



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ
CAMPUS LUIZ MENEGHEL

LUCAS VIVEIROS DA SILVA

**Desenvolvimento de um sistema de caronas
integrado com *Facebook* para plataforma
*Android***

Bandeirantes

2013

LUCAS VIVEIROS DA SILVA

**Desenvolvimento de um sistema de caronas
integrado com *Facebook* para plataforma
*Android***

Trabalho de conclusão de curso
apresentado à Universidade
Estadual do Norte do Paraná –
campus Luiz Meneghel – como
requisito parcial para obtenção do
grau em Bacharel em Sistemas de
Informação.

Orientador: Prof. Me. Carlos
Eduardo Ribeiro

Bandeirantes

2013

LUCAS VIVEIROS DA SILVA

**Desenvolvimento de um sistema de caronas
integrado com *Facebook* para plataforma
*Android***

Trabalho de conclusão de curso
apresentado à Universidade
Estadual do Norte do Paraná –
campus Luiz Meneghel – como
requisito parcial para obtenção do
grau em Bacharel em Sistemas de
Informação.

COMISSÃO EXAMINADORA

Prof. Me. Carlos Eduardo Ribeiro
UENP – *Campus* Luiz Meneghel

Prof. Me Ricardo Gonçalves Coelho
UENP – *Campus* Luiz Meneghel

Prof. Estevan Braz Brandt Costa
UENP – *Campus* Luiz Meneghel

Bandeirantes, 06 de Dezembro de 2013

Dedico este trabalho primeiramente a Deus e minha família, especialmente meus pais, que sempre me deram força, coragem e constante apoio para seguir em busca de meus objetivos.

AGRADECIMENTOS

Primeiramente agradeço a Deus, o centro e o fundamento de tudo em minha vida, por renovar a cada momento a minha força e disposição e pelo discernimento concedido ao longo dessa jornada.

Agradeço à minha mãe Lucia Regina de Azevedo da Silva e ao meu pai Sebastião Viveiros da Silva e familiares, que fazem parte da conclusão desse projeto, o tempo todo contribuíram com paciência, carinho, compreensão e muito bom humor, em momento algum desacreditaram do meu potencial.

Agradeço ao meu orientador Biluka uma pessoa simplesmente fantástica que teve atenção e paciência nas horas difíceis e pelas orientações que contribuíram para minha formação e realização deste trabalho. Cujos ensinamentos foram, são e serão de grande valia para o meu futuro acadêmico e profissional.

Agradeço a todos os professores da instituição e aos professores Ricardo Gonçalves Coelho e Estevan Braz Brandt Costa pela ajuda com críticas construtivas que vieram acrescentar neste trabalho.

A todos os meus colegas e aqueles que não foram citados, mas fizeram parte da minha vida nesses quatro anos e merecem um agradecimento.

“Eu aprendi... que todos querem viver no topo da montanha, mas toda felicidade e crescimento ocorre quando você esta escalando-a”
(William Shakespeare)

RESUMO

O mercado para venda e demanda por aplicativo de dispositivos móveis está em constante crescimento, possibilitando que o desenvolvimento de aplicações se torne uma grande vantagem. O trabalho proposto explora algumas funcionalidades do sistema operacional *Android* assim como a integração com a plataforma *Facebook* que viabiliza o compartilhamento de informações entre os usuários e seus amigos. A ideia principal é criar um serviço de divulgação de caronas, que os usuários possam adicionar o destino da viagem e pesquisar por caronas oferecidas, na qual as rotas possam ser compartilhadas utilizando o serviço de integração com a rede social.

Palavras-chaves: Android, Facebook, Integração, API Graph.

ABSTRACT

The market for supply and demand by application of mobile devices is constantly growing, allowing the development of applications becomes a big advantage. The proposed work explores some features of the Android operating system as well as integration with the Facebook platform that enables the sharing of information between users and their friends. The main idea is to create a service for the dissemination of rides, so users can add the destination of the Virgin and search for rides offered, in which routes can be shared using the service integration with the social network.

Keywords: Android, Facebook, Integration, Graph API.

LISTA DE FIGURAS

Figura 1 - Organização da Estrutura Android.....	19
Figura 2 - Ciclo de vida Android	20
Figura 3 - Diagrama de casos de uso	24
Figura 4 - Caso de uso: <i>Login Facebook</i>	25
Figura 5 - Caso de uso: Oferecer Carona	26
Figura 6 - Caso de uso: Caronas disponíveis.....	27
Figura 7 - Diagrama de Classe.....	28
Figura 8 - Diagrama de sequência: Login Facebook.....	30
Figura 9 - Diagrama de sequência: Cadastro da Carona	31
Figura 10 - Diagrama de sequência: Postar Carona no Facebook.....	32
Figura 11 - Diagrama de sequência: Lista Carona (Remover)	33
Figura 12 - AndroidManifest.xml.....	35
Figura 13 – Código fonte da tela Splash Screen	36
Figura 14 - Tela Splash Screen.....	37
Figura 15 - Criar novo aplicativo.....	39
Figura 16 - App ID e App Secret	39
Figura 17 - Informações básicas do aplicativo	40
Figura 18 – Código fonte responsável pelo Acess Token	42
Figura 19 - Tela de login	43
Figura 20 - Tela de autenticação do Facebook	44
Figura 21 - Tela de autorização do Facebook.....	44
Figura 22 - Código fonte dados do usuário	45
Figura 23 - Tela principal.....	46
Figura 24 - Tela de Cadastro de Carona.....	47
Figura 25 - Tela de compartilhamento da carona.....	48
Figura 26 - Exemplo de publicação da carona no Facebook	49
Figura 27 - Tela Lista de Caronas	50
Figura 28 - Minhas Caronas.....	51

LISTA DE SIGLAS

API *Application Programming Interface*

OHA *Open Handset Alliance*

SDK *Software Development Kit*

Sumário

1.	INTRODUÇÃO.....	12
1.1	Justificativa.....	12
1.2	Objetivos.....	13
1.2.1	Objetivo Geral.....	13
1.2.2	Objetivos Específicos:.....	13
2.	FUNDAMENTAÇÃO TEÓRICA.....	15
2.1	<i>Android</i>	15
2.1.1	Arquitetura.....	17
2.1.2	Ciclo de vida.....	19
2.2	Facebook.....	21
2.2.1	Android Software Development Kit.....	22
3.	DOCUMENTAÇÃO.....	24
3.1	Casos de Uso.....	24
3.2	Diagramas de classes.....	28
3.3	Diagramas de sequência.....	30
3.3.1	Login Facebook.....	30
3.3.2	Cadastro da Carona.....	31
3.3.3	Postar Carona no Facebook.....	32
3.3.4	Listar Carona.....	33
4.	DESENVOLVIMENTO.....	34
4.1	Técnicas e ferramentas utilizadas.....	34
4.2	Configuração do aplicativo.....	35
4.3	Tela Splash Screen.....	36
4.4	Integração Facebook SDK com aplicativo.....	38
4.4.1	Criar novo aplicativo.....	38
4.4.2	Login Facebook.....	41
4.5	Funcionalidades do aplicativo.....	46
4.5.1	Cadastrar Caronas e publicação no Facebook.....	47
4.5.2	Caronas oferecidas.....	50
5.	CONCLUSÃO.....	52
	REFERÊNCIAS.....	54

1. INTRODUÇÃO

O crescimento dos dispositivos móveis está caminhando para se tornar a principal tecnologia utilizada no dia a dia (SARDENBERG, 2012). Com o seu desenvolvimento, as pessoas deixam de usar seus computadores e notebooks devido a sua praticidade e expansão no mercado.

Segundo Gusmão (2013), destacando-se no mercado hoje o sistema operacional *Android* é plataforma mais utilizada para dispositivos móveis. A cada dia são ativados em todo mundo mais de um milhão de dispositivos.

Dessa maneira, esta tecnologia vem progredindo de acordo com a demanda, com isso cada vez mais são criadas novos recursos e funções aperfeiçoadas, tornando-se verdadeiros computadores pessoais portáteis, com o seu nível de processamento, capacidade de armazenamento, sua brevidade de desempenho de memória, conectividade em ambientes públicos e privados.

A grande acessibilidade que o *Android* proporciona, facilita a comunicação e integração destes e expandindo a gama de recursos a serem utilizadas.

Explorando estas tecnologias o presente trabalho abordará o desenvolvimento do aplicativo móvel, com o intuito de que os seus usuários possam adicionar seu destino e pesquisar por caronas oferecidas, na qual as rotas poderão ser compartilhadas utilizando o serviço de integração com a rede social *Facebook*.

1.1 Justificativa

Atualmente o uso compartilhado de automóveis vem crescendo a cada dia, isso por que as pessoas estão tentando reduzir os gastos gerados pelos mesmos, conseqüentemente este compartilhamento reduz o número de automóveis em circulação, diminuindo a emissão de gases poluentes e gerando menos congestionamento.

“O aumento de caronas poderá promover relações interpessoais entre os participantes, bem como contribuir para a diminuição do número de veículos automotores nas ruas e avenidas dos grandes centros urbanos. Sua aderência poderá trazer inúmeros benefícios à saúde, como: diminuir o stress do dia-a-dia, melhorar a qualidade de vida da população – sabendo que menos carros é sinônimo de menos horas perdidas no trânsito. Tudo isso em conjunto trará grande contribuição para o meio ambiente, tornando o ar que respiramos mais puro”. (OLIVEIRA, 2013)

Nos Estados Unidos, a carona solidária é incentivada para resolver problemas como o congestionamento de carros nas ruas, reduzindo custos tanto para viagens curtas ou de longa distância. (MACEDO, 2013)

Dessa maneira, algumas empresas no Brasil incentivam o uso da carona. Sendo que pessoas que trabalham no mesmo ambiente, estão lado a lado no trânsito, vão para o mesmo destino, mas será que sabem disso? Logo, é possível ajudá-las a utilizar um menor número de carros e gozarem dos benefícios já mencionados.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um aplicativo de publicação e pesquisas de caronas para a plataforma *Android*.

1.2.2 Objetivos Específicos:

Os objetivos específicos do trabalho são:

- Desenvolver um aplicativo *mobile* usando a plataforma de desenvolvimento *Android*;
- Permitir que os usuários postem seus destinos e que possam consultar caronas oferecidas;

- Permitir o compartilhamento das rotas na rede social *Facebook*;
- Auxiliar os usuários na consulta de disponibilidade de carona.

2. FUNDAMENTAÇÃO TEÓRICA

Acessar a Internet e outros recursos computacionais por meio de dispositivos móveis facilita o acesso às informações em qualquer lugar, com alcance amplo a qualquer hora. (ALCANTARA; VIEIRA, 2013)

Os recursos e serviços móveis começam a surgir para promover a liberdade no uso e consumo de informação por parte dos usuários finais que adquirem os smartphones, os quais possuem facilidade de uso e recursos cada vez mais modernos. (ALCANTARA; VIEIRA, 2013)

Atualmente o mercado de dispositivos móveis, conta com alguns sistemas operacionais entre eles pode-se citar: Windows Mobile, Android, iPhone OS. Neste capítulo são apresentadas as principais tecnologias que foram utilizadas no decorrer deste trabalho.

2.1 *Android*

Em outubro de 2003, começou a história do Sistema Operacional *Android*, quando os criadores Andy Robin, Rich Miner e Chis White fundaram a *Android Inc.* em Palo Alto, na Califórnia, para a criação de sistemas operacionais em dispositivos móveis. Em 2005 a empresa *Google* comprou a *Android*.

Atualmente, a plataforma *Google Android* é mantida pela Open Handset Alliance (OHA), formado por um grupo de mais de 40 empresas que se uniram para inovar e acelerar o desenvolvimento de aplicações e serviços para dispositivos móveis, trazendo aos usuários uma experiência mais rica em termos de recursos e menos custosa financeiramente para o mercado (LIMA, 2011).

Segundo esse grupo, o *Android* foi construído com uma meta específica: ser a primeira, completa e livre plataforma criada especialmente para dispositivos móveis. (GOOGLE, 2013).

O *Android* consiste em uma plataforma de desenvolvimento para aplicativos móveis. Baseada em um sistema operacional *Linux*, é a primeira plataforma para aplicações móveis completamente livre e de código aberto (*open source*) (LECHETA, 2010). A plataforma consiste em uma pilha de

software para dispositivos móveis, incluindo um sistema operacional, *middleware* e aplicativos. Para desenvolvimento de aplicações para esta plataforma, usa-se o *Software Development Kit do Android* (GOOGLE, 2008).

Sua estrutura para funcionamento do sistema operacional consiste basicamente em duas camadas. A primeira camada é constituída por um *Kernel Linux*, que provê a abstração do hardware do dispositivo móvel, como gerenciamento dos processos do sistema, de memória e manutenção do sistema de arquivos. Na segunda camada há um ambiente de execução de aplicações chamado "*Dalvik Virtual Machine*", uma máquina virtual de grande importância que executa as aplicações do sistema, dando possibilidade para que se possa testar os aplicativos a serem desenvolvidos.

Para que se tenha melhor desempenho do aplicativo, antes de desenvolver, algumas decisões como escolher a versão do sistema operacional é o elemento crucial para começar o aplicativo. Afinal, com o surgimento de cada versão, vêm suas atualizações e correção de *bugs*, para tanto precisa ser analisado qual versão que os usuários que mais utiliza. (CAMPOS, 2011).

A primeira versão foi lançada em Abril de 2009 com *Application Programming Interface* (API) (nível 4), sob o nome de Cupcake, na versão 1.5. Durante a escrita deste trabalho, o Android conta com a versão Jelly Bean, versão melhorada em velocidade e simplicidade, lançada Junho de 2012.

A ênfase do trabalho está na integração do aplicativo com *Facebook*, o mesmo oferece uma gama de funções e serviços para o desenvolvedor. Porém, com algumas limitações na qual ser obrigado a usar uma API superior do nível 8, versão 2.2. Na criação do aplicativo, a arquitetura do *Android* garante flexibilidade ao desenvolvedor, dinamismo e a possibilidade de explorar a fundo as capacidades do equipamento, pois permite a utilização sem restrições de qualquer funcionalidade do dispositivo móvel.

2.1.1 Arquitetura

Segundo Dobler (2010), a arquitetura do Android (ANROID-ARCHITECTURE) é dividida em 5 camadas, as quais são: *Application*, *Application Framework*, *Libraries*, *Android Runtime* e o *Linux Kernel*. Cada camada é descrita a seguir em mais detalhes:

- *Application*: o Android possui um conjunto de aplicativos básicos, como um cliente de e-mail, calendário, mapas, navegador, contatos e outros. Todos os programas são escritos utilizando a linguagem de programação Java. Uma aplicação pode ser construída utilizando-se quatro blocos principais:
 - *Activity* (Atividade): todo o processo do sistema operacional *Android* é voltado para as atividades, que seriam as telas dos aplicativos. Na qual são muito mais além do que simplesmente telas, possuindo um ciclo de vida representada na Figura 2, possibilitando ao usuário usar várias aplicações rodando ao mesmo tempo;
 - *Intent Receiver* (Receptor de Intenção): Trata a reação de eventos externos, que seria um mecanismo de alerta de uma nova mensagem;
 - *Service* (Serviço): Código sem interfaces de usuários, que rodam em background, possuindo ciclo de vida próprio;
 - *Content Provider* (Provedor de Conteúdo): Para compartilhar dados entre aplicativos ou podendo armazenar e recuperar dados em um repositório.
- *Application Framework*: por fornecer uma plataforma de desenvolvimento aberta, o Android oferece aos desenvolvedores a capacidade de criar aplicações extremamente ricas e inovadoras. De acordo com essa camada apresentada na Figura 1 encontra-se no desenvolvimento do trabalho a *Activity Manager* responsável por interagir com as atividades gerais do

funcionamento do sistema e classe *Notification Manager* serve para notificar usuários por algum evento acontecido;

- *Libraries*: possui um conjunto de bibliotecas C/C++ utilizadas por diversos componentes do sistema operacional. Estas capacidades são expostas aos desenvolvedores por meio da estrutura de aplicativos. Nela encontra-se *SQLite* responsável por fazer comunicação do Banco de Dados local do aplicativo;
- *Android Runtime*: possui um conjunto de bibliotecas que fornece a maioria das funcionalidades disponíveis nas principais bibliotecas da linguagem de programação Java. Também nesta camada está a Dalvik Virtual Machine compilador dos aplicativos;
- *Linux Kernel*: O Android se baseia na versão 2.6 do Kernel do Linux para o sistema central de serviços, como segurança, gerenciamento de memória, gerenciamento. No decorrer do desenvolvimento foram utilizadas alguns desses drivers presentes nesta camada, como exemplo a WifiDriver, serviço que disponibiliza a comunicação do aplicativo com a Internet.

É apresentada na Figura 1 a organização da arquitetura do Android:



Figura 1 - Organização da Estrutura Android

Fonte: Dobler, 2010

O conjunto da arquitetura traz variadas opções de bibliotecas e funcionalidades para o desenvolvimento da aplicação, com isso o ciclo de vida é de extrema importância na hora de usar as funcionalidades oferecidas.

2.1.2 Ciclo de vida

Segundo Developer (2013), a *Activity* do sistema *Android* é responsável por iniciar um determinado código, que instancia invocando métodos de retornos que correspondem às fases específicas do seu ciclo de vida. Ao contrário de outros paradigmas de programação no qual os aplicativos são lançados com um método *main()*.

Durante a execução de uma atividade, o sistema chama um conjunto básico de métodos formando um ciclo de vida em uma sequência semelhante a uma pirâmide. Ou seja, cada fase do ciclo de vida da atividade é uma etapa separada na pirâmide. No momento que o sistema cria uma nova instância de atividade, cada método de retorno move o estado de atividade um passo em direção ao topo. O topo da pirâmide é o ponto em que a atividade está sendo executada em primeiro plano e o usuário pode interagir com ele.

À medida que o usuário começa a deixar a atividade, o sistema chama outros métodos que movem o estado de atividade de volta para baixo da pirâmide, a fim de destruí-la. Em alguns casos, a atividade passará para parte de baixo da pirâmide e esperar (por exemplo, quando o usuário inicia outro aplicativo), a atividade pode se mover de volta ao topo (se o usuário retorna à atividade) e volta onde o usuário parou.

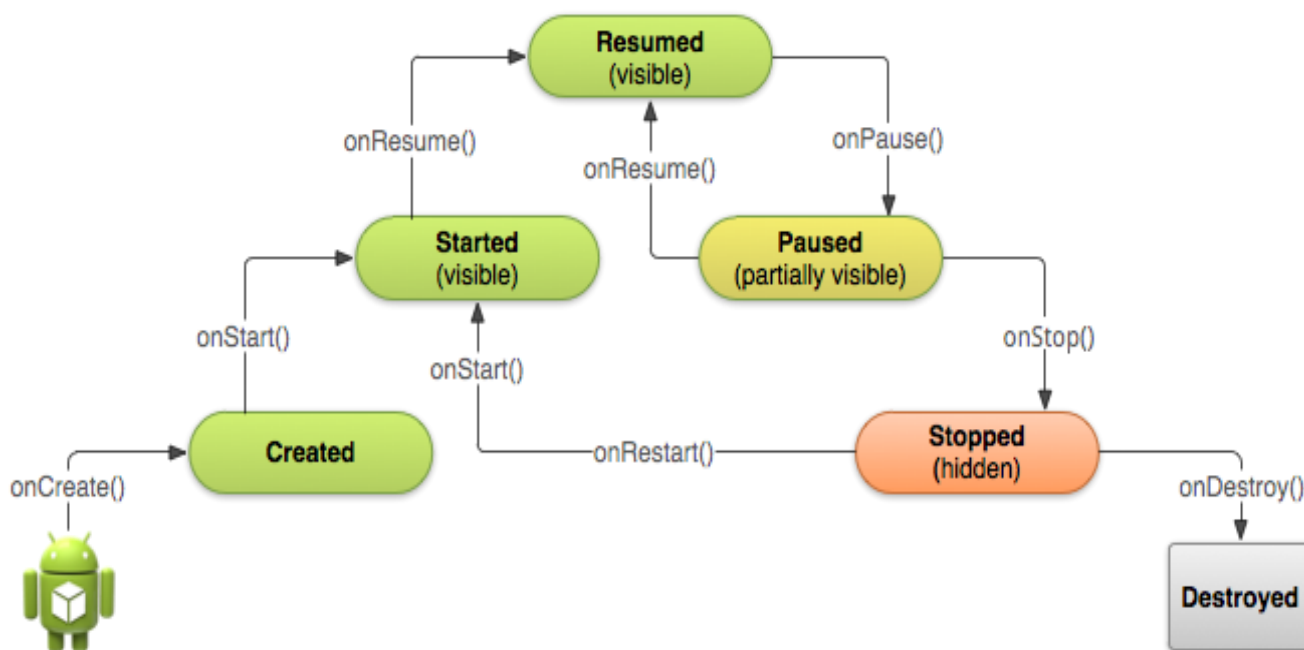


Figura 2 - Ciclo de vida Android
Fonte: Google Developers

Dependendo da complexidade da atividade, não é preciso implementar todos os métodos do ciclo de vida. No entanto, implementar os métodos de ciclo de vida da atividade corretamente garante que o aplicativo se comporta bem em várias formas, isso garante que:

- Não falhar ao recebimento de uma chamada de telefone ou mudar para outro aplicativo enquanto estiver usando o aplicativo;
- Não perder o progresso do usuário se ele deixar o seu aplicativo e retornar a ele em um momento posterior;
- Não falhar ou perder o progresso do usuário quando a tela gira entre a orientação horizontal e vertical.

Há várias situações em que uma atividade pode estar em transições entre diferentes estados que estão ilustrados na Figura 2. No entanto, apenas três destes estados podem ser estáticos:

- Resumed: estado em que atividade está em primeiro plano e o usuário pode interagir com ele;
- Paused: estado em que atividade fez uma pausa e não recebe entrada do usuário e não pode executar qualquer código;
- Stopped: estado em que atividade está completamente oculta e não visível.

Outros estados são transitórios que o sistema executa rapidamente, chamando o próximo método de retorno do ciclo de vida. Isso é tudo para o ciclo de vida de uma atividade básica de um aplicativo.

O Sistema Operacional *Android* possibilita aos seus desenvolvedores amplas formas de integrações dos aplicativos com as redes sociais. Conseguindo que a informação seja compartilhada das melhores e diversas maneiras.

2.2 Facebook

Em fevereiro de 2004 foi criado o Facebook por Mark Zuckerberg, Dustin Moskovitz, Chris Hughes e Eduardo Saverin, acadêmicos da Universidade de Harvard, na qual se tornou uma rede social que desde o início tem o objetivo de configurar um espaço no qual as pessoas possam encontrar umas às outras, postando opiniões e fotografias. (SANTANA, 2013)

Um dos principais fatores que contribui para o crescimento do *Facebook* foi à possibilidade no desenvolvimento de que aplicativos de terceiros possam ser integrados, possibilitando assim que esses aplicativos tenham um aspecto visual semelhante ao próprio *Facebook* e utilizem dados e funcionalidades disponíveis na rede social. (OLIVEIRA, 2010)

2.2.1 Android Software Development Kit

A plataforma do *Facebook* suporta a integração com outros aplicativos de diferentes plataformas, a seguir é apresentado um quadro dos principais plataformas suportadas.

<u>SDK</u>	<u>Descrições</u>	<u>Plataforma</u>
Android SDK	Integra os recursos da API Graph do Facebook com aplicativos móveis criados sobre o sistema operacional Android.	Aplicativos Móveis
iOS SDK (iPhone & iPad)	Este SDK fornece suporte da API Graph para aplicativos do iPad, iPhone e iPod Touch escritos na linguagem Objective-C. Permite realizar chamadas para API Graph do Facebook, como exibir caixas de diálogos da plataforma do Facebook.	Aplicativos Móveis
PHP SDK	Fornecer suporte aos aplicativos Web criado em linguagem PHP. Esta biblioteca ajuda adicionar botões de chamadas como "LoginFacebook" e dá suporte a API Graph.	Web Sites
JavaScript SDK	Permite que o acesso a todos os recursos da API Graph e caixas de diálogos através de JavaScript.	Web Sites

Segundo o *Facebook* (2013), o *Facebook SDK para Android* é a maneira de integrar seu aplicativo nativo para *Android* com a plataforma. O SDK torna mais fácil para adicionar a autenticações com a rede social, ler e escrever no *Facebook APIs*, compartilhando e suporte para elementos de interface nativos. Algumas das funções serão citadas abaixo:

- Autenticar: Implementar *Facebook login*, pedir ao usuário para as permissões às suas necessidades dos aplicativos, lidar com as alterações da sessão e sair do usuário;
- Personalizar: personalizar a experiência do usuário com a sua foto do perfil e nome quando logar;
- Amigos: Mostrar lista de amigos do usuário;
- Locais Próximos: Exibir uma lista de lugares próximos e permitir que o usuário marcar sua localização;
- Publicar uma Ação *Open Graph*: Publicar atividade de seu aplicativo para linha do tempo e feed de notícias.

Com as funcionalidades descritas, o presente trabalho utiliza o sistema de fazer o *login* com a conta do usuário, requisitando ao usuário apenas informações básicas. Possibilitando ao usuário o compartilhamento de conteúdo do aplicativo.

3. DOCUMENTAÇÃO

3.1 Casos de Uso

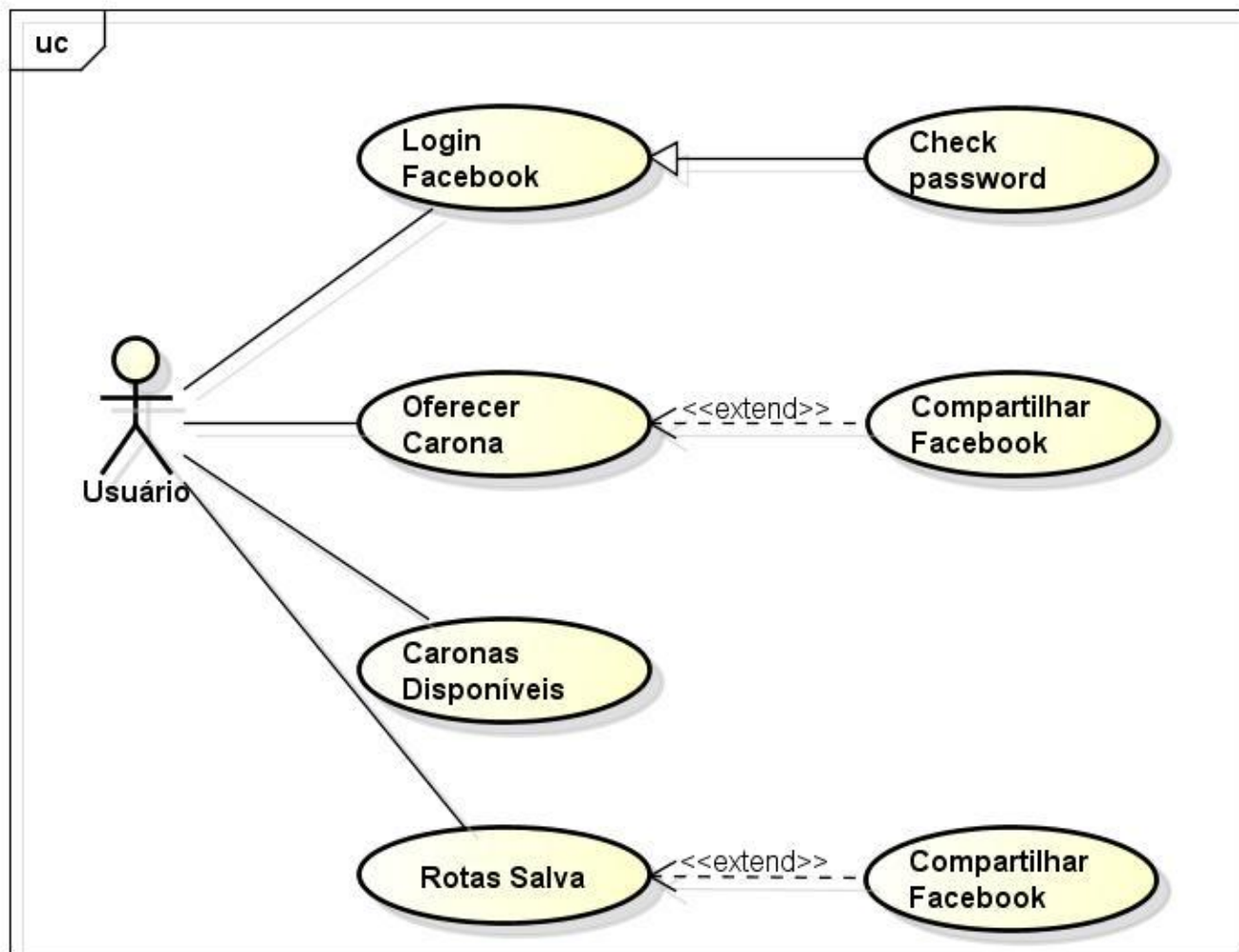


Figura 3 - Diagrama de casos de uso

Como observado na Figura 4, o modelo de caso de uso trata o Ator Usuário como o principal ator do sistema. Será descrito cada caso de uso de acordo com o ator do aplicativo.

Número: 01

Caso de Uso: Login

Descrição: Este caso de uso descreve o Login do aplicativo.

Ator: Usuário

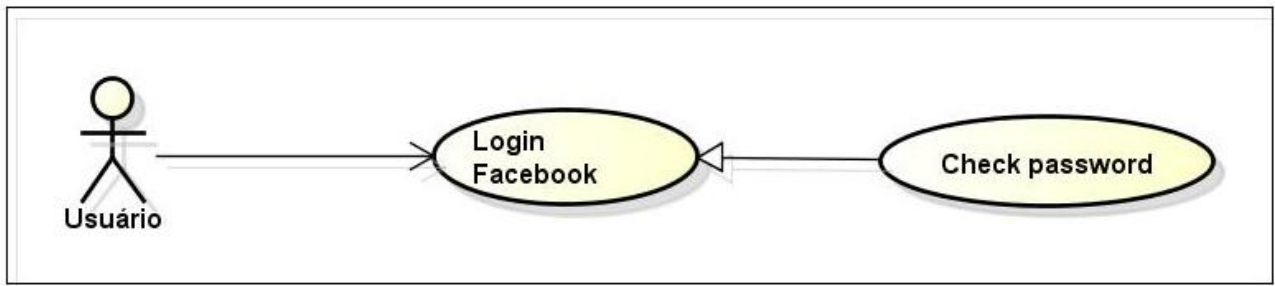


Figura 4 - Caso de uso: *Login Facebook*

Curso Normal:

1. Usuário informa os dados de sua conta do *Facebook*;
2. Sistema faz *Check Password*;
3. Usuário será logado no aplicativo.

Curso Alternativo do passo 2:

2. O sistema encontrou dados incorretos;
 - 2.1 O sistema informa dados incorretos e aguarda a entrada da conta;
 - 2.2 Encerra o caso de uso.

Número: 02

Caso de Uso: Oferecer Carona

Descrição: Este caso de uso descreve o cadastramento de uma carona oferecida.

Ator: Usuário

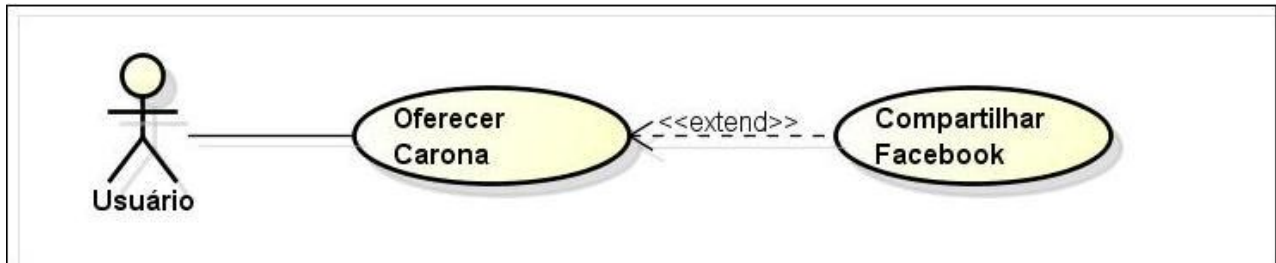


Figura 5 - Caso de uso: Oferecer Carona

Como pré-requisito, o usuário tem que estar logado com Facebook para ter acesso ao aplicativo.

Curso Normal:

1. Aparecerá para o usuário a tela para cadastrar uma carona;
2. Usuário deve informa os dados da carona, preenchendo os campos: origem, destino, horas, data e quantidade de vagas;
3. Sistema exibe opções para o usuário: Compartilhar *Facebook*, Cancelar e Salvar;
4. Após escolher uma das opções acima o sistema exibe a página de cadastro novamente com os campos vazios.

Curso Alternativo do passo 2:

2. O sistema encontra campos obrigatórios não preenchidos;
 - 2.1 O sistema informa a falha ao usuário e aguarda a entrada dos dados;
 - 2.2 Encerra o caso de uso.

Curso Alternativo do passo 3:

- 3.1 O usuário Compartilhar a sua rota no *Facebook*, e o sistema salva no aplicativo a carona compartilhada;
- 3.2 O usuário pode apenas salvar sua rota no aplicativo;

- 3.3 O usuário pode cancelar o cadastro da carona;
- 3.4 Encerra o caso de uso.

Número: 03

Caso de Uso: Caronas Disponíveis

Descrição: Este caso de uso descreve a lista das caronas oferecida.

Ator: Usuário



Figura 6 - Caso de uso: Caronas disponíveis

Curso Normal:

1. Usuário visualiza uma lista de caronas oferecidas;
2. Usuário pode buscar por uma cidade específica através da lista de caronas disponíveis;
3. Encerra caso de uso;

As descrições de casos de uso são de grande importância para sistemas e aplicativos, podendo ser aplicados para captar comportamento pretendido do sistema que está sendo desenvolvido, sem ser necessário especificar como esse comportamento é implementado. (BOOCH; RUMBAUGH; JACOBSON, 2005).

3.2 Diagramas de classes

Os diagramas de classes é a representação da estrutura para mostrar um conjunto de classes e seus relacionamentos, são importantes não só para visualização e documentação de modelos estruturais, mas também para a construção de sistemas executáveis por intermédio de engenharia de produção. (BOOCH; RUMBAUGH; JACOBSON, 2005).

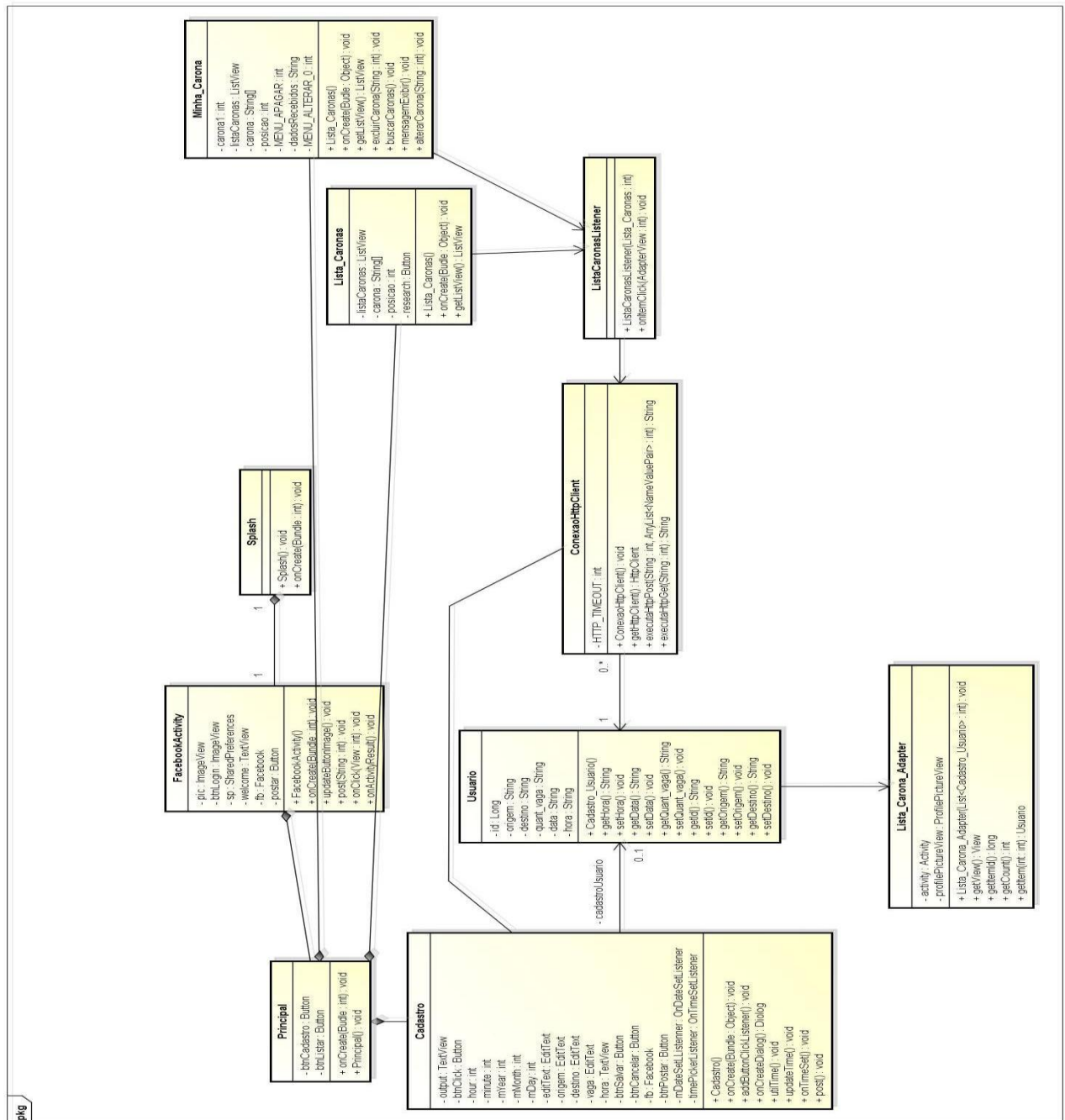


Figura 7 - Diagrama de Classe

O aplicativo *Android* fica embarcado em um dispositivo móvel, sendo estruturado em suas classes com as seguintes atribuições.

Splash: responsável por fazer apresentação do aplicativo, tela inicial que é exibida ao abrir o aplicativo.

FacebookActivity: responsável por integrar com o Facebook, solicitando para o usuário a tela de login. Ao ser o logado usuário aceita os termos de uso, disponibilizando ao aplicativo seus dados, como: id do usuário, foto e nome.

Principal: Tela menu para o usuário com opções de cadastro e lista de todas as caronas e a caronas do próprio usuário.

Cadastro: classe que instancia os objetos do .xml da tela de cadastro da carona e tem as funcionalidades de salvar, cancelar e publicar no Facebook uma carona.

Usuario: responsável por criar os objetos.

Lista_Carona_Adapter: classe que pega os valores digitados pelos usuários, tendo responsabilidade de controlar o conteúdo de uma lista.

Lista_Caronas: classe que lista todas as caronas.

Minha_Carona: classe que lista as caronas somente dos usuários, na qual o usuário pode excluir ou alterar suas caronas.

ListaCaronasListener: responsável por montar a lista.

ConexaoHttpClient: classe que faz comunicação com os arquivos *php* para comunicação com a base de dados.

3.3 Diagramas de sequência

A ênfase no diagrama de sequência é a ordenação temporal das mensagens. Os objetos colocados à sua esquerda são os que iniciam a interação e os da direita são os subordinados que vão crescendo conforme a linha do tempo. (BOOCH; RUMBAUGH; JACOBSON, 2005).

3.3.1 Login Facebook

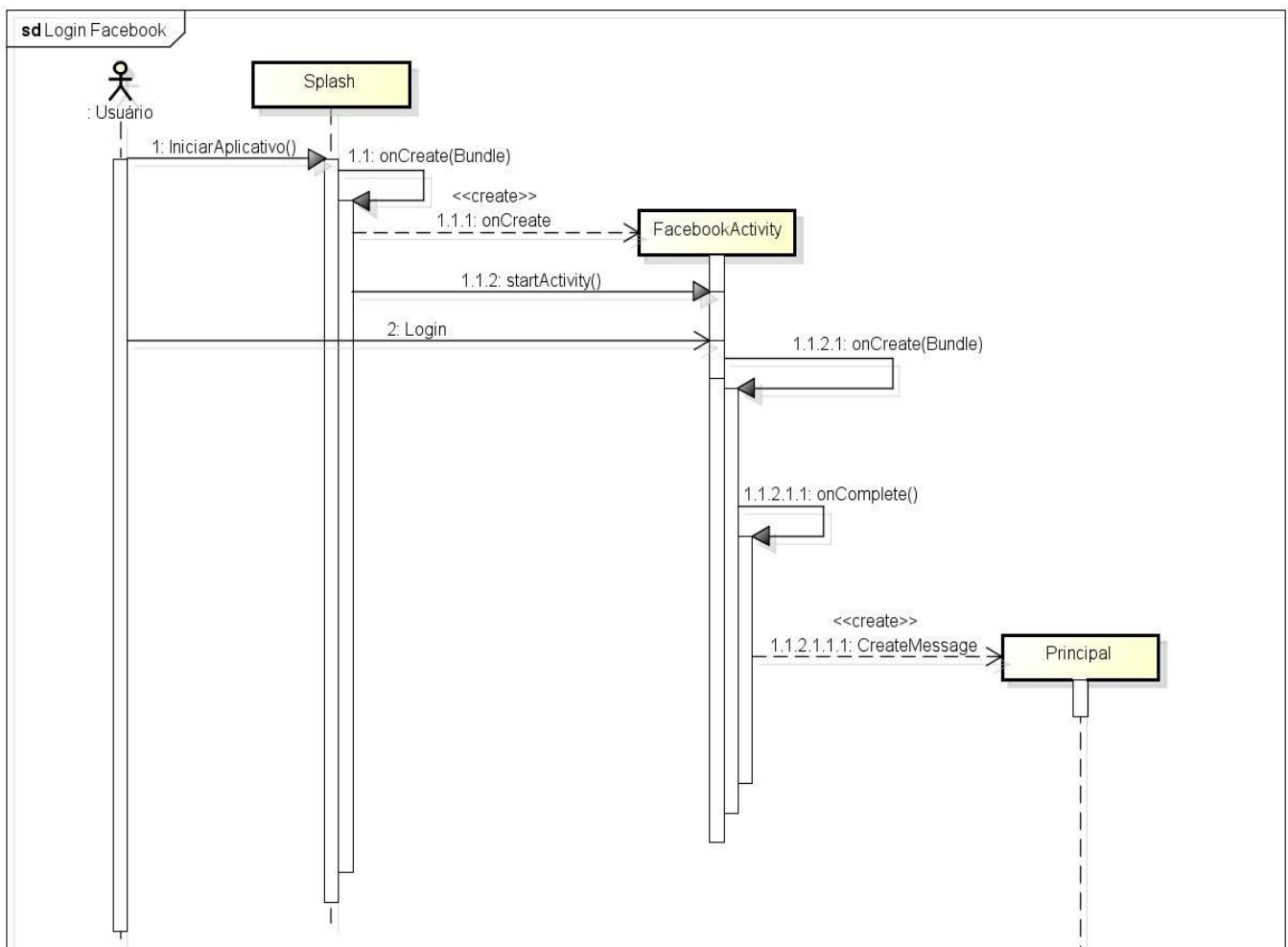


Figura 8 - Diagrama de sequência: Login Facebook

Na Figura 8 é apresentado o diagrama de sequencia do caso *Login Facebook*, na qual é iniciado o aplicativo com apresentação da tela *Splash*, logo após é iniciada a tela *FacebookActivity* pelo método(*startActivity()*). Ao carregar a tela é solicitado ao usuário o *Login Facebook*, que feita a verificação correta o usuário tem acesso a tela principal do sistema.

3.3.2 Cadastro da Carona

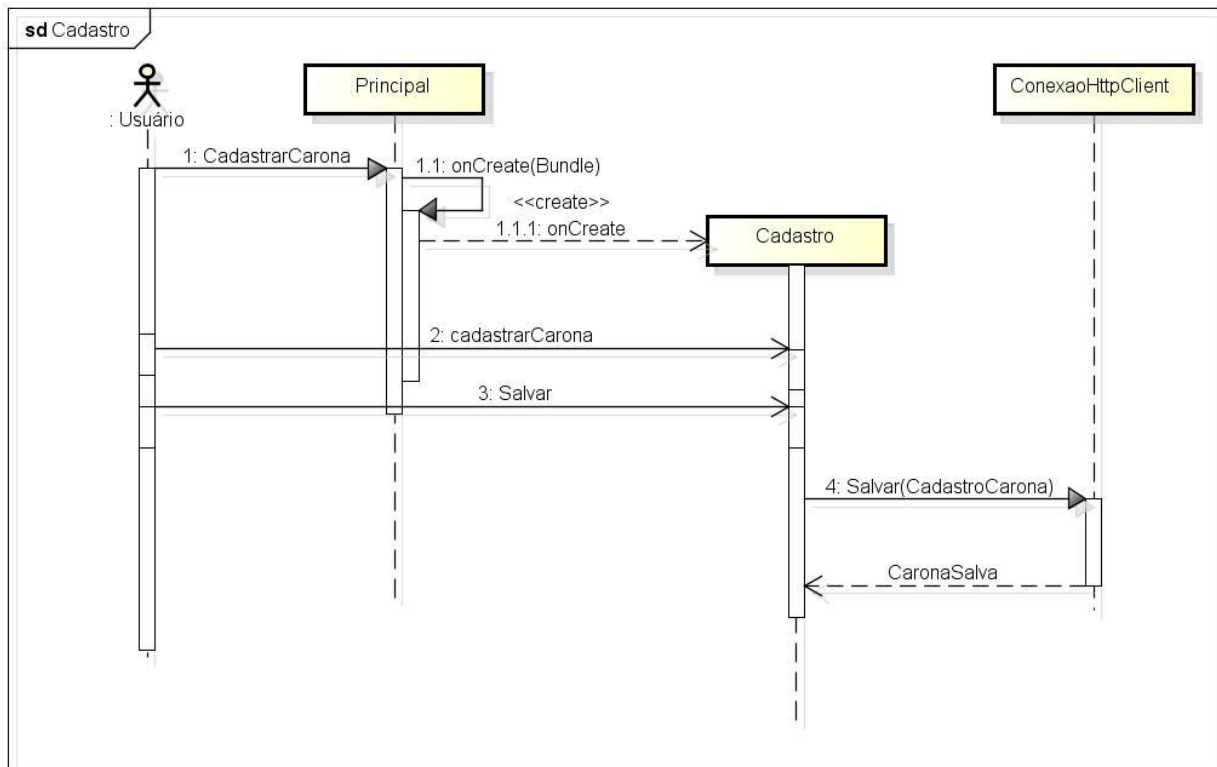


Figura 9 - Diagrama de sequência: Cadastro da Carona

Na Figura 9 é apresentado o cadastro da carona, na qual é iniciada a tela de cadastro (*onCreate*), que o usuário preenche os dados solicitado de uma carona e salva a mesma, na qual é feita solicitação *HTTP* com base de dados externo do aplicativo.

3.3.3 Postar Carona no Facebook

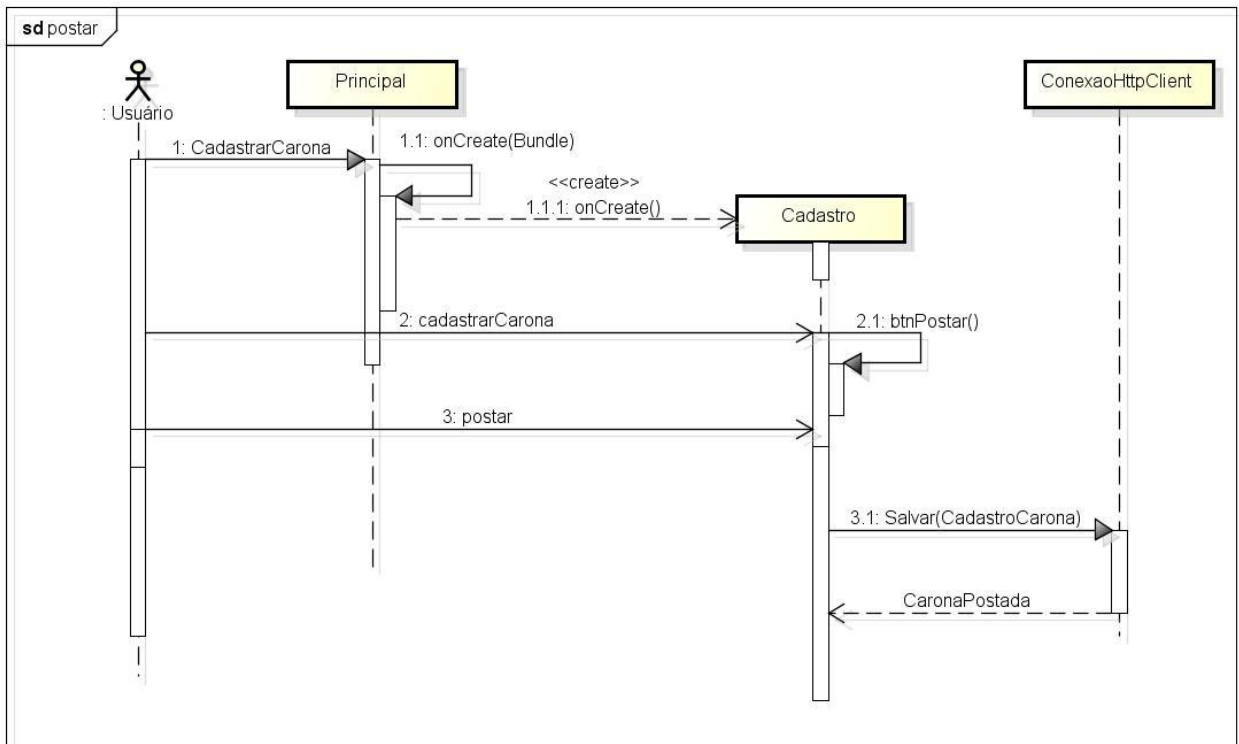


Figura 10 - Diagrama de sequência: Postar Carona no Facebook

Na Figura 10 é apresentada uma postagem da carona, na qual é iniciada a tela de cadastro (*onCreate*), que o usuário preenche os dados solicitado de uma carona e posta a mesma, uma pagina com as informação da carona e um campo para digitar alguma informação desejada é mostrada ao usuário, nesta mesma função é feita solicitação *HTTP* com base de dados externo do aplicativo para salvar os dados no aplicativo.

3.3.4 Listar Carona

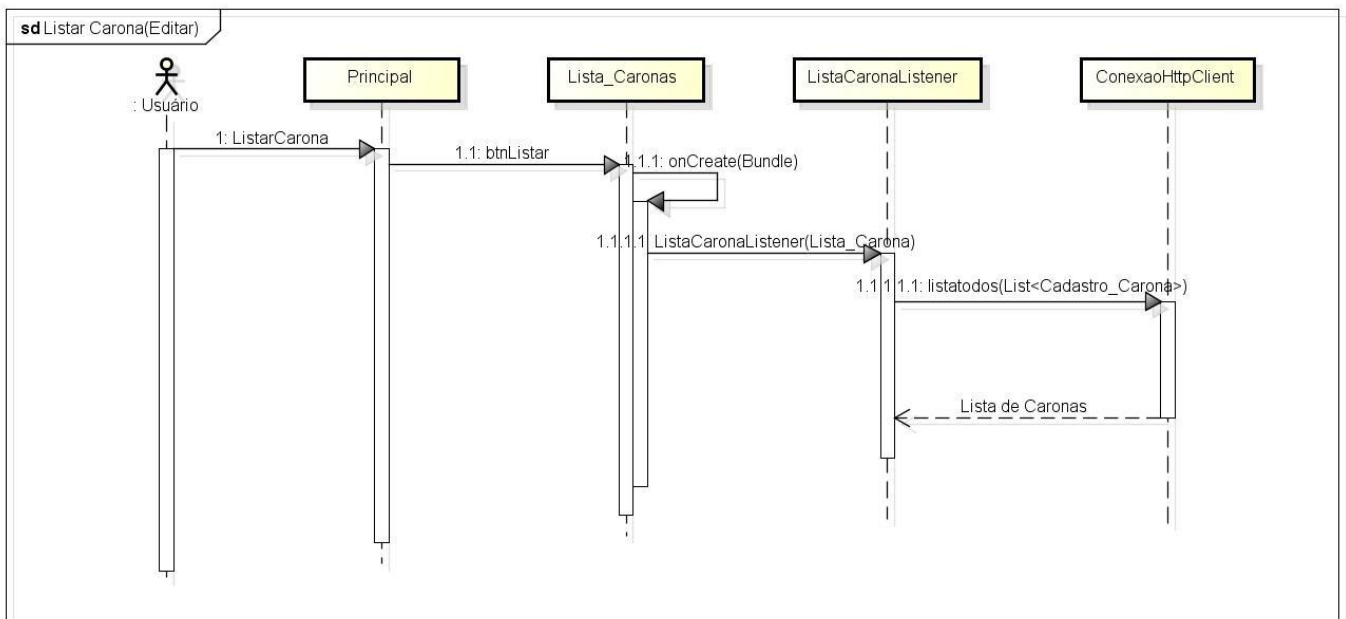


Figura 11 - Diagrama de sequência: Lista Carona (Remove)

Na Figura 11 é apresentado a lista de caronas cadastradas por todos os usuário, na qual é iniciada a tela *Listar_Caronas* (*onCreate*), passando os valores pela classe *ListaCaronaListener* responsável por formar a lista dos dados que é feita solicitação *HTTP* com base de dados externo do aplicativo.

4. DESENVOLVIMENTO

Neste capítulo é apresentado o software proposto, contendo as bibliotecas, técnicas e ferramentas utilizadas para o desenvolvimento da aplicação, junto com as descrições das telas e trechos de código para um melhor entendimento, as mesmas serão executadas em uma máquina virtual do *Android*.

4.1 Técnicas e ferramentas utilizadas

O objetivo final deste trabalho foi o desenvolvimento da aplicação Carona Legal. Este consiste em um sistema de divulgação de caronas, cadastrando suas rotas entre a sua origem até seu destino, podendo realizar pesquisas de caronas publicadas no aplicativo, integrando o serviço com a rede social Facebook.

A API do Facebook é bastante consistente e permite a interações com diversas plataformas e a criação de aplicativos diversos. No caso deste trabalho, foi utilizado o SDK para Android que a partir de sua conexão é disponibilizado alguns dados da pessoa.

A escolha do ambiente de desenvolvimento Eclipse se deu devido este ser o oficial da plataforma *Android* e a existência do *Android Development Tools* (ADT). Este plug-in facilita o desenvolvimento das aplicações, gerando o projeto com os componentes necessários de início, além de disponibilizar ferramentas de debug e log.

O pacote de desenvolvimento *Android* utilizado inclui um sistema operacional, um *middleware* e aplicativos de características essenciais, dentre os quais se encontram o emulador e um *debugger*, juntamente com toda a biblioteca de desenvolvimento para dispositivos móveis que utilizam o sistema operacional *Android*.

4.2 Configuração do aplicativo

Toda aplicação Android deve ter um arquivo AndroidManifest.xml em seu diretório raiz. Esse arquivo apresenta informações essenciais sobre a aplicação para o sistema operacional, que deve possuir informações do sistema antes que possa executar qualquer solicitação do código do aplicativo (MEDNIEKS,2009).

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3   package="android.app.carona"
4   android:versionCode="1"
5   android:versionName="1.0" >
6
7
8   <uses-sdk
9     android:minSdkVersion="8"
10    android:targetSdkVersion="18" />
11
12
13   <uses-permission android:name="android.permission.INTERNET" />
14   <uses-permission android:name="android.permission.VIBRATE" />
15   <uses-permission android:name="android.permission.GET_ACCOUNTS" />
16
17
18   <application
19     android:allowBackup="true"
20     android:icon="@drawable/ico"
21     android:label="@string/app_name"
22     android:theme="@style/AppTheme" >
23
24
25     <activity
26       android:name=".FacebookActivity"
27       android:label="@string/app_name" >
28       <intent-filter>
29         <action android:name="android.app.carona.FACEBOOKACTIVITY" />
30
31         <category android:name="android.intent.category.DEFAULT" />
32       </intent-filter>
33     </activity>
34
35

```

Figura 12 - AndroidManifest.xml

Na Figura 14 são apresentadas algumas informações, como:

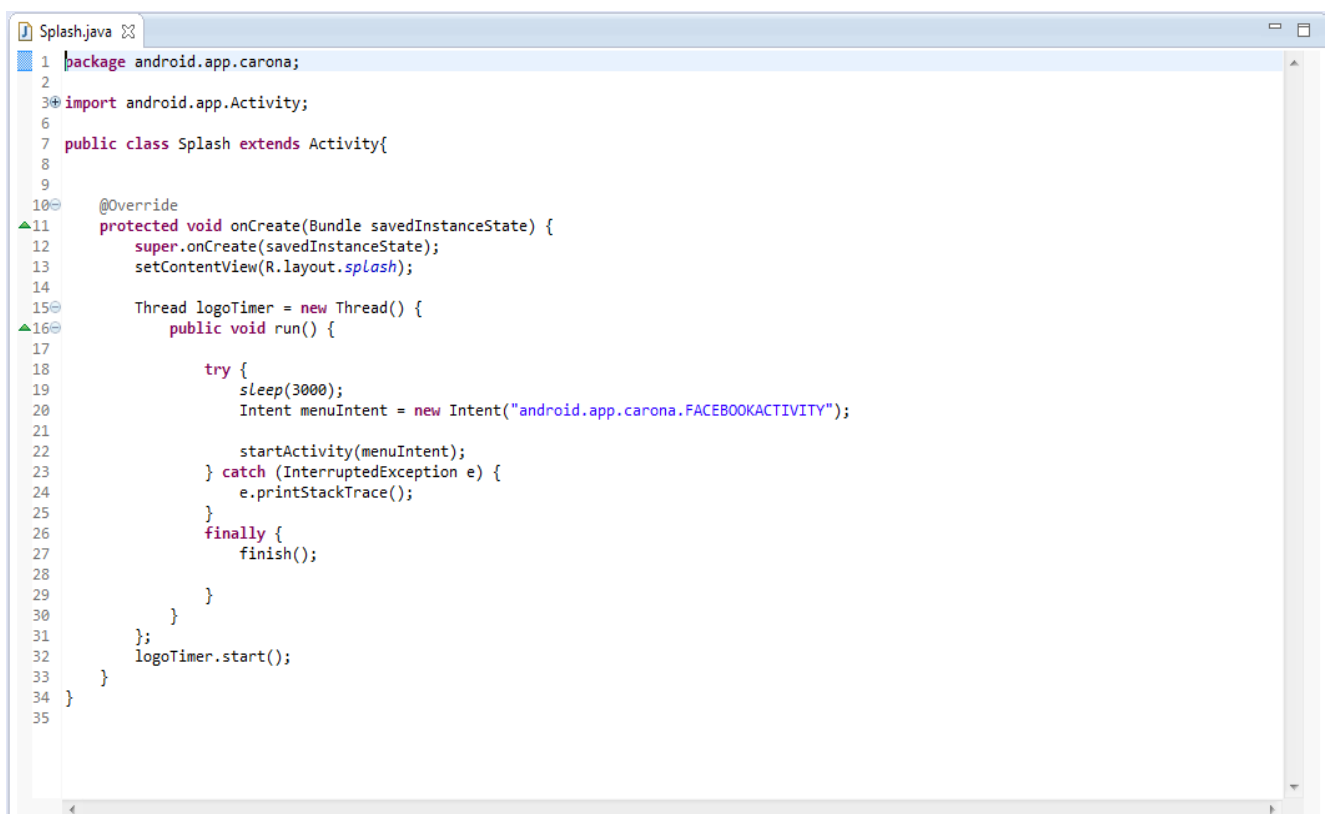
- Pacote Java para a aplicação. O nome do pacote “*android.app.carona*” serve como um identificador único para a aplicação;
- Nível mínimo e máximo da *API* do *Android* que o aplicativo requer;
- São declaradas permissões que o aplicativo precisa ao ser executado, que neste caso são “*INTERNET*”, “*VIBRATE*” e “*GET_ACCOUNTS*”;
- Descrição de todas as classes criadas.

Para cada alteração do aplicativo, como criação de uma nova classe, permissões, ou mudanças de nível de versão, é preciso que se define neste arquivo *AndroidManifest.xml*.

4.3 Tela Splash Screen

A tela de apresentação de um aplicativo *Android* é denominada como *Splash Screen*, está é a tela inicial carregada quando é aberta uma aplicação. Geralmente para exibir o logo do aplicativo ou de alguma determinada empresa.

Esta tela pode ter duas funcionalidades, fazer propagando do logo ou ser realizadas alguns processos antes de iniciar a aplicação. No caso deste trabalho tem a simples funcionalidade de exibir o logo da aplicação.



```
1 package android.app.carona;
2
3 import android.app.Activity;
4
5
6
7 public class Splash extends Activity{
8
9
10 @Override
11 protected void onCreate(Bundle savedInstanceState) {
12     super.onCreate(savedInstanceState);
13     setContentView(R.layout.splash);
14
15     Thread logoTimer = new Thread() {
16     public void run() {
17
18         try {
19             sleep(3000);
20             Intent menuIntent = new Intent("android.app.carona.FACEBOOKACTIVITY");
21
22             startActivity(menuIntent);
23         } catch (InterruptedException e) {
24             e.printStackTrace();
25         }
26         finally {
27             finish();
28         }
29     }
30     };
31     logoTimer.start();
32 }
33 }
34 }
35 }
```

Figura 13 – Código fonte da tela Splash Screen

Assim que se inicial o aplicativo a tela *splash* é executada, como apresentada na Figura 14, esta classe apenas é executada por apenas três segundos denominado pelo comando “*sleep(3000)*”, após a apresentação do

logo é iniciada a principal classe do aplicativo `“android.app.carona.FACEBOOKACTIVITY”`

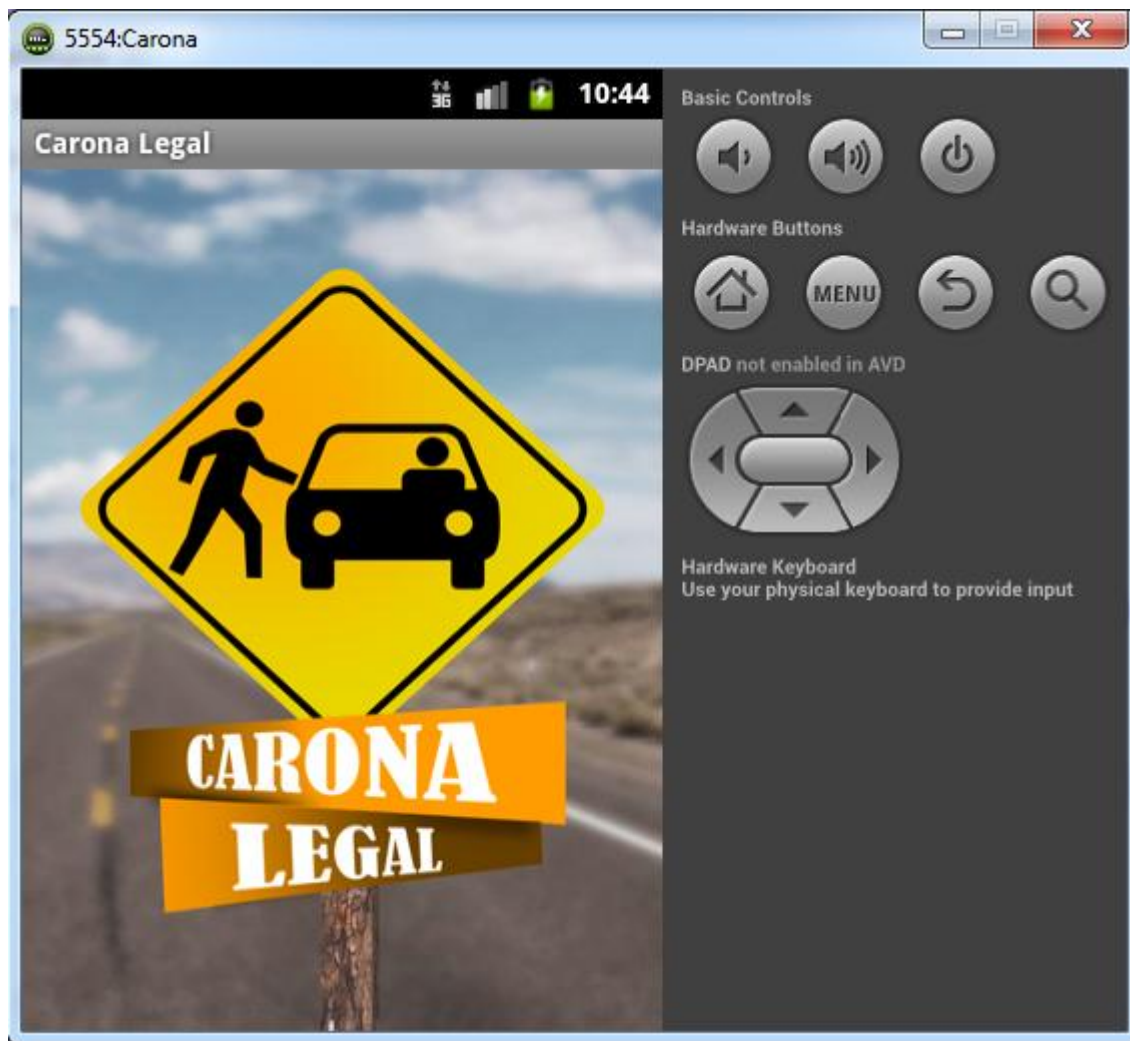


Figura 14 - Tela Splash Screen

Por não ter nenhum processo a ser executado, será preciso apenas três segundos para apresentação do logo do aplicativo, conforme na Figura 16. Ao término é iniciada a tela de Login do Facebook, que será descrita na próxima seção.

4.4 Integração Facebook SDK com aplicativo

Na página do Facebook Developers é disponibilizado o SDK, segmento que auxilia os desenvolvedores de aplicativos e páginas web. Constituído por uma série de ferramentas desenvolvidas para que programas externos tenham uma melhor integração com a Facebook API. Algumas restrições e exigências são impostas para que possa ser usado o Facebook SDK, como:

- Ter baixado o SDK do Android;
- Instalar ADT plugin para o eclipse;
- Usar uma versão superior do Android 2.2 (API 8).

A API principal da Facebook SDK é a API Graph que permite a leitura e gravação de dados a partir da rede do Facebook.

Graph considera como um objeto um grupo de dados sobre um usuário: seu perfil, lista de amigos, postagens, fotos, preferências, entre outros. Mantém também dados dos relacionamentos e conexões entre eles. Aos objetos, é atribuído um ID exclusivo, e eles são facilmente acessíveis.

Com a API Graph, é possível recuperar objetos, excluir e publicar, atualizar e filtrar resultados e até descobrir dinamicamente conexões/relacionamentos de um objeto.

Por padrão, os aplicativos têm acesso aos dados públicos do usuário, e para acessar dados privados, deve ser solicitada as permissões do usuário, chamadas de permissões estendidas. Os dados públicos serão descritos na seção de *Login* do *Facebook*.

4.4.1 Criar novo aplicativo

No site do Facebook Developers é disponibilizado para download SDK para Android, na qual deve ser importado no pacote que está sendo desenvolvido o aplicativo.

Após a importação deve ser criado um novo aplicativo, inserindo as informações básicas, conforme na Figura 15.

Figura 15 - Criar novo aplicativo

Uma vez criado o aplicativo, é gerada um *App ID* (*Identificador do aplicativo*) e *App Secret* (*Segredo do Aplicativo, código verificador de segurança*) responsável por identificar a aplicação, na qual é utilizada para adicionar no projeto, conforme visto na Figura 16.

Figura 16 - App ID e App Secret

Após ter gerado a identificação do aplicativo, é preciso assinar essa aplicação com uma chave gerada no arquivo de debug.

Cada aplicativo Android precisa ser criada uma assinatura digital, por questões de verificação de segurança para autenticidade do Facebook. Para que possa assinar um aplicativo, é preciso de uma chave hash encontrada em seu computador.

A chave é gratuita e simples de ser solicitada. Ao executar uma aplicação Android através Eclipse é gerado um arquivo debug.keystrok é utilizado para extrair com a ferramenta KeyTool disponível no Java SDK.

Nas configurações básicas é preciso ainda fornecer alguns dados, como nome do pacote, classe principal e chave hash. Essas informações devem estar exatamente iguais no desenvolvimento do aplicativo.

The screenshot shows the Facebook Developers console for the application 'Carona Legal'. The left sidebar contains navigation options like 'Configurações', 'Básicas', 'Privilégios de desenvolvedores', etc. The main content area is titled 'Aplicativos > Carona Legal > Básicas'. It displays the app's logo, name, App ID (391300480980207), and App Secret (4c26876cdc6185570dfe77f6e5fa5e1d). A warning indicates the app is in Sandbox Mode. Below this, the 'Informações básicas' section contains input fields for Display Name (Carona Legal), Namespace (facebookcarona), Contact Email (lvs.viveiros@gmail.com), and App Domains. The Sandbox Mode is currently set to 'Ativada'. A modal window titled 'Aplicativo nativo do Android' is open, showing configuration for the native app with Package Name (android.app.carona), Class Name (FacebookActivity), and Key Hashes (yF1nG9fjMdx298j8MhMgQ3mL9Y4=).

Figura 17 - Informações básicas do aplicativo

Nesta página mostrada na Figura 19 o desenvolvedor conta com uma série de configurações adicionais, como:

- Privilégios de desenvolvedores: Espaço que pode ser definido as pessoas que estão trabalhando no projeto, podendo ter os cargos definidos como administradores, desenvolvedores, testadores ou usuários testes.
- Permissões: Conjunto de configurações de permissões para o aplicativo quando for publicado na Central de Aplicativos;
- Pagamentos: Serviço de acompanhamento de pagamentos quando for publicado na Central de Aplicativos;
- Atualização em tempo real: Acompanhamento em tempo real do aplicativo em sincronia com o Facebook;
- Avançadas: Configurações avançadas que permite algumas configurações de como restrições de países e de idade;

Finalmente, após ter configurado a aplicação na página do *Facebook Developers*, pode se então usar as funcionalidades disponibilizadas para começar a desenvolver o aplicativo desejado.

4.4.2 Login Facebook

Segundo Facebook, é descrito na documentação para desenvolvimento ao se criar aplicações na qual que se integram a plataforma do Facebook é necessário que essas aplicações solicitem permissões para que possam obter informações dos usuários do Facebook. Algumas dessas solicitações que podem ser enviados a partir de aplicativos (FACEBOOK, 2013).

- Solicitações de usuários: Os pedidos são enviados por solicitações para usuário através de uma caixa de diálogo utilizando plataforma Facebook;
- Solicitações de aplicações: São solicitadas permissões para os usuários. Sendo possível apenas utilizando a API Graph, no qual permite enviar informações em nome de um usuário para a plataforma do Facebook.

Dessa forma a autenticação é um processo de examinar se a pessoa é de fato aquela correspondente da sua conta, para ser exibido seu conteúdo correspondente.

O serviço de *login* do *Facebook* é representado por um *access token*, para que possa ser obter todos os dados públicos do usuário, a figura 20 é apresentada principal classe do aplicativo, responsável por efetuar conexão com login e passando por parâmetros APP_ID identificação da aplicação que será responsável em efetuar a integração dos dados.

```

FacebookActivity.java
1 package android.app.carona;
2
3 import java.io.IOException;
35
36 public class FacebookActivity extends Activity implements
37     android.view.View.OnClickListener {
38
39     ImageView pic, button;
40     SharedPreferences sp;
41     TextView welcome;
42     Facebook fb;
43     Button postar;
44     Button cadastrar, listar;
45     Button listacarona, consulta;
46     Button conexao;
47
48 @SuppressWarnings("deprecation")
49 @Override
50 protected void onCreate(Bundle savedInstanceState) {
51     super.onCreate(savedInstanceState);
52     setContentView(R.layout.activity_facebook);
53
54     String APP_ID = getString(R.string.APP_ID);
55     fb = new Facebook(APP_ID);
56     sp = getPreferences(MODE_PRIVATE);
57     String access_token = sp.getString("access_token", null);
58     long expires = sp.getLong("access_expires", 0);
59
60     if (access_token != null) {
61         fb.setAccessToken(access_token);
62     }
63     if (expires != 0) {
64         fb.setAccessExpires(expires);
65     }

```

Figura 18 – Código fonte responsável pelo Acess Token

Ao ser feita a identificação do aplicativo pelo parâmetro *APP_ID*, conforme visto na linha 54 da Figura 18, é iniciada a aplicação estabelecendo conexão do *login* através do *access_token*. A classe *FacebookActivity* é a principal classe do projeto, é aquela declarada na criação do aplicativo na página do *Facebook Developers*.

Na figura 19 é apresentada a execução desta classe, na qual tem uma das principais funcionalidades do aplicativo. Contendo, o logo do aplicativo com o botão de *login* do *Facebook*.



Figura 19 - Tela de login

Por ser uma aplicação fora da plataforma do *Facebook* ao usar as ferramentas disponibilizadas pela *API Graph* é necessário que o usuário autentique sua conta, para que a aplicação possa integrar seus dados com a rede social, disponibilizando os dados públicos de cada usuário.

Ao clicar no botão *Login* é solicitado ao usuário que preencha o formulário de *login do Facebook*, porém esta autenticação servirá apenas para conceder a aplicação o *token* de acesso do usuário, como apresentada na Figura 20 tela de autenticação para o usuário do Facebook.

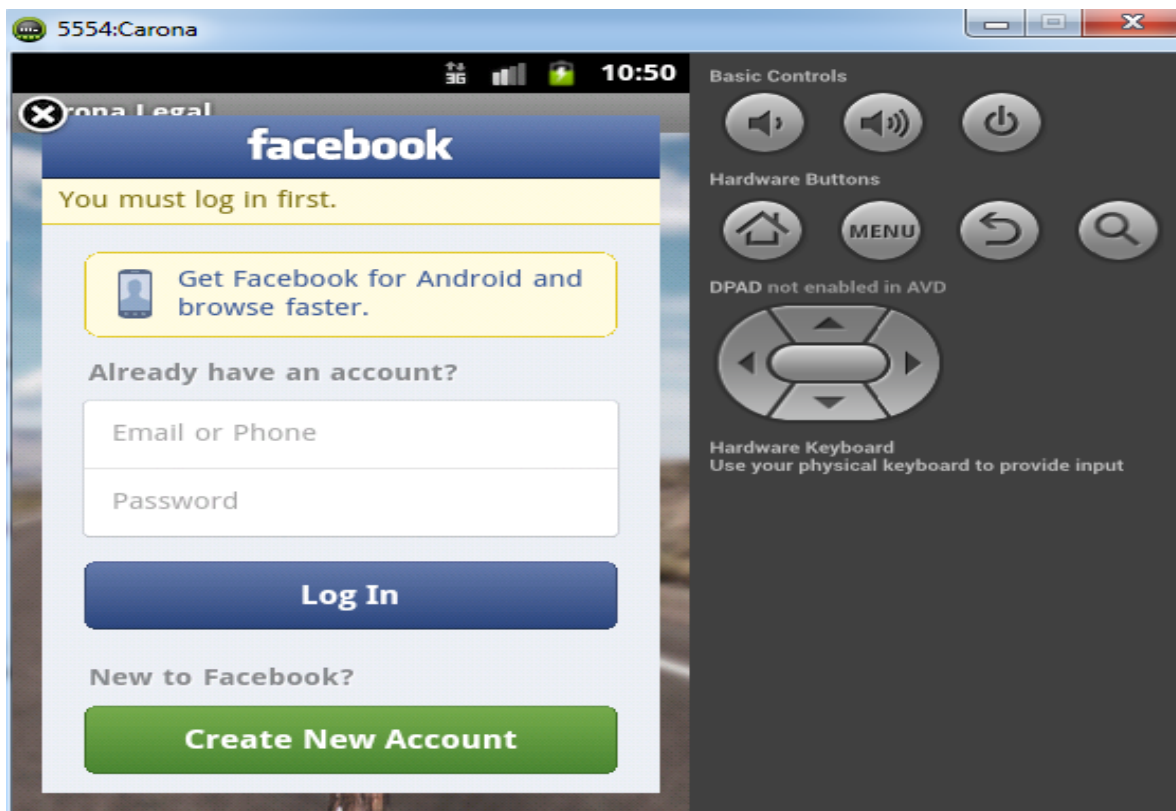


Figura 20 - Tela de autenticação do Facebook



Figura 21 - Tela de autorização do Facebook

Após efetuar o login com sua senha, a aplicação irá solicitar a permissão de acesso a alguns dados do usuário logado, ou então aparecerá que o usuário já autorizou a aplicação, como foi apresentado na Figura 21.

Neste caso os dados solicitados são apenas id (numeração única de casa usuário para sua identificação da rede social), nome, sua respectiva foto, e permissão para que possa ser postados conteúdos em seu mural.

O dados são obtidos através do método *updateButtonImage()*, que o mesmo tem a funcionalidade de verifica se usuário foi logado corretamente, feito isto os dados como *id*, *name* e *img_url* (foto do usuário) são visíveis para a aplicação, como pode ser visto na Figura 22.



```

90 public void updateButtonImage() {
91     if (fb.isSessionValid()) {
92
93
94         button.setImageResource(R.drawable.logout_button);
95         pic.setVisibility(ImageView.VISIBLE);
96         JSONObject obj = null;
97         URL img_url = null;
98
99         try {
100
101             String jsonUser = fb.request("me");
102             obj = Util.parseJson(jsonUser);
103             final String id = obj.optString("id");
104             final String name = obj.optString("name");
105             welcome.setText(" Bem - Vindo " + name);
106             img_url = new URL("http://graph.facebook.com/"+id+"/picture?type=small");
107             Bitmap bmp = BitmapFactory.decodeStream(img_url.openConnection().getInputStream());
108             pic.setImageBitmap(bmp);
109
110
111         } catch (FacebookError e) {
112             e.printStackTrace();
113         } catch (JSONException e) {
114             e.printStackTrace();
115         } catch (MalformedURLException e) {
116             e.printStackTrace();
117         } catch (IOException e) {
118             e.printStackTrace();
119         }
120     }

```

Figura 22 - Código fonte dados do usuário

Estes são os dados públicos de qualquer usuário, porém tenha-se a necessidade da permissão dos mesmos para que possa ser usada no aplicativo.

O próximo passo do aplicativo é a execução da tela principal, contendo os dados do usuário da rede social, com algumas funcionalidades do aplicativo para divulgação de caronas, como pode ser vista na figura 23.

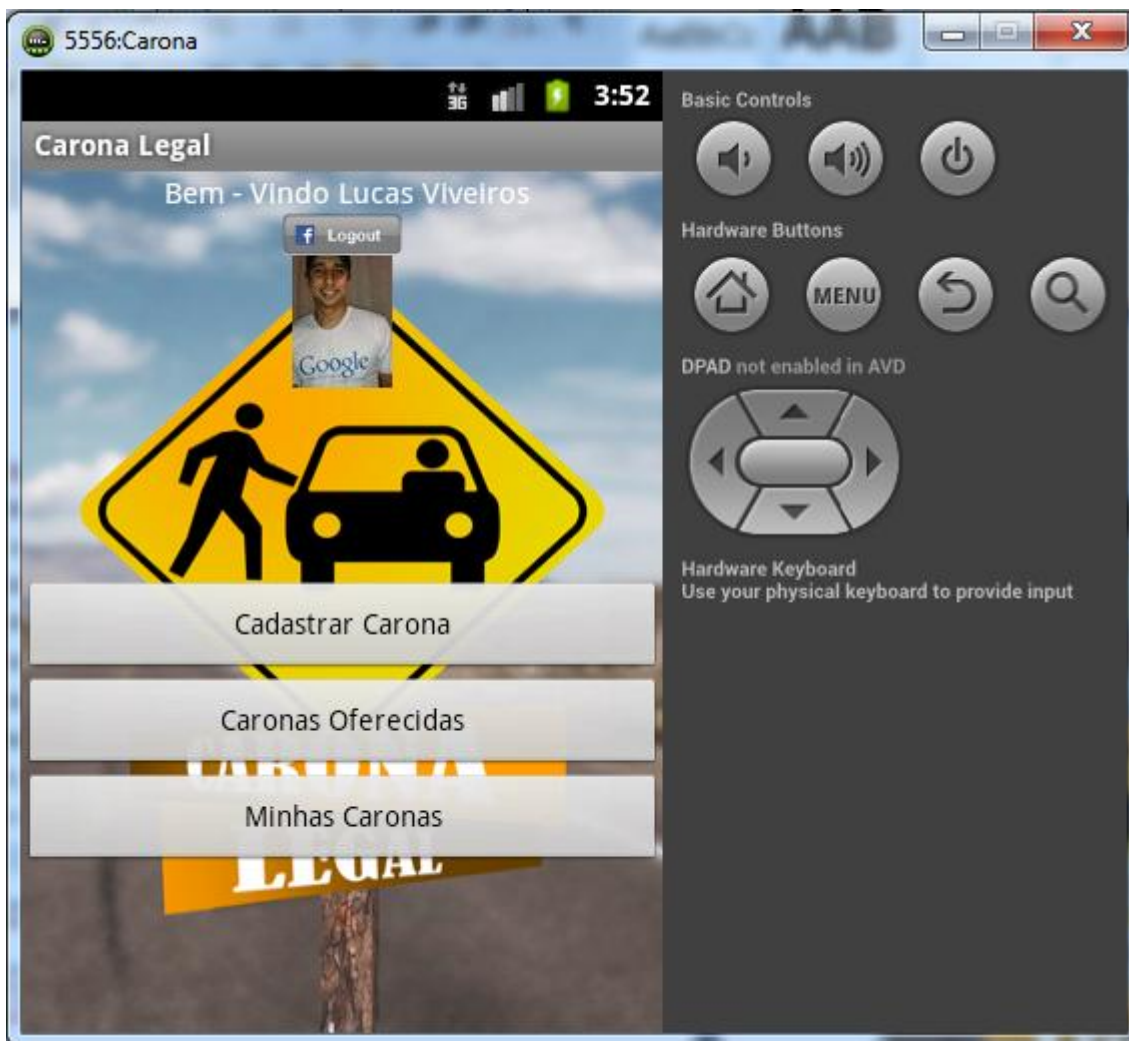


Figura 23 - Tela principal

Além dos dados da rede social, nesta tela principal encontra-se o botão de Logout para que o usuário possa desconectar da aplicação. As funcionalidades de cadastro da carona, caronas oferecidas e minhas caronas que serão detalhadas na próxima seção.

4.5 Funcionalidades do aplicativo

O aplicativo tem o objetivo de cadastrar uma carona para que possa ser publicada tanto na aplicação ou para os seus amigos da rede social Facebook. Para o armazenamento dos dados do aplicativo foi utilizado um servidor de banco de dados *mysql* para que todas as requisições do sistema sejam armazenados. Na próxima seção serão detalhadas essas funcionalidades do aplicativo Carona Legal.

4.5.1 Cadastrar Caronas e publicação no Facebook

A tela principal do aplicativo que se encontra a funcionalidade de cadastrar uma nova carona, ao clicar no botão é exibida uma tela de cadastro com os seguintes campos a ser preenchidos pelo usuário, origem (cidade de partida), destino (cidade na qual se pretende chegar), Horas (horário de partida da carona), quantidade de vagas e a data.

Os campos de origem e destino são do tipo *AutoCompleteTextView* que mostra sugestões de conclusão automaticamente enquanto o usuário está digitando. A lista de sugestões é exibida a partir de um adaptador de dados, contendo todas as cidades dos estados do Paraná e São Paulo.

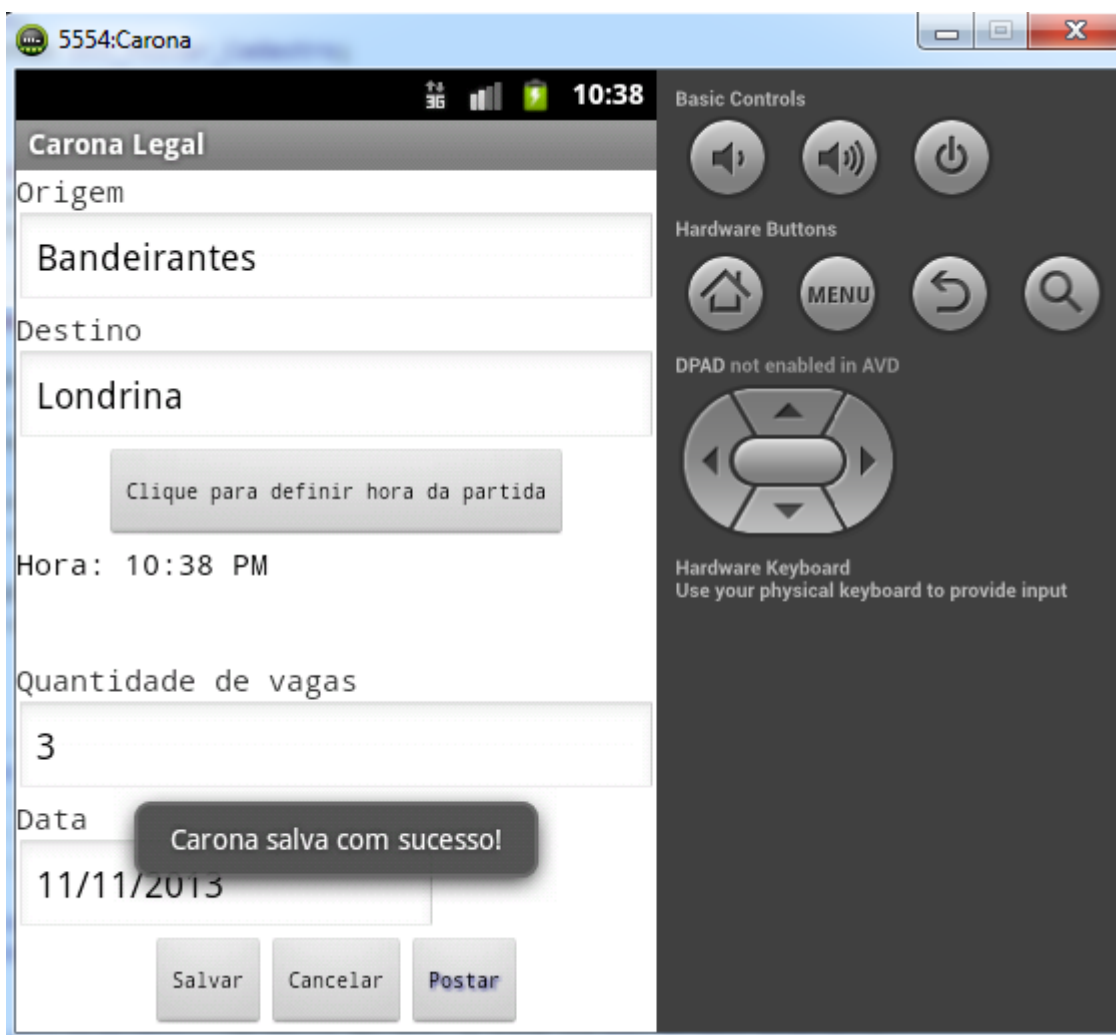


Figura 24 - Tela de Cadastro de Carona

A Figura 24 é apresentada uma carona sendo cadastrada com todos os campos preenchidos. Outra funcionalidade é poder cancelar, nesta função o sistema apenas cancela as operações feitas voltando para tela principal do aplicativo.

O botão postar tem a funcionalidade compartilhar a carona na rede social, sendo que o mesmo devem ter todos os campos obrigatoriamente preenchidos.

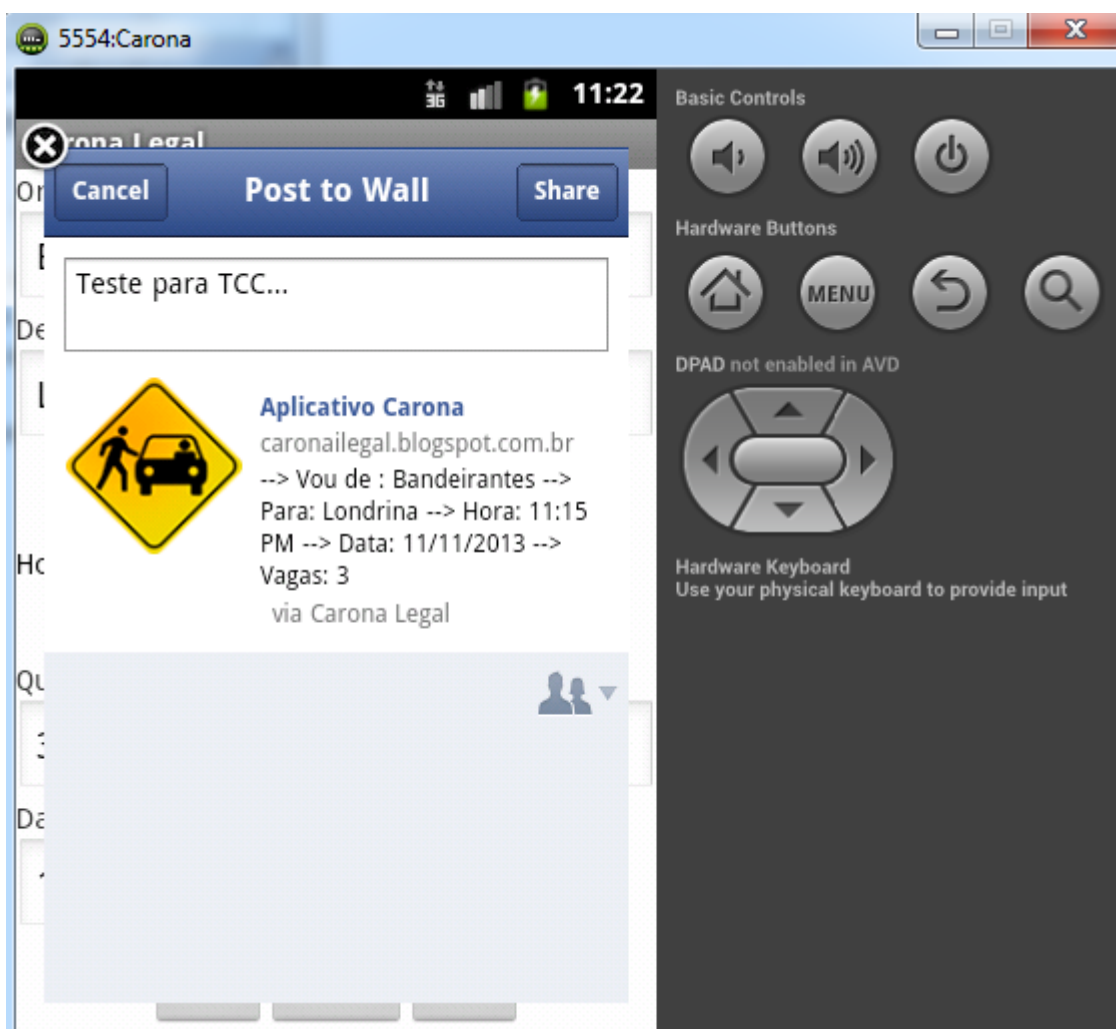


Figura 25 - Tela de compartilhamento da carona

A Figura 25 é representada a tela responsável em publicar caronas do usuário logado. Neste exemplo foi adicionada uma mensagem (Teste para TCC) e carregado à carona cadastrada pelo usuário.

Quando efetuada a publicação através do serviço integrador é possível visualizar na pagina oficial do *Facebook* do usuário o compartilhamento da carona.



Figura 26 - Exemplo de publicação da carona no Facebook

Conforme visto na Figura 26 foi realizado um exemplo de publicação na rede social, mostrando a carona compartilhada através do aplicativo Carona Legal.

4.5.2 Caronas oferecidas

Todas as caronas cadastradas e compartilhadas são salvas em um banco de dados externo que possibilita a consulta das mesmas por todos os usuários.

Como visto na Figura 27 a lista de todas as caronas dos usuários com seus respectivos nome e foto do *Facebook*, nesta tela contém um campo de pesquisa, na qual o usuário pode procurar por uma cidade de origem ou destino específica.



Figura 27 - Tela Lista de Caronas

O usuário pode apenas clicar no botão Pesquisar/Atualizar para a consulta de novas caronas.

Para que o usuário tenha controle, o sistema oferece uma tela específica contendo suas próprias caronas cadastradas e estas caronas podem ser excluídas ou alteradas pelo usuário. Na Figura 28 é apresentada esta tela de caronas cadastradas pelo usuário.

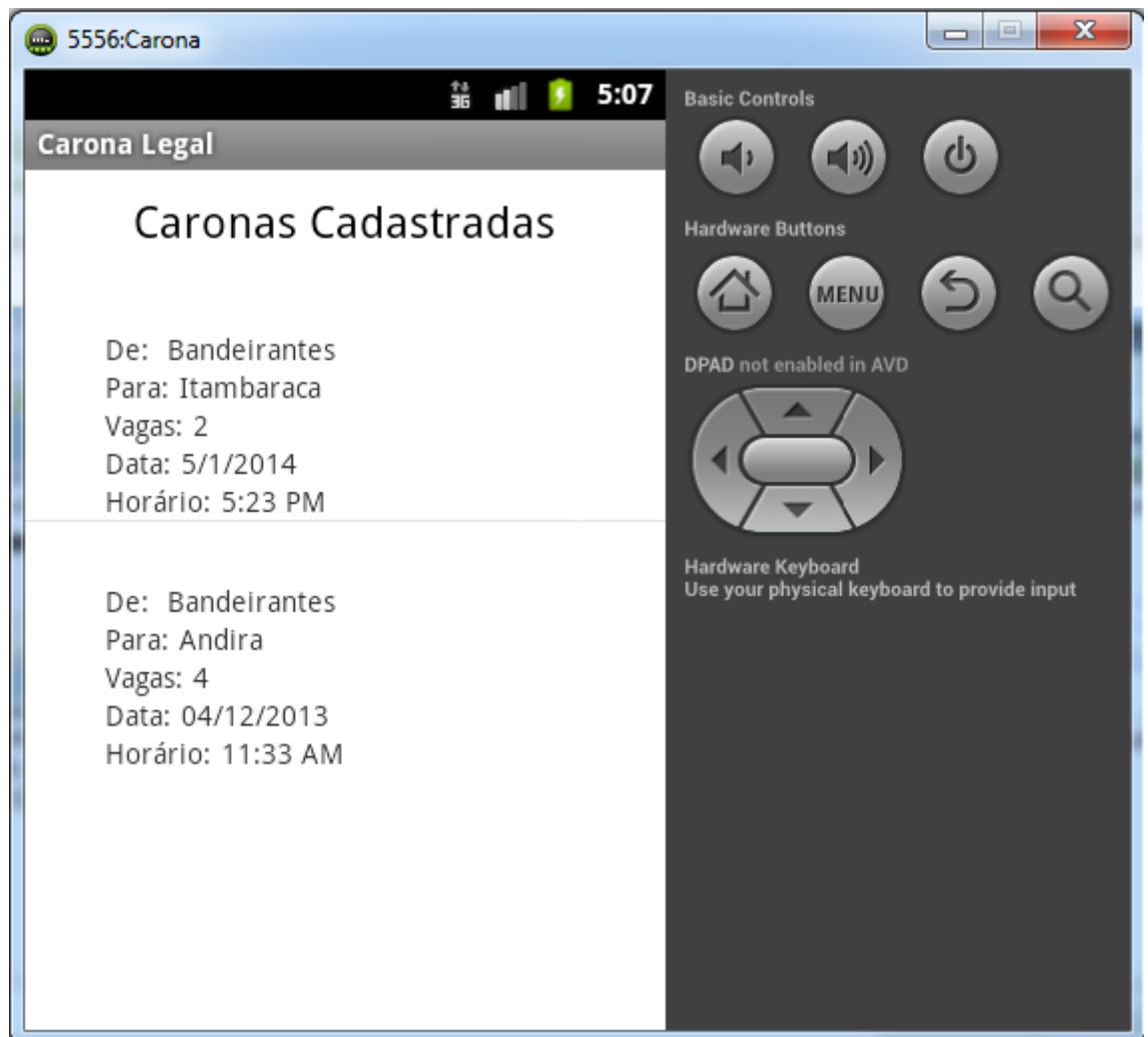


Figura 28 - Minhas Caronas

5. CONCLUSÃO

Este trabalho teve como objetivo desenvolver um aplicativo para o dispositivo móvel *Android*, onde sua principal função é divulgar informações sobre as caronas que deseja oferecer. Esta divulgação é realizada através de dois modos, tais como: publicação na linha do tempo do *Facebook* do usuário que esteja utilizando o aplicativo de caronas; busca por caronas através do próprio aplicativo aqui desenvolvido.

Com o fim do trabalho proposto, pode-se concluir que:

- O uso compartilhado dos automóveis possibilita uma redução dos gastos quando comparados a outros serviços de transporte;
- O aumento de caronas reduz o número de veículos em circulação, evitando engarrafamento, diminuindo o stress do dia a dia;
- O aplicativo se faz necessário ao ponto de vista de integração de novos meios para divulgação de caronas, possibilitando aos usuários variadas formas de se interagir com novas tecnologias;
- Trabalhar com integração da rede social possibilitou uma grande visibilidade e confiança da aplicação, ao invés de um cadastro de usuário convencional;
- A integração entre as tecnologias *Android* e *Facebook* traz resultados positivos no sentido de que é possível criar cada vez mais aplicações versáteis;

O ganho de experiência que se obteve com o desenvolvimento do aplicativo possibilitou a produção de um software móvel com banco de dados compartilhado, possibilitando que os usuários façam a busca de caronas centralizadas em apenas um aplicativo e que também possam fazer suas buscas caronas em um só lugar. Os dados são padronizados, assim todas as postagens vindas do aplicativo têm um mesmo modelo. Quanto à segurança o uso da *API* do *Facebook*, fornece ao usuário uma confiabilidade ao fornecer suas informações.

Assim, por meio de uma filosofia de liberdade e acessibilidade, o sistema operacional móvel *Android* vai se consolidando como o principal do

segmento, contribuindo de forma efetiva para o sucesso deste novo e promissor nicho de mercado.

REFERÊNCIAS

ALCANTARA, Carlos Augusto Almeida; VIEIRA, Anderson Luiz Nogueira. **TECNOLOGIA MÓVEL: UMA TENDÊNCIA, UMA REALIDADE.** Disponível em: <<http://arxiv.org/ftp/arxiv/papers/1105/1105.3715.pdf>>. Acesso em: 09 ago. 2013.

BOOCH, Grandy; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do usuário.** 2. ed. Rio de Janeiro: Campus, 2005.

CAMPOS, Augusto. **Para qual versão do Android devo desenvolver?.** 2011. Disponível em: <<http://imasters.com.br/artigo/24159/android/para-qual-versao-do-android-devo-desenvolver/>>. Acesso em: 06 jul. 2013.

DEVELOPER, Android. **Android, the world's most popular mobile platform.** Disponível em: <<http://developer.android.com/about/index.html>>. Acesso em: 27 maio 2013.

DEVELOPERS, Android. **Platform Versions.** Disponível em: <<http://developer.android.com/about/dashboards/index.html>>. Acesso em: 30 maio 2013.

DOBLER, Rodrigo Jaureguy, **Injeção de Falhas de Comunicação para Validação de Aplicações no Ambiente Android,** Porto Alegre, 2010. 18 p.

FACEBOOK. **Documentation.** Disponível em: <<https://developers.facebook.com/docs/>>. Acesso em: 07 ago. 2013.

GUSMÃO, Gustavo. **Android chega a 81% dos smartphones no mercado.** Disponível em: <http://info.abril.com.br/noticias/tecnologia-pessoal/2013/11/android-chega-a-81-dos-smartphones-no-mercado-diz-estudo-1.shtml?utm_source=redesabril_info&utm_medium=facebook&utm_campaign=redesabril_info>. Acesso em: 13 nov. 2013.

GOOGLE. **What is Android.** Disponível em <<http://code.google.com/android/what-is-android.html>>. Acesso em Setembro de 2013.

JARDIM, Robson Bresolin. **Portal Colaborativo Para Construções de Mapas Sobre Evolução de Doenças Epidemiológicas**. Disponível em: <http://www.ulbra.inf.br/joomla/images/documentos/TCCs/2011_01/TCCII_SI_RobsonJardim.pdf>. Acesso em: 05 ago. 2013.

LECHETA, Ricardo R.. **Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. São Paulo: Novatec, 2010.

LIMA, Fábio de Jesus, **Desenvolvimento de Aplicações para Plataforma Google Android**, São Paulo, 2011, 100p.

MACEDO, Letícia. **Empresas precisam incentivar carona solidária**. Disponível em: <<http://g1.globo.com/sao-paulo/noticia/2010/08/empresas-precisam-incentivar-carona-solidaria-diz-especialista.html>>. Acesso em: 17 jun. 2013.

MEDNIEKS, Zigurd; MEIKE, Blake. **Desenvolvimento de Aplicações Android**. São Paulo: Novatec, 2009, 1ª ed.

MOBILTEC, Marketing. **Utilização de API de mapas**. Disponível em: <<http://www.mobiltec.com.br/blog/index.php/utilizacao-de-api-de-mapas/>>. Acesso em: 11 jun. 2013.

OLIVEIRA, Sandra Costa de. **Educação Ambiental para Promoção da Saúde com Trânsito Solidário**. Disponível em: <<http://www5.usp.br/23695/pesquisa-da-fsp-avalia-programa-carona-solidaria/>>. Acesso em: 16 jun. 2013.

OLIVEIRA, Eduardo Alencar de. **Ser Social - Um aplicativo para divulgação de trabalhos sociais no Facebook**. Disponível em: <<http://www.campinas.sp.gov.br/uploads/pdf/1348281601.pdf>>. Acesso em: 06 ago. 2013.

PEREIRA, Lúcio Camilo Oliva; SILVA, Michel Lourenço da. **Android para desenvolvedores**. Disponível em: <http://books.google.com.br/books?hl=en&lr=&id=8u9wJowXfdUC&oi=fnd&pg=PA1&dq=Introdu%C3%A7%C3%A3o+google+maps+API+to+android&ots=LSjg15Ynr0&sig=6Jind7qTibUkAWoMzn_Elb4rCjE#v=onepage&q=Introdu%C3%A7%C3%A3o%20google%20maps%20API%20to%20android&f=false>. Acesso em: 06 jun. 2013.

SANTANA, Ana Lucia. **História do Facebook.** Disponível em: <<http://www.infoescola.com/internet/historia-do-facebook/>>. Acesso em: 06 ago. 2013.

SARDENBERG, Ronaldo. **Brasil ultrapassa um celular por habitante.** Disponível em: <<http://www.anatel.gov.br>>. Acesso em: 07 ago. 2013.