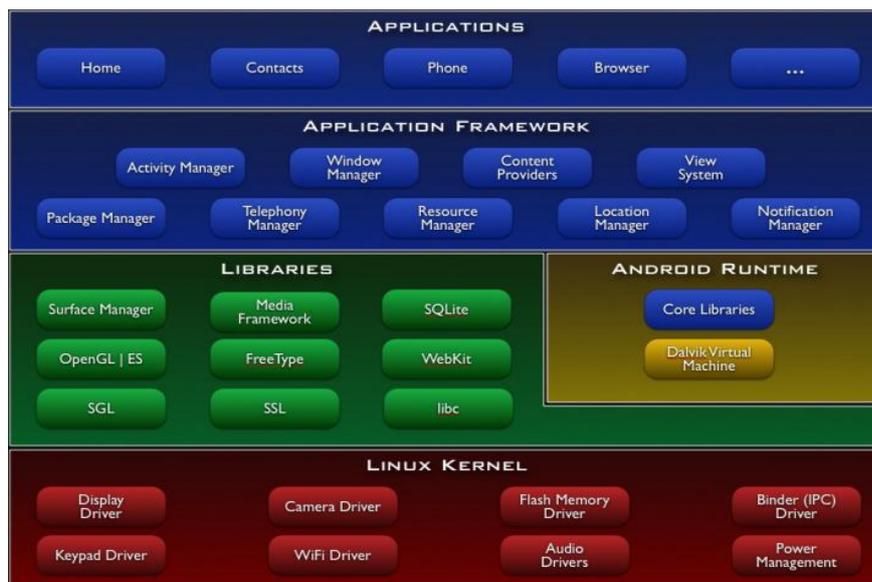


Fig. 12 - Arquitetura Android

A arquitetura Android é composta por várias camadas (Google, 2010) e cada camada possui programas que suportam funções específicas no sistema operativo.

1. **Camada Applications** - todas as aplicações em Android são desenvolvidas em Java, que por sua vez é interpretado pela *virtual machine* Dalvik, são alocadas nesta camada tais como os contatos e o *browser*. As aplicações desenvolvidas por terceiros conseguem aceder às mesmas funções que a equipa do Android.
2. **Camada Application Framework** - nesta camada o sistema operativo fornece um conjunto substancial de pacotes do Java 5 Standard Edition e outras APIs específicas do Android, tais como notificações ou o sistema telefónico.
3. **Camada System Libraries** - todas as livrarias do Android são escritas em C/C++ e fornecem todas as funcionalidades do sistema operativo e são chamadas através de uma interface Java.
4. **Camada Android Runtime** - aqui está incluído um conjunto de livrarias do núcleo Java e também a *virtual machine* Dalvik onde são executados os processos da aplicação. A *virtual machine* Dalvik corre os ficheiros num formato otimizado de baixo consumo de memória, são os ficheiros *Dalvik Executable* (.dex) que resultam da compilação.

Sendo uma plataforma totalmente aberta e direcionada para dispositivos móveis, a OHA tem alguns princípios para suportar estes ideais:

1. **Aberto** – Tenta ser amigo do programador, deixando o aparelho aberto para ser utilizado pelas aplicações. Assim como o Android ser *opensource* e ser possível de implementar em qualquer *hardware*.
2. **Aplicações tratadas por igual** – As aplicações de base do Android e as aplicações externas são tratadas da mesma maneira. Partilham a mesma API, têm o mesmo nível de acesso e podem fazer qualquer coisa que as outras façam.
3. **Sem fronteiras** – Não esconde nenhuma funcionalidade ao programador, este pode facilmente aceder ao sistema telefónico, GPS e contatos.
4. **Rápido e fácil desenvolvimento** – A *framework* de aplicações torna mais fácil algumas funcionalidades como o acesso ao GPS, acelerando assim o processo de desenvolvimento.

## 5.2. Eclipse e o Android SDK

O Eclipse é uma plataforma de integração de ferramentas, um IDE com suporte para várias linguagens, não apenas o Java, e tem uma comunidade de *opensource* muito ativa. O Eclipse fornece uma interface gráfica comum para todas as ferramentas, está implementado para poder ser executado em vários sistemas operativos, como Windows, Linux ou MacOS e tem uma arquitetura para descobrir, carregar e executar *plug-ins* de maneira a dar as ferramentas necessárias a cada utilizador. Foram estes fatores que levaram a Google a desenvolver o *plug-in* necessário para a criação de aplicações para o seu sistema operativo Android, juntamente com o emulador AVD.

O AndroidSDK é um conjunto de ferramentas mantidas pelo Google destinadas à criação de aplicações para o sistema operativo Android e contém bibliotecas: o ADB, o emulador, o AVD, e um conjunto de aplicações exemplo com o respetivo código disponível para consulta. Estas ferramentas podem ser acedidas diretamente a partir da linha de comandos ou através o *plug-in* disponível para o Eclipse, o ADT. O SDK é o único requisito para poder desenvolver para Android.

### 5.3. Técnicas de Programação Android

Apresentam-se algumas das funcionalidades do Android que foram importantes no desenvolvimento do protótipo.

#### 5.3.1. Conceitos gerais -Activities, Intents e AndroidManifest

As *Activities* consistem numa classe e correspondem a um ecrã sendo sua responsabilidade apresentar a interface gráfica e fazer a gestão dos eventos que ocorrem. Uma aplicação pode ser considerada como uma pilha de *Activities*. Para navegar entre ecrãs inicia-se ou termina-se uma *Activity*, sendo que a *Activity* que é apresentada vai para o topo dessa pilha. Para fazer a gestão de estados há um ciclo de vida associado às *Activities*.

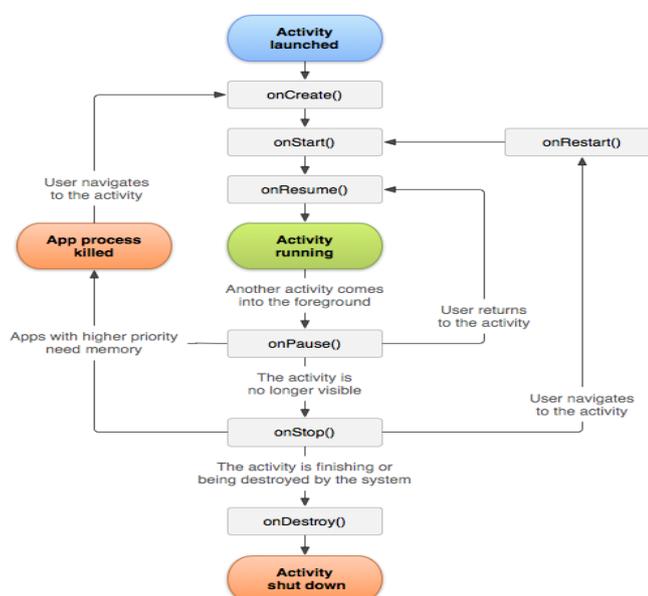


Ilustração 11 - Ciclo de Vida de uma Activity

Os principais métodos do ciclo de vida das *Activities* são:

1. **OnCreate()** – É o primeiro método a ser executado quando uma *Activity* é lançada. Por norma é a responsável por carregar os *layouts* XML ou os *layouts* carregados em *Runtime* e outras operações de inicialização. Neste método, a *Activity* ainda não está a ser apresentada ao utilizador.

2. **OnStart()** – É chamado logo após do onCreate(). Geralmente quando a *Activity* é sobreposta por outra na pilha e retorna a ser apresentada no ecrã este é o método chamado.
3. **OnResume()** – Este método é sempre chamado quando a *Activity* está no topo da pilha. É neste estado que a *Activity* é apresentada no ecrã.
4. **OnPause()** – É o primeiro método a ser executado quando a *Activity* é sobreposta por outra.
5. **OnStop()** – É o último método a ser chamado antes da *Activity* ser considerada “morta”.

Como referido anteriormente, os *layouts* das *Activities* podem ser definidos em *Runtime* ou a partir de ficheiros XML.

Se optar por criar em *Runtime*, é criada uma variável do tipo que se desejar e define-se a suas propriedades. O exemplo que se segue é da criação dum botão com o texto “Ok”.

```
ButtonbtOk = new Button(this);  
  
btOk.setText("Ok: ");
```

No caso de utilizar o ficheiro XML para definição do *layout*, se não for necessário aceder ao componente em *Runtime*, a descrição no ficheiro XML é suficiente. Caso contrário é necessário criar uma variável e associá-la ao componente criado no ficheiro XML, como mostra o exemplo:

```
Button btOk;  
  
btOk= (Button) findViewById(R.id.bt_ok);
```

No decorrer do desenvolvimento da aplicação todos os *layouts* foram criados em ficheiros XML, para a parte lógica da aplicação estar separada das interfaces gráficas.

Para a navegação entre *Activities* normalmente são utilizados *Intents*, que consistem em objetos da classe *Intent* e contêm a descrição da operação a ser realizada. Podem seguir

juntamente com os *Intents* atributos como o *Bundle*, que permite o envio de dados adicionais. Há dois tipos de *Intents*:

1. **Explícitos** – É identificado qual o componente a ser chamado e é chamada a classe que deve correr para que esse componente execute;
2. **Implícitos** – Não é identificado qual o componente, mas contém informação necessária para que o sistema operativo determine qual dos componentes disponíveis melhor se adequa aos requisitos do *Intent*.

Cada projeto Android contém um ficheiro *AndroidManifest.xml*, guardado na raiz do projeto e é obrigatório a sua presença em todos os projetos. É neste ficheiro que fica definida a estrutura, os meta dados e os componentes utilizados da aplicação. É aqui que são declaradas as *Activities*, os *Receivers*, a versão do SDK e as respetivas permissões, como, acesso à Internet ou ao estado da rede.

### 5.3.2. AsyncTask

Na plataforma Android, a *thread* principal de uma *Activity*, também conhecida como *UI thread*, é responsável por processar eventos do utilizador. Caso algum evento executado na *UI thread* leve algum tempo a ser executado é lançado um ANR, que consiste numa caixa de diálogo a perguntar se queremos fechar a aplicação ou esperar que acabe a tarefa. A alternativa para não surgirem ANRs é executar as tarefas em *background*. Uma das maneiras disponíveis de executar tarefas em *background*, é a utilização de *AsyncTasks*. Através destas é possível colocar uma tarefa a executar em *background* e atualizar a interface gráfica, de maneira a transmitir informação ao utilizador. Tem três métodos principais:

1. **onPreExecute():** é chamado antes de a tarefa começar, é executado na *UI thread* e é usado para a preparar, como por exemplo, dar a informação ao utilizador que a tarefa vai ser executada, exibindo uma barra de progresso ou de *loading*.
2. **doInBackground():** este método é chamado logo após o anterior e aqui é colocado todo o código que vai ser executado em *background*. Quando finaliza envia o resultado ao *onPostExecute()*.

3. **onPostExecute():** chamado imediatamente após a tarefa ser finalizada, e atualiza a interface gráfica consoante o resultado recebido. À semelhança do `onPostExecute()`, é executado na UI *thread*.

### 5.3.3. Services

São serviços utilizados por outros componentes que não possuem interface gráfico para o utilizador. Por norma são executados em *background*, são serviços que necessitam de continuar a correr mesmo que o utilizador siga para outra aplicação. Os *Services* expõem uma interface para que outros componentes se possam ligar e executem o que estes oferecem.

### 5.3.4. BroadcastReceivers

Foram utilizados no desenvolvimento diversas vezes mas principalmente para a funcionalidade de submissão automática de apólices.

Os *BroadcastReceivers* são utilizados para a receção de anúncios referentes externos, como por exemplo uma notificação, a chegada de uma mensagem ou que todo o processo *boot* do telemóvel já foi realizado.

### 5.3.5. SQLite

Um dos principais objetivos da aplicação é permitir ao utilizador, mesmo não havendo ligação à *internet*, que continue a executar o seu trabalho.

Partindo deste pressuposto, a aplicação tem uma base de dados local que armazena os dados necessários com recurso à utilização do suporte nativo ao SQLite. Trata-se de uma biblioteca de base de dados em SQL cuja principal função é atuar como um SGBD. Na plataforma Android, cada aplicação pode criar diversas bases de dados, ficando estas localizadas no diretório de dados da aplicação.

## **6. Conclusões**

O objetivo principal deste trabalho era o desenvolvimento de um protótipo que permitisse automatizar o processo de vendas da Combined Insurance de forma a ter ganhos de eficácia e eficiência ao nível dos processos.

Os testes realizados confirmaram a redução do tempo gasto em trabalho administrativo e durante o processo de venda em si.

Ao terminar este trabalho pode-se afirmar que os objetivos propostos foram alcançados com sucesso.

### **6.1. Trabalho Futuro**

Tratando-se de uma prova de conceito, o protótipo desenvolvido tem diversas melhorias que podem ser feitas.

A nível de interface poderá ser desenvolvido um mais apelativo e interativo com o utilizador.

Como referido nos testes, foi identificada uma dificuldade na assinatura. A assinatura num suporte digital deve ser testada e comparada em ecrãs capacitivos e ecrãs resistivos.

Os próximos passos deverão ser o teste da aplicação em ambiente de produção recorrendo a um piloto de testes com um grupo restrito de agentes comerciais, sendo previsível que com uma utilização mais intensiva da aplicação os agentes comerciais poderiam ter mais ideias para melhorar as funcionalidades da aplicação.

A aplicação desenvolvida poderá constituir a base para uma plataforma mais alargada que incorpore mais módulos e funcionalidades para a gestão de todo o ciclo de vendas da Combined Insurance.

## 7. Bibliografia

- Anacom. (s.d.). *Anacom*. Obtido em 06 de 01 de 2012, de Anacom: <http://www.anacom.pt>
- Android Developers*. (s.d.). Obtido em 12 de 04 de 2013, de Android Developers: <http://developer.android.com/reference/android/os/AsyncTask.html>
- Ballard, E. (2007). *Designing the Mobile User Experience*. John Wiley & Sons Ltd.
- Booth, D. (2005). *WS-ARCH*. Obtido em 28 de 12 de 2011, de W3: <http://www.w3.org/tr/ws-arch>
- Brahler, S. (2012). *Analysis of the android architecture*.
- Brooks, C. (2002). *An Introduction to Web Services*.
- Castells, M. (2007). *Comunicação Móvel e Sociedade uma Perspectiva Global*. Comunicação Calouste Gulbenkian.
- Castells, M. (2011). *A Sociedade em Rede, A Era da Informação: Economia, Sociedade e Cultura*. Lisboa: Edições Fundação Calouste Gulbenkian.
- Chen, L. N. (2008). A socio-technical perspective of mobile work. *Information Knowledge Systems Management*, 41-60.
- Chen, M. (2007). *Empowering collaborative commerce with Web services enabled*.
- Chinnici, R. (2002). *Web Services Description Language (WSDL) Version 1.2*. Obtido em 28 de 12 de 2011, de W3: <http://www.w3.org/TR/2002/WD-wsdl12-20020709/>
- Christensen, E. (2001). *Web Services Description Language (WSDL) 1.1*. Obtido em 28 de 12 de 2011, de W3: <http://www.w3.org/TR/wsdl>
- Crosstalk. (2006). *The Journal of Defense Software Engineering*. Crosstalk.
- Curl. (2008). Hand-held computers in clinical audit: A comparison with established paper and pencil methods. *International Journal of Healthcare Quality Assurance*.
- Day, G. (1994). *The capabilities of market-driven organizations*. *Journal of Marketing*.

Engineering, I. o. (1998). *IEEE Recommended Practice for Software Requirements Specifications*. New York: IEEE std.

Fling, B. (2009). *Mobile Design and Development*,. Steven Weiss.

Gondert, S. (1993). *Automation: The 10 biggest mistakes of SFA (and how avoid them)*. Sales and Marketing Management.

Hunter, G., & Perreault Jr., W. (2006). *Sales technology, information effectiveness and sales performance*. Journal of Personal Selling and Sales Management.

Nah, F. S. (2005). The value of mobile applications: a study on a public utility company. In *Communications of the ACM* (pp. 85-90). ACM.

OWASP. (s.d.). *OWASP*. Obtido em 28 de 12 de 2011, de OWASP: [https://www.owasp.org/index.php/Web\\_Services#What\\_are\\_Web\\_Services.3F](https://www.owasp.org/index.php/Web_Services#What_are_Web_Services.3F)

Price, R. (1996). *Technology and strategic advantage*. California Management Review.

Rivera. (2008). Propective, radomized evaluation of a personal digital assistant-based research tool in the emergency department. *Medical Informatic and Decision Making*.

Sommerville, I. (1995). *Software Engineering*. Wokigham: Addison-Wesley.

Stol. (2008). *Challenges in Using Open Source Software in Product Development: A Review of the Literature*.

Ward, J. (1995). *Principles of Information Systems Management*.