LunchBOX – Sistema online para reservas de refeições[1]

Iago Ferreira Frozza[2]

Ricardo Vanni Dallasen[3]

RESUMO

Neste trabalho foi desenvolvido um sistema para gerenciamento de entrega de refeições prontas. A principal motivação para a criação do sistema foi atender uma demanda em potencial não explorada, visto que as opções disponíveis no mercado carecem de funcionalidades e disponibilidade. Foram desenvolvidos um aplicativo para a plataforma móvel Android e um servidor. O aplicativo apresenta para o usuário um menu semanal de opções para que sejam selecionados quais dias ele deseja receber sua refeição. O servidor recebe as requisições do aplicativo e gera um relatório para o administrador, contendo os dias selecionados em que cada usuário desejar receber as refeições. Nos testes realizados o sistema funcionou de maneira adequada.

Palavras-chave: Android. Aplicativo Móvel. Marmita. Web Service.

1 INTRODUÇÃO

Cada vez mais as pessoas passam menos tempo em casa, gerando um aumento da demanda por comida pronta. Segundo dados da FGV-SP 2016 o Brasil conta com 168 milhões de dispositivos, gerando a média de 1,6 *smartphone* para cada habitante, com isso é de se esperar que alguns hábitos sejam alterados, como o fato de se realizar pedidos através de outras formas, além do tradicional telefone (MEIRELLES, 2016).

---

[1] Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Câmpus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet, na cidade de Passo Fundo, em 2016.

[2] Aluno do curso de Tecnologia de Sistemas para Internet no Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense de Passo Fundo (IFSul). E-mail: azzorf@gmail.com

[3] Orientador, professor do Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense de Passo Fundo (IFSul). E-mail: ricardo.dallasen@passofundo.ifsul.edu.br

Os aplicativos mais populares na região visam atender a demanda de tele entrega de refeições rápidas e noturnas, como por exemplo, iFood, Devorando, Delivery Much, entre outros. Existem poucas soluções para reserva de refeições cotidianas, conhecida popularmente como vianda ou marmita. Neste ramo existem poucas empresas que estão utilizando este tipo de tecnologia e nenhuma está presente na região do norte do Rio Grande do Sul. Visando atender esta demanda em potencial, este projeto se propõe em desenvolver um sistema de gerenciamento de pedidos, sendo este sistema dividido entre um cliente móvel e um servidor de gerenciamento.

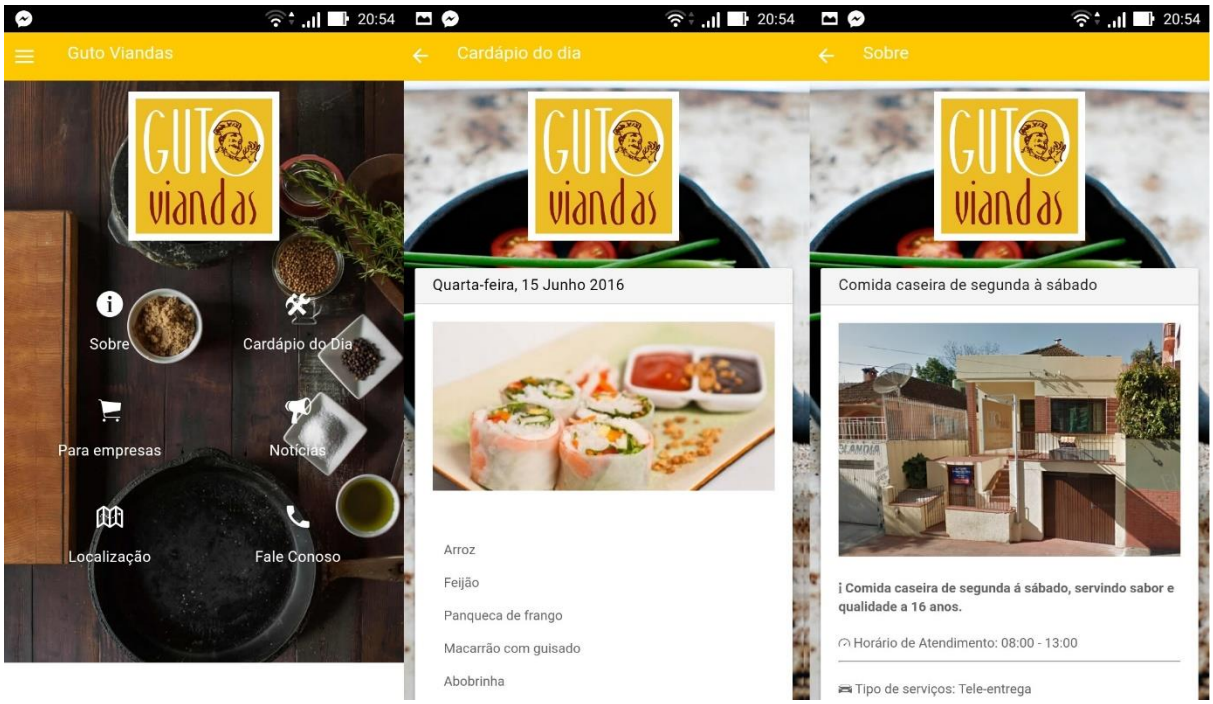
A divisão de capítulos desse artigo seguiu a seguinte ordem: na seção 2 é apresentado uma breve descrição dos aplicativos já presentes no mercado, nas seções 3 e 4 é apresentado o referencial teórico sobre o desenvolvimento do cliente e servidor, respectivamente. A seção 5 mostra como foi feito o desenvolvimento de todo o sistema e a seção 6 traz a conclusão sobre o trabalho desenvolvido.

2 APLICATIVOS PRESENTES NO MERCADO

Para o desenvolvimento deste trabalho foram analisados dois aplicativos já disponíveis para a plataforma Android na *PlayStore*: Guto Viandas e SóMarmitas.

O aplicativo Guto Viandas serve somente como referência para o cardápio do dia, não apresentando uma programação ou a possibilidade do usuário reservar a refeição. Na Figura 1 são mostradas algumas telas deste aplicativo onde estas funcionalidades são exibidas.

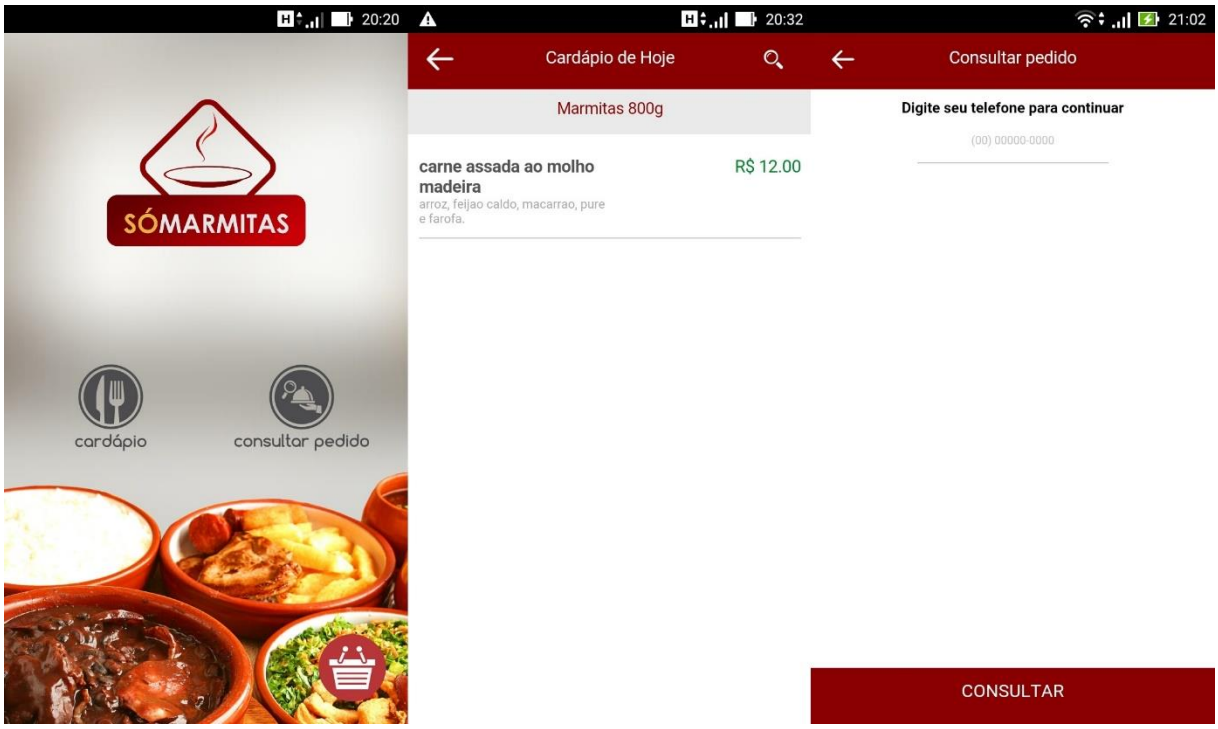
Figura 1 - Principais telas do aplicativo Guto Viandas.



Fonte: Guto Viandas, 2016.

O aplicativo SóMarmitas possui algumas funcionalidades complementares em relação ao Guto Viandas. Ele além de apresentar a opção de cardápio do dia também conta com a funcionalidade de compra de refeição. Entretanto não possui um sistema de agendamento semanal, sendo as telas do aplicativo mostradas na Figura 2.

Figura 2 - Principais telas do aplicativo Só Marmitas.



Fonte: SóMarmitas, 2016.

3 CLIENTE

O cliente foi desenvolvido para a plataforma Android, visto que o mesmo é o atual líder em sistemas operacionais para dispositivos móveis, segundo pesquisa feita pela IDC representada pela Tabela 1.

Tabela 1 – Divisão de mercado na área de SO para Dispositivos Móveis.

PERÍODO	ANDROID	IOS	WINDOWS PHONE	BLACKBERRY OS	OTHERS	PERÍODO
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%	2015Q3
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%	2014Q3
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%	2013Q3
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%	2012Q3

Fonte: IDC, 2015.

3.1 ANDROID

Segundo Amadeo (2014) o Android é um sistema operacional móvel criado inicialmente pela Android, Inc. e tem como base o *kernel* do Linux. Em 2005, ele foi adquirido pela Google e, atualmente, está sendo desenvolvida por desenvolvedores da Google juntamente com a OHA (*Open Handset Alliance*).

O objetivo principal do grupo é definir uma plataforma única e aberta para celulares deixando o usuário satisfeito com o produto final, além de desenvolver uma plataforma moderna e flexível para o desenvolvimento de aplicações corporativas (LECHETA, 2010). O mercado corporativo está se desenvolvendo muito e com isso existe a necessidade das aplicações moveis para agilizar os negócios e integrar as mesmas com o dia-a-dia e seus sistemas *back-end* (LECHETA, 2010).

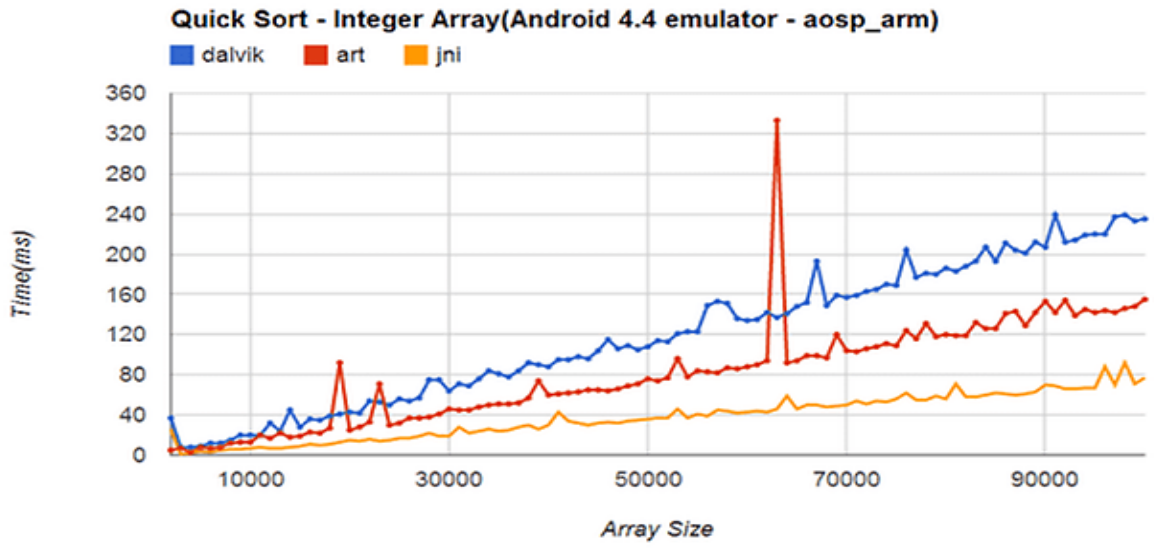
3.1.1 JVM

Para o desenvolvimento de aplicações Android é empregada a linguagem Java, junto de todos seus recursos presentes. Porém, como o Android não possui uma Máquina Virtual Java (Java Virtual Machine, JVM), é utilizada uma máquina virtual chamada Dalvik. Esta máquina virtual é otimizada especificamente para desenvolvimento de aplicações móveis (LECHETA, 2013).

Como consta no site do AndroidPit (2015), Dalvik e Android *Runtime* (ART) são as Maquinas Virtuais (Virtual Machine, VM) presentes no Android. A Dalvik utiliza a técnica de compilação em tempo real, conhecida como *Just In Time* (JIT), fazendo com que a tradução do *bytecode* seja feita durante a execução do aplicativo. Já a máquina virtual ART foi implementada na versão 4.4 do Android e trouxe algumas melhorias, sendo a principal a alteração do modo de tradução do *bytecode.* Anteriormente estra tradução era feita utilizando a técnica JIT, no ART a técnica utilizada é a “antes do tempo” (*ahead-of-time,* AOT). A principal diferença é que esta técnica é executada antes da execução do aplicativo, e não durante como acontece com JIT, trazendo um aumento de velocidade de execução de até 2 vezes em relação ao Dalvik (AndroidPit, 2015). Outra vantagem que vem com a técnica AOT é o menor uso do processador, que reduz o consumo de energia, aumentando a autonomia da bateria. Além destas alterações, a ART recebeu melhorias no sistema de “coleta de

lixo" (*garbage collector*) e um melhor detalhamento no sistema de exceções e reportes de erros. Na Figura 3 são mostrados o tempo de execução de um mesmo aplicativo nas diferentes VMs - Dalvik, ART e a Java *Native Interface* (JNI).

Figura 3 – Comparação de Tempo de Execução entre as VMs.



Fonte: ANDROIDPIT, 2014.

3.1.2 SQLite

O SQLite é o banco de dados utilizado pelo Android. O SQLite é um *software* autocontido, isto é, não necessita de configurações. Também não necessita de um servidor separado, ao contrário dos outros bancos de dados, sendo capaz de ler e escrever diretamente no disco rígido (SQLite, 2016). Como o SQLite é um banco de dados completo, com tabelas, índices, gatilhos e visões que estão contidas em um único arquivo, seu acesso é feito através de um terminal ou pelo prompt de comando.

3.2 JSON

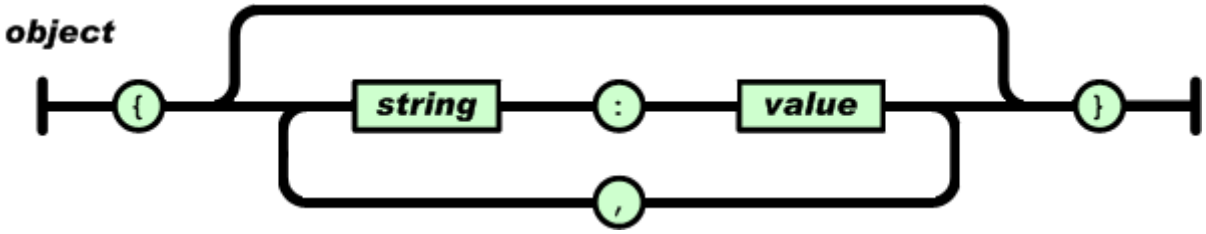
O JSON (2016) é formato de texto completamente independente de linguagem, de simples formatação para troca de dados, fazendo com que a leitura e escrita por humanos seja de fácil compressão e reprodução, e de fácil interpretação para

máquinas. Estas propriedades estas que fazem do JSON um formato ideal para a troca de dados (JSON, 2016).

Ele é constituído em duas estruturas: Objetos e Vetores. O Objeto é uma coleção de pares nome/valor, em outras linguagens pode ser caracterizado como um *object*, *record*, *struct*, dicionário, *hash table*, *keyed list*, ou *arrays* associativas. O vetor é uma lista ordenada de valores, que na maioria das linguagens é caracterizada como um *array*, vetor, lista ou sequência.

No formato JSON o conteúdo é apresentado de duas formas: objeto e *array*. O objeto é constituído de atributos com nome/valor, ficando contido entre chaves “{...}” e cada nome é seguindo por dois pontos “:” e divididos por virgula “,”, como mostrado na Figura 4.

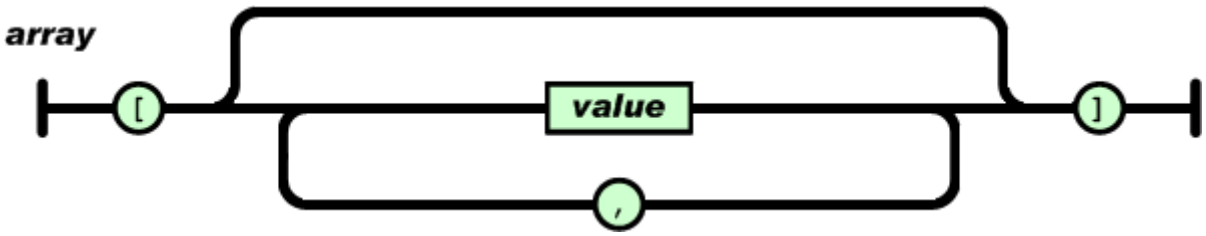
Figura 4 – Definição de um objeto JSON.



Fonte: JSON, 2016.

O *array*, que nada mais é de que uma coleção de valores ordenados, compreendidos entre colchetes “[...]” e com os valores separados por vírgula “,”, como mostrado na Figura 5.

Figura 5 – Definição de um array JSON.



Fonte: JSON, 2016.

4 SERVIDOR

O servidor tem como objetivo armazenar os dados gerados e também fazer o intermédio entre banco de dados e o cliente móvel. O sistema de gerenciamento de banco de dados utilizado neste sistema foi o MySQL. No banco foram utilizadas duas tabelas, sendo elas: *usuarios*, *refeicoes*. Já o web service, responsável por fazer a conversão e autenticação dos dados foi programado em PHP, recebendo uma requisição do aplicativo móvel e retornando um JSON.

4.1 PHP

O PHP é uma linguagem utilizada para o desenvolvimento Web. PHP é a abreviação para "*PHP: Hypertext Preprocessor*", que originalmente significava "*Personal Home Page Tools*", mas conforme seus objetivos foram expandidos, um novo nome foi escolhido pela comunidade. O PHP é utilizado para a criação de *scripts* para o servidor, podendo ser incorporado em HTML ou utilizado como um binário independente (CONVERSE; PARK, 2003). O PHP tem pouco a ver com *layout*, eventos ou qualquer coisa relacionada a aparência de uma página *Web*. De fato, a maior parte do que o PHP realiza é invisível para o usuário final (CONVERSE; PARK, 2003).

4.2 BOOTSTRAP

O Bootstrap é um *framework front-end* que auxilia o desenvolvedor e o designer a construir mais rapidamente uma página web. O *framework* contém configuração para *Cascading Style Sheets* (CSS) global e componentes integrados. Foi inicialmente desenvolvido por funcionários do Twitter e em agosto de 2011 teve seu código fonte liberado como um projeto de código aberto (BOOTSRAP, 2016).

4.3 WEB SERVICE

Web services são utilizados como uma forma de integrar e unificar um sistema, fazendo com que o mesmo possa efetuar uma chamada para um serviço que está

disponível em outro sistema, a fim de obter informações. Estas chamadas podem enviar ou receber informações em diversos formatos, sendo os mais populares, XML e JSON. (LECHETA, 2015).

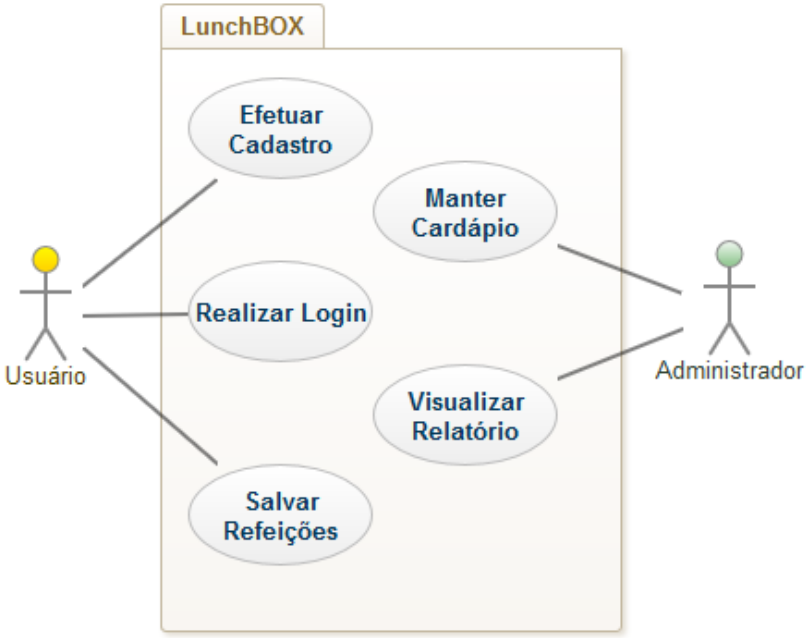
A utilização de web services permite um acesso aos dados padronizado e independente da linguagem de programação. Este conceito trouxe um grande avanço no modo qual os sistemas são construídos, pois permite uma maior flexibilidade (LECHETA, 2015).

Do ponto de vista técnico, web services constituem-se em softwares de baixo acoplamento, reutilizáveis, com componentes feitos para serem facilmente acessados pela internet. Já do ponto de vista conceitual, a utilização da tecnologia representa um modo para integrar tarefas que são partes de um processo através da internet (ABINADER; ABILIO, 2006).

5 IMPLEMENTAÇÃO

O desenvolvimento do projeto foi dividido em duas partes: cliente e servidor. O cliente é o aplicativo desenvolvido para o sistema operacional móvel Android. O servidor é a parte que engloba o banco de dados e o web service, que é responsável por fazer a conexão entre aplicativo e servidor. Na Figura 6 é mostrado o diagrama de casos de uso do sistema.

Figura 6 – Diagrama de Casos de Uso.



Fonte: Do Autor.

O usuário ao acessar o aplicativo, tem acesso a três ações: realizar *login*; efetuar cadastro e salvar as refeições selecionadas para a semana. O administrador tem acesso a um relatório dos usuários e a uma página para fazer alterações no cardápio.

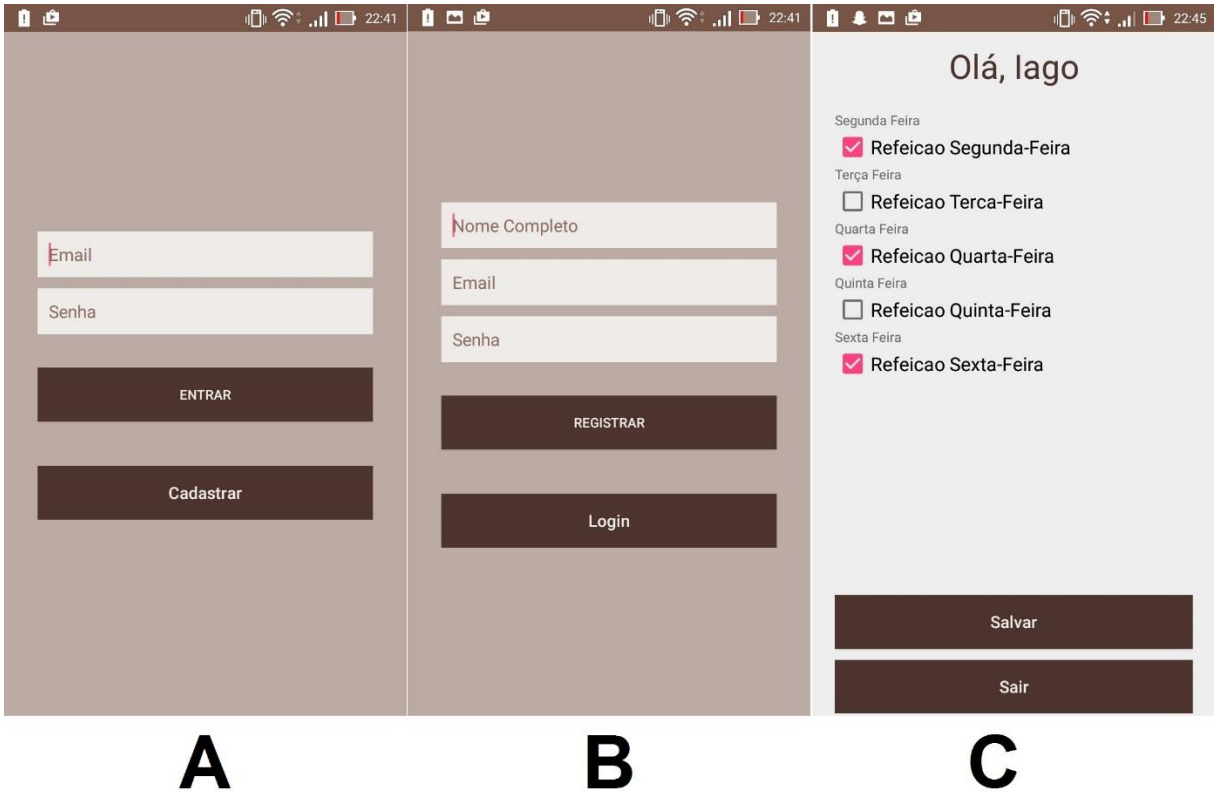
5.1 CLIENTE

O aplicativo foi desenvolvido para a plataforma móvel Android, com o ambiente de desenvolvimento integrado Android Studio. O aplicativo é composto por três telas, conforme mostrado na Figura 7. A parte de cadastro contém um formulário simples para cadastro, sendo os três campos obrigatórios. Após o cadastro, o usuário é enviado para a página de *login* de acesso ao sistema.

Após a execução do login o usuário tem acesso à página principal, onde são mostrados cinco *checkboxes* com as refeições (previamente cadastradas no sistema). O usuário tem a opção de selecionar os dias em que pretende utilizar o serviço e pode

salvar as suas escolhas. Os dados serão então enviados para o servidor, que faz o registro dos dados.

Figura 7 – LunchBOX: Telas do aplicativo.



Fonte: Do Autor.

Como podemos observar na tela de *login* na Figura 7A, existem dois campos para inserção de dados aonde após clicar no botão “entrar” o mesmo envia uma requisição para o servidor. Caso a autenticação seja aceita, o aplicativo recebe um JSON contendo os dados do usuário e salva no banco de dados local do Android, fazendo as próximas consultas diretamente na base local.

Na parte de cadastro devemos preencher três campos, sendo eles: Nome, E-mail e Senha. Após o usuário clicar em “registrar”, uma requisição é enviada para o servidor para verificar se o endereço de e-mail informado já está cadastrado, caso não esteja, uma nova conta é criada, Figura 7B.

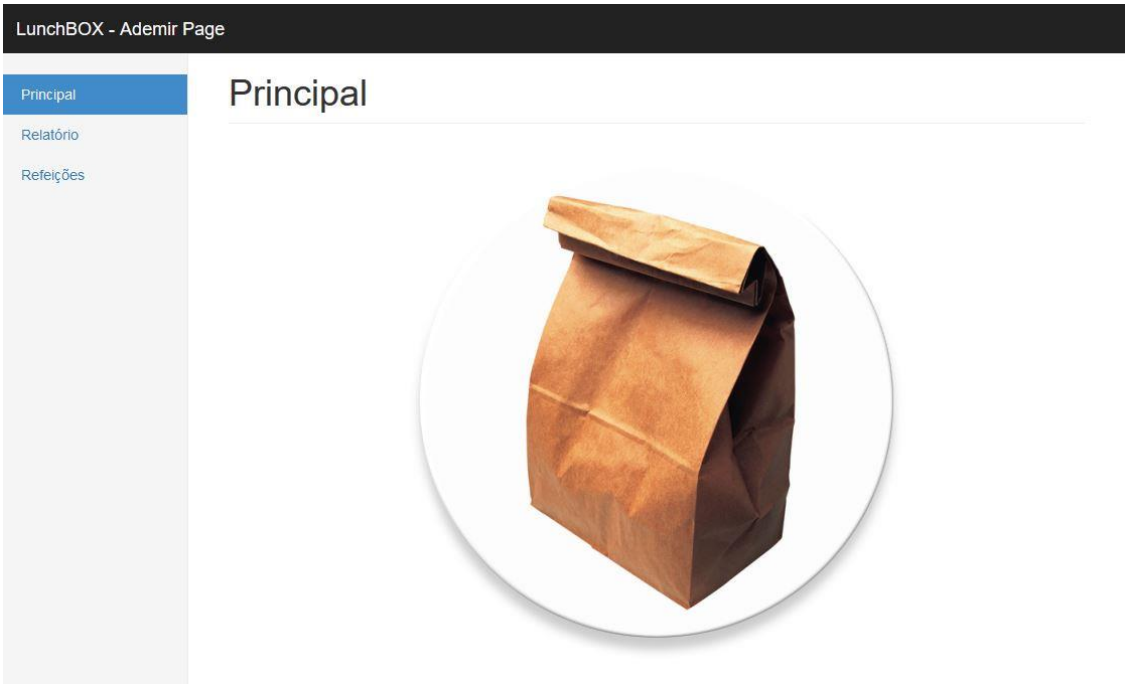
Após o usuário efetuar o *login*, é mostrada a tela principal, mostrada pela Figura 7C, onde são exibidas as refeições cadastradas para a semana, podendo selecionar os dias em que pretende consumir e também salvar sua escolha, enviando uma

requisição para o servidor. Isto faz com que os dias selecionados sejam salvos no banco de dados.

5.2 SERVIDOR

O servidor foi desenvolvido para atender as requisições do aplicativo móvel. O serviço foi desenvolvido em PHP, dividido entre banco de dados e web service. O banco de dados contém as informações referentes aos usuários, as refeições cadastradas e a relação entre usuário e dias desejados pelos usuários. O web service faz a comunicação entre servidor e cliente, recebendo as requisições geradas pelo cliente, tratando e então devolvendo uma resposta em formato JSON. Na Figura 8 é mostrada a tela principal da página do servidor.

Figura 8 – Sistema do Administrador: Página Principal.



Fonte: Do Autor.

Como podemos ver na Figura 8, o sistema do administrador é composto por três páginas: “principal”, “relatório” e “refeições”. A página principal é estática e apenas serve para referência ao sistema. Para o administrador alterar as refeições, o mesmo deve acessar a página de “refeições”. Esta página contém um formulário com todos

os respectivos dias da semana e um botão para “salvar". Assim que acessada, a página carrega as opções já cadastradas no sistema e então o usuário pode alterar o conteúdo cadastrado, conforme mostrado na Figura 9.

Figura 9 – Sistema do Administrador: Página de Refeições.

Fonte: Do Autor.

Na página de relatório, é mostrado um pequeno relatório, contendo os usuários cadastrados e as refeições selecionadas pelo usuário durante a semana. Isto visa auxiliar na gestão e no controle sobre quantidade de insumos necessários para atender a demanda. Na Figura 10 é mostrado um relatório gerado pelo sistema.

Figura 10 - Sistema do Administrador: Página de Relatórios.

Fonte: Do Autor.

Para o web service foi implementado um sistema que o servidor recebe os parâmetros, faz a consulta no banco de dados e então retorna um JSON. Na Figura 11 mostramos o sistema utilizado para a parte de cadastro do aplicativo. Neste exemplo são enviados o nome, e-mail e senha para o web service, após verificar se o e-mail já não se encontra na base de dados, é criado o novo usuário. Recebemos como retorno um JSON contendo as informações cadastradas.

Figura 11 – Código utilizado para o cadastro de usuário

```php
<?php
require_once 'include/Funcoes.php';
$db = new Funcoes();
$resposta = array("error" => FALSE);
if (isset($_POST["nome"]) && isset($_POST["email"]) && isset($_POST["senha"])) {
    $nome = $_POST['name'];
    $email = $_POST['email'];
    $senha = $_POST['senha'];
    if ($db->usuarioExiste($email)) {
        $resposta["error"] = TRUE;
        $resposta["error_msg"] = "Usuário já cadastrado";
        echo json_encode($resposta);
    } else {
        $usuario = $db->armazenarUsuario($nome, $email, $senha);
        if ($user) {
            $resposta["error"] = FALSE;
            $resposta["id"] = $usuario["id"];
            $resposta["usuario"]["name"] = $usuario["name"];
            $resposta["usuario"]["email"] = $usuario["email"];
            $resposta["usuario"]["data_criacao"] = $usuario["data_criacao"];
            echo json_encode($resposta);
        } else {
            $resposta["error"] = TRUE;
            $resposta["error_msg"] = "Erro, não foi possivel criar o usuário.";
            echo json_encode($resposta);
        }
    }
} else {
    $resposta["error"] = TRUE;
    $resposta["error_msg"] = "Preencha todos os campos.";
    echo json_encode($resposta);
}
?>
```

Fonte: Do Autor.

6 CONSIDERAÇÕES FINAIS

O desenvolvimento deste aplicativo visa atender a uma demanda em potencial que não é atendida de maneira adequada pelos sistemas disponíveis. O aplicativo desenvolvido apresenta alguns diferenciais em relação aos aplicativos analisados no

trabalho. O principal diferencial do aplicativo desenvolvido em relação aos disponíveis é o sistema de agendamento semanal. O sistema gera um relatório para o proprietário do estabelecimento contendo a relação das pessoas e os dias selecionados para receber a refeição. Nos testes o aplicativo funcionou de forma regular.

Para o futuro serão adicionadas melhorias para aperfeiçoar a usabilidade do sistema e também para aumentar sua funcionalidade. Para melhorar a usabilidade serão realizadas alterações no *design* do aplicativo e da página de relatórios do servidor. No aplicativo será disponibilizada a opção para que os usuários possam escolher quais alimentos desejam receber em sua refeição. Também será desenvolvido um sistema para identificação dos recipientes por código de barras.

ABSTRACT

This work developed a system for managing delivery of packed lunch. The main motivation for creating this system was a demand. Since the options available in the market lack of features and availability. It was developed an application for the Android mobile platform and a server. The application was made for the user to access weekly menu options and select which days they wanted to receive meals. The server receives requests from the application and generates a report for the administrator, containing selected days in which the user wants to receive meals. In tests the system worked properly.

Keywords: Android. Mobile App. Packed Lunch. Web Service.

REFERÊNCIAS

ABINADER, Abilio; DUEIRE, Rafael; *Web Services em Java.* Rio de Janeiro, Brasport, 2006.

AMADEO, Ron. *The history of Android*. Disponível em: <http://arstechnica.com/gadgets/2014/06/building-android-a-40000-word-history-of-googles-mobile-os/>. Acesso em: 02 jun. 2016.

BOOTSTRAP. *What is Bootstrap? We Have the Answers.* Disponível em < https://bootstrapbay.com/blog/what-is-bootstrap/>. Acesso em: 11 ma. 2016.

CONVERSE, Tim; PARK, Joyce. *PHP a Bíblia.* Rio de Janeiro, Elsevier, 2003.

GUTO VIANDAS. *Guto Viandas Cardápio do Dia!* Disponível em: < https://play.google.com/store/apps/details?id=br.com.gutoviandas>. Acesso em: 10 mai. 2016.

IDC. *Smartphone OS Market Share, Q2 2015*. Disponível em <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. Acesso em: 11 ma. 2016.

JSON. *Introdução ao JSON*. Disponível em: <http://www.json.org/json-pt.html>. Acesso em: 10 jun. 2016.

LECHETA, Ricardo R. *Google Android - 3ª Edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK.* São Paulo, Novatec Editora, 2013.

LECHETA, Ricardo R. *Web Services RESTful: Aprenda a criar web services RESTful em Java na nuvem do Google*. São Paulo, Novatec Editora, 2015.

MEIRELESS, Fernando S. *27ª Pesquisa Anual do Uso de TI, 2016.* Disponível em <http://eaesp.fgvsp.br/sites/eaesp.fgvsp.br/files/pesti2016gvciappt.pdf>. Acesso em: 28 jun. 2016.

SÓ MARMITAS. *Só Marmitas*. Disponível em: < https://play.google.com/store/apps/details?id=com.somarmitas.pedido.online>. Acesso em: 9 mai. 2016.

SQLITE. About SQLite. Disponível em: <http://www.sqlite.org/about.html>. Acesso em: 02 jun. 2016.