



UNIVERSIDADE FEDERAL DE MATO GROSSO
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM
SISTEMAS DE INFORMAÇÃO

**RELATÓRIO DE ESTÁGIO SUPERVISIONADO
AUTOMATIZAÇÃO DO GERENCIAMENTO DE
PROTOCOLOS DO MINISTÉRIO PÚBLICO UTILIZANDO
ANDROID**

ANDRE FELIPE DA COSTA LANA

CUIABÁ – MT

2014

UNIVERSIDADE FEDERAL DE MATO GROSSO
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM
SISTEMAS DE INFORMAÇÃO

**RELÁTÓRIO DE ESTÁGIO SUPERVISIONADO
AUTOMATIZAÇÃO DO GERENCIAMENTO DE
PROTOCOLOS DO MINISTÉRIO PÚBLICO UTILIZANDO
ANDROID**

ANDRE FELIPE DA COSTA LANA

Relatório apresentado Instituto de
Computação da Universidade Federal de
Mato Grosso, para obtenção do título de
Bacharel em Sistemas de Informação.

CUIABÁ – MT

2014

UNIVERSIDADE FEDERAL DE MATO GROSSO
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM
SISTEMAS DE INFORMAÇÃO

ANDRE FELIPE DA COSTA LANA

Relatório de Estágio Supervisionado apresentado à Coordenação do Curso de
Sistemas de Informação como uma das exigências para obtenção do título de
Bacharel em Sistemas de Informação da Universidade Federal de Mato Grosso

Aprovado por:

Prof. MSc. Nilton Hideki Takagi
Instituto de Computação
(Coordenador de Estágios)

Prof. MSc. Daniel Avila Vecchiato
Instituto de Computação
(ORIENTADOR)

Esp. Thiago Sampaio Figueiredo
(SUPERVISOR)

DEDICATÓRIA

À Deus graças por ter me dado a oportunidade de concluir mais esta fase em minha vida, sempre me proporcionando sabedoria nas decisões tomadas.

À minha família, meus pais José e Maria, meu irmãos Hermógenes e Marconni por terem me apoiado não só durante o período do curso de Sistemas de Informação, mas durante toda minha vida.

AGRADECIMENTOS

Agradeço em primeiro lugar, a Deus que iluminou meu caminho durante essa longa caminhada. Ao Ministério Público do Estado de Mato Grosso pelo ambiente profissional, criativo e amigável que proporciona.

A Universidade Federal de Mato Grosso e seu corpo docente que abriram meus horizontes para o mercado de trabalho.

Ao meu orientador MSc. Daniel Avila Vecchiato, pela orientação, apoio e confiança.

Aos meus pais, pelo amor, incentivo e apoio incondicional.

Agradeço a todos que direta ou indiretamente fizeram parte da minha formação.

SUMÁRIO

LISTA DE FIGURAS	7
LISTA DE TABELAS.....	8
LISTA DE SIGLAS E ABREVIATURAS	9
RESUMO	11
1. INTRODUÇÃO	12
1.1 <i>Objetivo Geral</i>	13
1.2 <i>Objetivo específico</i>	13
2. REVISÃO DE LITERATURA	14
2.1 <i>Desenvolvimento para dispositivo móveis com Android</i>	14
2.1.1 <i>Arquitetura Android</i>	14
2.2 <i>Protocolo de segurança HTTPS</i>	16
2.3 <i>Testes automatizados</i>	18
2.4 <i>Serviços Web</i>	20
2.4.1 <i>Comunicação</i>	21
2.5 <i>Desenvolvimento Ágil de Software</i>	23
3. MATERIAS, TÉCNICAS E MÉTODOS	25
3.1 <i>ADT-Bundle</i>	26
3.2 <i>JUnit 4</i>	27
3.3 <i>Emulador Genymotion</i>	27
3.4 <i>JBoss RESTEasy</i>	28
4. RESULTADOS	29
5. DIFICULDADES ENCONTRADAS	43
6. CONCLUSÕES.....	44
7. REFERÊNCIAS BIBLIOGRÁFICAS.....	45

LISTA DE FIGURAS

FIGURA 1 - ARQUITETURA ANDROID (ANDROID DEVELOPERS, 2012).	15
FIGURA 2 - EXEMPLO DE ACESSO VIA HTTPS.....	17
FIGURA 3 - CICLO DE UM TDD (IMOBILIS, 2013).....	19
FIGURA 4 - COMUNICAÇÃO VIA WEB SERVICE (SANTOS, 2013).....	21
FIGURA 5 - EXEMPLO DE TROCA DE MENSAGENS UTILIZANDO ARQUITETURA SOAP (DEVX, 2014)....	22
FIGURA 6 – AMBIENTE DE DESENVOLVIMENTO ADT BUNDLE.....	26
FIGURA 7 - <i>LOGON</i>	29
FIGURA 8 - SELEÇÃO DE LOCAIS.....	30
FIGURA 9 - ÁREA DE TRABALHO	30
FIGURA 10 - ÁREA DE TRABALHO (<i>TABLET</i>).....	31
FIGURA 11 - DETALHE DO PRAZO	32
FIGURA 12 - DETALHE DO PRAZO (<i>TABLET</i>)	33
FIGURA 13 - DETALHE DO PROTOCOLO (ABA DETALHE).....	34
FIGURA 14 - DETALHE DO PROTOCOLO (ABA CLASSIFICAÇÃO)	34
FIGURA 15 - DETALHE DO PROTOCOLO (ABA PARTES)	35
FIGURA 16 - DETALHE DO PROTOCOLO (ABA DOCUMENTOS)	36
FIGURA 17 - DETALHE DO PROTOCOLO (ABA MOVIMENTOS).....	37
FIGURA 18 - DETALHE DO PROTOCOLO NO <i>TABLET</i> (MENU MOVIMENTOS).....	38
FIGURA 19 - DETALHE DO PROTOCOLO NO <i>TABLET</i> (MENU PARTES)	39
FIGURA 20 - DETALHE DO PROTOCOLO NO <i>TABLET</i> (MENU DOCUMENTOS)	40
FIGURA 21 - EXCLUIR PRAZO	41
FIGURA 22 - EXCLUIR PRAZO (<i>TABLET</i>)	42

LISTA DE TABELAS

TABELA 1.....	25
---------------	----

LISTA DE SIGLAS E ABREVIATURAS

ABNT	Associação Brasileira de Normas Técnicas
SIMP	Sistema Integrado do Ministério Público
CPU	<i>Central Processing Unit</i> – Unidade Central de Processamento
API	<i>Application Programming Interface</i> – Interface de Programação de Aplicações
BSD	<i>Berkeley Software Distribution</i> – Berkeley Distribuição de Software
SGL	Bibliotecas do <i>Android Software Development Kit</i> para gráficos 2D
HTTP	<i>Hypertext Transfer Protocol</i> – Protocolo de Transferência de Hipertexto
HTTPS	<i>Hypertext Transfer Protocol Secure</i> – Protocolo de Transferência de Hipertexto Segura
SSL	<i>Secure Socket Layer</i> – Camada de Socket Seguro
TLS	<i>Transport Layer Security</i> – Camada de Transporte Segura
NIST	<i>National Institute of Standards and Technology</i> – Instituto Nacional de Padrões e Tecnologia.
TDD	<i>Test Driven Development</i> – Desenvolvimento Orientado a Testes
W3C	<i>World Wide Web Consortium</i>
SOAP	<i>Simple Object Access Protocol</i> – Protocolo Simples de Acesso a Objetos
XML	<i>eXtensible Markup Language</i>
SMTP	<i>Simple Mail Transfer Protocol</i> – Protocolo de Transferência de Correio Simples
FTP	<i>File Transfer Protocol</i> – Protocolo de Transferência de Arquivos
RMI	<i>Remote Method Invocation</i> – Invocação de Método Remoto
IIOP	<i>Internet Inter-Orb Protocol</i>
REST	<i>Representational state transfer</i>
URL	<i>Uniform Resource Locator</i>

ADT	<i>Android Development Tools</i> – Ferramentas de Desenvolvimento Android
IDE	<i>Integrated Development Environment</i> – Ambiente Integrado de Desenvolvimento
ARM	<i>Advanced RISC Machine</i>
JDK	<i>Java Development Kit</i> – Kit de Desenvolvimento Java
EJB	<i>Enterprise JavaBeans</i>

RESUMO

Este relatório de estágio descreve as atividades realizadas durante o período de estágio supervisionado realizado no Ministério Público de Mato Grosso pela empresa DSS Construção e Tecnologia do discente André Felipe da Costa Lana, supervisionado por Tiago Sampaio Figueiredo e orientado por Daniel Avila Vecchiato.

A atividade executada durante o período de estágio na empresa foi a implementação de um aplicativo para dispositivos móveis que utiliza a plataforma Android para auxiliar no gerenciamento de protocolos pelos funcionários do Ministério Público do Estado de Mato Grosso. Este aplicativo, denominado Sistema Integrado do Ministério Público foi desenvolvido utilizando a linguagem de programação Java, também se fez necessário o uso do framework REStEasy para o desenvolvimento dos serviços web, a biblioteca JUnit para os testes automatizados e o ambiente de desenvolvimento ADT Bundle.

Durante todo o processo de desenvolvimento trabalhamos com o framework ágil de desenvolvimento denominada SCRUM.

1. INTRODUÇÃO

O relatório em questão, trata-se de um relatório de estágio supervisionado onde estarão especificados os conceitos e práticas utilizadas no decorrer do estágio. O estágio supervisionado foi realizado na DSS Construção e Tecnologia com o intuito de desenvolver um aplicativo para dispositivos móveis, especificamente para Android, baseado em uma versão web já existente, que pudesse auxiliar os funcionários do Ministério Público do Estado de Mato Grosso no gerenciamento dos protocolos e seus prazos.

O aplicativo foi desenvolvido desde o início por uma equipe formada por três desenvolvedores, dois analistas, um gerente de projeto e um *Web designer*. O framework de desenvolvimento utilizado durante todo o processo foi o SCRUM, em que de intervalos de 15 dias era disponibilizada uma versão funcional do aplicativo para que os analistas pudessem realizar os testes pertinentes e dar um *feedback* da versão disponibilizada.

Este documento está organizado da seguinte forma: O Capítulo 3 discorre sobre a revisão de literatura que irá detalhar toda a teoria utilizada ao longo do estágio. Logo após, no Capítulo 3, são relatados os materiais, técnicas e métodos que irão especificar as ferramentas utilizadas para a realização dos trabalhos. Os resultados do trabalho realizado no estágio estarão logo depois, no Capítulo 4, demonstrando o aplicativo final com imagens e descrição das funcionalidades. O Capítulo 5 é onde estão descritas as dificuldades encontradas durante a realização do estágio, e por fim temos a conclusão especificando a importância da realização desse estágio supervisionado.

1.1 Objetivo Geral

O objetivo do estágio era automatizar o gerenciamento de protocolos do Ministério Público do estado de Mato Grosso utilizando a plataforma Android, para isso, foram realizadas atividades de codificação do *Front-end* do aplicativo utilizando o código nativo do Android que é o Java, outra atividade foi a implementação do *Back-end* que envolve o desenvolvimento de serviços *web* também desenvolvidos em Java, análise de usabilidade do aplicativo e desenvolvimento dos testes automatizados.

1.2 Objetivo específico

Realizar as seguintes atividades:

- Aplicar de forma prática os conhecimentos adquiridos durante todo o curso de Sistemas de Informação.
- Desenvolver um aplicativo Android que execute em aparelhos com a versão 3.0 ou superior.
- Desenvolver *web service* para que o aplicativo Android desenvolvido possa realizar a comunicação com o servidor.
- Realizar o consumo de *web service* de maneira segura, utilizando protocolo SSL.
- Implementar testes automatizados das funcionalidades desenvolvidas no aplicativo Android.

2. REVISÃO DE LITERATURA

Neste Capítulo estão descritos os principais temas relacionados ao trabalho feito durante o estágio supervisionado.

2.1 Desenvolvimento para dispositivo móveis com Android

A computação móvel é o fator de maior disrupção no desenvolvimento na história da humanidade (NUDELMAN, 2013) e uma das grandes responsáveis por isso é a *Open Handset Alliance*, que é uma aliança de várias empresas com o objetivo de desenvolver padrões para a telefonia móvel. As principais empresas que compõem a *Open Handset Alliance* são: Nvidia, HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile e Google.

A *Open Handset Alliance* tem como líder, a Google e foi criada em 2007, mesmo ano em que criou a plataforma de código aberto Android que atualmente é o Sistema Operacional mais utilizado em smartphones e o Brasil está entre os cinco mercados de maior comercialização de aplicativos Android, junto com a Europa, Índia, Coréia do Sul e Rússia (CANALTECH, 2013). Com todo esse sucesso o Android conquistou o interesse das empresas, desenvolvedores e público em geral.

O framework Android foi desenvolvido em Java e é executado em um kernel Linux. Os aplicativos são desenvolvidos em Java. Segundo (DIMARZIO, 2008) Android inclui também aceleração 3D, motores gráficos, suporte de banco de dados (SQLite), e um navegador web integrado.

2.1.1 Arquitetura Android

A arquitetura da plataforma Android tem por objetivo simplificar a reutilização dos componentes fazendo com que outros programas utilizem as funções desse aplicativo.

Vale lembrar que o desenvolvedor tem acesso total à mesma estrutura de API's usada nos aplicativos nativos, podendo desta forma aproveitá-las conforme

achar (GOOGLE, 2007). Na realidade, Android é uma arquitetura composta por cinco camadas como é mostrado na Figura 1.

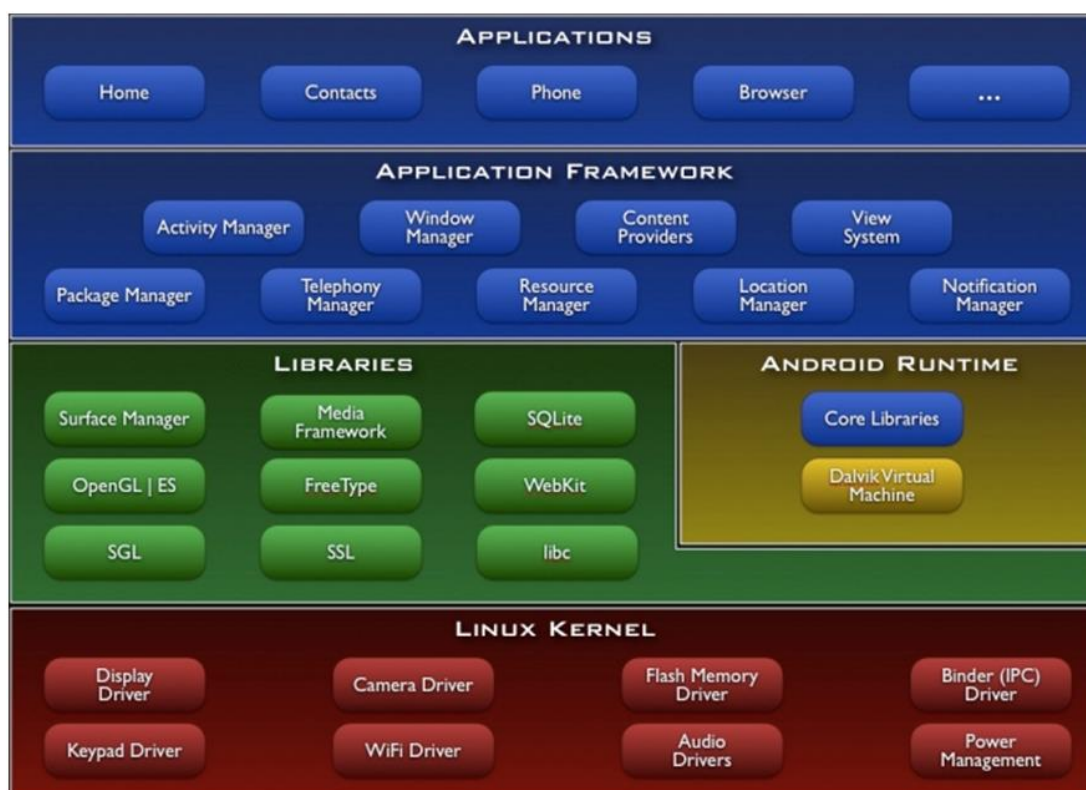


Figura 1 - Arquitetura Android (Android Developers, 2012).

Acima do **Linux Kernel** ficam as **bibliotecas** C/C++ utilizadas por diversos dos componentes do sistema, como: uma implementação da biblioteca padrão do C (libc), mas com licença BSD e otimizada para dispositivos embarcados; bibliotecas para suporte e formatos de áudio, vídeo e imagens; um gerenciador que intermedia o acesso ao *display* e compõe camadas de imagem 2D e 3D; o motor para navegadores WebKit; bibliotecas para gráficos 2D (SGL) e 3D (OpenGL ES); um renderizador de fontes bitmap e vetoriais; e o banco de dados relacional SQLite [Android Developers 2012].

No Android, aplicações escritas em Java são executadas em sua própria máquina virtual, que por sua vez é executada em seu próprio processo no Linux, isolando-a de outras aplicações e facilitando o controle de recursos [Android Developers 2012]. O **Android Runtime** é composto pela máquina virtual chamada Dalvik VM, onde as aplicações são executadas, e por um conjunto de bibliotecas que

fornece boa parte das funcionalidades encontradas nas bibliotecas padrão do Java [Android Developers 2012].

Na camada acima, escrita em Java, fica o **framework de aplicações**, que fornece todas as funcionalidades necessárias para a construção de aplicativos, através das bibliotecas nativas. As aplicações Android podem possuir quatro tipos de componentes: *activities*, *services*, *content providers* e *broadcast receivers*. Além destas peças fundamentais em uma aplicação, existem os recursos, que são compostos por *layouts*, *strings*, *estilos* e *imagens* e o arquivo de manifesto, que declara os componentes da aplicação e os recursos do dispositivo que ela irá utilizar [Android Developers 2012].

2.2 Protocolo de segurança HTTPS

O HTTPS (*Hypertext Transfer Protocol Secure*) nada mais é do que uma extensão do protocolo HTTP que utiliza criptografia nas operações de transferência de dados. Esse protocolo foi desenvolvido pela Netscape Communications Corporation em 2000 com o objetivo de fornecer comunicações seguras (ANUSH,2012). A necessidade de SSL veio depois da introdução de *sniffers*, que são programas que capturam o tráfego de dados na rede, inclusive e-mails, senhas, movimentações bancárias, etc. Quando esses dados são criptografados, torna-se praticamente impossível para os *sniffers* saberem o que e sobre o que são os dados trafegados. Com isso os bancos, serviços de e-mail, sites de *e-commerce* mudaram suas conexões para HTTPS que atualmente é suportado por quase todos os browsers.

HTTPS não é um protocolo mas sim um protocolo HTTP que opera em conjunto com SSL (*Secure Socket Layer*) e TLS (*Transport Layer Security*) que são mecanismos de criptografia. O protocolo HTTPS utilizado para enviar mensagens fornece a autenticação desses usuários, que buscam acesso a fila de mensagens via servidor *web* bem como permite a configuração da conexão segura entre um emissor e seu receptor.

Enquanto comandos HTTPS são uma boa idéia para transferências seguras, certamente também há limites do que se pode fazer com HTTPS. O nível de proteção que um HTTPS provê, dependerá da sua implementação pelo: *browser*, servidor e

dos algoritmos criptográficos que são suportados. HTTPS fornece proteção somente para ataques de espionagem e de interceptação de dados.

Para criar um servidor *web* capaz de processar requisições HTTPS, é necessária a criação de um certificado como mostra a Figura 2. A autenticação HTTPS utiliza dois tipos de certificados, conforme previsto pela autoridade de certificação:

- Certificados de Servidor
 - Chaves criadas por meio de criptografia. Eles basicamente são cadeias de texto cifradas utilizadas para garantir a conexão segura entre um grupo de usuários
- Certificados de Cliente
 - Informações pessoais indicam o usuário, garantindo a identificação de um cliente para um servidor SSL.

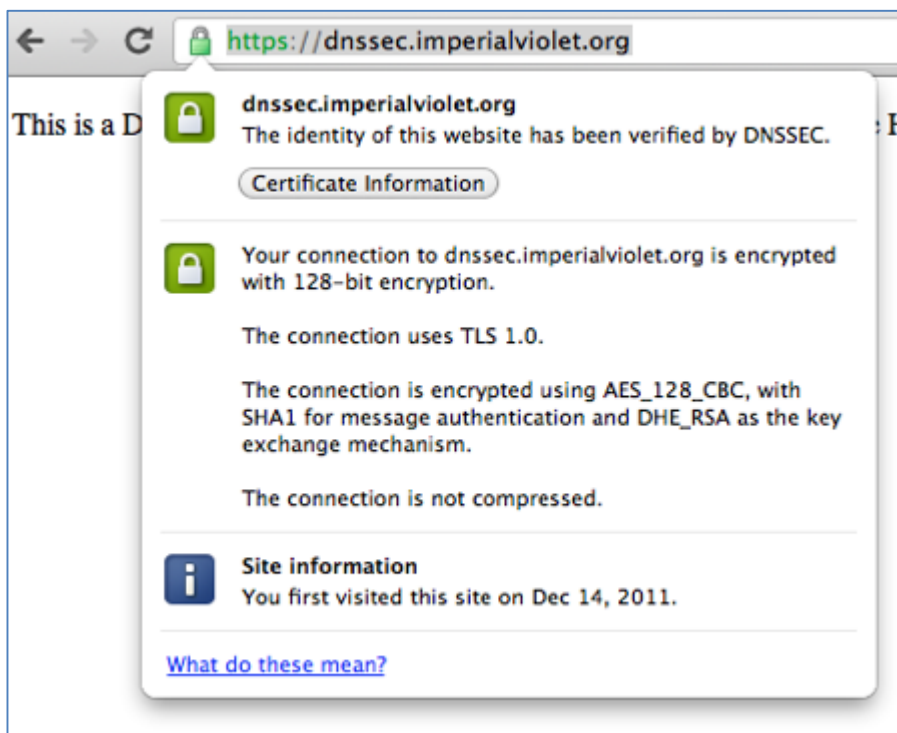


Figura 2 - Exemplo de acesso via HTTPS

2.3 Testes automatizados

Gerenciar a qualidade dos softwares produzidos é uma tarefa que vem se tornando cada dia mais difícil, além do crescimento da complexidade do produto a quantidade de plataformas disponíveis vem aumentando a cada dia fugindo do convencional *desktop* e se espalhando por diversos dispositivos móveis que possuem diferentes sistemas operacionais, tamanhos de tela e versões do sistema operacional. Gerenciar isso tudo manualmente demandaria muito tempo, recurso e consequentemente aumentaria o custo do desenvolvimento, segundo BORGES um estudo publicado (NSIT, 2013) pelo *National Institute of Standards and Technology* (NIST) revelou que os erros de software custam cerca de 60 bilhões de dólares a economia norte americana a cada ano.

Para gerenciar os testes de forma mais eficiente surgiram os testes automatizados ou TDD que é um estilo de desenvolvimento de software ágil derivado do método *Extreme Programming* (XP) (BECK, 2000) e do Manifesto Ágil e foi criado para antecipar a identificação e correção de falhas durante o desenvolvimento do software (BORGES, 2006). Segundo BECK, um método ágil é comparável ao ato de dirigir um carro: você deve observar a estrada e fazer correções contínuas para se manter no caminho. Seguindo este pensamento o desenvolvimento dos testes seria o responsável por fazer as correções contínuas e manter o veículo no rumo certo impedindo que o mesmo se perca.

Remover por completo os erros de uma aplicação é algo quase impossível, mas, utilizando-se da técnica de TDD bem elaborada é possível reduzir de forma considerável a quantidade de problemas em um software. Existem três formas essenciais para lidar com as falhas

- *Fault-avoidance* (Evitar Falhas)
 - Com atividades apropriadas de especificação, projeto, implementação e manutenção sempre visando evitar falhas em primeiro lugar. Inclui o uso de métodos de construção de software avançados, métodos formais e reuso de blocos de software confiáveis.

- *Fault-elimination* (Eliminar falhas)
 - compensação analítica de erros cometidos durante a especificação, projeto e implementação. Inclui verificação, validação e teste.
- *Fault-tolerance* (Tolerância a falhas)
 - tratamento de exceções e compensação em tempo real de problemas residuais como mudanças fora da especificação no ambiente operacional, erros de usuário, etc.

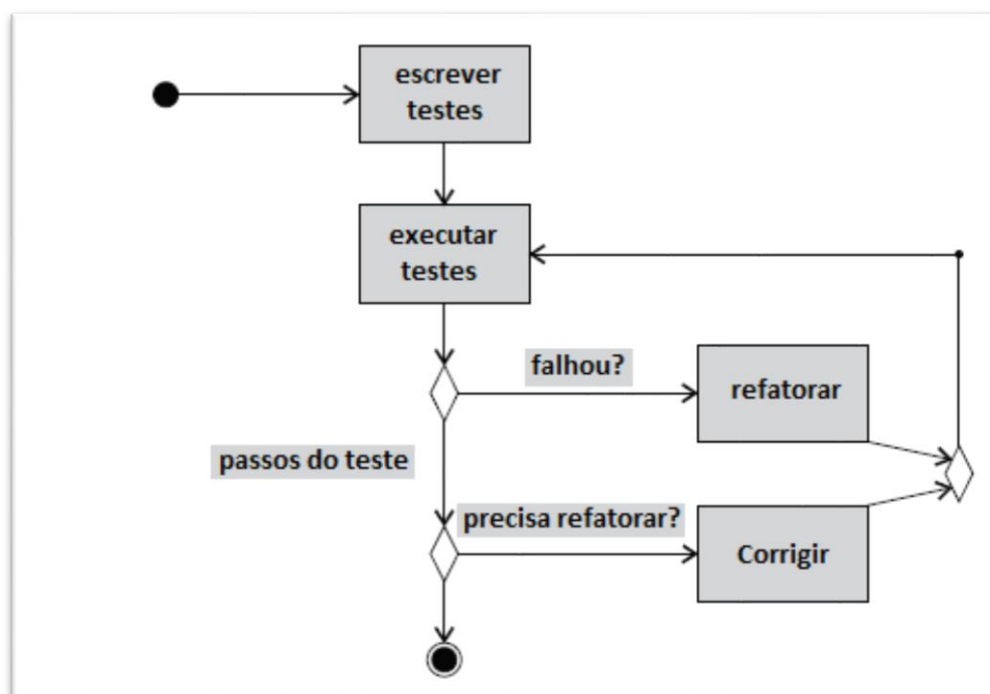


Figura 3 - Ciclo de um TDD (IMOBILIS, 2013).

Como podemos ver na Figura 3 a maneira mais eficiente de se implementar Testes Automatizados é escrever os testes antes do código pois quando o teste é escrito depois que o código esteja pronto, a tendência é que esse teste seja de baixa qualidade já que de forma inconsciente escrevemos testes para rodar no código produzido.

Testes automatizados proporcionam inúmeros benefícios tanto para a equipe de desenvolvimento quanto aos clientes, principalmente por especificar que os testes devem ser projetados antes de implementar a funcionalidade em si, e para projetar os

testes é preciso entender o que realmente deve ser testado e como consequência faz com que o engenheiro tenha um melhor entendimento sobre a funcionalidade.

Outro ponto importante que é interessante quando se usa TDD é a valorização do produto, um produto que já possui sua cadeia de testes implementadas representa uma entrega de maior valor agregado ao produto trazendo benefícios para ambos os lados, pois a qualidade em geral do produto será maior e além disso a equipe de desenvolvimento terá um aumento na produtividade já que a maioria das falhas foi detectada previamente e corrigidas durante a fase de teste.

Portanto, a conclusão que pode-se chegar em relação ao uso do TDD é de que ele é um dos principais responsáveis pelo sucesso dos projetos que utilizam um método ágil de desenvolvimento. Deve-se frisar também que com o uso dessa técnica, que detecta falhas ainda na etapa do desenvolvimento graças ao contínuo *feedback* dado ao desenvolvedor, é possível tornar o software menos complexo e aumentar sua manutenibilidade.

2.4 Serviços Web

O W3C define um Serviço Web como um sistema de *software* desenvolvido para permitir interações máquina-máquina através de uma rede. É uma interface descrita para ser consumida por máquinas. Outros sistemas interagem com o Serviço Web através de mensagens SOAP, geralmente enviadas através de HTTP em conjunto com outros padrões relacionados à web (W3C, 2004). A interação de outros sistemas com os Serviços Web se dá por meios de mensagens SOAP, em geral utilizando o protocolo HTTP com conteúdo XML em conjunto com outros padrões relacionados a Web. Serviços Web permitem a utilização de diferentes formas de transmissão de dados pela rede. Logo, a arquitetura deles pode trabalhar com protocolos, tais como HTTP, SMTP, FTP, RMI/IIOP ou protocolos de mensagem proprietários (W3C, 2004).

Serviços Web permitem que uma aplicação faça o consumo de um serviço sem a necessidade de conhecer os detalhes internos como codificação, linguagens de programação utilizadas, lógicas, etc. Isso acaba facilitando a reutilização de código e simplifica os sistemas. Essa tecnologia também é utilizada para integrar sistemas antigos a sistemas recentes com o objetivo de aproveitar a base de código e regras de

negócio que já foram escritas. Abaixo na Figura 4 podemos ver basicamente como é a troca de mensagens entre um cliente e um servidor através de Serviços Web.

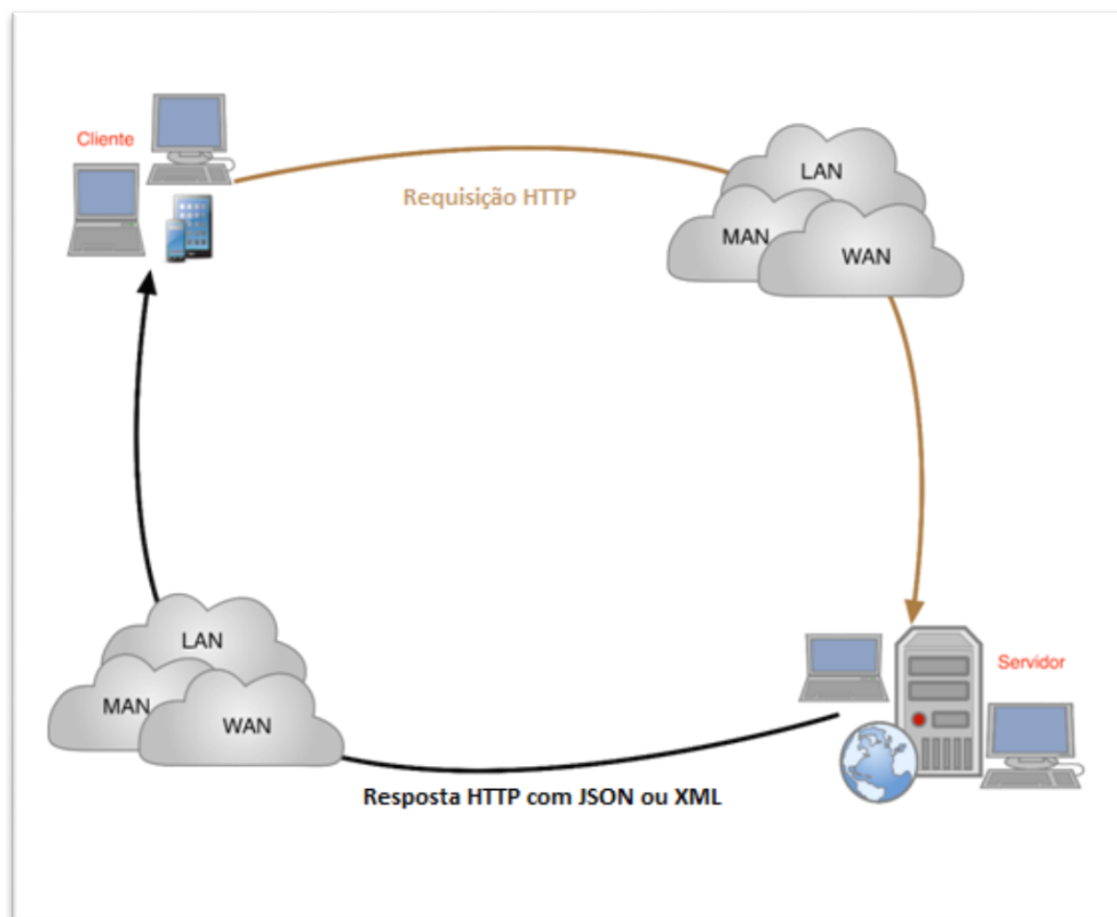


Figura 4 - Comunicação via Web Service (SANTOS, 2013).

2.4.1 Comunicação

Aplicações corporativas geralmente precisam se comunicar com algum servidor web, para realizar o sincronismo das informações. Essa comunicação pode ser feita de diversas formas, utilizando o protocolo HTTP, *sockets* ou *web services* (LECHETA, 2010).

A comunicação entre dois sistemas via *web service* ocorre através da troca de mensagens e para que essa comunicação possa de fato ocorrer essas máquinas devem estar conectadas de alguma forma e além disso utilizar um protocolo de comunicação comum entre elas para que o conteúdo da mensagem possa ser decifrado. O protocolo SOAP é responsável por definir um padrão para a comunicação, que são

documentos XML e além disso SOAP também tem a responsabilidade de especificar uma linguagem para descrever os métodos e funções que o serviço (WSDL) disponibiliza. A organização que desenvolveu a especificação do SOAP, foi a W3C e além de desenvolver ela recomenda o SOAP o que significa que tornou-se um padrão e já passou por uma série de testes. É importante ressaltar que o protocolo SOAP é um protocolo que não mantém estado entre as mensagens trocadas, ou seja, cada mensagem deve possuir todas as informações necessárias para que o serviço possa processá-la.

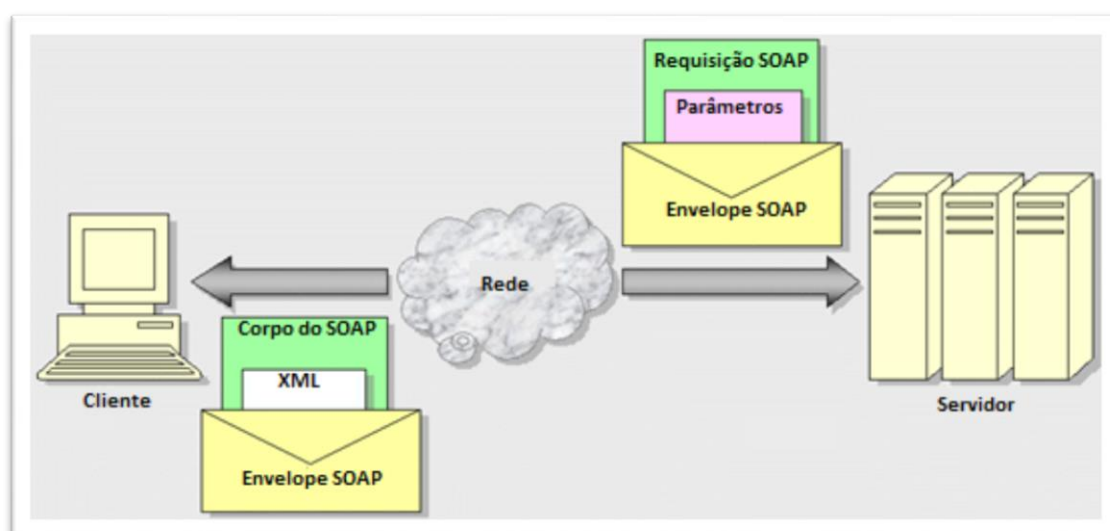


Figura 5 - Exemplo de troca de mensagens utilizando arquitetura SOAP (DEVX, 2014)

Além do SOAP, existe uma arquitetura muito utilizada atualmente para o desenvolvimento de Serviços Web que é a REST, grandes empresas como o Flickr e o Twitter utilizam também a arquitetura REST. Segundo MACEDO (MACEDO, 2010) o mais importante em uma arquitetura REST são as URL's que são mais conhecidas como (RESTful url's) e os *resources* (recursos). Um *resource* é um recurso seja ele uma entidade ou objeto, estes que compõem a informação a ser transferida por meio de um XML.

Portanto, Serviços Web disponibilizam interoperabilidade, simplicidade e flexibilidade com um baixo custo de processamento, ao utilizar mecanismos como o protocolo HTTP (*Hypertext Transfer Protocol*) (Fielding et al., 1999) para transporte e a linguagem XML para representação e padronização de suas informações.

2.5 Desenvolvimento Ágil de Software

Organizações de desenvolvimento de software geralmente enfrentam grandes desafios na produção de seus sistemas. Grande parte disso está relacionado à volatilidade de seus requisitos (BECK, 2000). A todo momento são solicitadas mudanças e novas funcionalidades causando impacto no software a ser elaborado. Surge então a necessidade de se adaptar a essas mudanças rapidamente e é nesse contexto que aparecem as metodologias ágeis de desenvolvimento que irão auxiliar na dinâmica para acompanhar as mudanças de forma mais rápida. Essa realidade tem se destacado no mercado produtor de *software*, onde cada vez mais, as metodologias ágeis ganham espaço (MILLER, 2009).

Proposto em 2001, o Manifesto Ágil (ÁGIL, 2001) é uma declaração de padrões que são fundamentais para o desenvolvimento ágil de software, ele possui quatro conceitos fundamentais, são eles:

1. Indivíduos e interações ao invés de processos e ferramentas;
2. Software executável ao invés de documentação;
3. Colaboração do cliente ao invés de negociação de contratos;
4. Respostas rápidas a mudanças ao invés de seguir planos.

O Manifesto Ágil possui também 12 princípios que são:

- Garantir a satisfação do cliente entregando rapidamente e continuamente softwares funcionais;
- Softwares funcionais são entregues frequentemente (semanas, ao invés de meses);
- Softwares funcionais são a principal medida de progresso do projeto;
- Até mesmo mudanças tardias de escopo no projeto são bem-vindas;
- Cooperação constante entre pessoas que entendem do 'negócio' e desenvolvedores;

- Projetos surgem através de indivíduos motivados, e que deve existir uma relação de confiança;
- Design do software deve prezar pela excelência técnica;
- Simplicidade;
- Rápida adaptação às mudanças;
- Indivíduos e interações mais do que processos e ferramentas;
- Software funcional mais do que documentação extensa;
- Colaboração com clientes mais do que negociação de contratos;
- Responder a mudanças mais do que seguir um plano.

As metodologias ágeis, apesar de serem relativamente novas, já apresentam resultados relevantes (SOARES, 2004), o que faz um comparativo das metodologias ágeis e tradicionais, e mostra que os projetos que implementam um processo ágil tiveram resultados mais satisfatórios em relação a cumprimento de prazos, qualidade e até custo. Outro fator importante que o artigo demonstra a partir de seus estudos que a quantidade de projetos e equipes que utilizam metodologias ágeis para o processo de desenvolvimento tem crescido.

3. MATERIAS, TÉCNICAS E MÉTODOS

Durante a realização do estágio supervisionado, foram utilizados uma série de ferramentas para que os trabalhos pudessem ser realizados em um determinado período. Pode-se visualizar na Tabela 1 a relação entre atividade e tempo que levou para ser executada, e logo após estão listadas todas as ferramentas utilizadas durante a realização do estágio supervisionado.

Atividade Executada	Tempo Dedicado
Desenvolvimento e testes da funcionalidade de logon do sistema.	12 horas
Desenvolvimento da funcionalidade que permite ao usuário selecionar o local ao qual deseja realizar o logon.	15 horas
Desenvolvimento da consulta de protocolos.	15 horas
Desenvolvimento do calendário que exibe os prazos de cada promotoria/procuradoria.	30 horas
Desenvolvimento do detalhe do prazo selecionado no calendário.	15 horas
Desenvolvimento das ações que a tela de detalhe do prazo possuem como: excluir, prorrogar e excluir	33 horas
Desenvolvimento da tela do detalhe do protocolo.	24 horas
Desenvolvimento do front-end e da funcionalidade de movimentar um protocolo.	18 horas
Ajuste do layout da aplicação para que se ajuste adequadamente a dispositivos com telas e densidades diferentes.	12 horas
Desenvolvimento do Controle de Sessão.	24 horas
Desenvolvimento da comunicação da aplicação com o webservice através de protocolo HTTPS.	36 horas
Desenvolver testes automatizados para o <i>Login</i> .	12 horas
Desenvolver testes automatizados para a <i>Home</i> .	12 horas
Desenvolver testes automatizados para a funcionalidade de Movimentar.	24 horas
Desenvolver testes automatizados para as ações do prazo.	18 horas
Desenvolver testes automatizados para a consulta de protocolos.	12 horas

Tabela 1

3.1 ADT-Bundle

O ADT Bundle (BUNDLE, 2014), visível na Figura 6, é um ambiente de desenvolvimento de aplicações móveis para Android que permite que desenvolvedores produzam aplicativos com muita praticidade e ele se resume ao Eclipse IDE com o Android SDK pré configurado.

O Android SDK inclui uma variedade de ferramentas que ajudam a desenvolver aplicações móveis para a plataforma Android. As ferramentas são classificadas em dois grupos: ferramentas do SDK e ferramentas da plataforma. Ferramentas do SDK são independentes da plataforma e são obrigatórias independentemente da plataforma Android que será desenvolvida. E ferramentas de plataforma são ferramentas personalizadas para suportar características particulares de uma ou mais plataformas Android.

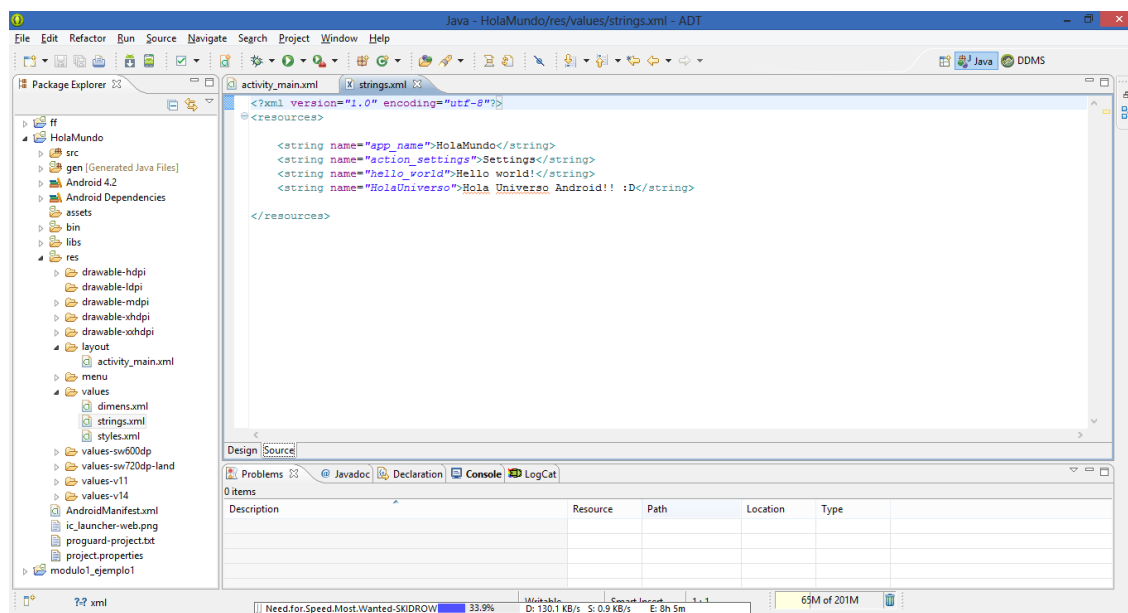


Figura 6 – Ambiente de desenvolvimento ADT Bundle

A IDE conforme mostra a Figura 6, foi utilizada porquê já vem pré configurada para o desenvolvimento de aplicativos para dispositivos móveis Android, eliminando assim o tempo que seria necessário para a configuração de

bibliotecas e ferramentas. O uso da IDE foi possível por que o fabricante disponibiliza gratuitamente sob a licença do *Android Software Development Kit* a IDE para qualquer pessoa interessada em utilizá-la.

3.2 JUnit 4

JUnit (JUNIT, 2014) é um framework de código aberto que dá suporte a criação de testes automatizados utilizando a linguagem de programação Java. O Objetivo do JUnit é facilitar a criação dos casos de testes, evitar escrever testes duplicados, e escrever testes reutilizáveis.

O *framework* é simples para a criação das classes de testes. Estas classes contêm um ou mais métodos para que sejam realizados os testes, podendo ser organizados de forma hierárquica, de forma que o sistema seja testado em partes separadas, algumas integradas ou até mesmo todas de uma só vez.

Essa ferramenta foi utilizada porque possibilita a verificação dos testes de forma textual e gráfica e também permite a reutilização de casos de testes. A utilização da mesma foi possível porquê a fabricante disponibiliza a ferramenta de forma gratuita para qualquer um que tiver interesse. Utilizando o JUnit foi possível perceber a rapidez e precisão na execução dos testes, já que o teste automatizado quando escrito de forma certa uma vez, não estará sujeito a erro humano.

3.3 Emulador Genymotion

Genymotion (GENYMOBILE, 2013) é um emulador que vem com imagens pré-configuradas do Android (x86 com OpenGL hardware acceleration) e por isso tem uma performance semelhante a de um dispositivo móvel físico. O Genymotion é uma alternativa ao emulador incluso no ADT Bundle, sua grande vantagem é rodar em cima de uma imagem Intel x86 enquanto o emulador do ADT Bundle roda em cima de uma arquitetura ARM, o que o torna mais lento e acaba tomando tempo no momento de desenvolver.

Utilizei o Genymotion nesse projeto por ele ser um emulador simples, realista, fácil de trabalhar e acima de tudo tem um tempo de resposta excelente. A

utilização do emulador se deu porque a fabricante disponibiliza uma versão gratuita apesar de ser uma versão um pouco mais limitada, aplicou-se perfeitamente ao projeto em questão. Com o Genymotion foi possível testar com precisão o aplicativo em geral para vários tamanhos de tela, resoluções diferentes e versões de Android diferentes.

3.4 JBoss RESTEasy

RESTEasy (RESTEASY, 2014) é um projeto da JBoss que prove vários frameworks que auxiliam no desenvolvimento de Web Service utilizando arquitetura REST (JBoss, 2013). O RESTEasy implementa a especificação JAX-RS onde define um conjunto de API para o desenvolvimento de serviços web seguindo a arquitetura REST. RESTEasy tem como principais características:

- Implementa integralmente a especificação JAX-RS
- Roda no servidor web Tomcat ou em qualquer servidor de aplicação desde que o mesmo utilize JDK 6 (Java Development Kit) ou superior.
- Integra com tecnologias EJB, Seam, Guice, Spring e JUnit

Foi necessário utilizar o RESTEasy no estágio para realizar a comunicação entre o servidor e o aplicativo, e a maneira mais eficiente de se fazer isso foi através de *web services* REST, que são simples de consumir, simples de codificar e consequentemente simples de manter. O uso do RESTEasy foi possível por ser um projeto *open source*.

4. RESULTADOS

A seguir serão apresentadas as principais telas do sistema desenvolvido e um descritivo de cada uma.

A Figura 7 é a tela inicial do aplicativo e engloba todo o controle de acesso, ao digitar um nome de usuário e uma senha, o dispositivo irá checar se as informações de login e senha realmente são válidas através de uma consulta na base de dados por meio do *web service* desenvolvido com o framework JBoss RESTEasy.

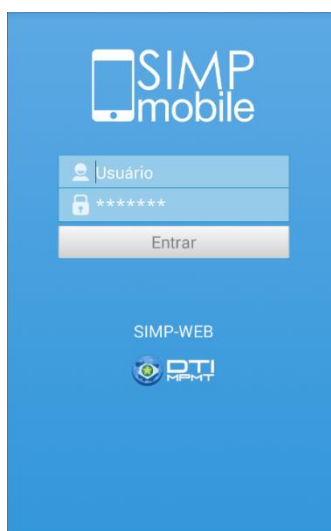


Figura 7 - Logon

A Figura 8 é o próximo passo após o login. Qualquer membro do Ministério Público do Estado de Mato Grosso está vinculado a um ou mais locais, caso o membro que esteja utilizando o sistema tenha seu usuário vinculado a mais de um local, a Figura 8 será exibida para que o mesmo possa selecionar o local que deseja gerenciar.



Figura 8 - Seleção de locais

A Figura 9 representa a área de trabalho do aplicativo, e tem o intuito de promover a visualização e manipulação dos prazos contidos em um determinado feito.

O Calendário exibido na Figura 9 acaba ajudando o usuário a se programar já que é uma tela em que ele pode acompanhar seus prazos basta navegar no calendário selecionando o dia desejado. Caso o dia selecionado possua um círculo azul ao redor do dia, então há pelo menos um prazo com o vencimento naquela data. Se existir um ou mais prazos para o dia selecionado, os mesmos serão exibidos logo abaixo do calendário em uma listagem.



Figura 9 - Área de trabalho

A Figura 10 representa a área de trabalho do aplicativo, mas na versão para *tablet*. Assim como a Figura 9, ela também tem o objetivo de ajudar o usuário a se programar. A única diferença dela para a versão para *smartphone* é na disposição das informações, pois devido às dimensões dos *tablets* em geral que possuem uma tela muito maior do que os *smartphones* foi preciso aproveitar melhor os espaços. Vale frisar que houve uma mudança nos dias do calendário, ao invés de exibir círculos nos dias que possuem prazo, são exibidos pequenos marcadores com a cor do tipo do prazo. Assim o usuário pode identificar os tipos de prazo que tem em determinado dia sem precisar clicar no mesmo.

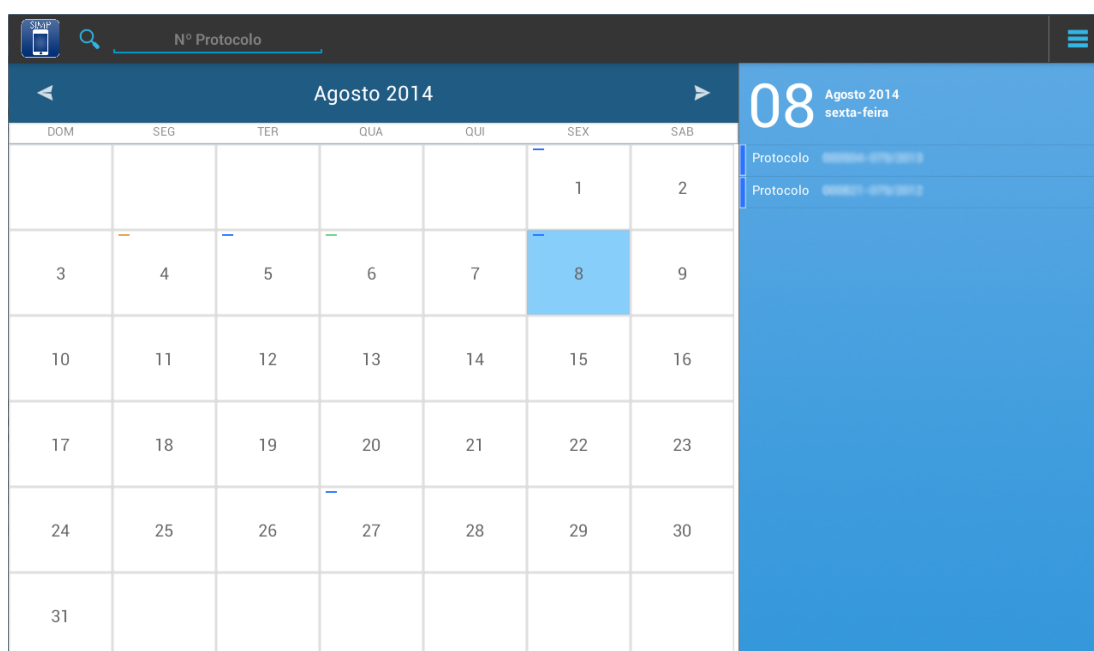


Figura 10 - Área de trabalho (tablet)

Quando algum prazo da área de Trabalho é selecionado no *smartphone* seus detalhes são exibidos através da tela de Detalhe do Prazo (Figura 11). A tela de Detalhe do Prazo é responsável por detalhar o prazo com todas as suas informações. Deste ponto também é possível executar ações sobre o prazo em questão, basta ir até o *menu* do canto superior direito e serão exibidas as opções: excluir, encerrar e prorrogar. Na parte superior da tela é possível inserir um número de protocolo qualquer para consulta.

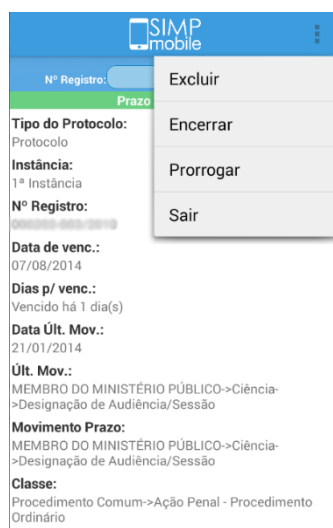


Figura 11 - Detalhe do prazo

A Figura 12 é a tela exibida quando se está utilizando o SIMP de algum *tablet* e algum prazo qualquer na área de trabalho é selecionado. Como se pode ver diferente do *smartphone* a Figura 12 não é redirecionada para uma nova tela, mas sim exibe as informações na própria área de trabalho, utilizando apenas a lateral esquerda da tela e exibindo de forma parecida com uma gaveta com as opções de encerrar prazo, excluir prazo e prorrogar prazo em um *menu* superior esquerdo. Outra ação que pode ser realizada deste ponto é a consulta de protocolos a partir do canto superior esquerdo digitando o mesmo e clicando na imagem da lupa localizada a sua esquerda.

Figura 12 - Detalhe do prazo (tablet)

O Detalhe do protocolo, representado da Figura 13 a Figura 20, contém as principais informações de um protocolo. Na parte superior da Figura 13 até a Figura 20 é informada qual procuradoria ou promotoria está destinada a trabalhar naquele protocolo e também informa o procurador ou promotor que responde por aquele protocolo. Outras informações são exibidas logo abaixo como: detalhes, classificação, partes, documentos e movimentos. Além disso, ainda permite a consulta de um novo protocolo inserindo um número de protocolo no campo superior da tela.

A Figura 13 é uma imagem específica para o acesso via *smartphones*, onde é exibida a aba Detalhe que possui informações básicas de um protocolo como o seu local atual, que é o local onde se encontra o protocolo atualmente (departamentos, gabinetes, promotorias, procuradorias), a tela também mostra a pessoa que está com o protocolo em seu poder. Além disso, é exibida também a data que o protocolo foi instaurado e um breve resumo do protocolo. Algumas informações na Figura 13 como número único e número do processo, que são informações para a consulta desse protocolo na base de dados do Tribunal de Justiça de Mato Grosso.



Figura 13 - Detalhe do protocolo (aba detalhe)

A aba classificação para *smartphones*, localizada no detalhe do protocolo pode ser vista na Figura 14, ela é responsável por exibir informações sobre classificação taxonômica do protocolo. Taxonomia é uma lista de classes e de assuntos providas pelo Conselho Nacional do Ministério Público e através dessa lista é que são definidas as características jurídicas que identificam: área, classe e assunto do protocolo, que são justamente as informações exibidas na aba de classificação.

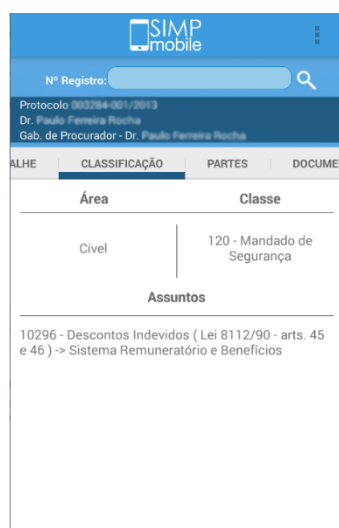


Figura 14 - Detalhe do protocolo (aba classificação)

A versão para *smartphones*, também possui na tela de detalhe do protocolo, a aba Partes (Figura 15), que lista as partes (toda pessoa física ou jurídica) envolvidas no processo. As partes sempre são definidas no cadastro do protocolo em que é obrigatório incluir um ou mais denunciante(s) e um ou mais denunciado(s). Com base nas informações cadastradas na inclusão do protocolo é que são exibidas as partes envolvidas no protocolo na Figura 15. A tela de partes é dividida em duas seções, uma para o(s) denunciante(s) e outra para o(s) denunciado(s), e para cada parte é exibido o nome e o(s) assunto(s) na qual essa parte está envolvida

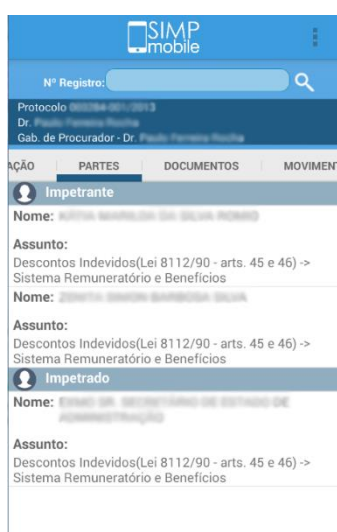


Figura 15 - Detalhe do protocolo (aba partes)

A Figura 16 mostra a aba documentos em sua versão para *smartphone*, também localizada na tela de detalhe do protocolo. Protocolos podem ou não conter arquivos anexados a eles como ofícios, relatórios, documentos etc.

O objetivo dessa aba visível é possibilitar ao usuário realizar o *download* desses arquivos anexados ao protocolo e além de realizar o *download* a tela possibilita ainda a visualização dos documentos já baixados. Como podemos ver na Figura 16 os documentos são exibidos em forma de lista e com um ícone a sua direita. Esse ícone pode variar de formato e de função, caso o arquivo já tenha sido baixado a imagem exibida é a de um documento e quando é tocada exibirá o arquivo, caso contrário será exibido um ícone com uma seta para baixo, que significa que o download ainda não foi feito e quando o ícone é tocado, o download se iniciará.

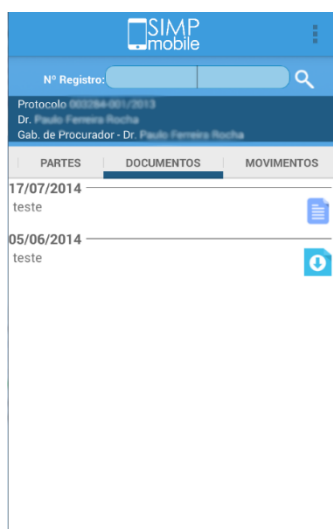
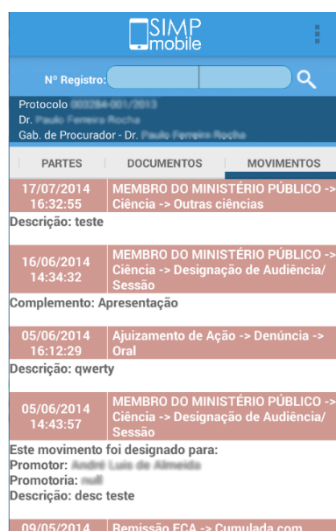


Figura 16 - Detalhe do protocolo (aba documentos)

Na Figura 17, existente na versão do SIMP para *smartphones* podemos visualizar os últimos movimentos realizados naquele protocolo. Toda e qualquer evolução ou atividade realizada em um processo é representada através de um movimento. É principalmente através dos movimentos que a principal premissa do sistema SIMP é atendida: O protocolo eletrônico tem que representar fielmente tudo aquilo que acontece no físico.



The screenshot shows the SIMP mobile application interface. At the top, there is a search bar labeled 'Nº Registro:' with a magnifying glass icon. Below this, the protocol details are displayed: 'Protocolo 000284-001/2013', 'Dr. Paulo Fernando Rocha', and 'Gab. de Procurador - Dr. Paulo Fernando Rocha'. The interface has three tabs: 'PARTES', 'DOCUMENTOS', and 'MOVIMENTOS', with 'MOVIMENTOS' being the active tab. The movements are listed in a table with columns for date and time, and a description of the movement. The movements are as follows:

DATA/HORA	DESCRIÇÃO
17/07/2014 16:32:55	MEMBRO DO MINISTÉRIO PÚBLICO -> Ciência -> Outras ciências
Descrição: teste	
16/06/2014 14:34:32	MEMBRO DO MINISTÉRIO PÚBLICO -> Ciência -> Designação de Audiência/Sessão
Complemento: Apresentação	
05/06/2014 16:12:29	Ajuizamento de Ação -> Denúncia -> Oral
Descrição: qwerty	
05/06/2014 14:43:57	MEMBRO DO MINISTÉRIO PÚBLICO -> Ciência -> Designação de Audiência/Sessão
Este movimento foi designado para: Promotor: <i>João Luiz de Almeida</i> Promotora: null Descrição: desc teste	
09/05/2014	Remissão ECA -> Cumulada com

Figura 17 - Detalhe do protocolo (aba movimentos)

A interface do detalhe do protocolo na versão para *tablets* sofre uma grande alteração de layout, pois as informações de detalhes, classificação, partes, movimentos e documentos não estão mais divididos em abas e sim em menus. A consulta de protocolos ainda pode ser feita na parte superior esquerda como mostrada na Figura 18.

Podemos visualizar melhor como ficou a estrutura da tela detalhe do protocolo na Figura 18. Os detalhes básicos do protocolo e sua classificação taxonômica ficam sempre visíveis do lado esquerdo, já as informações do lado direito são alteradas de acordo com a opção selecionada na barra de tarefas localizada no canto superior direito. No caso da Figura 18 a opção selecionada na barra de tarefas foi a de *Movimentos* fazendo com que os últimos movimentos do protocolo aparecessem na lateral direita da tela. As informações exibidas são as mesmas que na versão para *smartphones*, a única mudança é na disposição da informação.

SIMP **Nº Protocolo** **Partes** **Documentos** **Movimentos**

Protocolo **Dr. Paulo Francisco Rosário** **Movimentar**

Gab. de Procurador - Dr. Paulo Francisco Rosário

Número Único 9812345-00/2012-03T00:00:00-03:00

Processo/Protocolo TJ 123456789

Local Atual Em trânsito

Detentor Atual

Data Instauração 2012-12-03T00:00:00-03:00

Resumo

Último Movimento MEMBRO DO MINISTÉRIO PÚBLICO -> Ciência -> Outras ciências

Área Civil

Classe 120 - Mandado de Segurança

Assuntos 10296 - Descontos Indevidos (Lei 8112/90 - arts. 45 e 46) -> Sistema Remuneratório e Benefícios

Movimentos

17/07/2014	MEMBRO DO MINISTÉRIO PÚBLICO -> Ciência -> Outras ciências
Descrição: teste	
16/06/2014 14:34:32	MEMBRO DO MINISTÉRIO PÚBLICO -> Ciência -> Designação de Audiência/Sessão
Complemento: Apresentação	
05/06/2014	Ajuizamento de Ação -> Denúncia -> Oral
Descrição: qwerty	
05/06/2014 14:43:57	MEMBRO DO MINISTÉRIO PÚBLICO -> Ciência -> Designação de Audiência/Sessão
Este movimento foi designado para:	
Promotor: [nome]	
Promotoria: [nome]	
Descrição: desc teste	
09/05/2014	Remissão ECA -> Cumulada com Medida Sócio-educativa -> Advertência
Descrição: adv, cumulada	

Figura 18 - Detalhe do protocolo no *tablet* (menu movimentos)

A Figura 19 é uma imagem da tela de detalhe do protocolo acessada por um *tablet* e com o *menu Partes* selecionado. Ao clicar na opção *Partes* da barra de tarefas localizada no superior da tela apenas a lateral direita da mesma é alterada, o restante das informações continua sendo exibidas normalmente.

As informações das partes são exibidas assim como na versão para *smartphone* dividido em duas seções (denunciado(s) e denunciante(s)) e listando as partes envolvidas dentro da seção pertencente.

The screenshot shows the SIMP (Sistema Integrado de Movimentação Processual) interface on a tablet. The top navigation bar includes the SIMP logo, a search icon, a text input for 'Nº Protocolo', and icons for 'Partes', 'Documentos', 'Movimentos', and a menu icon. Below this, a sub-header shows 'Protocolo' and 'Dr. Paulo Henrique Rocha' (Gab. de Procurador - Dr. Paulo Henrique Rocha). A 'Movimentar' button is on the right.

The left sidebar contains the following fields:

- Número Único:** 0000000-00.0000.0.0000
- Processo/Protocolo TJ:** 0.0.0000.0000
- Local Atual:** Em trânsito
- Detentor Atual:**
- Data Instauração:** 2012-12-03T00:00:00-00:00
- Resumo:**
- Último Movimento:** MEMBRO DO MINISTÉRIO PÚBLICO -> Ciência -> Outras ciências
- Área:** Civil
- Classe:** 120 - Mandado de Segurança
- Assuntos:** 10296 - Descontos Indevidos (Lei 8112/90 - arts. 45 e 46) -> Sistema Remuneratório e Benefícios

The right main area is titled 'Partes' and contains two sections:

- Impetrante:**
 - Nome:** MARIA MARILIA DA SILVA FROES
 - Assunto:** Descontos Indevidos(Lei 8112/90 - arts. 45 e 46) -> Sistema Remuneratório e Benefícios
 - Nome:** MARIA MARILIA DA SILVA FROES
 - Assunto:** Descontos Indevidos(Lei 8112/90 - arts. 45 e 46) -> Sistema Remuneratório e Benefícios
- Impetrado:**
 - Nome:** PAULO HENRIQUE DA SILVA FROES
 - Assunto:** Descontos Indevidos(Lei 8112/90 - arts. 45 e 46) -> Sistema Remuneratório e Benefícios

Figura 19 - Detalhe do protocolo no *tablet* (menu partes)

Podemos visualizar na Figura 20 a tela de detalhe do protocolo acessada via *tablet*, mais especificamente com a opção *Documentos* selecionada. Assim como a Figura 19 a única parte modificada ao clicar na opção é a lateral direita da tela.

Os documentos continuam sendo exibidos como na versão do SIMP para *smartphone* através de uma lista, onde é possível saber quais documentos já foram baixados (através do ícone), e a partir disso pode-se visualizar ou realizar o *download* do documento para depois visualizá-lo.

SIMP Nº Protocolo Partes Documentos Movimentos Movimentar

Protocolo **Dr. Paulo Fernando Rocha**
983284-001/2012 Gab. de Procurador - Dr. Paulo Fernando Rocha

Número Único 0117906-00.2012.8.11.0000	Processo/Protocolo TJ 0117906/2012
--	--

Local Atual
Em trânsito

Detentor Atual

Data Instauração
2012-12-03T00:00:00-00:00

Resumo

Último Movimento
MEMBRO DO MINISTÉRIO PÚBLICO -> Ciência -> Outras ciências

Área
Cível

Classe
120 - Mandado de Segurança

Assuntos
10296 - Descontos Indevidos (Lei 8112/90 - arts. 45 e 46) -> Sistema Remuneratório e Benefícios

Documentos

17/07/2014	teste	252KB	
05/06/2014	teste	252KB	

Figura 20 - Detalhe do protocolo no tablet (menu documentos)

A tela de exclusão de prazo na versão para *smartphone* pode ser vista na Figura 21 e como o próprio nome já diz, tem a função de apagar um prazo de protocolo.

Os protocolos cadastrados no SIMP podem ou não possuir prazos, isso irá depender de sua classificação taxonômica que é baseada na tabela de taxonomia elaborada pelo Conselho Nacional dos Ministérios Públicos. Caso essa classificação taxonômica exija um prazo, ao concluir o cadastro desse protocolo ele irá definir o prazo deste protocolo automaticamente, assim sendo ele será exibido na área de trabalho (Figura 9 e Figura 10).

A Figura 21 exhibe a tela de exclusão de prazo de protocolo, nela são mostradas informações básicas do prazo como a data de abertura e a data de vencimento, e qual foi seu último movimento. Para apagar um prazo é obrigatório inserir uma justificativa, depois de inserida basta clicar no botão confirmar para que o prazo seja excluído.

The screenshot shows the 'SIMP mobile' application interface. At the top, there is a blue header with the logo and a search bar labeled 'Nº Registro:'. Below the header, a grey bar contains the text 'Excluir Prazo'. The main content area displays the following information: 'Protocolo: 000000-000/2010', 'Data de Abertura: 24/04/2014', 'Data de vencimento: 07/08/2014', and 'Movimento: MEMBRO DO MINISTÉRIO PÚBLICO->Ciência->Designação de Audiência/Sessão'. Below this is a text area labeled 'Justificativa:'. At the bottom, there is a red warning message: 'Após sua exclusão, este PRAZO não ficará mais visível aos usuários', followed by a grey button labeled 'Confirmar'.

Figura 21 - Excluir prazo

A Figura 22 mostra como excluir um prazo na versão para *tablet* no SIMP . Diferentemente da versão para *smartphones*, o aplicativo não é redirecionado para outra tela, ele realiza a exclusão sem precisar sair da área de trabalho através de uma espécie de gaveta exibida na lateral esquerda. Porém o processo de exclusão é o mesmo que na versão para *smartphone* exibindo as mesmas informações como: data de abertura, data de vencimento, número do protocolo que terá seu prazo excluído e o último movimento realizado no protocolo. Para concluir a exclusão basta inserir a justificativa e tocar no botão *Confirmar* localizada no *menu* superior esquerdo, caso contrário basta clicar no botão cancelar que a área de trabalho será mostrada novamente.

Após a exclusão do prazo, o mesmo será excluído do calendário de prazos da área de trabalho.





Excluir Prazo

✓ Confirmar

✕ Cancelar

Protocolo: 000000-0000-2010

Data de Abertura: 24/04/2014

Data de vencimento: 07/08/2014

Sigiloso:

Movimento: AUDIÊNCIA DE INSTRUÇÃO E JULGAMENTO

Justificativa:

Após sua exclusão, este ALERTA não ficará mais visível aos usuários

QUISEXSAB

	1	2
7	8	9
14	15	16
21	22	23
28	29	30

07

Agosto 2014

quinta-feira

Movimento: 000000-0000-2010

Figura 22 - Excluir Prazo (tablet)

5. DIFICULDADES ENCONTRADAS

Desenvolver um aplicativo móvel para diversos tamanhos de tela, inúmeras resoluções, variadas versões da plataforma Android com certeza foi a maior dificuldade encontrada nesse projeto. Primeiramente porquê não tínhamos todas as variações de aparelhos Android possíveis para testarmos, e mesmo que tivéssemos não seria viável testar em todas elas porquê não teríamos tempo o suficiente. A saída foi realizar os testes nas versões da plataforma Android mais utilizadas pelo público alvo do aplicativo que são os promotores, procuradores e outros membros do Ministério Público do Estado de Mato Grosso. Para descobrir qual era a versão mais utilizada, consultamos todos os futuros usuários do sistema, e com base nessa pesquisa chegamos à conclusão de que o aplicativo deveria ser implementado para executar no mínimo na versão 3.0 do Android que foi a versão mais antiga encontrada na pesquisa.

Outra dificuldade encontrada foi na criação do layout das telas. Algumas tinham que exibir uma grande quantidade de informação, e foi extremamente difícil encontrar uma maneira que não deixasse a tela com uma impressão muito carregada.

6. CONCLUSÕES

A criação de um aplicativo móvel para controle dos protocolos virtuais do Ministério Público do Estado de Mato Grosso é algo novo, visto que é o primeiro aplicativo móvel desenvolvido pela equipe de desenvolvimento do órgão.

A determinação da equipe, aliada a metodologia ágil de desenvolvimento aplicação, e a implementação de testes automatizados garantiram o sucesso e a qualidade do aplicativo SIMP que foi desenvolvido no tempo planejado. Outro fator que garantiu a qualidade do aplicativo desenvolvido foi a preocupação com o código seguro a todo o momento.

Todo o trabalho desenvolvido durante o período de estágio e elaboração desse relatório apresentou resultados muito positivos para o meu processo de formação como profissional da área de Sistemas de Informação. Ao longo do estágio obtive experiências não só na área de desenvolvimento na qual atuei especificamente, mas também na área de análise. Conheci também novos frameworks como JUnit e RESTEasy.

7. REFERÊNCIAS BIBLIOGRÁFICAS

ÁGIL, MANIFESTO; 2001. Manifesto para o desenvolvimento ágil de software, Disponível por www em <http://manifestoagil.com.br> (acessado em 30 de junho de 2014).

ANDROID, DEVELOPERS. The developer's guide, Disponível por www em <http://developer.android.com/guide/index.html> (acessado em 29 de junho de 2014).

ANUSH, HTTPS Hypertext Transfer Protocol Secure, Disponível por www em <http://techstream.org/Web-Development/Hypertext-Transfer-Protocol-Secure> (acessado em 29 de junho de 2014).

BECK, KENT; 2000. *eXtreme Programming Explained*. 1ª Edição Addison Wesley.

BECK, KENT; 2002. *Test Driven Development: By Example*. 1ª Edição Addison Wesley.

BORGES, EDUARDO N.; 2006. Conceitos e Benefícios do Test Driven Development, Universidade Federal do Rio Grande do Sul(UFRS).BUNDLE, ADT; 2014. Get the Android SDK, Disponível por www em <http://developer.android.com/sdk> (acessado em 30 de junho de 2014).

CANALTECH, Google Play ultrapassa App Store em downloads de aplicativos. 2013, Disponível por www em <http://canaltech.com.br/noticia/apps/Google-Play-ultrapassa-App-Store-no-volume-de-downloads-de-aplicativos> (acessado em 30 de junho de 2014).

CINDRAL, BELINE. Sistemas Operacionais Para Celulares. Disponível por www em <http://www.techtudo.com.br/artigos/noticia/2011/01/afinal-o-que-e-android.html> (acessado em 28 de junho de 2014).

DEVX.; 2014. SOAP Web Service Request, Disponível por www em <http://www.devx.com/supportitems/showSupportItem.php?co=20369&supportitem=figure2> (acessado em 28 de junho de 2014).

DIMARZIO, JEROME F.; 2008. *Android: A programmer's Guide*. 1ª Edição New York: McGrawHill.

GENYMOBILE, SAS.; 2013. Genymotion, Disponível por www em <http://www.genymotion.com> (acessado em 28 de junho de 2014).

GOOGLE, INC. Dev Guide. Documento Eletrônico, Disponível por www em <http://developer.android.com/guide/topics/fundamentals.html> (acessado em 28 de junho de 2014).

IMOBILIS, LABORATÓRIO. Desenvolvimento Android Baseado em Testes, Disponível por www em <http://www.decom.ufop.br/imobilis/?p=1763> (acessado em 28 de junho de 2014).

JBOSS, INC, RESTEasy. Disponível por www em <http://www.jboss.org/resteasy> (acessado em 27 de junho de 2014).

JUNIT; 2014. Disponível por www em <http://junit.org> (acessado em 27 de junho de 2014).

LECHETA, RICARDO R.; 2010. *Google Android: Aprenda a criar aplicações para dispositivos móveis com Android SDK*. 2ª Edição. São Paulo: Novatec Editora LTDA.

MACEDO, ALEXANDRE; 2010, Características de um web service RESTful, Disponível por <http://www.k19.com.br/artigos/criando-um-webservice-restful-em-java> (acessado em 30 de junho de 2014).

MILLER, L. D.; 2009. *Agile user experience SIG*. Conference on Human Factors in Computing Systems, Boston.

NSIT; 2013 National Institute of Standards and Technology, Disponível por www em <http://www.nist.gov> (acessado em 30 de junho de 2014)

NUDELMAN, GREG; 2013. *Padrões de Projeto para o Android*. 1ª Edição. São Paulo, SP Brasil: Novatec Editora LTDA.

RESTEASY, JBOSS; 2014 RESTEasy, Disponível por www em <http://resterasy.jboss.org> (acessado em 30 de junho de 2014)

SANTOS, ALLAN C.; 2013. Web Services em aplicações Android e iOS, Disponível por www em <http://www.devmedia.com.br/web-services-em-aplicacoes-android-e-ios/28901> (acessado em 28 de junho de 2014).

SOARES, MICHEL S.; 2004 Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software, Universidade Presidente Antônio Carlos.

W3C, Web Services Architecture. 2004, Disponível por www em <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211> (acessado em 30 de junho de 2014)