

**FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RIO GRANDE DO  
NORTE**

**PÓS- GRADUAÇÃO EM SISTEMAS CORPORATIVOS**

**UTILIZAÇÃO DA PLATAFORMA ANDROID PARA O DESENVOLVIMENTO  
DE UM APLICATIVO DE RESERVAS ON-LINE.**

**ARLINDO RODRIGUES DA SILVA NETO**

**NATAL/RN**

**2011**

ARLINDO RODRIGUES DA SILVA NETO

UTILIZAÇÃO DA PLATAFORMA ANDROID PARA O DESENVOLVIMENTO  
DE UM APLICATIVO DE RESERVAS ON-LINE.

Relatório técnico-científico, apresentado para obtenção do grau de Especialista no Curso de Especialização em Desenvolvimento de Softwares Corporativos da Faculdade Natalense para o Desenvolvimento do RN, FARN.

Orientador: Ricardo Wendell Rodrigues

NATAL/RN

2011

## RESUMO

As aplicações para dispositivos móveis estão trazendo mudanças significativas ao dia-a-dia das pessoas, seja para diversão, ou até mesmo no meio corporativo. Dessa forma, ao passo que os sistemas evoluem e facilitam a execução das tarefas diárias, pode-se verificar também que as inovações tecnológicas possibilitam que novas soluções sejam desenvolvidas.

Baseando-se no site de reservas on-line, o presente trabalho tem o intuito de desenvolver um aplicativo de reservas para *smartphones*, fazendo a utilização da plataforma android.

Palavras chave: Android, smartphones, Reservas.

## **ABSTRACT**

The mobile applications are bringing significant changes to day-to-day lives, whether for fun, or even in the middle corporate. Thus, while the systems evolve and facilitate the execution of daily tasks, can also be seen that technological innovations enable new solutions to be developed.

Based on the booking website online, this paper aims to develop an application for smartphones, making use of the android platform.

Keywords: Android, smartphones, Booking.

## LISTA DE ILUSTRAÇÕES

|                                                                                                       |    |
|-------------------------------------------------------------------------------------------------------|----|
| Figura 1 – Site do topReservas. ....                                                                  | 8  |
| Figura 2 – Arquitetura do Android. ....                                                               | 10 |
| Figura 3 – Ciclo de vida da Activity. ....                                                            | 11 |
| Figura 4 – Vitrine virtual do restaurante. ....                                                       | 13 |
| Figura 5 – Livro eletrônico de reservas. ....                                                         | 14 |
| Figura 6 – Instalação do SDK do Android. ....                                                         | 15 |
| Figura 7 – Instalação do plugin ADT no eclipse. ....                                                  | 15 |
| Figura 8 – Configuração do emulador. ....                                                             | 16 |
| Figura 9 – Emulador do android. ....                                                                  | 16 |
| Figura 10 – Organização do projeto. ....                                                              | 17 |
| Figura 11 – Heranças da classe AbstractActivity ....                                                  | 18 |
| Figura 12 – Heranças das classes AbstractListActivity e AbstractMapActivity. ....                     | 19 |
| Figura 13 – Exemplificação da integração via WebService.....                                          | 19 |
| Figura 14 – Conexão via WebService utilizando KSOAP2.....                                             | 20 |
| Figura 15 – Realização do login via WebService. ....                                                  | 21 |
| Figura 16 – Classe JsonUtil.java, responsável por converter objetos Java em JSON e<br>vice-versa..... | 22 |
| Figura 17 – Tela inicial do aplicativo. ....                                                          | 23 |
| Figura 18 – Busca de restaurantes. ....                                                               | 24 |
| Figura 19 – Resultado da busca de restaurantes.....                                                   | 24 |
| Figura 20 – Tela de login,.....                                                                       | 25 |
| Figura 21 – Tela de cadastro.....                                                                     | 25 |
| Figura 22 – Tela de busca de mesas. ....                                                              | 26 |
| Figura 23 – Posição atual do usuário. ....                                                            | 27 |
| Figura 24 – Restaurantes próximos. ....                                                               | 27 |
| Figura 25 – Query para retornar os restaurantes próximos.....                                         | 28 |
| Figura 26 – Método para calcular distância entre dois pontos. ....                                    | 28 |
| Figura 27 - Obtém a posição atual do usuário.....                                                     | 29 |

## SUMÁRIO

|       |                                    |    |
|-------|------------------------------------|----|
| 1     | INTRODUÇÃO.....                    | 7  |
| 1.1   | OBJETIVO .....                     | 8  |
| 2     | FUNDAMENTAÇÃO TEÓRICA .....        | 9  |
| 2.1   | PHP.....                           | 9  |
| 2.2   | Web Services .....                 | 9  |
| 2.3   | JSON.....                          | 9  |
| 2.4   | ANDROID.....                       | 10 |
| 2.5   | ACTIVITY .....                     | 11 |
| 3     | IMPLEMENTAÇÃO.....                 | 12 |
| 3.1   | SITE TOP RESERVAS .....            | 12 |
| 3.1.1 | Visão do Usuário .....             | 12 |
| 3.1.2 | Visão do Estabelecimento .....     | 13 |
| 3.2   | AMBIENTE DE DESENVOLVIMENTO .....  | 14 |
| 3.3   | ORGANIZAÇÃO DAS CAMADAS.....       | 17 |
| 3.4   | PRINCIPAIS CLASSES ENVOLVIDAS..... | 18 |
| 3.5   | COMUNICAÇÃO COM O SERVIDOR .....   | 19 |
| 3.6   | TELAS DO USUÁRIO .....             | 23 |
| 3.6.1 | Busca de Restaurantes .....        | 23 |
| 3.6.2 | Reservar .....                     | 24 |
| 3.6.3 | Mais próximos.....                 | 26 |
| 4     | CONCLUSÃO .....                    | 30 |
| 5     | REFERÊNCIAS .....                  | 31 |

## 1 INTRODUÇÃO

Com o grande número de sites de compras coletivas, os restaurantes vêm disponibilizando diversas promoções, assim, atraindo muitos clientes. Com isso, os restaurantes estão ficando cada vez mais cheios e não controlando de forma eficaz suas reservas, tornando o atendimento ineficiente e demorado.

Diante destes problemas, foi desenvolvido o site [www.topreservas.com.br](http://www.topreservas.com.br), que tem como objetivo disponibilizar os restaurantes da cidade para que os usuários possam realizar reservas on-line.

Visando o crescimento e expansão do serviço de reservas on-line, teve-se a idéia de desenvolver o sistema para celulares *smartphones*, assim tornaria ainda mais cômodo para o cliente, e sem contar com o aumento do número de clientes para os restaurantes.

As aplicações para dispositivos móveis estão trazendo mudanças significativas ao dia-a-dia das pessoas, seja para diversão, ou até mesmo no meio corporativo. Dessa forma, ao passo que os sistemas evoluem e facilitam a execução das tarefas diárias, pode-se verificar também que as inovações tecnológicas possibilitam que novas soluções sejam desenvolvidas.

A escolha de uma plataforma ideal para o desenvolvimento de um projeto significa optar por uma solução que forneça os melhores benefícios, em termos de custos, eficiência e tempo de desenvolvimento esperados para a finalização do projeto.

Segundo Darcey (2010), Plataforma Android: É gratuito e fácil de desenvolver, está disponível para milhões de potenciais utilizadores em todo o mundo, e tem menos limitações do que outras plataformas. Ainda assim, a plataforma Android é relativamente jovem e ainda não atingiu seu potencial completo. Em outras palavras, pode ser um pouco de uma viagem atribulada, mas ainda há tempo para saltar sobre esse movimento, crie algumas aplicações e faça seu nome.

Sendo assim, o presente trabalho pretende exemplificar a criação de um aplicativo de reservas on-line, fazendo o uso da plataforma Android, que incorpora o sistema operacional da Google para dispositivos móveis.

O capítulo 2 deste trabalho apresenta as tecnologias utilizadas para elaboração do mesmo. O capítulo 3 aborda a implementação da ferramenta e da suas funcionalidades. O capítulo 4, conclui-se com os objetivos do projeto. No capítulo 5 finaliza com as referências encontradas para desenvolvimento do presente trabalho.

## 1.1 OBJETIVO

O objetivo deste trabalho é disponibilizar uma solução móvel, baseada no site [topreservas.com.br](http://topreservas.com.br), figura 1, para os usuários realizarem reservas de mesas em restaurantes em tempo real, tornando o processo muito mais ágil e cômodo, evitando telefones ocupados, restaurantes lotados, entre outros transtornos.

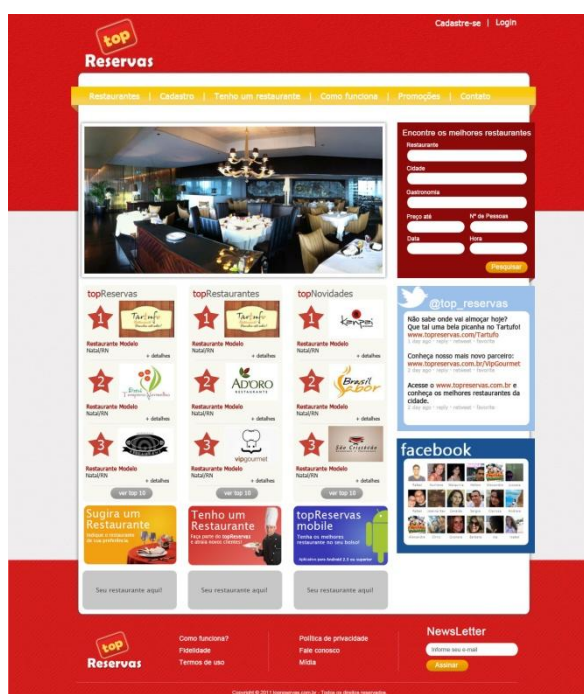


Figura 1 – Site do topReservas.  
Fonte: O Autor.

## 2 FUNDAMENTAÇÃO TEÓRICA

Para a construção do aplicativo de reservas on-line para *smartphones*, serão utilizadas as tecnologias Android e PHP em conjunto, que iram interagir através de um webservice, possibilitando consultas e envio de comandos para o servidor.

### 2.1 PHP

PHP (*Hypertext Preprocessor*) é uma linguagem de script open source de uso geral, utilizada especialmente para o desenvolvimento de aplicações Web.

### 2.2 Web Services

De acordo com BARRY, Web Service é um serviço disponível na internet, que utiliza a linguagem XML para enviar e receber mensagens através dos protocolos HTTP ou SMTP, que seja bem descrito para interação com outros sistemas e tenha um mecanismo simples para sua publicação. Em outras palavras, é um componente que permite a interoperabilidade entre sistemas de diferentes plataformas. A figura 2 – 3 ilustra a interação entre um provedor e um consumidor de um serviço.

### 2.3 JSON

Segundo MACEDO, JSON (JavaScript Object Notation) é uma padrão para formatação de dados bem simples, derivado da sintaxe de objetos em Javascript. O principal uso atualmente do JSON é ser uma alternativa ao XML na transmissão de dados entre cliente e servidor, que foi popularizado com o uso de AJAX.

Em geral, o formato JSON costuma ser menor que o equivalente em XML, e isso é uma vantagem quando é necessário que os dados sejam transmitidos pela rede. Mas a vantagem principal é que o formato do JSON é mais simples de ser interpretado.

## 2.4 ANDROID

O Android é uma plataforma para smartphones, baseada no sistema operacional Linux, possui diversos componentes, com uma variada disponibilidade de bibliotecas e interface gráfica, além de disponibilizar ferramentas para a criação de aplicativos (Lecheta, 2009).

A plataforma inclui desde Sistema Operacional (Android SO), *middleware* e aplicativos importantes. O SDK do Android fornece as ferramentas e APIs necessárias para começar a desenvolver aplicativos que executam em dispositivos com Android.

Desenvolvido pela *Open Handset Alliance*, Android é, segundo GOOGLE, é considerada uma plataforma móvel completa, livre e aberta, apresentando características incorporadas de outros sistemas, além de inúmeros conceitos inovadores para o segmento móvel.

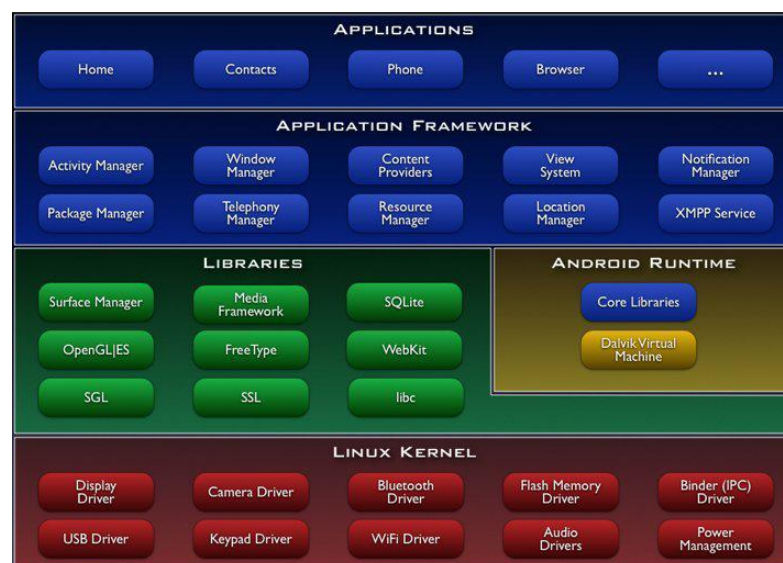


Figura 2 – Arquitetura do Android.  
Fonte: Google Inc.

## 2.5 ACTIVITY

Uma Activity é um componente do aplicativo que fornece uma tela com a qual os usuários podem interagir, a fim de fazer algo. Para cada atividade é dada uma janela na qual se desenha sua interface de usuário.

Um aplicativo normalmente consiste de múltiplas atividades que são frouxamente ligadas uns aos outros. Normalmente uma atividade em um aplicativo é especificada como a atividade “principal”, que é apresentada ao usuário ao iniciar o aplicativo pela primeira vez. Cada atividade pode começar outra atividade, a fim de executar ações diferentes. Cada vez que começa uma nova atividade, a atividade anterior é interrompida, mas o sistema preserva a atividade em uma pilha (a “pilha de volta”). Quando uma nova atividade começa, é empurrada para a pilha de volta e leva o foco do usuário. A pilha de volta usa “last in, first out” como mecanismo de fila, então, quando o usuário está em uma atividade e pressiona a tecla BACK, a atividade é removida da pilha (e destruída) e retoma a atividade anterior. DUARTE (2010).

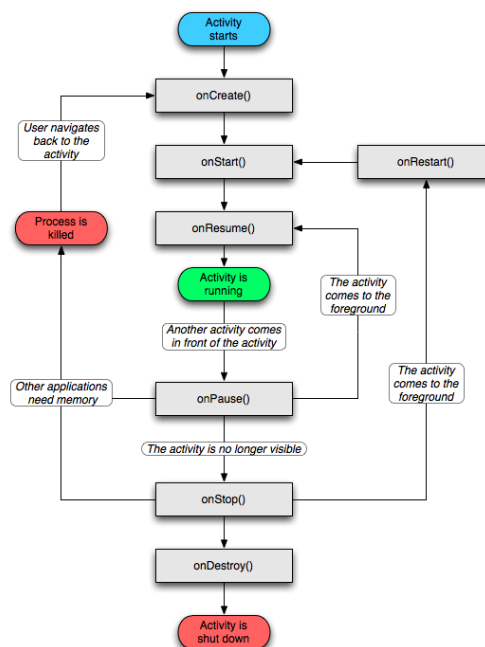


Figura 3 – Ciclo de vida da Activity.  
Fonte: Google Inc.

### **3 IMPLEMENTAÇÃO**

Neste capítulo terá uma descrição breve do funcionamento do site topReservas, que será utilizado como base para elaboração deste trabalho, e em seguida será mostrada a organização e implementação do projeto.

#### **3.1 SITE TOP RESERVAS**

O topReservas é um serviço gratuito de reservas on-line. Atualmente seu foco são os estabelecimentos gastronômicos.

Possui uma variedade de restaurantes, servindo também de guia astronômico, onde o usuário pode buscá-los de acordo com suas preferências: localização, gastronomia, preço ou disponibilidade.

##### **3.1.1 Visão do Usuário**

O usuário acessando o site: [www.topreservas.com.br](http://www.topreservas.com.br) terá acesso a todos os estabelecimentos gastronômicos cadastrados. O usuário poderá escolher o estabelecimento de acordo com suas preferências, utilizando a busca ou por sugestão dos links promocionais.

Uma vez escolhido o estabelecimento, o usuário acessa sua vitrine virtual, como mostra na figura 2. Nela ele irá observar as fotos do estabelecimento, sua descrição, horários de funcionamento, localização, a avaliação do estabelecimento feita por outros clientes, entre outras informações. Caso o usuário deseje realizar suas reserva, basta apenas fazer seu login (ou cadastro, caso ainda não tenha), informar a data, horário e nº de lugares desejados, confirmar os dados e pronto. Caso o estabelecimento não tenha mais disponibilidade o usuário receberá um aviso e sugestões de outros horários.

Concluída a marcação da reserva, o usuário receberá uma notificação por e-mail confirmando a sua reserva.

No dia e horário escolhido, o usuário irá ao estabelecimento e consolidar sua reserva. Após isto estará apto a qualificar o estabelecimento acessando o site do topReservas.

A qualquer momento (antes do horário da reserva), o usuário pode cancelar sua reserva. Para isso, basta acessar sua conta no site do topReservas e cancelar a reserva desejada. Automaticamente esta reserva será cancelada no livro de reservas on-line do estabelecimento.

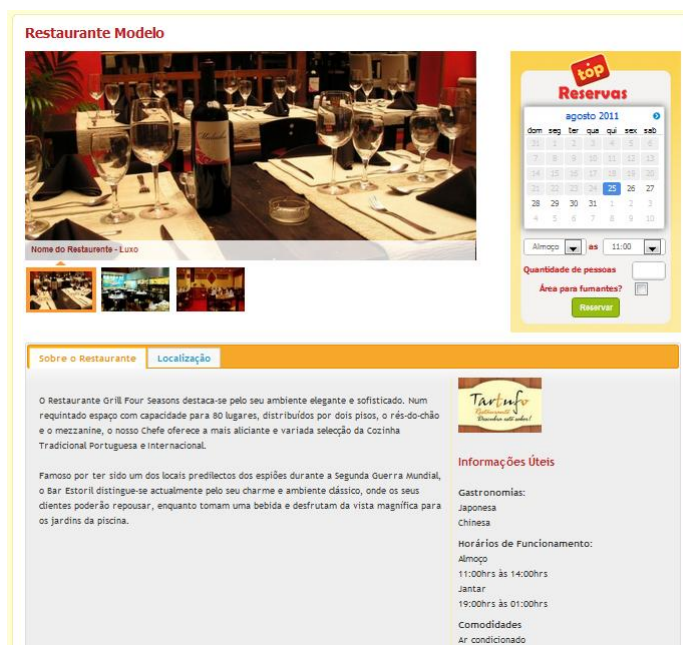


Figura 4 – Vitrine virtual do restaurante.  
Fonte: O Autor.

### 3.1.2 Visão do Estabelecimento

Uma vez aderido ao serviço topReservas, o estabelecimento ganhará sua vitrine virtual onde postará suas informações, e o livro eletrônico de reservas on-line. Assim, poderá cadastrar as reservas solicitadas via telefone, ou no próprio estabelecimento. A medida que as reservas são feitas via vitrine virtual, elas vão sendo inseridas automaticamente no livro eletrônico de reservas do estabelecimento.

Quando o cliente comparecer ao estabelecimento afim de honrar sua reserva o estabelecimento deve informar no livro eletrônico que o usuário “chegou” (compareceu) e após a refeição que “saiu”, Como mostra na figura 3.

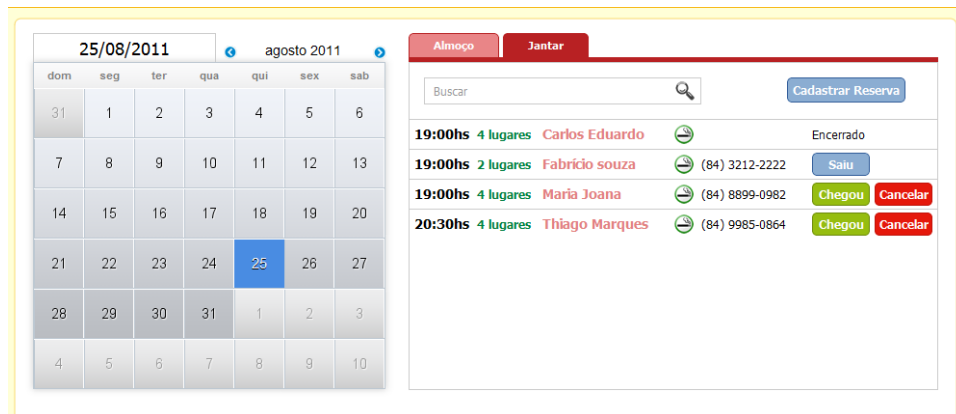
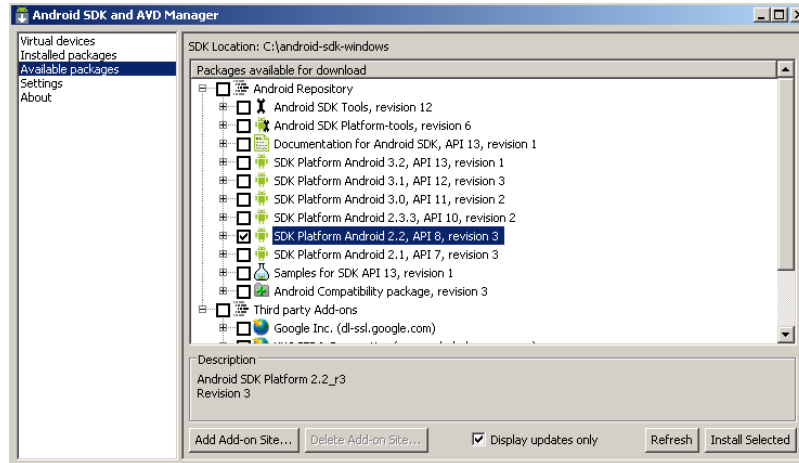


Figura 5 – Livro eletrônico de reservas.  
Fonte: O Autor.

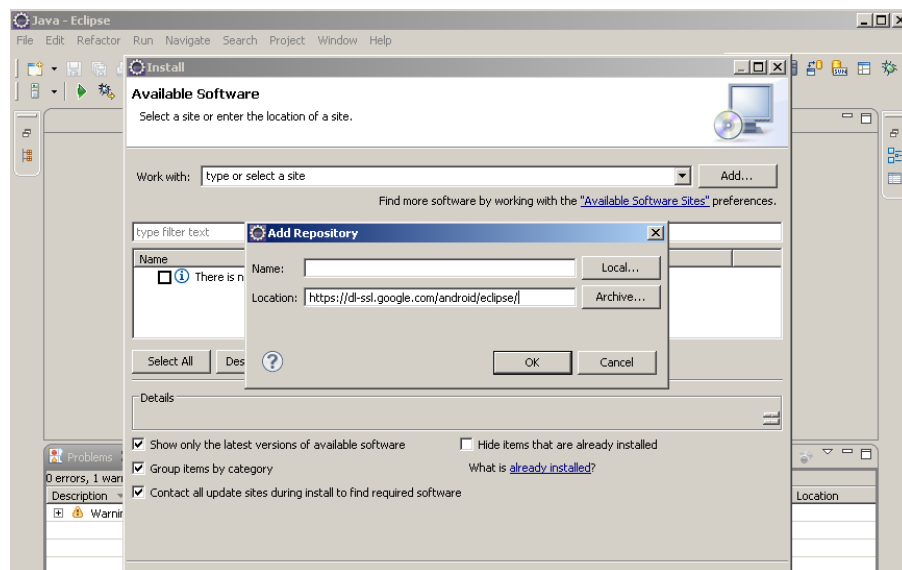
### 3.2 AMBIENTE DE DESENVOLVIMENTO

A construção de aplicativos Android pode ser realizada com as mesmas ferramentas utilizadas no desenvolvimento de aplicativos Java. O framework de desenvolvimento do Android oferece bibliotecas e funcionalidades específicas da plataforma para criação de aplicações móveis e algumas das principais bibliotecas já conhecidas e utilizadas no desenvolvimento de aplicativos Java.

Para a elaboração do aplicativo, é necessária a utilização da IDE Eclipse, que pode ser obtido no site oficial do eclipse <http://www.eclipse.org/downloads/>, o SDK do Android Revisão 11 ou superior, exibido na figura 4, e o plugin ADT do Android para o Eclipse. Na figura 5 mostra a instalação do plugin ADT. Estes podem ser obtidos em <http://developer.android.com/sdk/index.html>. É necessário também que tenha instalado a Máquina Virtual Java.

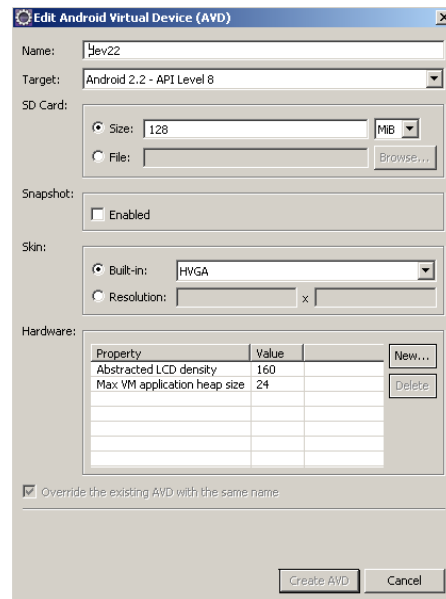


*Figura 6 – Instalação do SDK do Android.  
Fonte: O Autor.*



*Figura 7 – Instalação do plugin ADT no eclipse.  
Fonte: O Autor.*

Para finalizar, deve-se criar o emulador (AVD - Android Virtual Device), para realizar os testes necessários da aplicação. Como mostra a figura 6. Na figura 7 mostra o emulador em funcionamento.



*Figura 8 – Configuração do emulador.  
Fonte: O Autor.*



*Figura 9 – Emulador do android.  
Fonte: O Autor.*

O emulador Android, como mostra na figura 8, foi utilizado para testar e executar a aplicação sem a necessidade de um dispositivo físico. O emulador consegue simular tanto o hardware quanto o software de um dispositivo móvel, inclusive simulando o recebimento e realização de chamadas. Pode também invocar

outros aplicativos dentro do sistema Android, acessar a rede do computador, executar vídeos e músicas, armazenar e recuperar dados.

### 3.3 ORGANIZAÇÃO DAS CAMADAS

O projeto em si foi dividido em 5 pacotes, que foram organizados seguindo a lógica de divisão de camadas. Como mostra na Figura 12.

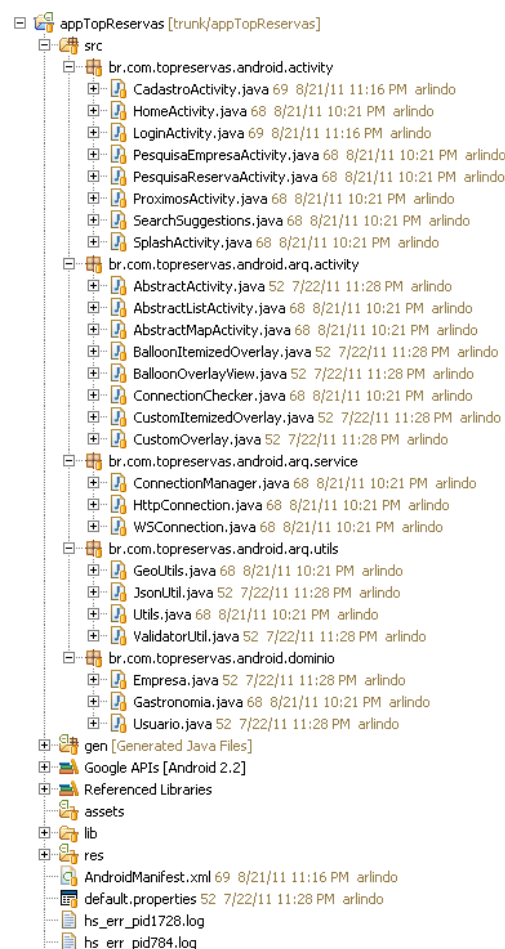


Figura 10 – Organização do projeto.  
Fonte: O Autor.

### 3.4 PRINCIPAIS CLASSES ENVOLVIDAS

Como mostra na figura 10, foi implementada a classe `AbstractActivity`, que herda a classe `Activity`. A `AbstractActivity` será a `Activity` “pai” de todas as demais `Activities` do projeto. Nesta conterá todas as operações necessárias para o uso exclusivo de uma `Activity`.

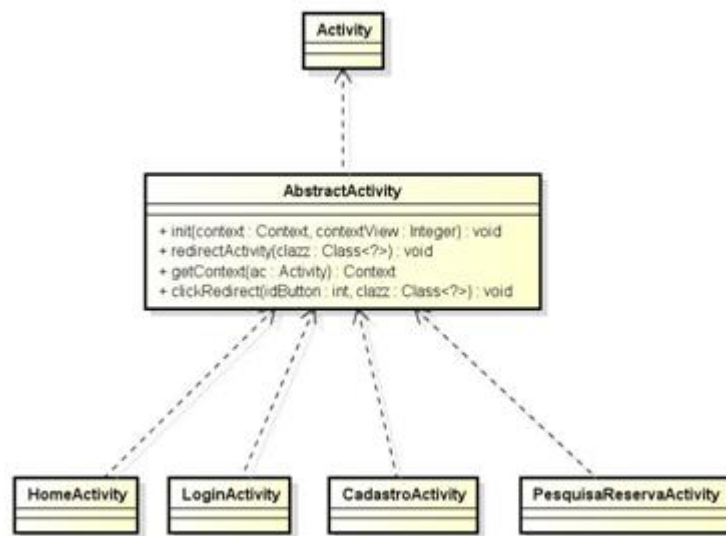


Figura 11 – Heranças da classe `AbstractActivity`  
Fonte: O Autor.

Da mesma forma foi criada a classe intermediária `AbstractListActivity`, onde todas as classes que herdam esta classe, terão o comportamento de uma `ListActivity`. Acontece o mesmo com `AbstractMapActivity`, todas as classes herdadas terão o comportamento de um `MapActivity`, porém com funcionalidades adicionais.



Figura 12 – Heranças das classes *AbstractListActivity* e *AbstractMapActivity*.  
Fonte: O Autor.

### 3.5 COMUNICAÇÃO COM O SERVIDOR

Para realizar a integração foi disponibilizado pelo site, topreservas, webservice que facilitará a integração entre o aplicativo android e o site. A figura 10 exemplifica a integração.

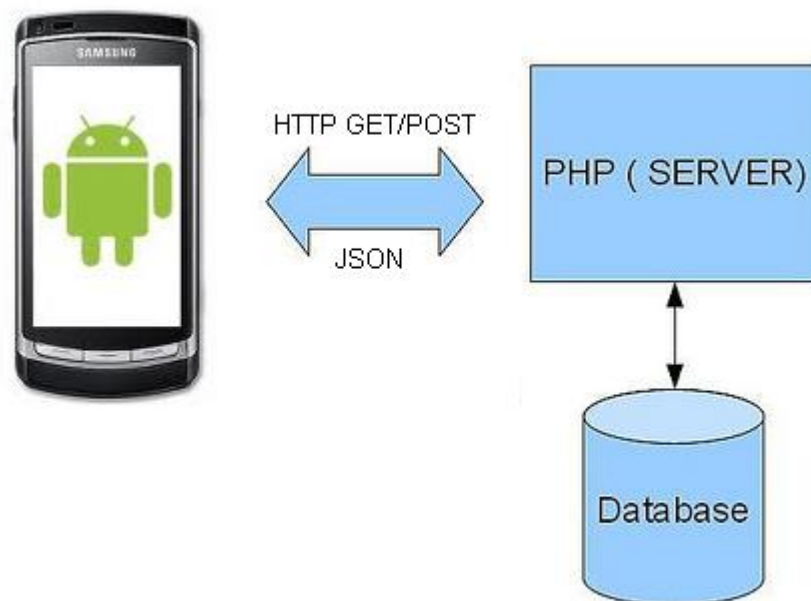


Figura 13 – Exemplificação da integração via Webservice.  
Fonte: O Autor.

Foi utilizada a biblioteca KSOAP2, muito conhecida em Java ME para consumo de aplicações. Como Android já vem integrado no mínimo com o Java 5.0, é possível facilmente rodar bibliotecas que já funcionam para Java SE, como é o caso do KSOAP2. Na figura 11 mostra a realização da conexão utilizando o KSOAP2.

```
/**
 * Método que realiza a comunicação com o webservice
 * @param method
 * @param param
 * @return
 */
private static String call(String method, String param) {
    try {
        SoapObject request = new SoapObject(NAMESPACE, OPERATION);

        request.addProperty("method", method);
        request.addProperty("params", param);

        SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(
            SoapEnvelope.VER11);

        envelope.setOutputSoapObject(request);

        HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);

        androidHttpTransport.call(OPERATION, envelope);

        String result = (String) envelope.getResponse();

        return result;
    } catch (Exception e) {
        Log.e("Erro WSConnection", e.getMessage());
        return null;
    }
}
```

*Figura 14 – Conexão via Webservice utilizando KSOAP2.  
Fonte: O Autor.*

Para exemplificar o uso do web service, a figura 11 mostra o trecho de código responsável para a realização do login.

```

/**
 * Realiza o login
 * @param login
 * @param senha
 * @return
 */
public Usuario login(String login, String senha) {
    try {
        Usuario usuario = new Usuario(login, senha);
        String param = JsonUtil.objectToJSON(usuario);

        String result = call("login", param);
        usuario = (Usuario) JsonUtil.JSONToObject(result, Usuario.class);

        return usuario;
    } catch (Exception e) {
        Log.e("Erro", e.getMessage());
        return null;
    }
}

```

*Figura 15 – Realização do login via WebService.  
Fonte: O Autor.*

Para a conversão de objetos Java em JSON foi utilizada a biblioteca Gson, também pode ser usada para converter uma string JSON para um objeto Java equivalente. Na figura 12 exemplifica a conversão de um objeto Java para JSON.

```

/**
 * Classe com operações úteis
 *
 * @author Arlindo
 *
 */
public class JsonUtil {

    /**
     * Converte um Objeto para JSON
     * @param obj
     * @return
     */
    public static String objectToJSON(Object obj){
        Gson gson = new Gson();
        return gson.toJson(obj);
    }

    /**
     * Converte de JSON para a Classe informada
     * @param json
     * @param clazz
     * @return
     */
    public static Object JSONToObject(String json, Class<?> clazz){
        try {
            Gson gson = new Gson();
            return gson.fromJson(json, clazz);
        } catch (JsonSyntaxException e){
            Log.e("Erro na Conversão JSON", e.getMessage());
            return null;
        }
    }
}

```

Figura 16 – Classe *JsonUtil.java*, responsável por converter objetos Java em JSON e virce-versa.  
Fonte: O Autor.

### 3.6 TELAS DO USUÁRIO

Ao iniciar o aplicativo serão exibidas algumas opções, como mostra a figura 13, onde através destas o usuário poderá busca um restaurante, realizar uma reserva diretamente a um restaurante específico ou buscar por restaurantes próximos a eles, onde esta última opção utilizará os recursos do GPS, para obter a opção atual do usuário.



*Figura 17 – Tela inicial do aplicativo.  
Fonte: O Autor.*

#### 3.6.1 Busca de Restaurantes

Para realizar a busca, é necessário informar o nome do restaurante no espaço indicado, como mostra a figura 14. Depois de informado o nome e selecionado a opção de buscar no ícone da lupa, serão exibidos os possíveis restaurantes que possuem o texto que foi informado, onde nesta listagem, serão exibidos, distância, da posição atual até ele, o tempo que levará para chegar até ele andando, o nome e o endereço, como exemplifica na figura 15.



Figura 18 – Busca de restaurantes.  
Fonte: O Autor.



Figura 19 – Resultado da busca de restaurantes.  
Fonte: O Autor.

### 3.6.2 Reservar

Para a realização da reserva é necessário escolher a opção “Reservar”, após selecionada, caso não esteja logado, será exibida a tela de login, para que o usuário seja identificado pelo sistema, como mostra na figura 16.



topReservas

**top**  
**Reservas**

Seja bem vindo!

Usuário:

Senha:

Entrar Cadastre-se!

*Figura 20 – Tela de login,  
Fonte: O Autor.*

Caso o usuário não possua cadastro, será possível realizar um novo cadastro selecionando a opção “Cadastre-se”, então será exibida a tela de cadastro como mostra a figura 17.



topReservas

**Cadastro**

Nome

E-mail

Celular

Senha

Confirma  
senha

Cadastrar

*Figura 21 – Tela de cadastro.  
Fonte: O Autor.*

Após realizado o login, será exibida a tela de busca de mesas, figura 18, nesta tela será possível realizar a busca de restaurantes que possuam mesas disponíveis para reserva. A busca poderá ser filtrada por número de pessoas, data da reserva, hora, gastronomia ou preço.



*Figura 22 – Tela de busca de mesas.  
Fonte: O Autor.*

Na próxima tela exibida, serão listados os restaurantes disponíveis para reserva, da mesma forma que na busca, figura 15. Para finalizar é necessário selecionar o restaurante de preferência e assim estará concluída a reserva.

### **3.6.3 Mais próximos**

Para realizar a busca de restaurantes mais próximos a localidade do usuário, é necessário a ativação do GPS, assim será possível obter a localização atual do usuário e consequentemente a posição dos restaurantes próximos. A figura 19 mostra a exibição da posição atual do usuário.



Figura 23 – Posição atual do usuário.  
Fonte: O Autor.

Caso exista restaurantes cadastrados no raio de 1 KM da posição atual do usuário, serão exibidos no mapa, como mostra a figura 20.

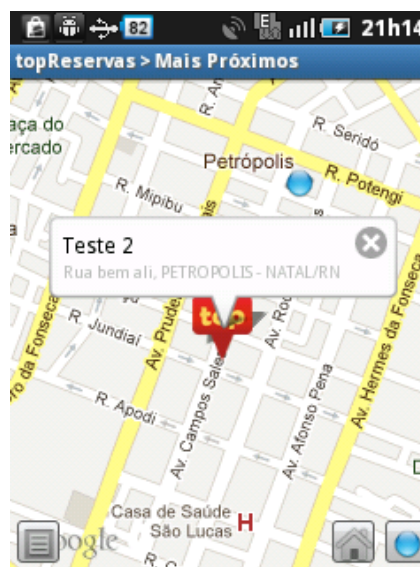


Figura 24 – Restaurantes próximos.  
Fonte: O Autor.

```

SELECT *, 6371 * 2 * ASIN(SQRT(
POWER(SIN((:orig_lat - abs(dest.lat)) * pi()/180 / 2),
2) + COS(:orig_lat * pi()/180 ) * COS(abs(dest.lat) *
pi()/180) * POWER(SIN((:orig_lon - dest.lon) *
pi()/180 / 2), 2) )) as distance
FROM empresas dest
having distance < :dist
ORDER BY distance limit 10

```

*Figura 25 – Query para retornar os restaurantes próximos.  
Fonte: O Autor.*

Para realizar a consultas das empresas próximas, foi utilizada a query SQL exibida na figura 22, onde realiza o cálculo de proximidade baseado no geoposicionamento, onde o valor 6371 equivale à quantidade de quilômetros da circunferência da terra.

A obtenção da distância dos restaurantes a posição atual do usuário, figura 24, foi obtida através do método `distanceKM()`, da classe `GeoUtils`. Na figura 23 mostra o código para realização do cálculo.

```

/**
 * Operações com geoposicionamento.
 */
public class GeoUtils {
    private static int EARTH_RADIUS_KM = 6371;

    public static int MILLION = 1000000;

    /**
     * Retorna a distancia em quilometros entre dois pontos da terra
     * @param lat1
     * @param lon1
     * @param lat2
     * @param lon2
     * @return.
     */
    public static double distanceKm(double lat1, double lon1, double lat2, double lon2) {
        double lat1Rad = Math.toRadians(lat1);
        double lat2Rad = Math.toRadians(lat2);
        double deltaLonRad = Math.toRadians(lon2 - lon1);

        return Math.acos(Math.sin(lat1Rad) * Math.sin(lat2Rad) + Math.cos(lat1Rad) * Math.cos(lat2Rad)
            * Math.cos(deltaLonRad))
            * EARTH_RADIUS_KM;
    }
}

```

*Figura 26 – Método para calcular distância entre dois pontos.  
Fonte: O Autor.*

```

//obtem a posição atual
locOverlay = new MyLocationOverlay(this, getMapView()) {
    @Override
    public void onLocationChanged(Location loc) {
        super.onLocationChanged(loc);
        location = new GeoPoint((int) (loc.getLatitude() * 1E6), (int) (loc.getLongitude() * 1E6));

        SharedPreferences.Editor editor = prefs.edit();
        editor.putInt("location_lat", (int) (loc.getLatitude() * 1E6));
        editor.putInt("location_lon", (int) (loc.getLongitude() * 1E6));
        editor.commit();
    }
};

```

*Figura 27 - Obtém a posição atual do usuário.  
Fonte: O Autor.*

## 4 CONCLUSÃO

O objetivo deste trabalho foi mostrar uma nova forma de aplicação de reservas on-line, para o usuário que procura restaurantes menos ocupados de forma mais ágil e eficaz, utilizando seu smartphone.

A tendência é aumentar cada vez mais a quantidade de aplicativos para android, assim facilitando ainda mais a vida dos usuário assíduos, e agilizando o processo de informatização e de integração social.

O aplicativo ainda não tomou sua forma final, mas já fica a idéia para que possa ser utilizadas em diversas outras áreas.

## 5 REFERÊNCIAS

AMORIM, A. D. (2011). *Android, uma visão geral*.

BARRY, D. K. (2009). *Service Architecture*. Acesso em 02 de Setembro de 2011, disponível em [http://www.service-architecture.com/web-services/articles/web\\_services\\_definition.html](http://www.service-architecture.com/web-services/articles/web_services_definition.html).

GARMENTA, M. (2011). *Learning Android*. O'Reilly.

Google Inc. (2011). *Android Code*. Acesso em 20 de Agosto de 2011, disponível em Android: <http://code.google.com/android>.

Google Inc. (2011). *What is Android*. Acesso em 20 de Agosto de 2011, disponível em <http://developer.android.com/guide/basics/what-is-android.html>.

MACEDO, A. (2010). *K19 Treinamentos*. Acesso em 02 de Setembro de 2011, disponível em <http://www.k19.com.br/artigos/webservice-restful-utilizando-json/>.

MCKENZIE, D. (2011). *Smashing Magazine*. Acesso em 26 de Setembro de 2011, disponível em <http://coding.smashingmagazine.com/2011/06/30/designingfor-android/>.

MURPHY, M. L. (2011). *Android Programming Tutorials*. CommonsWare.

PHP. (2011). *PHP*. Acesso em 25 de Agosto de 2011, disponível em [http://www.php.net/manual/pt\\_BR/intro-what-is.php](http://www.php.net/manual/pt_BR/intro-what-is.php)