

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

MAURICIO COPETTI

**SISTEMA PARA ESTUDO E SIMULAÇÃO DA PROVA TEÓRICA DO
DETRAN EM DISPOSITIVOS ANDROID**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2014**

MAURICIO COPETTI

**SISTEMA PARA ESTUDO E SIMULAÇÃO DA PROVA TEÓRICA DO
DETRAN EM DISPOSITIVOS ANDROID**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.


Orientador: Prof. Andrei Carniel.

**PATO BRANCO
2014**

ATA Nº: 255

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO MAURICIO COPETTI.

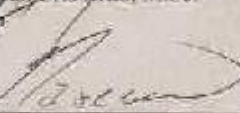
Às 15:30 hrs do dia 18 de dezembro de 2014, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Andrei Carniel (Orientador), Robilson Cris Brito (Convocado) e Marcelo Teixeira (Convocado), para avaliar o Trabalho de Diplomação do aluno Mauricio Copetti, matrícula 1295640, sob o título Sistema para Estudo e Simulação da Prova Teórica do DETRAN em Dispositivos Android, como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho APROVADO. Às 16:05 hrs foi encerrada a sessão.



Prof. Andrei Carniel, Esp.
Orientador




Prof. Robilson Cris Brito, M.Sc.
Convocado



Prof. Marcelo Teixeira, Dr.
Convocado

Profª. Eliane Maria de Bortoli Favero, M.Sc.
Coordenadora do Trabalho de Diplomação



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

Dedico especialmente a todos os professores, que sempre estão dispostos a passar o seu conhecimento adiante.

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer ao meu orientador de estagio e TCC Andrei Carniel, que sem a sua ajuda eu não conseguiria chegar aonde eu cheguei.

Gustavo Copetti, Tiago Sanguanini, Lukas Schiochet, Matheus Maffioletti e Thomas Vitale, obrigado por estarem sempre ao meu lado e não me deixaram enlouquecer nesse momento de estresse e dificuldade. Pois sem amigos ninguém escolheria viver.

Obrigado aos meus pais e meus irmãos, que mesmo com todas as brigas e o excesso de cobrança me ajudaram a chegar até aqui, amo vocês incondicionalmente.

A minha redatora, psicóloga e namorada Thaynara Achre, que de longe foi a pessoa que mais me apoiou, do começo ao fim, mesmo com todos os momentos que qualquer pessoa no mundo não aguentaria a minha presença, obrigado por continuar ao meu lado.

Por fim gostaria de agradecer a todos os professores que contribuíram para a minha formação, mas um muito obrigado especial para a Beatriz Borsoi, que foi a melhor professora que eu já tive e que me ensinou a gostar de programação, e ao Robison Brito, que me ensinou a tecnologia que eu tanto amei e utilizei para fazer esse TCC.

Se você pensa que pode ou se pensa que não pode, de qualquer forma você está certo.

Henry Ford

RESUMO

COPETTI, Mauricio. Sistema para estudo e simulação do teste teórico do DETRAN em dispositivos Android. 2014. 64f. Trabalho de Conclusão de Curso - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Campus Pato Branco. Pato Branco, 2014.

O ser humano está em constante movimento seja deslocando-se de casa para o trabalho, nas viagens de férias, na visita aos amigos, entre outros, este estilo de vida está presente no cotidiano de muitas pessoas. Muitas vezes esse constante movimento envolve distâncias maiores entre os locais. Essa distância aumentou também nos grandes centros, onde o uso do automóvel se tornou muito mais do que um conforto, sendo uma necessidade para conseguir se locomover de sua casa para trabalho, por exemplo. Porém, para que uma pessoa possa dirigir é necessário que ela tenha uma carteira de habilitação, a qual inclui uma série de aulas e testes para que o futuro condutor possa adquirir a primeira habilitação. Dentre esses testes, destaca-se o teórico, onde seus conhecimentos sobre: mecânica, legislação, direção defensiva, primeiros socorros, meio ambiente e convívio social serão testados em uma prova muito abrangente, a qual exige muitas horas de estudo. Como forma de auxílio, o DETRAN fornece uma alternativa de estudo via Internet e voltado para computadores convencionais, sendo está não muito agradável de utilizar em outros ambientes diferentes, como um *smartphone*, por exemplo. Como forma de resolver esta situação, foi desenvolvido um *software* utilizando a tecnologia Android para simular e auxiliar nos estudos para prova teórica do DETRAN. O mesmo beneficiará um amplo numero de pessoas, desde quem está pensando em conseguir pela primeira vez a habilitação, até para candidatos a reciclagem da carteira, já que o software apresenta o modo estudo e o modo simulação da prova, em dispositivos móveis com plataforma Android. Este trabalho é a extensão de um artigo publicado no 3º CBIE ocorrido no ano de 2014.

Palavras-chave: Carteira de Habilitação. Dispositivos Móveis. Linguagem de Android. Simulado DETRAN.

ABSTRACT

COPETTI, Mauricio. System for study and simulation of the DETRAN's theoretical test in android devices. 2014. Course completion assignment – Higher education of analysis and systems development, Federal Technological University of Paraná, Campus Pato Branco. Pato Branco, 2014.

People are in constant movement. They move from home to work, on vacation trips, on visits to friends, among others. This lifestyle is present in the daily lives of many people and usually involves long distances, which is increased in large cities, where the use of cars has become much more than a comfort, but a necessity to be able to move quickly. However, for a person to drive it is necessary to obtain a driver's license, which requires a series of lessons and tests. Only then the future driver can acquire the first qualification. Among those tests, there is the theoretical step, during which the knowledge of the driver candidate is tested with respect to: mechanical, legislation, defensive driving, first aid, environment and social life. This is a very comprehensive test, which requires many hours of study. To assist the candidate to prepare himself, the DMV provides an online alternative that can be accessed via Internet using conventional computers. However, such option is not so pleasant to be exploited using a smartphone, for example. In order to address this inconvenience, we subsume our previous results and develop a software using Android technology to simulate and help candidates with the theoretical DMV test. Our software benefits a large number of people, from those who are thinking about getting the license for the first time to those candidates recycling their licenses.. Since the software presents the study mode and the mode simulation of proof in mobile Android platform . This work is the extension of an article published on 3rd CBIE occurred in 2014.

Keywords: Driving license. Mobile devices. Android platform. Test, DETRAN

LISTA DE FIGURAS

FIGURA 1 - INFOGRÁFICO DE CARRO E MOTO POR HABITANTES. 21	
FIGURA 2 - ESTATÍSTICA DE MORTES NO TRANSITO.	22
FIGURA 3 - HTC DREAM OU T-MOBILE G1, O PRIMEIRO CELULAR COM ANDROID	23
FIGURA 4 - TECLADO SOBREPONDO A QUESTÃO.	24
FIGURA 5 - RECURSOS PEDAGÓGICOS NO CELULAR.	26
FIGURA 6 - DETRAN SIMULADOS.....	27
FIGURA 7 - SIMULADO DETRAN SP	28
FIGURA 8 - SIMULADO DETRAN RJ.....	28
FIGURA 9 – TELA DE CONFIGURAÇÕES DO JAVA SE.	31
FIGURA 10 – TELA DE CONFIGURAÇÕES DO ANDROID DEVELOPER TOOLS	32
FIGURA 11 - IDE ECLIPSE	33
FIGURA 12 - ANDROID SDK MANAGER	33
FIGURA 13 - VISUAL PARADIGM.....	35
FIGURA 14 – LOGO RUBY ON RAILS	36
FIGURA 15 - CASO DE USO.....	42
FIGURA 16 - DIAGRAMA DE SEQUÊNCIA	42
FIGURA 17 - CASO DE USO WEB SERVICE	43
FIGURA 18 - REPRESENTAÇÃO DO APLICATIVO.....	43
FIGURA 19 - TELA INICIAL DO SISTEMA.....	44
FIGURA 20 - TELA DE ESCOLHA DE TEMAS	45
FIGURA 21 - TELA DO MODO ESTUDO	45
FIGURA 22 - TELA DE RESULTADO DO MÓDULO ESTUDO	46
FIGURA 23 - TELA DO MODO SIMULADO	47
FIGURA 24 - TELA DE REPROVADO DO MODULO SIMULADO	48
FIGURA 25 - TELA DE APROVADO DO MODULO SIMULADO	48
FIGURA 26 - AVISO CASO CONECTADO VIA OPERADORA	49
FIGURA 27 - AVISO DE ATUALIZAÇÃO	49

LISTAGEM

LISTAGEM 1 - GERAÇÃO DAS PERGUNTAS ALEATÓRIAS	50
LISTAGEM 2 - CÓDIGO DOS BOTÕES ANTERIOR E PRÓXIMA.....	51
LISTAGEM 3 - FUNÇÃO PARA TRANSFORMAR BASE64 EM BITMAP	52
LISTAGEM 4 - LAYOUT DA PAGINA DE PERGUNTAS	53
LISTAGEM 5 - MARCAÇÃO DA RESPOSTA CORRETA NO MÓDULO ESTUDO	54
LISTAGEM 6 - LÓGICA DO BOTÃO FINALIZAR.....	55
LISTAGEM 7 - CÓDIGO DO CRONÔMETRO	55
LISTAGEM 8 - SELEÇÃO DE PERGUNTAS	57
LISTAGEM 9 – RECUPERAÇÃO DAS PERGUNTAS DO WEB SERVICE.....	58
LISTAGEM 10 - VERIFICAÇÃO DA CONEXÃO UTILIZADA.....	58

LISTA DE QUADROS

QUADRO 1 - MATERIAIS UTILIZADOS.....	30
QUADRO 2 - REQUISITO FUNCIONAL ESTUDO SELETIVO.....	39
QUADRO 3 - REQUISITO FUNCIONAL SIMULADO.....	40
QUADRO 4 - REQUISITO FUNCIONAL ATUALIZAÇÃO VIA WEB SERVICE.....	41
QUADRO 5 - REQUISITO FUNCIONAL CADASTRO DE PERGUNTAS VIA NAVEGADOR.....	41
QUADRO 6 - REQUISITO FUNCIONAL BANCO DE DADOS <i>OFFLINE</i>	41

LISTA DE SIGLAS

ADT	Android Developer Tool
API	Application Programming Interface
AVD	Android Virtual Device
CBIE	Congresso Brasileiro de Informática na Educação
CNH	Carteira Nacional de Habilitação
HTTP	Hypertext Transfer Protocol
IBGE	Instituto Brasileiro de Geografia e Estatística
IDC	International Data Corporation
IDE	Integrated Development Environment
JDK	Java Development Kit
JDO	Java Data Objects
JOGL	Java OpenGL
JVM	Java Virtual Machine
LADV	Licença para Aprendizagem de Direção Veicular
MVC	Model View Controller
SDK	Software Development Kit
SE	Standard Edition
XML	Extensible Markup Language

LISTA DE ACRÔNIMOS

CONTRAN	Conselho Nacional de Trânsito
DETRAN	Departamento Estadual de Trânsito
DENATRAN	Departamento Nacional de Trânsito
GUI	Graphical User Interface
IPEA	Instituto de Pesquisa Econômica Aplicada
OHA	Open Handset Alliance
RENACH	Registro Nacional de Condutores Habilitados

SUMÁRIO

1 INTRODUÇÃO	14
1.1 CONSIDERAÇÕES INICIAIS	14
1.2 OBJETIVOS	16
1.2.1 Objetivo Geral	16
1.2.2 Objetivos Específicos	16
1.3 JUSTIFICATIVA	17
1.4 ESTRUTURA DO TRABALHO	17
2 REFERENCIAL TEÓRICO	19
2.1 PRIMEIRA HABILITAÇÃO	19
2.2 TECNOLOGIA ANDROID E O DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS	22
2.3 UTILIZAÇÃO DE DISPOSITIVOS MÓVEIS NO CONTEXTO EDUCACIONAL ..	24
2.4 TRABALHOS RELACIONADOS	26
3 MATERIAIS E MÉTODO	30
3.1 MATERIAIS	30
3.1.1 JAVA SE.....	30
3.1.2 ADT Bundle	31
3.1.2.1 Eclipse.....	32
3.1.2.2 Android SDK.....	33
3.1.3 SQLite	34
3.1.4 Visual Paradigm	34
3.1.5 Ruby on Rails	35
3.1.6 MySQL	36
3.2 MÉTODO.....	36
4 RESULTADOS	38
4.1 ESCOPO DO SISTEMA	38
4.2 MODELAGEM DO SISTEMA	38
4.3 APRESENTAÇÃO DO SISTEMA	43
4.4 IMPLEMENTAÇÃO DO SISTEMA	49
5 CONCLUSÃO	59
5.1 TRABALHOS FUTUROS	59
REFERÊNCIAS	61

1 INTRODUÇÃO

Neste capítulo serão abordadas as considerações iniciais juntamente com os objetivos e a justificativa, também será mostrada a estrutura do trabalho.

1.1 CONSIDERAÇÕES INICIAIS

Carros, motos, ônibus, caminhões, aviões, entre outros. Todos são veículos que podem ser utilizados para as mais diversas finalidades, sendo indispensáveis no dia a dia para se deslocar rapidamente a um determinado local, ir ao trabalho, estudo, lazer, levar grandes quantidades de cargas ou simplesmente por comodidade. As razões para utilizar qualquer tipo de veículo são as mais variadas.

Cerca de 80% da população brasileira se desloca com algum tipo de veículo (IPEA, 2011), sendo que destes 44% usam o transporte público, 23.8% carros e 12.6% motocicletas, ainda pode-se destacar as pessoas que utilizam destes veículos para trabalhar, como motoristas de ônibus, taxi ou de veículos de carga.

Mesmo com os números indicando que a maioria da população utiliza o transporte público, o número de pessoas que optam por utilizar o veículo próprio é muito grande, em especial nas pequenas cidades. Segundos dados do DENATRAN (2014), a frota nacional de veículos no ano de 2003 era de 2.251.022 veículos, em 2013 esse número subiu para 5.394.535 veículos, ou seja, um aumento de 139%.

Atualmente para conduzir qualquer veículo automotor rodoviário, existe a necessidade de um condutor habilitado por um órgão competente, este órgão é o DETRAN (2014).

No Brasil, para obter uma Carteira Nacional de Habilitação (CNH), é necessário atingir a maioridade (18 anos) e passar por testes psicológicos, os quais irão avaliar se o indivíduo pode ser habilitado, frequentar curso de formação de condutores e ao final deste é realizado a prova teórica, para verificar o aprendizado do aluno.

O teste teórico verifica conhecimentos sobre mecânica, legislação, direção defensiva, primeiros socorros, meio ambiente e convívio social. Consiste de trinta perguntas, que devem ser respondidas em até cinquenta minutos. Para ser

aprovado o candidato necessita conseguir um mínimo de 70% de acerto, ou seja, vinte e uma questões. (CONTRAN, 2004).

Devido ao grande número de áreas de abrangência para que o aluno seja habilitado na prova teórica, o DETRAN, a fim melhorar ainda mais os níveis de aprendizado sobre o trânsito, disponibiliza via site um meio para que todos os candidatos possam estudar e também testar seus conhecimentos.

Porém a solução ofertada foi desenvolvida para computadores convencionais, sendo que nem sempre o candidato possui acesso a um durante todo o tempo, perdendo a oportunidade de estudar enquanto espera um ônibus, está na fila do banco ou entre uma atividade e outra. Neste contexto, o uso de dispositivos móveis poderia aumentar a qualidade do aprendizado dos candidatos, fornecendo com materiais interativos (questões, simulados, etc) a qualquer momento.

O teste teórico é dividido em vários assuntos os quais representam as diversas situações que o condutor poderá vivenciar no dia a dia no trânsito. Como forma de teste ou até mesmo de estudo sobre um determinado assunto, o qual o usuário tem mais dificuldades em aprender, existe a necessidade de o usuário poder selecionar alguns temas, o que ajuda na compreensão de determinados temas onde o candidato possui maior dificuldade.

Dessa forma o *software* não se restringe a apenas candidatos a primeira habilitação, mas também para quem deseja realizar a reciclagem. A reciclagem é realizada por condutores habilitados que perderam sua carteira por descuido ou falta de conhecimento sobre legislação de trânsito, assim causando um excesso de infrações de trânsito, ou até por não terem conhecimento sobre as novas legislações.

Contudo, a legislação sofre algumas mudanças com o passar do tempo, como por exemplo, a modificação dos valores das multas, mudança nos pontos que o condutor vai levar na habilitação se fizer alguma infração, e até novas proibições que não existiam anteriormente. Isso implica em alteração de materiais de estudo e alterações das questões da prova teórica do DETRAN. Com a finalidade de que o condutor mantenha seus conhecimentos sempre renovados.

Neste contexto, o foco deste trabalho é disponibilizar um ambiente de estudo para a prova teórica do DETRAN e obtenção de CNH, por meio de material interativo e questões com respostas. Este desenvolvido para dispositivos móveis, o qual é uma extensão de um trabalho publicado no 3º Congresso Brasileiro de Informática

na Educação (CBIE), na cidade de Dourados em Mato Grosso do Sul (BRITO, Robison C.; CARNIEL, Andrei; COPETTI, Mauricio, 2014).

1.2 OBJETIVOS

Nesta seção serão apresentados os objetivos geral e específico do trabalho.

1.2.1 Objetivo Geral

Desenvolver um *software* para o estudo e a simulação do teste teórico do DETRAN, com suporte a imagens, método de estudo seletivo e sincronização com um servidor na Internet para atualização do banco de dados do dispositivo móvel.

1.2.2 Objetivos Específicos

- Implementação de um banco de dados para dispositivos móveis e web, com uma base ampla de perguntas e suas respectivas alternativas e respostas.
- Desenvolvimento de um módulo específico para simular a prova teórica que será aplicada no DETRAN, com estatísticas de acertos e erros no final do simulado.
- Implementação de suporte a perguntas com imagens.
- Seletor de categorias para estudos, o usuário poderá escolher uma ou mais categorias de estudos para testar seus conhecimentos.
- Utilização do framework Ruby on Rails para o desenvolvimento do aplicativo servidor para efetuar a atualização do banco de dados dos dispositivos móveis via HTTP.

1.3 JUSTIFICATIVA

Segundo IDC (2013) o recorde de vendas de *smartphones* foi superado no ano de 2013, com 68 aparelhos vendidos por minutos, onde 80% desses utilizam Android, o que demonstra um grande crescimento na utilização desses dispositivos. Além disso, os dispositivos estão expandindo suas capacidades (processamento, armazenamento, entre outros) cada vez mais, as vezes se igualando ou até mesmo ultrapassando a capacidade de um computador.

Com base nesses dados, e na praticidade de levar um *smartphone* para qualquer lugar, o *software* foi desenvolvido visando a utilização estes dispositivos. Outro ponto a favor é a facilidade de conexão com a Internet através da rede da operadora, a qual pode fornecer várias opções para acesso a rede, conforme a necessidade da pessoa (Wi-fi, GPRS, 3G, entre outras).

Com a constante modificação na legislação de trânsito, o aplicativo visando a constante atualização do condutor com as novas legislações de trânsito, irá disponibilizar novas perguntas e quando o usuário desejar, as perguntas vão ser sincronizadas. Assim o *software* irá se conectar a um servidor na Internet e efetuar o *download* das questões, evitando que o usuário tenha que baixar uma nova versão do aplicativo a cada inclusão de novas perguntas no servidor.

Também visando a importância no contexto educacional, este trabalho ressalta a importância do *Mobile Learning*, termo utilizado para o uso de conteúdo didático em dispositivos móveis que, segundo (MARÇAL, ANDRADE, RIOS, 2005, p. 03), tem como objetivos: melhorar os recursos para o aprendizado do aluno, aumentar as possibilidades de acesso ao conteúdo; expandir estratégias de aprendizado disponíveis e fornecer meios para o desenvolvimento de métodos inovadores de ensino.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em capítulos, dos quais este é o primeiro e apresenta as considerações iniciais referentes ao trabalho, os objetivos gerais e específicos, e a justificativa.

O Capítulo 2 apresenta o referencial teórico do trabalho, onde terá ênfase na primeira habilitação, em seguida, sobre a plataforma em que o *software* será desenvolvido, o Android.

No Capítulo 3 será abordado sobre os materiais utilizados no desenvolvimento do *software* juntamente com os seus conceitos.

No Capítulo 4 será falado sobre os resultados obtidos no desenvolvimento. Primeiramente uma visão geral do sistema, a modelagem e a descrição dele.

Por fim no Capítulo 5 está a conclusão com as considerações finais do trabalho.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados conceitos relacionados da primeira habilitação, em seguida, sobre a plataforma em que o *software* será desenvolvido, o sistema operacional Android e por fim sobre o contexto educacional.

2.1 PRIMEIRA HABILITAÇÃO

Independência e mobilidade são palavras chaves para conseguir realizar todas as tarefas do dia, principalmente quando a realização de uma tarefa envolve percorrer grandes distâncias. Nesse ponto muitas vezes encontramos a necessidade de ter uma CNH, a qual é necessária para que uma pessoa possa conduzir um veículo automotivo.

A aquisição da CNH envolve todo um processo de habilitação, onde o candidato deverá preencher previamente os seguintes requisitos: ser penalmente imputável, saber ler e escrever, possuir documento de identidade e CPF. Depois do cadastramento no RENACH, o candidato passara por “Avaliação Psicológica, Exame de Aptidão Física e Mental, Curso Teórico-técnico, Exame Teórico-técnico, Curso de Prática de Direção Veicular e Exame de Prática de Direção Veicular, nesta ordem” (CONTRAN, 2004).

Caso o candidato faça o pedido da habilitação nas categorias “A” e “B” ao mesmo tempo, ele só precisará passar por um exame de aptidão física e mental e avaliação psicológica. Porém essas devem ser renovadas a cada cinco anos, exceto para pessoas com mais de sessenta e cinco anos que devem refazer os testes a cada três anos. Já os portadores de deficiências físicas e/ou mental terão o prazo a ser definido pelo perito. Se o candidato quiser adicionar categorias na sua habilitação deverá refazer os exames de Aptidão física e mental.

Próximo passo após ser aprovado no teste físico e mental e avaliação psicológica, o candidato deverá realizar um curso teórico de 30 horas de aula onde será abordado diversos assuntos. São eles:

- 12 horas aulas de Legislação de Trânsito – sinalização viária; penalidades e crimes de trânsito; direitos e deveres do cidadão; normas de circulação e conduta;

- 8 horas aula de Direção defensiva – Condições adversas; como evitar acidentes; cuidados com os demais usuários da via; estado físico e mental do condutor; situações de risco;
- 4 horas aula de Noções de Primeiros Socorros – sinalização do local do acidente; acionamento de recurso em caso de acidentes; verificação das condições gerais da vítima; cuidados com a vítima;
- 4 horas aula de Meio Ambiente e Convívio Social – o veículo como agente poluidor do meio ambiente; emissão de partículas; emissão sonora; relacionamento interpessoal;
- 2 horas aula de Mecânica – equipamentos de uso obrigatório do veículo e sua utilização; noções de manuseio; manutenção do veículo; soluções para eventualidades mais comuns.

Depois de concluída as aulas teóricas, o candidato deverá fazer o teste teórico a ser realizado no DETRAN. Caso o candidato seja aprovado ele receberá a Licença para Aprendizagem de Direção Veicular (LADV), necessário para que o candidato possa iniciar aulas práticas acompanhado por um instrutor de uma auto escola devidamente credenciada. O processo é composto de 15 horas aulas, onde segundo o CONTRAN será ensinado sobre:

O veículo: funcionamento, equipamentos obrigatórios e sistemas;
Prática na via pública: direção defensiva, normas de circulação e conduta, parada e estacionamento, observação da sinalização e comunicação;
Os pedestres, os ciclistas e demais atores do processo de circulação.
(CONTRAN, 2004)

Caso o condutor já era habilitado e perdeu a habilitação (seja por descuido, ou por não ter conhecimento total das leis de trânsito, excesso de infrações, entre outros) ele terá que passar por um processo de reciclagem. Nesta situação serão ministradas 30 horas aula teóricas, sendo: legislação de trânsito (12 horas aula); direção defensiva (8 horas aula); primeiros socorros (4 horas aula) e relacionamento interpessoal (6 horas aula).

Para obter a CNH novamente, o candidato terá que fazer um exame e uma avaliação, sendo eles: o exame de aptidão física e mental, e a avaliação psicológica, que serão feitos por profissionais da área. Em seguida o candidato será submetido a prova teórica, que será aplicado pelo DETRAN de cada estado. Para a aprovação o candidato deverá obter no mínimo 70% de acerto, ou seja, no mínimo 21 questões

corretas, de um total de 30. Por último existe o teste prático que deve ser realizado na obtenção, adição ou mudança de categoria, este teste consiste em duas etapas para veículos com quatro ou mais rodas e uma etapa para motocicletas.

Sendo a primeira etapa para veículos com quatro ou mais rodas, é a baliza que deve ser completada em um determinado tempo, dependendo da categoria que o candidato está realizando o teste. Sendo eles, de 2 a 5 minutos para categoria “B”; 3 a 6 minutos para categoria “C” e “D” e 6 a 9 para a categoria “E”. Ainda nesta etapa, o candidato deve conduzir veículo em área pública, ambos os testes deveram ser acompanhados por no mínimo dois membros da comissão.

Já a segunda etapa, é unicamente para motocicleta e será realizada em um circuito fechado com alguns obstáculos. Essas são as etapas padrões para obter a primeira habilitação. Lembrando que não são obrigatórias, já que o candidato pode optar por realizar somente uma das etapas.

O número de habilitados vem crescendo e isso está diretamente relacionado com o numero de veículos nas estradas, pois sem ela não é possível dirigir legalmente no país. Somente no Brasil segundo IBGE (2012), o número de automóveis chegou a 42.682.049 enquanto a população era de 193.946.886, ou seja, isso significa 4.5 habitantes para cada carro.

A figura abaixo mostra o número de carros e motos para cada habitante em 2014.

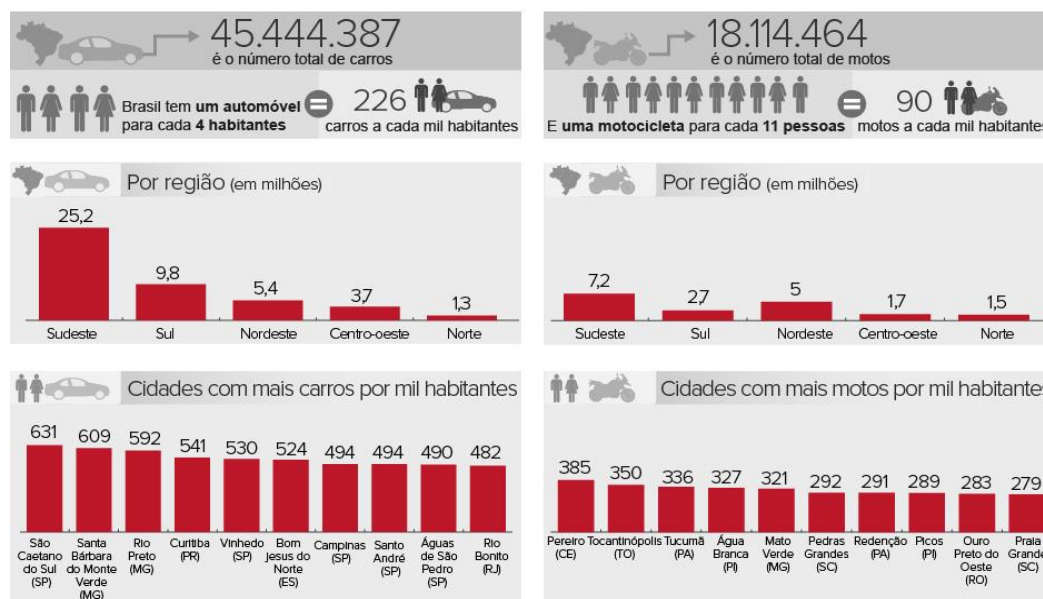


Figura 1 - Infográfico de carro e moto por habitantes.
Fonte: REIS, Thiago (2014).

Com esse grande aumento do número de automóveis, ocorre que algumas vezes nem todos os motoristas estão preparados para dirigir nas estradas, o que gera um crescimento no número de acidentes e consecutivamente mortes, como pode ser visto na Figura 2.

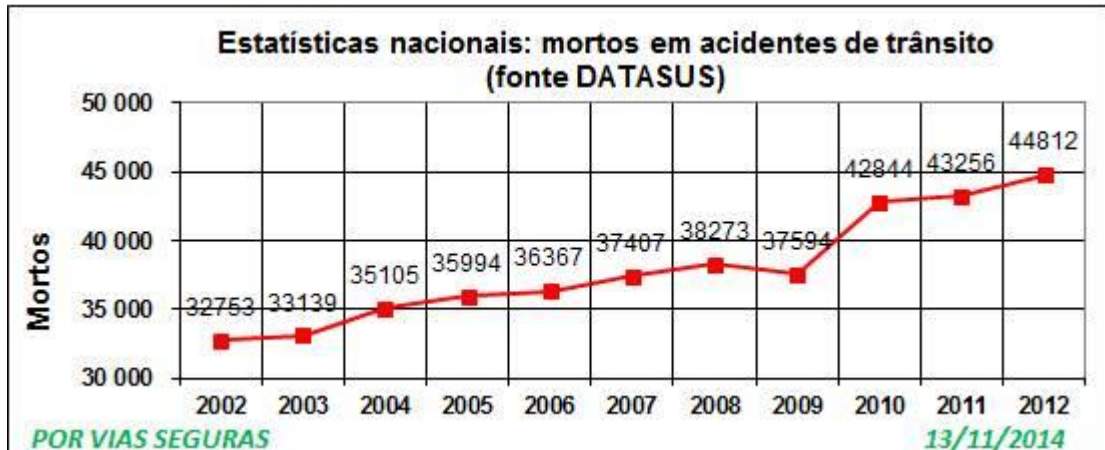


Figura 2 - Estatística de mortes no trânsito.
Fonte: DATASUS (2014).

Como pode ser visto no gráfico, o número de mortes no trânsito vem crescendo. Com isso o *software* visa auxiliar o motorista com questões como: “O motorista que não mantém a distância de seguimento pode envolver-se em colisão com o:... ”; neste caso auxiliando também a todo usuário que deseje apenas testar novamente seus conhecimentos ou relembrar seu aprendizado.

2.2 TECNOLOGIA ANDROID E O DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS

O Android é uma tecnologia desenvolvida para dispositivos móveis, que foi construída inicialmente por Andy Rubinera, Nick Sears e Chris White, em outubro de 2003 na cidade de Palo Alto, Califórnia – EUA. Dois anos mais tarde a Google adquiriu a companhia e começou o desenvolvimento com o código aberto, em cima do *kernel* do Linux versão 2.6. O projeto Android hoje está ligado à *Open Handset Alliance* (OHA) fundada em 2007, que é um consórcio de empresas de tecnologia composta por Google, NVidia, Samsung e várias outras fabricantes de dispositivos e empresas de telefonia, que se uniram para fomentar o crescimento da plataforma Android (OGLIARI; BRITO, 2014).

O primeiro aparelho com o sistema Android foi o HTC Dream, que foi lançado em 2008. A partir desse momento o número de dispositivos só aumentou, chegando a uma estimativa de venda de um bilhão de unidades em 2014 (GARTNER, 2014).



Figura 3 - HTC Dream ou T-Mobile G1, o primeiro celular com Android

Na parte de instalação de aplicativos, o Android disponibiliza uma loja virtual que permite baixar, comprar ou vender aplicativos, filmes, músicas e *E-books*, chamado de Google Play. Mas não se restringe somente a isso, se o usuário desejar pode navegar por inúmeros sites na Internet. E no caso dos desenvolvedores, eles podem vender seu *software* ficando com 70% lucro e os outros 30% vão para parceiros de distribuição e para taxas operacionais.

Mesmo tendo uma grande compatibilidade com as diversas tecnologias, ainda existem problemas na hora de usar sites criados para uso em *desktop* em dispositivos móveis, como no próprio site do DETRAN que, se utilizado em um celular, encontrará problemas no design, como é possível ver na Figura 3, onde o teclado está colocado sobre a pergunta o que impossibilita ao usuário ler toda a questão, e também existe o incomodo de ter que constantemente alterar o *zoom* da tela para conseguir visualizar o conteúdo total da página.



Figura 4 - Teclado sobrepondo a questão.

Esse problema ocorre pois, a página de simulado, foi desenvolvida para computadores convencionais e não para dispositivos móveis. Se fosse desenvolvido um *software* para plataforma móvel, a navegabilidade e a interface seriam mais atrativas, assim o usuário não encontraria problemas na utilização do aplicativo.

2.3 UTILIZAÇÃO DE DISPOSITIVOS MÓVEIS NO CONTEXTO EDUCACIONAL

O Brasil é o 4º maior mercado consumidor de *smartphones* de mundo (TOZETTO, 2014). Destes, segundo um estudo realizado pela IDC no último trimestre de 2013, 88,73% dos *smartphones* vendidos possuíam Sistema Operacional Android, 6% possuíam Windows Phone e 4,7% possuíam o sistema IOS (OLHARDIGITAL, 2014). Segundo Teleco (2014), no mês de dezembro de 2013 o número de *smartphones* no mercado correspondia a 2,5 milhões de aparelhos, contra 1,4 milhões de aparelhos celulares tradicionais, o que mostra a superioridade desta nova plataforma.

Neste contexto, empresas e instituições estão investindo no *mobile learning*, termo utilizado para o uso de conteúdo didático em dispositivos móveis. Segundo (MARÇAL, ANDRADE, RIOS, 2005, p. 03), os objetivos de aplicações *mobile learning* são:

- Melhorar os recursos para o aprendizado do aluno, que poderá contar com um dispositivo computacional para execução de tarefas, anotação de ideias, consulta de informações via Internet, registro de fatos através de câmera digital, gravação de sons e outras funcionalidades existentes;
- Prover acesso aos conteúdos didáticos em qualquer lugar e a qualquer momento, de acordo com a conectividade do dispositivo;
- Aumentar as possibilidades de acesso ao conteúdo, incrementando e incentivando a utilização dos serviços providos pela instituição, educacional ou empresarial;
- Expandir o corpo de professores e as estratégias de aprendizado disponíveis, através de novas tecnologias que dão suporte tanto à aprendizagem formal como à informal;
- Fornecer meios para o desenvolvimento de métodos inovadores de ensino e de treinamento, utilizando os novos recursos de computação e de mobilidade.

Um exemplo é a Fundação Vanzolini, que firmou uma parceria com escolas municipais da cidade de São Vicente (SP), onde o objetivo do projeto é incluir conteúdo didático móvel no currículo escolar do município, para que ocorra um maior interesse dos alunos. No entanto, muitos professores ainda têm certas dificuldades em utilizar os softwares para dispositivos móveis nas salas de aula.

Em uma enquete do portal (INSTITUTO CLARO, 2011) quase metade dos educadores afirmaram querer integrar a ferramenta para dispositivos móveis à sala de aula, resultado este apresentado na Figura 5.

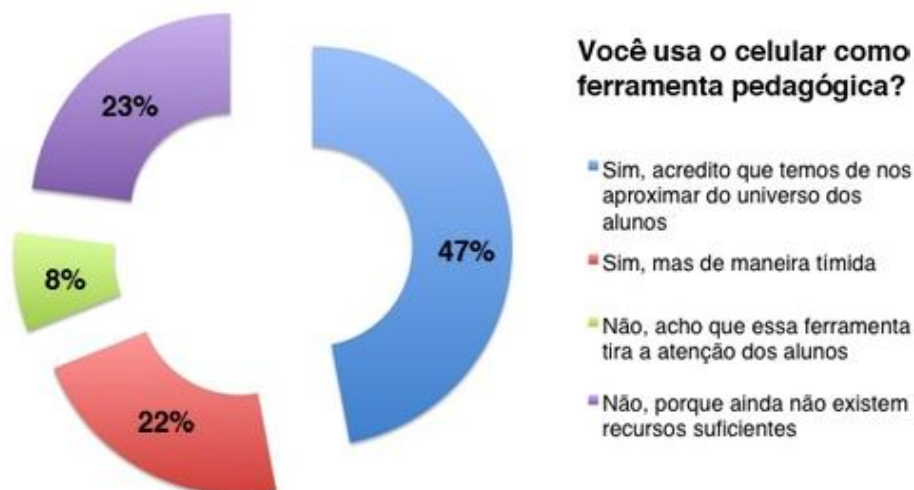


Figura 5 - Recursos Pedagógicos no Celular.

Fonte: Instituto Claro (2014).

De acordo com a Figura 5, pode-se observar que um número considerável de pessoas (23%) afirma que os celulares não continham recursos pedagógicos suficientes para seu bom aproveitamento nos estudos.

2.4 TRABALHOS RELACIONADOS

Existem alguns trabalhos relacionados no mercado. O simulado teórico do site do DETRAN, foi utilizado como base para o desenvolvimento do aplicativo, sendo este focado em computadores convencionais e não muito atrativo para a utilização em dispositivos móveis. A seguir serão mostrados alguns aplicativos semelhantes disponíveis na Google Play, loja de aplicativos da Google. Todos os aplicativos podem ser avaliados com uma nota de 1 a 5 pelos usuários.

DETRAN Simulados (Figura 6): um aplicativo simples, que disponibiliza nove simulados com trinta questões cada. Caso o usuário deseje ter mais tipos de provas é necessário baixar a versão *Premium*, pode ser usado sem a conexão com a internet, disponível para sistemas Android 2.3 ou superior, e com nota média de 4.1, avaliada por 3080 pessoas na Google Play.

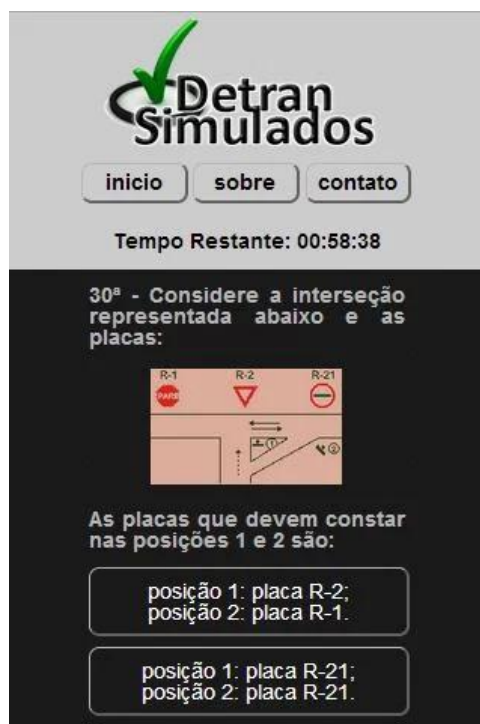


Figura 6 - Detran Simulados
Fonte: Google Play (2014)

Simulado DETRAN-SP (Figura 7): um aplicativo feito para a prova teórica do estado de São Paulo, disponibilizado apenas a funcionalidade de simulado, com trinta questões geradas aleatoriamente, mas para poder utilizar o mesmo é necessário estar conectado na Internet, disponível para sistemas Android 2.2 ou superior e iOS 6.1 ou posterior, e com nota de 4.2, avaliado por 4001 pessoas na Google Play.

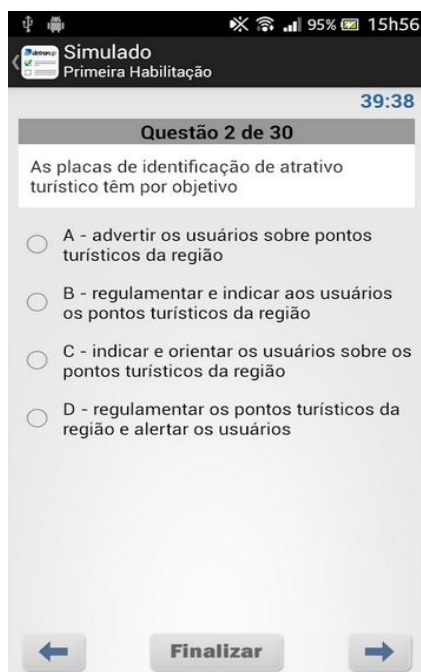


Figura 7 - Simulado DETRAN SP
Fonte: PRODESP (2014)

Simulado DETRAN RJ (Figura 8): aplicativo desenvolvido por Lucas R. Martins, feito para a prova teórica do Rio de Janeiro, pode ser utilizado sem conexão com a internet, disponível para aplicativos com Android 2.2 ou superior, com uma nota de 4.3, avaliado por 1730 pessoas na Google Play.

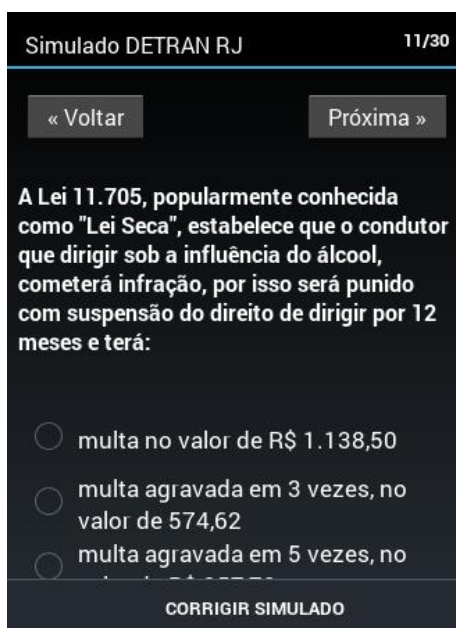


Figura 8 - Simulado DETRAN RJ
Fonte: MARTINS, Lucas R (2014)

O mais importante é que todos esses aplicativos citados anteriormente tiveram uma avaliação positiva. Ambos conseguiram uma nota superior a 4, e a maioria dos comentários foram positivos, destacando a importância que os dispositivos móveis estão ganhando no mercado.

3 MATERIAIS E MÉTODO

Este capítulo aborda os materiais utilizados no desenvolvimento do *software*, juntamente com os seus conceitos.

3.1 MATERIAIS

Para o desenvolvimento do projeto foram utilizadas diversas ferramentas, apresentadas no Quadro 1.

Materiais			
Software	Versão	Licença	Propriedade
Java SE	1.7.0_15-b03	Gratuito	Oracle
ADT Bundle	22.3.0.v2013102420 05-887826	Gratuito	Google
Android Plataform/ API Level	15	Gratuito	Google
SQLite	3.4.0	Gratuito	SQLite Consortium
Eclipse	Juno Service Release 2	Gratuito	Eclipse Foundation
Visual Paradigm Standard Edition	11.1	Trial (30 dias)	Visual Paradigm
Ruby on Rails	3.2	Gratuito	MIT
MySQL	14.14	Gratuito	Oracle

Quadro 1 - Materiais utilizados

3.1.1 JAVA SE

Java Platform Standard Edition permite desenvolver e implementar aplicativos Java em desktops e servidores. O Java SE é composto pela biblioteca Java e a JVM (*Java Virtual Machine*), necessária tanto para desenvolvedores como para usuários finais, pois a biblioteca Java interpreta o arquivo e a JVM roda os aplicativos. Java SE também possui uma variedade de APIs como, por exemplo: JDO (*Java Data Objects*), JOGL (*Java OpenGL*) e muito mais.

Vale ressaltar que as IDEs utilizam o Java SE para desenvolver e futuramente rodar as aplicações. Qualquer pessoa pode baixar o Java gratuitamente em JDK (2014).



Figura 9 – Tela de configurações do Java SE.

3.1.2 ADT Bundle

O ADT Bundle é composto pelo *plugin Android Developer Tools* (ADT) que fornece um conjunto de ferramentas que são integradas com a IDE Eclipse. Ele oferece acesso a muitos recursos que ajudam a desenvolver aplicativos Android rapidamente. O ADT também proporciona acesso via GUI (*Graphical User Interface*) para muitas das ferramentas do SDK (*Software Development Kit*).

O plugin ADT Bundle é composto por três ferramentas: IDE Eclipse, Android SDK e o AVD.

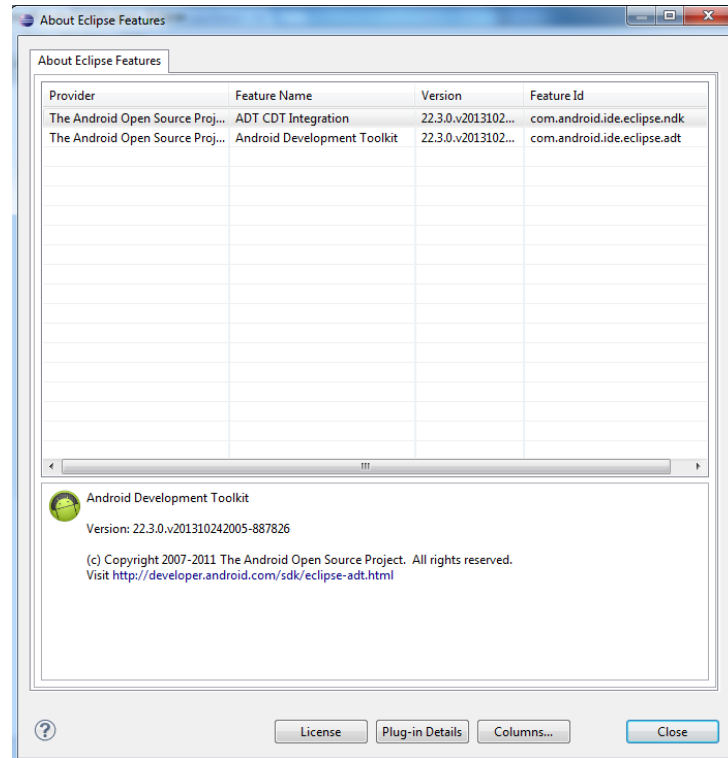


Figura 10 – Tela de configurações do Android Developer Tools

3.1.2.1 Eclipse

Eclipse é uma IDE de desenvolvimento Java, porém ela suporta várias outras linguagens por meio da instalação de *plugins*, ela foi feita em Java e segue o modelo *open source*. Atualmente ela faz parte do ADT Bundle (DEVELOPERS, 2014) e em 2012 foi a IDE de desenvolvimento mais popular entre os desenvolvedores Java no mundo (WHITE, Oliver, 2012).

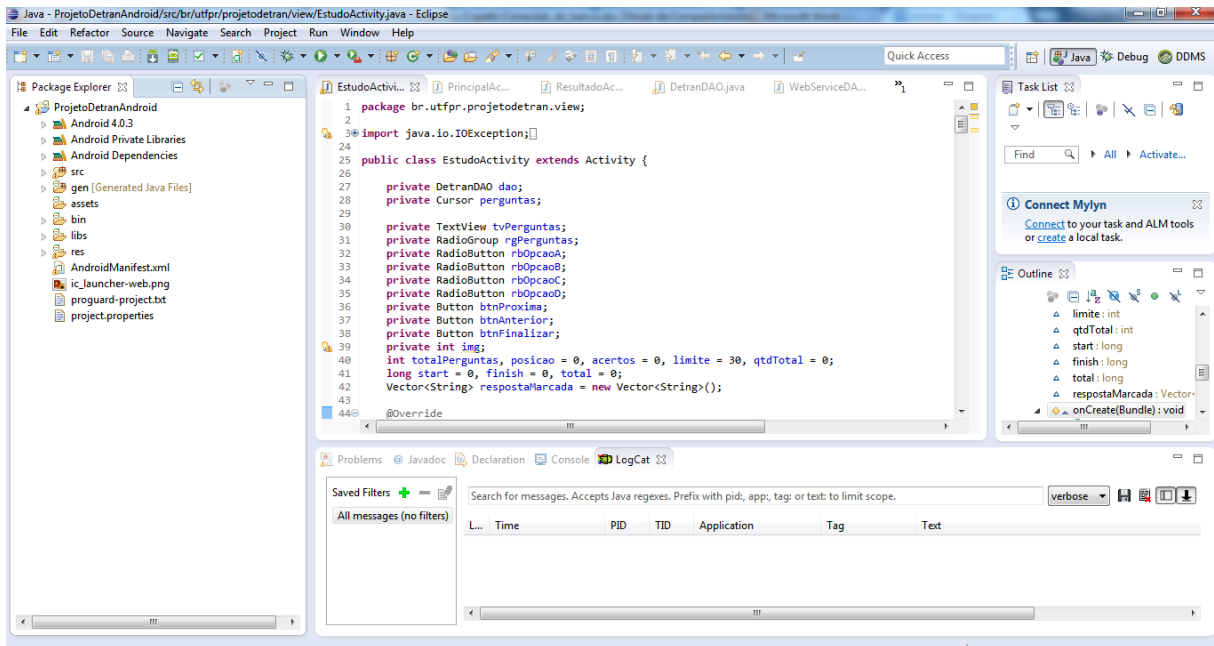


Figura 11 - IDE Eclipse

3.1.2.2 Android SDK

O ADT Bundle também contém o Android SDK que fornece as bibliotecas API (*Application Programming Interface*) e as ferramentas necessárias para construir, testar e debubar os aplicativos Android (DEVELOPERS, 2014) como se pode ver na Figura 12.

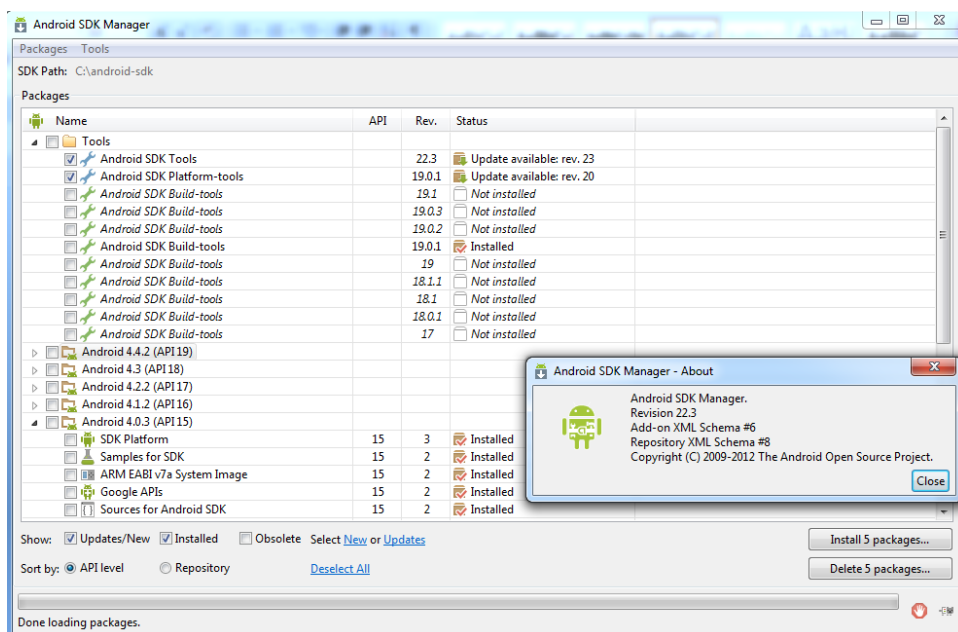


Figura 12 - Android SDK Manager

3.1.3 SQLite

SQLite é o banco de dados relacional padrão do Android, que está disponível desde a primeira API, ele é um banco de dados gratuito, que não precisa de configuração, pois é pré configurado de fábrica, e também não precisa de servidor porque é implementado diretamente no *smartphone/tablet*. O código do banco está em domínio público, portanto é livre para ser utilizado para qualquer fim, seja comercial ou não.

É um banco compacto, que com todos os recursos habilitados possui o tamanho inferior a 500KB, dependendo da plataforma e do compilador, tornando uma escolha popular para dispositivos com memória limitada, como celulares e MP3 *players*.

Algumas das características do banco são que as transações são atômicas mesmo depois de falhas de sistema ou falhas de energia, ou seja, uma operação ou um conjunto de operações em uma base de dados deve ser executada completamente em caso de sucesso, ou ser abortada completamente em caso de erro. Além disso, o SQLite ainda suporta banco de dados de *terabytes*, e é simples e de fácil manuseio (SQLITE, 2014).

3.1.4 Visual Paradigm

Visual Paradigm é uma ferramenta paga, com uma versão de teste de 30 dias, para modelagem de sistemas, UML (*Unified Modeling Language*), desenvolvimento de diagramas, entre outros. A ferramenta é comercial e também oferece suporte a transformação para código fonte de algumas linguagens como, por exemplo, C++ e Java. Facilitando a criação de classes, banco de dados e do projeto como um todo.



Figura 13 - Visual Paradigm

3.1.5 Ruby on Rails

Ruby on Rails é um *framework* de desenvolvimento web (gratuito e de código aberto) desenvolvido em 2003 por David Heinemeier Hansson (RUBYONRAILS, 2014), que aumenta a velocidade e facilita o desenvolvimento de sites orientados a banco de dados, uma vez que é possível criar aplicações com base em estruturas pré-definidas. As aplicações criadas utilizando o *framework* Rails são desenvolvidas com base no padrão de arquitetura MVC (padrão de projeto de software onde: *Model* ou Modelo, contém toda as regras, lógicas e funções; *View* ou Visualização, todas as saídas de dados; e a *Controller* ou Controladora, coordena as camadas).

O Rails é um "*meta-framework*" (ou seja, um framework de frameworks), composto pelos seguintes *frameworks*: Active Record, Action Pack, Action Mailer, Active Support (RUBYONRAILS, 2012).



Figura 14 – Logo Ruby on Rails

3.1.6 MySQL

O MySQL é um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface. É atualmente o segundo banco de dados mais popular segundo DB-ENGINES (2014). Entre os usuários do banco de dados MySQL estão: NASA, Nokia, Sony, U.S. Navy, Cisco Systems, Google e outros (MYSQL, 2014).

Algumas das suas principais características são a portabilidade, compatibilidade, desempenho e estabilidade, facilidade no manuseio, suporta *stored procedures, functions, triggers, cursors* e além de tudo é um *software* livre com base na GPL (MYSQL COMMUNITY, 2014).

3.2 MÉTODO

O desenvolvimento do *software* para Teste de Conhecimento para a Prova Teórica do DETRAN foi dividido em etapas, seguindo o modelo sequencial linear de Pressman (2006). Essas etapas são:

a) Requisitos: Os requisitos para o desenvolvimento deste sistema foram criados com base na análise e utilização da solução ofertada pelo DETRAN.

b) Análise: O caso de uso descreve o cenário que mostra as funcionalidades do aplicativo, para o qual seria proposto uma solução. Os casos de uso serão criados utilizando a ferramenta Visual Paradigm, para produzir o diagrama UML.

c) Projeto: Nesta fase foi criado um diagrama de classe utilizando a linguagem UML.

d) Desenvolvimento: O desenvolvimento será realizado utilizando as tecnologias: Java SE na versão 1.7.0_15-b03; ADT Bundle 22.3.0, com seus respectivos softwares, IDE Eclipse, Android SDK 4.0.3 (API 15) e o Android AVD também na versão 15 da API.

e) Teste: Testes são necessários para garantir o funcionamento esperado do aplicativo criado. Foi feito de 3 maneiras diferentes: A primeira foi diretamente no Android AVD, através da máquina virtual foi elaborado as primeiras versões e efetuado os primeiros testes; Após instalado em um *smartphone* convencional onde foi verificado a interface e usabilidade; E por fim a instalação em um tablet, já que muitos desses dispositivos também são compatíveis com o *software* produzido.

f) Implantação: Consistiu em testar o *software* no próprio emulador e em um *smartphone*.

4 RESULTADOS

Nesse capítulo será falado sobre os resultados obtidos no desenvolvimento. Primeiramente uma visão geral do sistema, sua modelagem e sua descrição.

4.1 ESCOPO DO SISTEMA

O sistema de simulado e estudo do DETRAN vai ajudar o candidato a se preparar para o exame teórico oficial. Disponibilizando dois métodos de aprendizado, o modo estudo e o modo simulado, além de um Web Service para atualização do banco de dados de perguntas, para que o usuário não precise atualizar o aplicativo toda vez que saia alguma pergunta nova.

No modo estudo as perguntas serão geradas aleatoriamente de um banco de dados com várias questões retiradas do próprio site do DETRAN. Esse modo não tem limite de tempo e se o usuário errar o próprio sistema vai marcar a resposta correta em verde, o que o ajuda a perceber onde estão seus erros e com isso aumentar a chance de melhorar e não reprovar no teste, e também contempla um seletor de perguntas por categoria, onde ele poderá escolher quais serão os temas das perguntas e quantas serão respondidas.

O modo simulado que acontece com limite de tempo e também irá gerar questões aleatórias do banco de dados, mas o resultado só será exibido no final do teste se o usuário foi aprovado ou não, e qual sua pontuação final, sem a possibilidade de saber qual questão errou e nem de escolher os temas das perguntas.

4.2 MODELAGEM DO SISTEMA

Para o desenvolvimento do sistema foram definidos alguns requisitos funcionais e não funcionais, estes listados nos Quadros 2, 3, 4, 5 e 6, respectivamente. Os requisitos funcionais descrevem as funcionalidades do sistema e os requisitos não-funcionais descrevem as restrições e regras para a elaboração do sistema.

RF 1 Requisito Funcional 1 – Modo Estudo Seletivo			
Descrição: Através desse módulo será efetuado o estudo da prova teórica do DETRAN.			
Requisitos Não-Funcionais			
Nome	Categoria	Desejável	Permanente
RNF 1.1 Mostrar a resposta correta após a primeira seleção.	Usabilidade	X	X
RNF 1.2 Não ter limite de tempo.	Usabilidade	X	X
RNF 1.3 Gerar perguntas aleatórias.	Usabilidade	X	X
RNF 1.4 Exibir resultados de acertos e tempo total de prova.	Usabilidade	X	X
RNF 1.5 Exibir imagens quando disponível.	Interface	X	X
RNF 1.6 Permitir escolha do tema e do número de perguntas.	Usabilidade	X	X
RNF 1.7 Funcionar em ambientes <i>touchscreen</i> .	Disponibilidade	X	X
RNF 1.8 Não precisar de conexão com a Internet.	Disponibilidade	X	X
RNF 1.9 Gerar o banco de dados automaticamente.	Desempenho	X	X
RNF 1.10 Permitir que o usuário altere quantas vezes quiser a resposta.	Usabilidade	X	X

Quadro 2 - Requisito Funcional Estudo Seletivo

RF 2 Requisito Funcional 2 – Modo Simulado			
Descrição: Através desse módulo será efetuado o simulado da prova do DETRAN.			
Requisitos Não-Funcionais			
Nome	Categoria	Desejável	Permanente

RNF 2.1 Não exibir a resposta correta.	Usabilidade	X	X
RNF 2.2 Limite de tempo de 50 minutos.	Usabilidade	X	X
RNF 2.3 Gerar perguntas aleatórias.	Usabilidade	X	X
RNF 2.4 Exibir resultados de acertos e o tempo gasto.	Usabilidade	X	X
RNF 2.5 Funcionar em ambientes <i>touchscreen</i> .	Disponibilidade	X	X
RNF 2.6 Não precisar de conexão com a Internet.	Disponibilidade	X	X
RNF 2.7 Gerar o banco de dados automaticamente.	Desempenho	X	X
RNF 2.8 Encerrar automaticamente depois de 50 minutos	Usabilidade	X	X
RNF 2.9 Exibir o tempo durante a execução do simulado.	Usabilidade	X	X
RNF 2.10 Permitir que o usuário altere quantas vezes quiser a resposta.	Usabilidade	X	X
RNF 2.11 Exibir imagens quando disponível.	Interface	X	X

Quadro 3 - Requisito Funcional Simulado

RF 3 Requisito Funcional 3 – Atualização via Web Service			
Descrição: Através desse módulo será efetuada a atualização das perguntas via Web Service.			
Requisitos Não-Funcionais			
Nome	Categoria	Desejável	Permanente
RNF 3.1 Exibir mensagem de alerta caso o usuário esteja na rede da operadora.	Segurança	X	X
RNF 3.2 Utilização do Ruby on Rails.	Praticidade	X	X

RNF 3.3 Utilização do protocolo HTTP.	Praticidade	X	X
---------------------------------------	-------------	---	---

Quadro 4 - Requisito funcional Atualização via Web Service

RF 4 Requisito Funcional 4 – Cadastro de perguntas via navegador de Internet			
Descrição: Através desse módulo o administrador irá adicionar as perguntas ao Web Service.			
Requisitos Não-Funcionais			
Nome	Categoria	Desejável	Permanente
RNF 4.1 Utilização do <i>framework</i> Ruby on Rails.	Usabilidade	X	X
RNF 4.2 Utilização do protocolo HTTP.	Usabilidade	X	X

Quadro 5 - Requisito Funcional Cadastro de perguntas via navegador

RF 5 Requisito Funcional 5 – Banco de dados <i>off-line</i>			
Descrição: Todas as perguntas devem estar salvas em um banco de dados <i>off-line</i> para utilização no software.			
Requisitos Não-Funcionais			
Nome	Categoria	Desejável	Permanente
RNF 5.1 Suportar armazenamento de imagens codificadas em Base 64.	Disponibilidade	X	X
RNF 5.2 Armazenar todas as perguntas.	Disponibilidade	X	X

Quadro 6 - Requisito Funcional Banco de dados *offline*

O diagrama de caso de uso representa agrupamentos de requisitos funcionais, já o diagrama de sequência representa a ordem das ações que devem ser tomadas pelo usuário. A Figura 15 apresenta o diagrama de caso de uso do sistema, e logo após, Figura 16, é apresentado o diagrama de sequência. Por fim o caso de uso do Web Service é apresentado, na Figura 17.

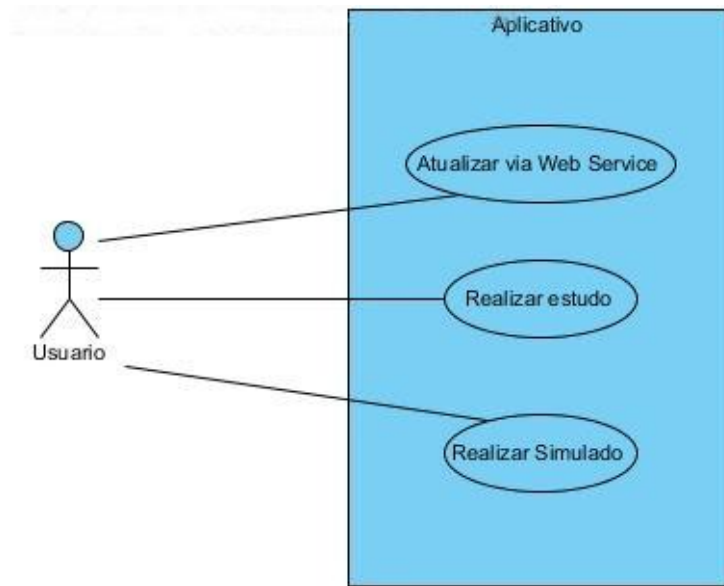


Figura 15 - Caso de uso

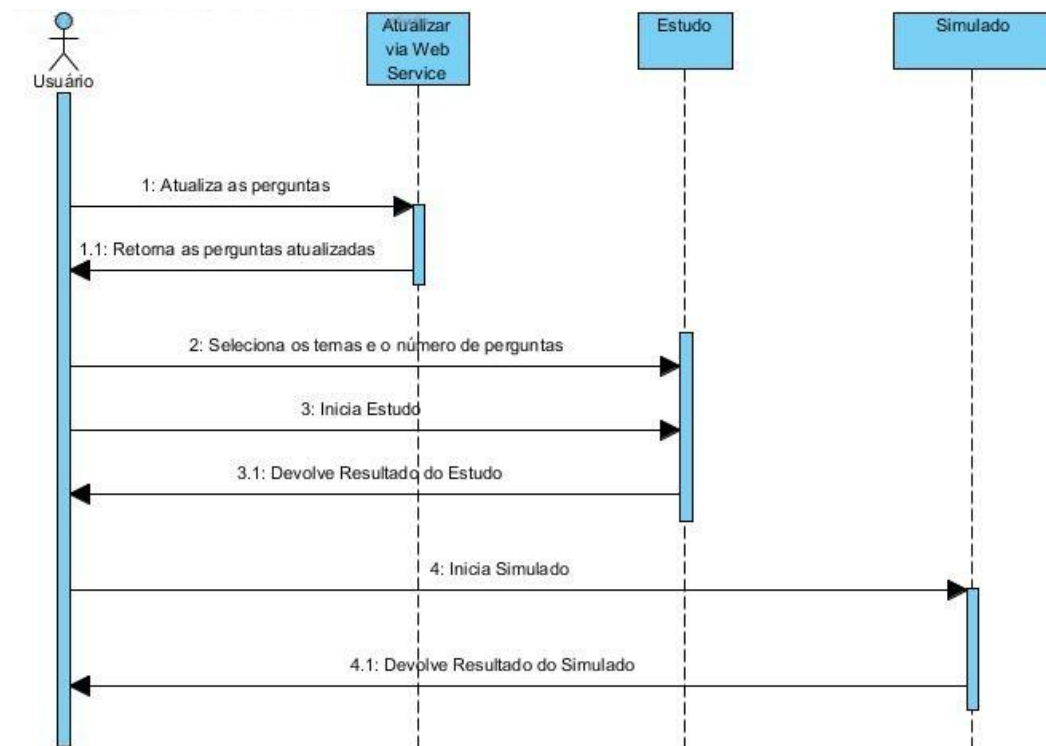


Figura 16 - Diagrama de sequência

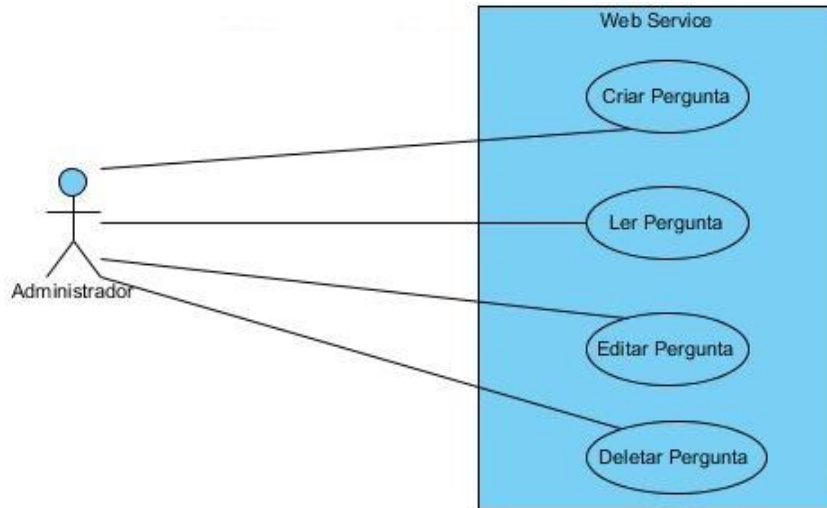


Figura 17 - Caso de uso Web Service

4.3 APRESENTAÇÃO DO SISTEMA

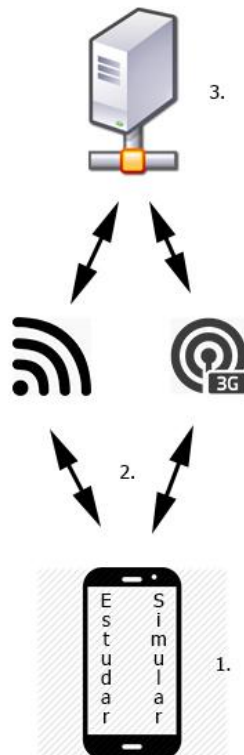


Figura 18 - Representação do aplicativo

Na Figura 18, os números representam:

1. *Smartphone* Android que vai realizar o Estudo ou o Simulado.
2. Meios de se comunicar com o servidor, que pode ser via WiFi ou 3G.
3. Servidor que armazena as perguntas e respostas para poder atualizar o banco de dados do *smartphone*.

O sistema para estudo e simulação da prova teórica do DETRAN foi desenvolvido com base no teste teórico disponível no site do DETRAN-PR (2014). O *software* possui 85 questões disponibilizadas diretamente no site do DETRAN, as quais serão utilizadas no *software*. Há dois módulos disponíveis, são eles: “Estudo Seletivo” e o “Simulado”.

A duas telas foram construídas de forma a permitir que as perguntas sejam exibidas com as alternativas de respostas em branco e marcação de uma única alternativa por questão. Após a primeira vez que o usuário selecionar uma alternativa não será mais possível deixar a questão em branco novamente, apenas trocar a opção (no caso do modo simulado, será permitido trocar a alternativa desde que esteja dentro do tempo previsto para a realização).

Todas as respostas por questões serão armazenadas no celular, assim se o usuário sair e resolver voltar para uma pergunta, ela estará marcada com a última alternativa assinalada pelo usuário.

Na Figura 19 é apresentada a tela inicial do aplicativo. Nessa tela o usuário vai ler uma série de instruções e deverá escolher qual dos módulos ele vai utilizar.

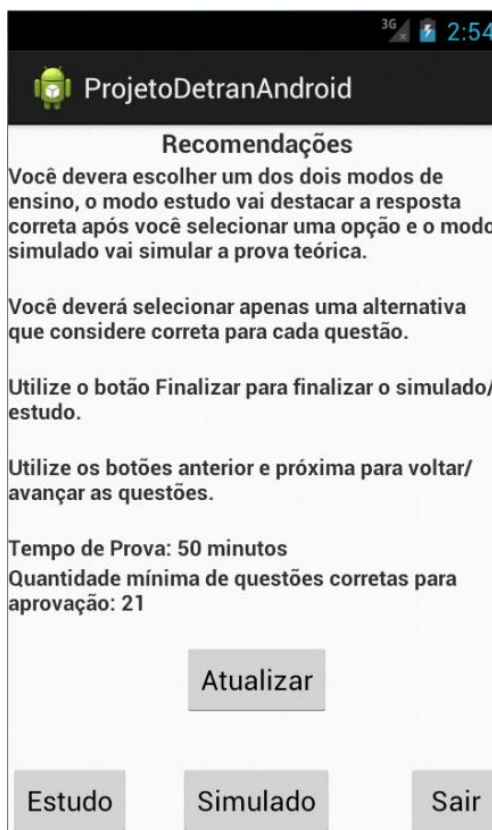


Figura 19 - Tela inicial do sistema

Na opção “Estudo” o usuário será redirecionado a tela de escolha dos temas a serem estudados, e a quantidade de perguntas disponível no banco de dados que o usuário deseja responder (Figura 20). O usuário deverá selecionar os temas que deseja estudar, e o aplicativo vai gerar as perguntas aleatórias do banco de dados com base na escolha anterior e exibidas uma a uma na tela do aplicativo.

Após a escolha dos temas e da quantidade de perguntas a serem respondidas, o usuário será redirecionado para a tela de resposta, conforme a Figura 21, mostrando as perguntas uma a uma com suas respectivas alternativas. O *software* vai auxiliar o usuário marcando a resposta correta com fundo verde caso ele erre ou acerte, o tempo para responder é ilimitado.

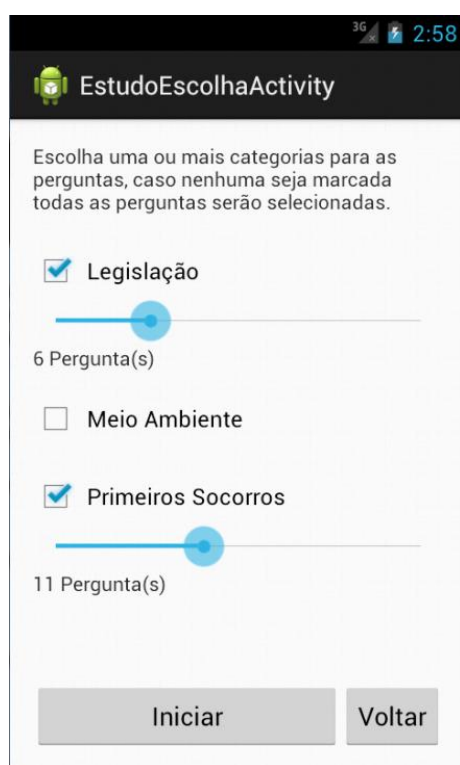


Figura 20 - Tela de escolha de temas

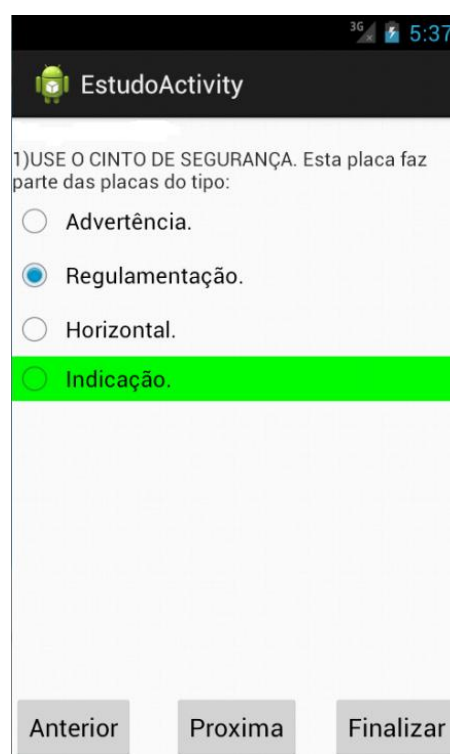


Figura 21 - Tela do modo Estudo

Os botões “Anterior” e “Próxima” possibilitam navegar entre as questões, facilitando ao usuário a troca de alternativa marcada, ou responder uma questão posteriormente. Ao clicar no botão Finalizar será exibido a tela de resultados com o número de acertos baseado nas alternativas marcadas e o tempo gasto, como pode ser observado na Figura 22. Vale ressaltar que o módulo estudo não tem limite de tempo, sendo apenas um dado para o usuário se auto avaliar.

O botão Menu Principal redireciona o usuário para a tela inicial, já o botão Sair fecha o aplicativo.

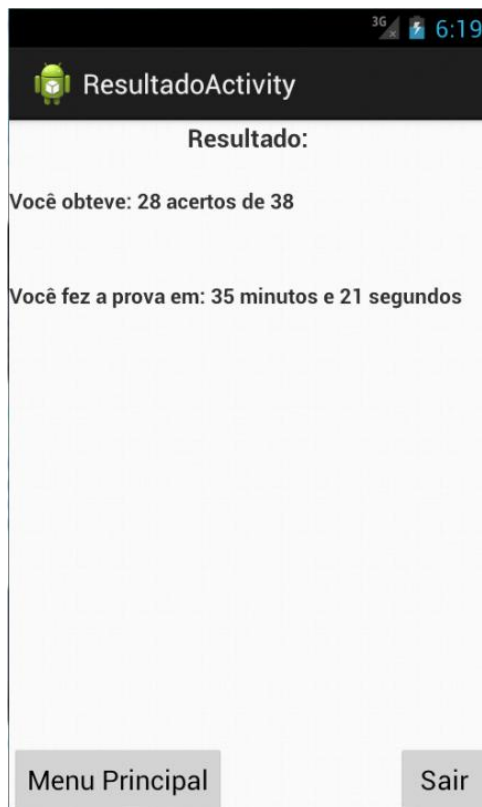


Figura 22 - Tela de resultado do módulo estudo

O módulo simulado é semelhante ao módulo estudo, porém, esse módulo não mostra a alternativa correta e possui um contador regressivo com o tempo relativo a prova teórica (50 minutos), isso para deixar o *software* mais próximo da representação do teste original. O *software* irá finalizar o módulo Simulado automaticamente, caso ainda esteja ativo e atingir o tempo limite.

O botão Simulado da tela principal vai redirecionar para a tela que simula o teste teórico (Figura 23).

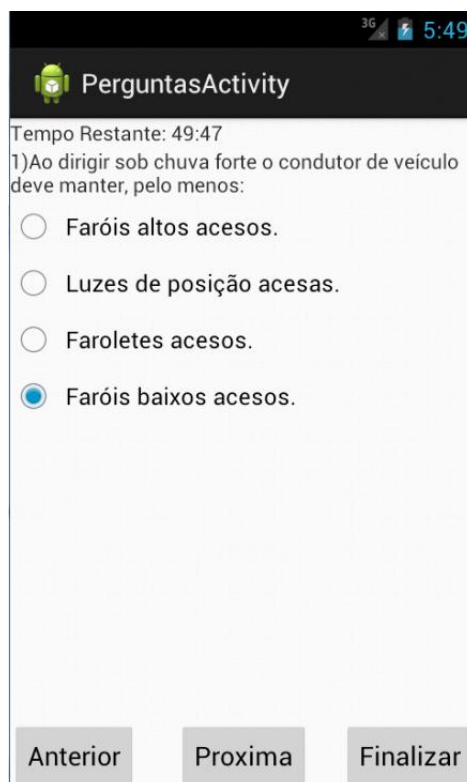


Figura 23 - Tela do modo simulado

Essa tela é parecida com a tela de estudo, porém, possui um cronômetro que limita o tempo que o usuário tem para fazer a prova, sendo como tempo total 50 minutos, e essa tela não mostra a resposta correta. Caso o limite de tempo seja atingido, o *software* finaliza o módulo automaticamente, verificando quantas perguntas estão certas e exibirá a tela de resultados. Os botões Anterior e Próxima voltam e avançam as questões. Já se o usuário clicar no botão Finalizar ele será redirecionado para a tela de resultados.

As Figuras 24 e 25 são as duas telas possíveis de resultado do módulo simulado, mostrando uma mensagem de aprovado ou reprovado com os seus acertos e o tempo que restou para fazer a prova.

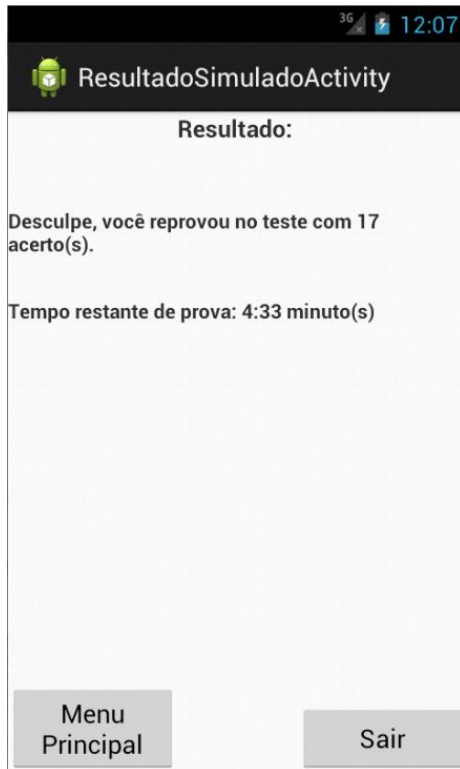


Figura 24 - Tela de reprovado do modulo simulado

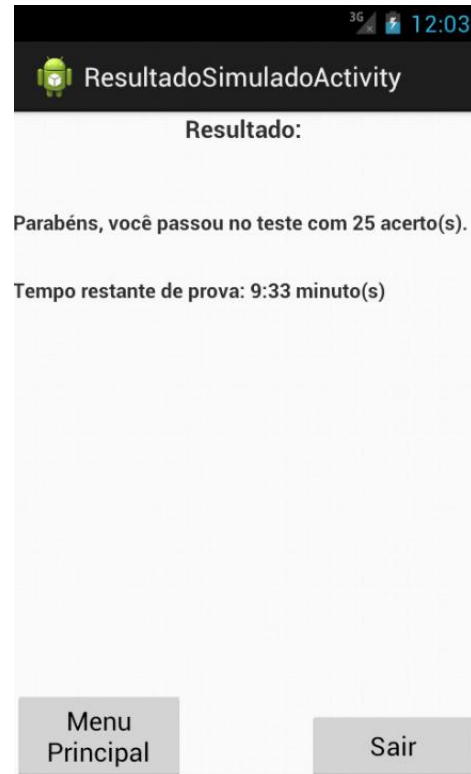


Figura 25 - Tela de aprovado do modulo simulado

Caso o usuário clique em atualizar no menu principal, há a possibilidade de duas telas distintas serem mostradas, se o usuário estiver utilizando rede móvel, a Figura 26 irá aparecer, e se o usuário clicar no botão sim ou se ele estiver conectado em rede Wi-fi a tela da Figura 27 vai aparecer.

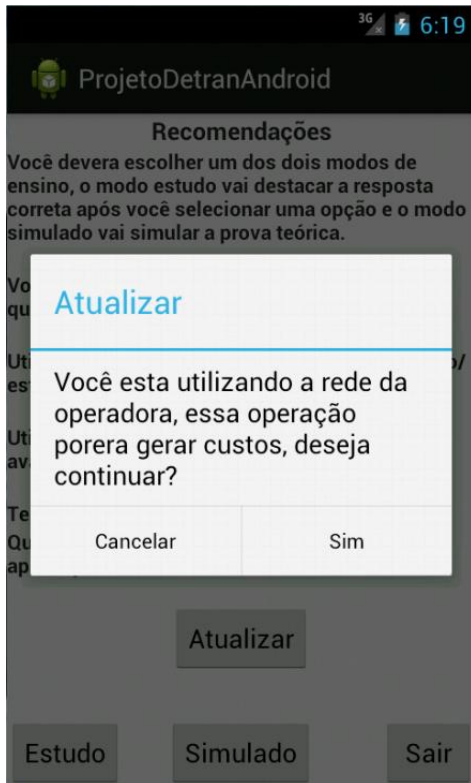


Figura 26 - Aviso caso conectada via operadora

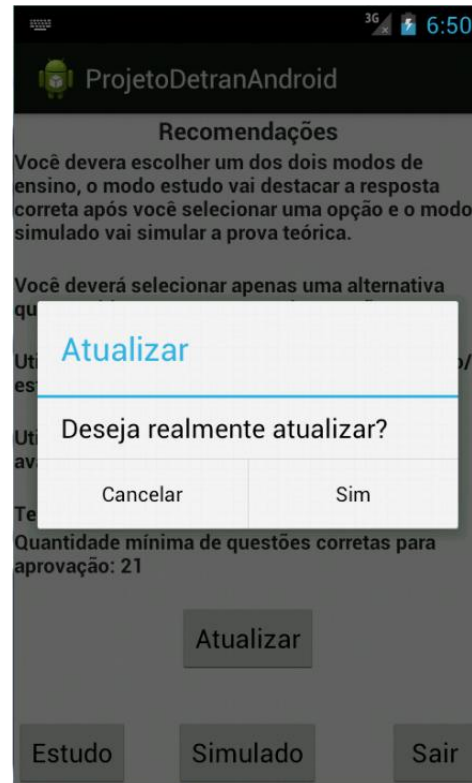


Figura 27 - Aviso de atualização

4.4 IMPLEMENTAÇÃO DO SISTEMA

Os dois módulos do *software*, Estudo e Simulado, possuem um banco de dados compartilhado com as perguntas e suas respectivas respostas, porém a recuperação das perguntas se difere. A recuperação das perguntas no módulo Simulado é feita através de um *select* completo do banco de dados e armazenada em uma variável, após isso será gerado 30 números aleatórios e armazenado no vetor de números inteiros "lista". Os números são validados de forma a não permitir que o mesmo número seja usada na atual escolha. Como pode ser verificado na Listagem 1.

```
dao = new DetranDAO(this);
perguntas = dao.listarPerguntas();
totalPerguntas = perguntas.getCount();

Random random = new Random();
Int numero;

vector<Integer> lista = new Vector<Integer>();
```

```

int i = 0;
while(i < 30){
    numero = random.nextInt(totalPerguntas);
    if(!lista.contains(numero)){
        lista.add(numero);
        i++;
    }
}

```

Listagem 1 - Geração das perguntas aleatórias

Na Listagem 2 pode-se ver as funções de clique dos botões “Anterior” e “Próxima”, primeiramente é reiniciado as cores dos componentes RadioButtons para a cor branca com a função `resetaPerguntas()`. Se a pergunta não for a última ele vai mover o cursor “perguntas” para a posição da próxima pergunta do vetor “lista” e apresentar a pergunta seguinte na tela com suas respectivas alternativas.

Porém como as alternativas poderão conter imagens (e essas imagens estão salvas como Base64 no banco de dados) o próprio aplicativo verifica se o tamanho da String é maior que 2000 caracteres, caso a alternativa seja uma imagem, então ele faz o processo de transformação, de Base64 para Bitmap, logo após ele coloca a imagem no seu Radio Button.

Caso o usuário já tenha marcado alguma pergunta o aplicativo busca no vetor “respostaMarcada” e exibe automaticamente na tela a alternativa que o usuário selecionou.

```

protected void btnProximaOnClick() {
    if (posicao < (limite - 1)) {
        resetaPerguntas();
        posicao++;

        perguntas.moveToPosition(posicao);
        tvPerguntas.setText(posicao + 1 + " " + perguntas.getString(1));

        if (perguntas.getString(2).length() >= 2000) {

            Bitmap img = getBitmapFromString(perguntas.getString(2));
            Drawable d = new BitmapDrawable(getResources(),img);
            rbOpcaoA.setCompoundDrawablesWithIntrinsicBounds(d, null, null,
null);

        } else {
            rbOpcaoA.setText(perguntas.getString(2));
        }

        if (perguntas.getString(3).length() >= 2000) {

```

```

        Bitmap img = getBitmapFromString(perguntas.getString(3));
        Drawable d = new BitmapDrawable(getResources(),img);
        rbOpcaoB.setCompoundDrawablesWithIntrinsicBounds(d, null, null,
null);

        } else {
            rbOpcaoB.setText(perguntas.getString(3));
        }

        if (perguntas.getString(4).length() >= 2000) {
            Bitmap img = getBitmapFromString(perguntas.getString(4));
            Drawable d = new BitmapDrawable(getResources(),img);
            rbOpcaoC.setCompoundDrawablesWithIntrinsicBounds(d, null, null,
null);
        } else {
            rbOpcaoC.setText(perguntas.getString(4));
        }

        if (perguntas.getString(5).length() >= 2000) {
            Bitmap img = getBitmapFromString(perguntas.getString(5));
            Drawable d = new BitmapDrawable(getResources(),img);
            rbOpcaoD.setCompoundDrawablesWithIntrinsicBounds(d, null, null,
null);
        } else {
            rbOpcaoD.setText(perguntas.getString(5));
        }

        if (respostaMarcada.get(posicao) == "a") {
            rbOpcaoA.setChecked(true);
        } else if (respostaMarcada.get(posicao) == "b") {
            rbOpcaoB.setChecked(true);
        } else if (respostaMarcada.get(posicao) == "c") {
            rbOpcaoC.setChecked(true);
        } else if (respostaMarcada.get(posicao) == "d") {
            rbOpcaoD.setChecked(true);
        } else {
            rgPerguntas.clearCheck();
        }

    }
}

```

Listagem 2 - Código dos botões Anterior e Próxima

Na Listagem 3, é feita a transformação de Base64 para Bitmap, para poder ser utilizado nas alternativas, ele faz uso da classe Base64, feita por HARDER, R. (2014) na versão 2.3.7.

```

private Bitmap getBitmapFromString(String string){
    try {
        byte[] decodedString = Base64.decode(string);
        Bitmap decodedByte = BitmapFactory.decodeByteArray(decodedString, 0,
decodedString.length);
        return decodedByte;
    } catch (IOException e) {

```

```

        System.out.print("ERRO");
        e.printStackTrace();
        return null;
    }
}

```

Listagem 3 - Função para transformar Base64 em Bitmap

A Listagem 4 mostra o XML da tela de respostas do módulo “Estudo”, que é um layout fixo, e a cada vez que o usuário navega nos botões “Anterior” e “Próximo” os valores das perguntas e alternativas mudam em tempo de execução, fazendo com que o aplicativo não precise criar 30 telas, mas sim reutilizar sempre a mesma, evitando assim o uso desnecessário de memória.

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/tvPergunta"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Questão 1: " />

    <RadioGroup
        android:id="@+id/rgPerguntas"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <RadioButton
            android:id="@+id/rbOpcaoA"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:onClick="onRadioButtonClicked"
            android:text="Alternativa A" />

        <RadioButton
            android:id="@+id/rbOpcaoB"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:onClick="onRadioButtonClicked"
            android:text="Alternativa B" />

        <RadioButton
            android:id="@+id/rbOpcaoC"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:onClick="onRadioButtonClicked"
            android:text="Alternativa C" />

        <RadioButton

```

```

android:id="@+id/rbOpcaoD"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:onClick="onRadioButtonClicked"
android:text="Alternativa D" />
</RadioGroup>

<RelativeLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="horizontal">

<Button
android:id="@+id/btnFinalizar"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:onClick="btnFinalizarOnClick"
android:layout_alignParentBottom="true"
android:layout_alignParentRight="true"
android:text="Finalizar" />

<Button
android:id="@+id/btnProxima"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_centerHorizontal="true"
android:onClick="btnProximaOnClick"
android:text="Proxima" />

<Button
android:id="@+id/btnAnterior"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_alignParentLeft="true"
android:onClick="btnAnteriorOnClick"
android:text="Anterior" />
</RelativeLayout>
</LinearLayout>

```

Listagem 4 - Layout da página de perguntas

Na Listagem 5 é feita a marcação da pergunta correta (fundo verde) pertencente apenas ao módulo Estudo. Quando o usuário seleciona uma alternativa automaticamente será exibida a resposta certa e também será salva no vetor “respostaMarcada” para que o usuário possa navegar entre as perguntas e alterar as alternativas marcadas de acordo com sua necessidade e posteriormente para verificar qual foi a pontuação final do mesmo.

```

Public void onRadioButtonClicked(View view) {
    boolean checked = ((RadioButton) view).isChecked();

    String marcaCorreta = perguntas.getString(6);

```

```

        if (marcaCorreta.equalsIgnoreCase("a"))
            rbOpcaoA.setBackgroundColor(Color.GREEN);
        elseif (marcaCorreta.equalsIgnoreCase("b"))
            rbOpcaoB.setBackgroundColor(Color.GREEN);
        elseif (marcaCorreta.equalsIgnoreCase("c"))
            rbOpcaoC.setBackgroundColor(Color.GREEN);
        else
            rbOpcaoD.setBackgroundColor(Color.GREEN);

        switch (view.getId()) {
        case R.id.rbOpcaoA:
            if (checked)
                respostaMarcada.add(posicao, "a");
            break;
        case R.id.rbOpcaoB:
            if (checked)
                respostaMarcada.add(posicao, "b");
            break;
        case R.id.rbOpcaoC:
            if (checked)
                respostaMarcada.add(posicao, "c");
            break;
        case R.id.rbOpcaoD:
            if (checked)
                respostaMarcada.add(posicao, "d");
            break;
        }
    }
}

```

Listagem 5 - Marcação da resposta correta no módulo Estudo

A Listagem 6 mostra a lógica para gerar a tela de Resultados do módulo Simulado. Primeiramente é calculado quantas perguntas o usuário acertou, comparando o vetor “respostaMarcada” com o cursor “perguntas”, após a contagem das perguntas corretas o aplicativo vai mandar uma *string* com a mensagem de sucesso ou fracasso, e o tempo restante da prova por parâmetro para a tela Resultado, por fim a *activity* “ResultadoSimuladoActivity” é chamada.

```

btnFinalizar.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        acertos = 0;
        for (posicao = 0; posicao<lista.size(); posicao++) {
            perguntas.moveToPosition(lista.elementAt(posicao));

            String retorno = perguntas.getString(6);
            String retorno2 = respostaMarcada.elementAt(posicao);
            perguntas.moveToNext();

            if (retorno.equalsIgnoreCase(retorno2) ) {
                acertos++;
            }
        }
    }
}

```

```

Intent intent = newIntent(v.getContext(),
ResultadoSimuladoActivity.class);
Bundle params = newBundle();

if (acertos >= 21) {
    String resposta = "Parabens, Você passou no teste com " +
acertos+ " acertos";
    params.putString("mensagem", resposta);

} else {
    String resposta = "Desculpa, Você reprovou no teste com "
+ acertos+ " acertos";
    params.putString("mensagem", resposta);

}

double tempo = Double.parseDouble(tempoFinal);
params.putDouble("tempo", tempo);
intent.putExtras(params);
finish();
startActivity(intent);
}
});

```

Listagem 6 - Lógica do botão Finalizar

CountDownTimer é uma classe da API nível 15, e está disponível desde a API level 1, implementa a funcionalidade de cronometro regressivo. A classe foi utilizada para fazer o temporizador do módulo Simulado, ela possui dois métodos, o onFinish e o onTick, o onTick exibe no TextView "tvTimer" da tela Simulado o tempo restante de prova a cada 1 segundo, e quando o tempo acabar o método onFinish é chamado automaticamente, o qual irá iniciar a sequência de passos para finalizar a prova, como pode ser visto na Listagem 7.

```

new CountdownTimer(3000000, 1000) {
    Public void onTick(long millisUntilFinished) {
        tvTimer.setText("Tempo Restante: "
+ ((millisUntilFinished / 1000) / 60) + ":"
+ (millisUntilFinished / 1000) % 60);
        tempoFinal = ((millisUntilFinished / 1000) / 60) + "." +
((millisUntilFinished / 1000) % 60);
    }

    Public void onFinish() {
        btnFinalizarOnClick();
    }
}.start();

```

Listagem 7 - Código do cronômetro

A tela de seleção de perguntas (Listagem 8) do módulo Estudo é feita baseada no numero de questões disponíveis, mesmo as categorias sendo estáticas, o numero de questões é recuperada com uma função contaPerguntas(), passando

por parâmetro a categoria desejada, assim as SeekBars vão de 0 até o número máximo de perguntas por categorias.

Quando o usuário selecionar as categorias e a quantidade de perguntas desejadas ele poderá iniciar o estudo, clicando no botão Iniciar, assim o *software* vai passar as categorias que ele deseja e a quantidade de perguntas por parâmetro para a tela de respostas.

```

barLegislacao.setMax(dao.contaPerguntas("legislacao"));
barMeioAmbiente.setMax(dao.contaPerguntas("ambiente"));
barPrimeirosSocorros.setMax(dao.contaPerguntas("socorros"));

barLegislacao.setOnSeekBarChangeListener(this);
barMeioAmbiente.setOnSeekBarChangeListener(this);
barPrimeirosSocorros.setOnSeekBarChangeListener(this);

btIniciar.setOnClickListener(new View.OnClickListener() {

@Override
public void onClick(View v) {
    Intent intent = new Intent(v.getContext(), EstudoActivity.class);
    Bundle params = new Bundle();

    params.putInt("qtdLegislacao", 0);
    params.putInt("qtdMeioAmbiente", 0);
    params.putInt("qtdPrimeirosSocorros", 0);

    if (cbLegislacao.isChecked() == false && cbMeioAmbiente.isChecked() == false
    && cbPrimeirosSocorros.isChecked() == false) {
        params.putString("categoria", "nenhuma");
    } else if (cbLegislacao.isChecked() == true && cbMeioAmbiente.isChecked() ==
    true && cbPrimeirosSocorros.isChecked() == true) {
        params.putString("categoria", "todas");
        params.putInt("qtdLegislacao", barLegislacao.getProgress());
        params.putInt("qtdMeioAmbiente", barMeioAmbiente.getProgress());
        params.putInt("qtdPrimeirosSocorros",
barPrimeirosSocorros.getProgress());
    } else if (cbLegislacao.isChecked() == true && cbMeioAmbiente.isChecked() ==
    true) {
        params.putString("categoria", "LM");
        params.putInt("qtdLegislacao", barLegislacao.getProgress());
        params.putInt("qtdMeioAmbiente", barMeioAmbiente.getProgress());
    } else if (cbLegislacao.isChecked() == true &&
cbPrimeirosSocorros.isChecked() == true) {
        params.putString("categoria", "LP");
        params.putInt("qtdLegislacao", barLegislacao.getProgress());
        params.putInt("qtdPrimeirosSocorros",
barPrimeirosSocorros.getProgress());
    } else if (cbMeioAmbiente.isChecked() == true &&
cbPrimeirosSocorros.isChecked() == true) {
        params.putString("categoria", "MP");
        params.putInt("qtdMeioAmbiente",
barMeioAmbiente.getProgress());
        params.putInt("qtdPrimeirosSocorros",
barPrimeirosSocorros.getProgress());
    }
}
}

```

```

    } else if (cbMeioAmbiente.isChecked() == true) {
        params.putString("categoria", "M");
        params.putInt("qtdMeioAmbiente",
            barMeioAmbiente.getProgress());
    } else if (cbLegislacao.isChecked() == true) {
        params.putString("categoria", "L");
        params.putInt("qtdLegislacao", barLegislacao.getProgress());
    } else if (cbPrimeirosSocorros.isChecked() == true) {
        params.putString("categoria", "P");
        params.putInt("qtdPrimeirosSocorros",
            barPrimeirosSocorros.getProgress());
    } else {
        params.putString("categoria", "nenhuma");
    }

    intent.putExtras(params);
    finish();
    startActivity(intent);
}

});

```

Listagem 8 - Seleção de perguntas

A Listagem 9 mostra o modo como o aplicativo conecta no Web Service, através da classe `WebServiceDAO`, após isso ele deleta e recria a tabela `Perguntas` para zerar os dados, então percorre o `JSONObject` com todas as perguntas recuperadas do servidor e insere eles linha a linha na tabela com a função `inserirPerguntas`.

```

try {
    JSONArray jperguntas = new WebServiceDAO().execute(pergunta).get();

    dao.dropTablePerguntas();
    dao.createTablePerguntas();

    try {
        for (int i = 0; i < jperguntas.length(); i++) {
            JSONObject jpergunta = jperguntas.getJSONObject(i);

            dao.inserirPerguntas(jpergunta.getString(PERGUNTA),
                jpergunta.getString(RESPOSTA_A),
                jpergunta.getString(RESPOSTA_B),
                jpergunta.getString(RESPOSTA_C),
                jpergunta.getString(RESPOSTA_D),
                jpergunta.getString(RESPOSTA_CORRETA),
                jpergunta.getString(CATEGORIA));
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
} catch (Exception ex) {
    ex.printStackTrace();
    System.out.print((ex.getMessage()));
}

```

```
}
```

Listagem 9 – Recuperação das perguntas do Web Service

A Listagem 10 mostra o código responsável por verificar qual método de conexão com a Internet o usuário está utilizando. Esta função será executada toda vez que o botão Atualizar for clicado, o qual irá exibir um Alert Dialog pedindo se o usuário realmente quer atualizar o banco de dados se estiver na Wi-fi e se estiver usando a rede da operadora e informando ao usuário que a atualização via rede da operadora pode gerar gastos.

```
public static String getNetworkClass(Context context) {
    ConnectivityManager cm = (ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo info = cm.getActiveNetworkInfo();
    if (info == null || !info.isConnected())
        return "-"; // not connected
    if (info.getType() == ConnectivityManager.TYPE_WIFI)
        return "WIFI";
    if (info.getType() == ConnectivityManager.TYPE_MOBILE) {
        return "MOBILE";
    }
    return "?";
}
```

Listagem 10 - Verificação da conexão utilizada

5 CONCLUSÃO

A informatização do método tradicional de estudo para a prova do DETRAN e o desenvolvimento de um aplicativo para dispositivos móveis com essa finalidade tem grande potencial para ajudar na transição do aprendizado em folha de papel, para o informatizado. E isso é uma grande contribuição para o *Mobile Learning*, já que o mesmo é um assunto ainda novo, porém muito importante.

Utilizando-se do grande crescimento do número de *smartphones* para facilitar o ensino sobre o trânsito como forma de preparar melhor ainda os futuros condutores de veículos, contribuindo assim para uma melhor educação no trânsito.

No presente trabalho foi apresentado um estudo sobre o desenvolvimento de um aplicativo para dispositivos móveis para o teste teórico da CNH, o qual foi baseado na ferramenta disponibilizada no próprio site do DETRAN, que oferece a forma de estudo para computadores conectados a Internet.

O *software* desenvolvido possui muitos benefícios a quem está pensando em conseguir sua CNH, tanto para os possíveis candidatos a serem habilitados quanto para os candidatos a reciclagem de carteira, já que ele oferece um modo de estudo e um de simulação da prova para dispositivos móveis com plataforma Android, a qual é a mais utilizada atualmente pelo público, e também por adicionar novas perguntas diretamente de um Web Service, sem a necessidade do usuário precisar atualizar o aplicativo toda a vez, com suporte de imagens, fazendo com que o aplicativo seja o mais real possível.

5.1 TRABALHOS FUTUROS

Como proposta de trabalhos futuros pretende-se desenvolver uma interface mais amigável, para que o usuário fique mais a vontade utilizando o aplicativo, e uma estatística dos testes, para que o usuário possa fazer um acompanhamento da sua evolução.

Outra proposta de trabalho futuro é aprimorar o Web Service, para que um futuro administrador possa transformar as imagens em Base 64 diretamente do navegador, sem precisar utilizar sites de terceiros.

E também pretende-se realizar um estudo sobre o progresso obtido nos alunos de auto escolas, onde será disponibilizado uma versão do software para os alunos e depois verificar o percentual de aumento de no aprendizado e aprovação dos alunos no teste teórico do DETRAN.

REFERÊNCIAS

- ANDROID. **Ferramentas de desenvolvimento Android**. 2014. Disponível em: <http://developer.android.com/tools/index.html>. Acesso em: 13 ago. 2014.
- BRITO, Robison C.; CARNIEL, Andrei; COPETTI, Mauricio. Sistema para Teste de Conhecimento para Prova Teórica do DETRAN: Um estudo de aceitação do aplicativo pelos usuários. **3º Congresso Brasileiro de Informática na Educação**. Dourados. Nov. 2014.
- CONTRAN. **Resolução número 168 do CONTRAN**. 2004. Disponível em: http://www.denatran.gov.br/download/Resolucoes/RESOLUCAO_CONTRAN_168.pdf. Acesso em: 16 jul. 2014.
- DATASUS. **Estatísticas nacionais de acidentes de trânsito**. 2014. Disponível em: http://www.vias-seguras.com/os_acidentes/estatisticas/estatisticas_nacionais. Acesso em: 05 dez. 2014.
- DB-ENGINES. **Ranking dos banco de dados mais populares**. 2014. Disponível em: <http://db-engines.com/en/ranking>. Acesso em: 05 dez. 2014.
- DENATRAN. **Frota nacional de veículos**. 2014. Disponível em: <http://www.denatran.gov.br/frota.htm>. Acesso em: 19 set. 2014.
- DETRAN-PR. **Teste teórico online do DETRAN-PR**. 2014. Disponível em: <http://www.simulado.detran.pr.gov.br/detrانprova/simularProva.do?action=iniciarProcesso>. Acesso em: 26 set. 2014.
- DETRAN. **Departamento Estadual de Trânsito**. 2014. Disponível em: <http://www.detran.pr.gov.br/>. Acesso em: 12 set. 2014.
- DEVELOPERS. **Desenvolvimento Android**. 2014. Disponível em: <http://developer.android.com/tools/index.html>. Acesso em: 18 ago. 2014.
- GARTNER. **Vendas mundiais de dispositivos móveis**. 2014. Disponível em: <http://www.gartner.com/newsroom/id/2791017>. Acesso em: 05 ago. 2014.
- GOOGLE PLAY. **DETRAN Simulados**. 2014. Disponível em: <https://play.google.com/store/apps/details?id=com.app.detransimulados>. Acesso em 10 dez. 2014.
- HARDER, Robert. **Classe Base 64**. 2014. Disponível em: <http://iharder.sourceforge.net/current/java/base64/>. Acesso em 19 jan. 2015.
- IBGE. **Frota nacional de veículos**. 2012. Disponível em: <http://cidades.ibge.gov.br/painel/frota.php?codmun=0>. Acesso em: 17 jul. 2014.
- IDC. **Estudo sobre o mercado Brasileiro de celulares**. 2014. Disponível em: <http://br.idclatin.com/releases/news.aspx?id=1613>. Acesso em: 14 jul. 2014.

INSTITUTO CLARO. **Uso de celulares na educação**. 2011. Disponível em: <https://www.institutoclaro.org.br/em-pauta/uso-de-celulares-na-educacao-aproxima-professores-do-universo-dos-alunos/>. Acesso em: 10 set. 2014.

IPEA. **Cerca de 45% dos brasileiros usam ônibus para se locomover**. 2011. Disponível em: http://www.ipea.gov.br/portal/index.php?option=com_content&view=article&id=6943&catid=159&Itemid=75. Acesso em: 11 jul. 2014.

JDK. **Download Java**. 2014. Disponível em: https://www.java.com/pt_BR/download/. Acesso em: 26 set. 2014.

MARÇAL, Edgar; ANDRADE, Rossana; RIOS, Riverson. **Aprendizagem utilizando dispositivos móveis com sistemas de realidade virtual. Novas Tecnologias na Educação**, Porto Alegre, V.3, N. 1, Maio 2005. Disponível em: http://lumenagencia.com.br/dcr/arquivos/a51_realidadevirtual_revisado.pdf. Acesso em: 11 set. 2014.

MARTINS, Lucas R. **Simulado DETRAN RJ**. 2014. Disponível em: <https://play.google.com/store/apps/details?id=info.lucasrmartins.simulado>. Acesso em 10 dez. 2014.

MYSQL. **Cientes do MySQL**. Disponível em: <http://www.mysql.com/customers/>. Acesso em: 05 dez. 2014.

MYSQL. **MySQL Open Source**. Disponível em: <http://dev.mysql.com/downloads/mysql/>. Acesso em: 05 Jan. 2015.

OGLIARI, Ricardo da Silva; BRITO, Robison Cris. **Android – Do Básico ao Avançado**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2014.

OLHARDIGITAL. Disponível em: <http://olhardigital.uol.com.br/pro/noticia/40338/40338>. Acesso em: 12 set. 2014.

PRESSMAN, R. S. **Engenharia de software**. 5 ed. São Paulo: McGraw Hill, 2002.

PRODESP. **Simulado DETRAN SP**. 2014. Disponível em: <https://play.google.com/store/apps/details?id=br.gov.sp.detran.simulado>. Acesso em 10 dez. 2014.

REIS, Thiago. **Aumento da frota no país**. 2014. Disponível em: <http://g1.globo.com/brasil/noticia/2014/03/com-aumento-da-frota-pais-tem-1-automovel-para-cada-4-habitantes.html>. Acesso em: 21 ago. 2014.

Ruby on Rails. **Rails, desenvolvimento web sem dor**. 2014. Disponível em: <http://www.rubyonrails.com.br/>. Acesso em: 04 nov. 2014.

Ruby on Rails. **Rails 3.2. has been released!**. 2012. Disponível em: <http://weblog.rubyonrails.org/2012/1/4/rails-3-2-0-rc2-has-been-released/>. Acesso em: 04 nov. 2014.

SQLITE. **Características do SQLite.** 2014. Disponível em: <http://www.sqlite.org/features.html>. Acesso em: 07 ago. 2014.

TELECO. **Vendas de Telefone Celular e Smartphone no Brasil.** Disponível em: <http://www.teleco.com.br/celprod.asp>. Acesso em: 12 set. 2014.

TOZETTO, Claudia. Disponível em: <http://veja.abril.com.br/noticia/vida-digital/windows-phone-supera-ios-e-assume-segunda-posicao-no-brasil>. Acesso em: 12 set. 2014.

WHITE, Oliver. **Popularidade das IDE's de desenvolvimento Java.** 2012. Disponível em: <http://zeroturnaround.com/rebellabs/developer-productivity-report-2012-java-tools-tech-devs-and-data/4/> Acesso em: 18 ago. 2014.