

APLICATIVO ANDROID INTEGRADO AO BANCO DE DADOS EXTERNO VIA WEB SERVICE: UM ESTUDO DE CASO

SILVIO LUIZ DOS SANTOS CUSTODIO

ORIENTADOR: PROF. CARLOS HENRIQUE LOUREIRO FEICHAS

COORIENTADOR: PROF. ANÍBAL EVARISTO FERNANDES

Resumo

O presente projeto trata-se de um aplicativo mobile que tem como principal funcionalidade fazer a interação com o banco de dados usando a tecnologia que envolve web service. Projeto voltado para o nicho de comércio alimentício como bares, restaurantes, padarias e fast food. Seguindo a necessidade da criação de novos serviços que se integrem ao sistema principal de uma forma estratégica a fim de agregar um diferencial diante de outros concorrentes do mesmo segmento de softwares o aplicativo disponibilizará funções de exibir cardápio, receber pedidos, informar o usuário o tempo para que seu pedido possa estar pronto e futuramente receber pagamentos via o próprio aplicativo. Serviço esse com o propósito de facilitar a interação com o cliente, pois evitará que o mesmo enfrente longas filas, problema esse que por muitas vezes afastam os clientes impactando negativamente nas vendas. Aplicativo elaborado para ser um diferencial de mercado, contribuindo para a fidelização de clientes em tempos de concorrência e disputas acirradas, onde a criatividade e inovação são ferramentas necessárias para qualquer tipo de negócio.

Palavras-chave: Aplicativo. Clientes. Pedido.

Abstract

This project is a mobile application done in Android is Google's operating system for mobile devices based on Linux. that will make the interaction with the database via Web service. This project is intended to serve a specific market niche, such as bars, restaurants, bakeries and fast food. The need for the creation of new services that integrate the main business management system, the well-known ERP (Enterprise Resource Planning), who perform a series of activities managed by a software or by people. These services act strategically in order to gain an advantage over other competitors in the same segment. Then came the idea of creating an application that can be easily integrated to any ERP system, offering the functions to show the menu, take orders, tell the user the time so that your request can be ready and receive payments through the application itself. facilitating customer purchase request, avoiding long queues that often away customers, hurting the company's sales increase. Application that was called "Fura Fila", being designed to be a market differentiator, contributing to customer loyalty in times of strong competitions, where creativity and innovation are the tools needed for any business.

Keywords: Application. Client. Orders.

Introdução

Os clientes no mundo de hoje estão em uma dinâmica diferenciada. Eles usam celulares, tablets ou outros dispositivos móveis que têm todas as informações que precisam. É por isso que os aplicativos são muito importantes no mercado atual.

Seus serviços fornecem informações e conteúdo de modo simplificado e ágil, que permitem aos clientes terem todas as informações na ponta dos dedos e acessível em várias plataformas. Não importa qual o negócio, um aplicativo móvel pode ajudar a obter e reter clientes e isso é um fato. É notório que em primeiro lugar os clientes vão procurar por um produto ou serviço online e se a empresa está online e tem um aplicativo que os usuários podem baixar para seus dispositivos, a empresa vai fazer realmente boa impressão. À primeira vista eles serão capazes de ver, abrir, interagir e comprar algo da empresa.

Podemos citar alguns dos benefícios de aplicativos: fidelização do cliente, reforço da marca, aumento da visibilidade e aumento da acessibilidade. Além de poder receber notificações de eventos especiais, acesso sobre os serviços oferecidos, realizar compras pelo aplicativo do próprio estabelecimento, entre outras vantagens.

O objetivo geral desse projeto é desenvolver um aplicativo móvel com foco na interação com o banco de dados podendo também interagir com qualquer Sistema ERP principal já existente, interação essa feita através de um Web Service, que é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis, no nosso projeto permitirá o compartilhamento de serviços de banco de dados, pedidos e futuramente a realização de pagamento pelo próprio aplicativo, tendo como objetivo específico os seguintes itens:

- a) Apresentar os elementos teóricos da economia, dos negócios e da inovação tecnológica para a criação de um aplicativo que atue como um fator diferencial e agregue valor ao sistema de ERP principal;
- b) Identificar as oportunidades de implementação de um software com foco na prestação de serviços para clientes do ramo alimentício, através dos dispositivos móveis;
- c) Idealizar um projeto inovador por meio da utilização de métodos adaptados ao ambiente econômico, bem como verificar sua viabilidade em termos de custos e retorno.

A metodologia e os procedimentos utilizados para a pesquisa e desenvolvimento deste trabalho será de caráter exploratório, oportunidade em que será possível utilizar um conjunto de ações para o alcance dos resultados. Pretende-se utilizar como base central os conceitos de

planos de negócios e empreendedorismo. Por outro lado, será desenvolvida uma pesquisa bibliográfica e documental, buscando alinhar os fatores teóricos ao projeto, como forma de dar suporte às suposições e previsões de viabilidade prática.

Uma justificativa plausível para esse projeto é que os aplicativos estão cada vez mais populares e se tornaram muito importantes nos últimos anos, eles passam a ter papel de suma importância na vida dos usuários. Aplicativos mobile são softwares que desempenham objetivos específicos em smartphones e tablets. É possível acessá-los por meio das “lojas de aplicativos”, como a App Store, Play Store, BlackBerry App World, Ovi Store, entre outros.

Com o surgimento da Internet os fundamentos econômicos, empresariais e tecnológicos da economia tradicional são desafiados, renovando o sistema e fazendo com que os empreendedores desenvolvam novos modelos para os negócios dentro desta nova economia (TURBAN et al., 2004).

A necessidade da integração entre aplicações, a utilização unificada de processos encontrados em diferentes sistemas e escritos em diferentes linguagens são alguns exemplos de carências encontradas em empresas nos dias de hoje. A fim de sanar estas questões, a tecnologia dos Web Services foi criada, permitindo assim, disponibilizar formas de integrar sistemas distintos, modularizar serviços e capacitar a integração e consumo de informações (Menéndez 2002).

De Sordi (2003) discorre que, buscando soluções eficazes, empresas de software desenvolveram um conjunto de sistemas de informação administrativos e financeiros integrados, conhecidos como Enterprise Resource Planning (ou simplesmente ERP), que pode ser traduzido livremente como Planejamento de Recursos Empresariais.

O ERP pode ser utilizado na modalidade de negócios conhecida como application service provider (ou apenas ASP), um provedor de serviços de aplicação, que trata a solução de sistema de informação na forma de prestação de serviço (DE SORDI, 2003).

Neste sentido, este modelo aplica-se indiretamente aos softwares de dispositivos móveis (aplicativos), visto sua semelhante funcionalidade. Igualmente, o ERP pode ser tido como utilizável, até certo ponto, nos novos modelos de negócio da economia digital da década vigente.

Acredita-se que esse projeto se torna relevante podendo se estender à pesquisa acadêmica e fora dela, pois aborda a integração de sistemas, explorando uma realidade de mercado que de uma forma direta ou indireta impõe sobre qualquer produto ou serviço, sugerindo a inovação, diferencial de mercado e trabalho em equipe.

1. Revisão bibliográfica/Fundamentação Teórica

O referencial teórico é o responsável por fornecer as bases que sustentam a análise das teorias sociais e econômicas intrínsecas ao empreendedorismo e ao contexto da inovação.

1.1 Tendências em Computação Móvel

Como foi possível observar o campo da Computação Móvel, vem evoluindo de forma muito rápida e tem crescido substancialmente a cada ano. Pesquisas demonstram que o futuro promete ser ainda mais promissor. Segundo um estudo publicado pela eWeek, a indústria de dispositivos móveis gerou 2 milhões de empregos no mundo em 2015, elevando o número de pessoas empregadas na indústria sem fio para 10 milhões de pessoas. Onde a receita das empresas móveis cresceu de US\$1.5 trilhões em 2011 para US\$ 1.9 trilhões em 2015.

A indústria da computação móvel até 2015 investiu 793 bilhões de dólares em capital e contribuir com 2.7 trilhões de dólares para o financiamento público em todo mundo. Além, do impacto econômico global, esta pesquisa também destacou que a indústria móvel é um fator significativo no crescimento das economias locais.

1.2 Crescimento do uso de aplicativos

Os aplicativos de compras e de utilitários em geral puxaram o crescimento do uso de smartphones em 2014 no mundo. É o que atesta um levantamento feito pela empresa de análise Flurry Inc., empresa que monitora mais de 600 mil apps dos sistemas Android e iOS instalados em mais de 1 bilhão de smartphones ao redor do mundo.

A companhia mede o uso de um aplicativo pela sua quantidade de sessões, ou seja, de vezes em que ele é aberto pelo usuário em seu smartphone. Os aplicativos da categoria "estilo de vida e shopping" foram os campeões de crescimento, com aumento de 174% no número de sessões em 2014. Somente em Android, apps de shopping cresceram sozinhos 220%. Em segundo lugar veio a categoria de utilitários e produtividade, com crescimento de 121%. Em seguida: mensagens e redes sociais (103%); saúde e bem-estar (89%); viagem (89%); esportes (74%); notícias e revistas (49%); entretenimento (33%); e jogos (30%).

Na média, houve um aumento de 76% no uso de aplicativos móveis no mundo, na rede da Flurry. Em 2014 a empresa registrou 2,079 trilhões de sessões de apps móveis globalmente. Seu recorde em um só dia foi em 31 de dezembro, com 8,5 bilhões de sessões, por conta do grande uso de apps de comunicação e compartilhamento de conteúdo na virada do ano.

1.3 Arquitetura Android

O Kernel em face da diversidade de arquiteturas de dispositivos móveis, assim como é em computadores e outros dispositivos eletrônicos, para que um sistema consiga operar com diferentes tipos de hardware, se fazem necessários os sistemas operacionais.

Segundo TANENBAUM (2003), um sistema operacional deve ser capaz de gerenciar o processador, memória e outros dispositivos de entrada e saída, além de fornecer, aos programas de usuário, uma interface mais simplificada com o hardware.

Bibliotecas (Libraries): Atividades como a abertura de arquivos, a realização de cálculos aritméticos, exibição de imagens, entre outras, são atividades básicas, que acabam não sendo o foco principal do desenvolvedor.

Execução (Runtime): Da mesma forma que com o Java, uma aplicação Android é interpretada pelo sistema. A diferença é que o Android gera códigos Dalvik Executáveis (.dex), e não os byte-code (.class) do Java, tais código são interpretados pela Máquina Virtual Dalvik (MVD) (PROJECT, 2012).

Após a compilação, todos os arquivos dex e outros recursos utilizados pela aplicação (exemplo: imagens, sons, etc.), são compactados em um arquivo do tipo .apk (Android Package File), sendo este arquivo, a aplicação finalizada e pronta para ser distribuída e instalada em qualquer dispositivo com Android, (LECHETA, 2009).

A MVD também é incluída no SDK do Android, onde transformam os códigos das classes Java (.class) em códigos Dalvik executáveis (.dex), que posteriormente serão executado pelo emulador Android (BORNSTEIN, 2008).

O Android possui um conjunto de bibliotecas, disponíveis para a criação de seus aplicativos como a System C Library, Media Libraries, Surface Manager, Lib Webcore, SGL, 3D libraries, Freetype e SQLite. Tais bibliotecas permitem a manipulação de vídeos, imagens, sons animações, banco de dados, etc. Conforme se observa na Figura 1, a arquitetura Android é dividida em Kernel, Runtime, Bibliotecas, framework e aplicativos.

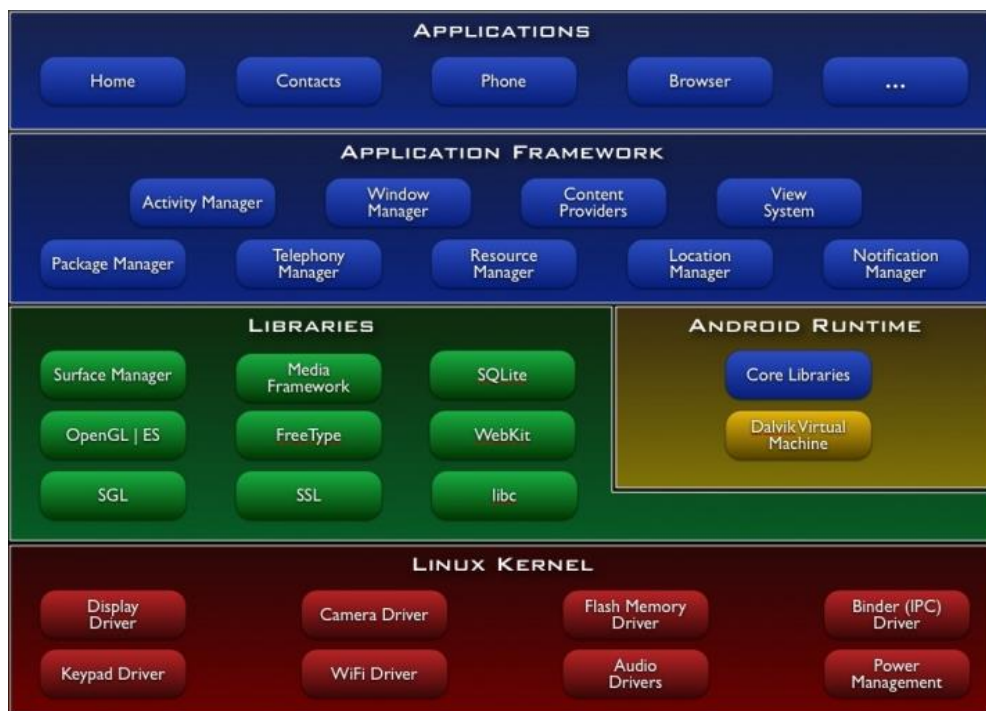


Figura 1: Arquitetura Android

Fonte: <http://developer.android.com/images/system-architecture.jpg>

No projeto inicialmente será utilizado o SQLite, que é um sistema gerenciador de banco de dados (SGBD) embutido no Android, de código aberto, que não possui servidor de processos separado.

O SQLite é um gerenciador compacto e disponível para várias plataformas, foi desenvolvido para ocupar o mínimo de memória possível, mas com grande quantidade de recursos, permite criação de várias tabelas, triggers, índices e views, tudo em apenas um arquivo. É indicado para dispositivos que possuem limitações de memória e processamento, como os celulares, por exemplo. (LECHETA, 2009)

1.4 Android Studio

O Android Studio é um novo ambiente de desenvolvimento Android com base no IntelliJ IDEA. Semelhante ao Eclipse com o plugin ADT, Android Studio fornece ferramentas de desenvolvimento integradas Android para desenvolvimento e depuração. Foi anunciado em 16 de Maio de 2013 na conferência Google I/O. Em junho de 2013, ele foi disponível em forma de beta para os usuários. Por ser um produto desenvolvido pela própria empresa responsável pelo Android no caso o Google, essa IDE consegue reunir ferramentas de criação que agilizam no processo criativo, tornando a usabilidade muito mais ágil, fácil e intuitivo.

A figura 2, a seguir representa a IDE Android Studio, que está sendo utilizada para o desenvolvimento do aplicativo.

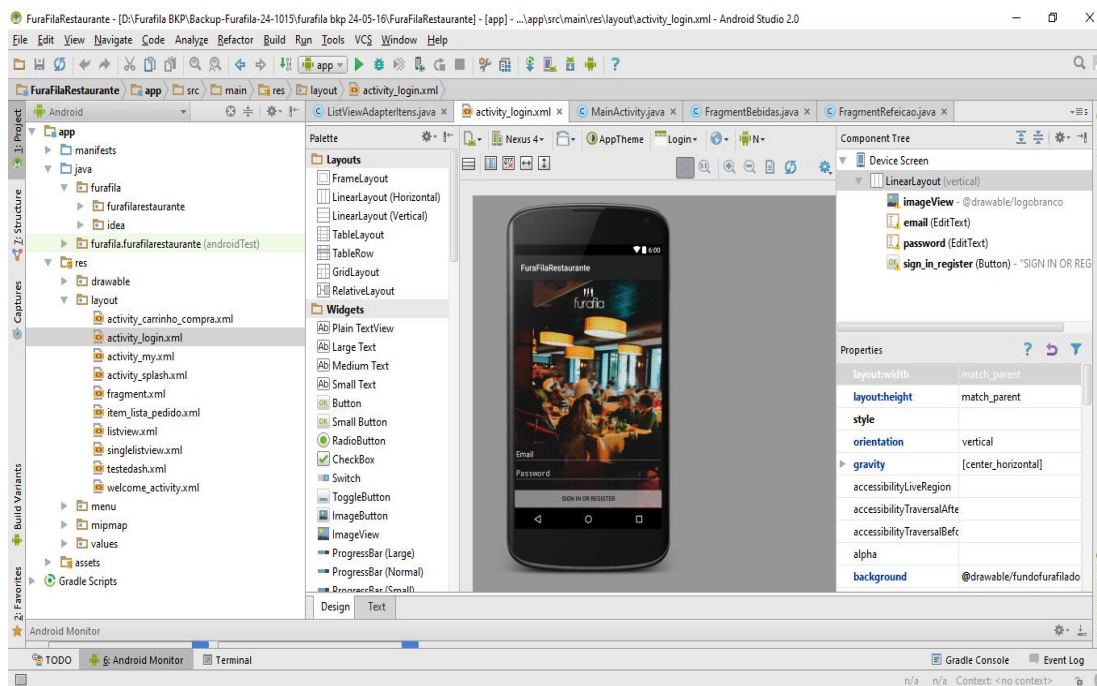


Figura 2: IDE Android Studio

Fonte: © 2000-2016 Google Android Studio 2.0

1.5 JAVA

Para começar a programar para o Android é necessário conhecer (ou aprender) a linguagem Java, pois foi a linguagem escolhida pela Google™.

Java é uma linguagem de programação orientada a objeto e desenvolvida pela Sun Microsystems na década de 90, que atualmente, pertence a Oracle. A característica mais marcante dessa linguagem é que programas criados nela não são compilados em código nativo da plataforma. Programas em Java são compilados para um bytecode, que é executado por uma máquina virtual, o que permite aos desenvolvedores criarem um programa uma única vez e depois executar este em qualquer uma das plataformas suportadas pela tecnologia.

JDK - como o Android faz uso da tecnologia Java, é necessário ter o Java Development Kit instalado. O JDK, abreviação para Java Development Kit, é um conjunto de utilitários cuja a finalidade é a permissão para criação de jogos e programas para a plataforma Java. Este pacote é disponibilizado pela Oracle, e nele vem todo o ambiente necessário para a criação e execução dos aplicativos java.

1.6 MySQL

O MySQL é um sistema de gestão de bases de dados relacionais, suporta a Linguagem de Consulta Estruturada ou SQL (Structured Query Language), é open source e um dos Sistema de Gerenciamento de Banco de Dados (SGBD) para utilização profissional mais utilizado (conta com mais de 5 milhões de instalações ativas) e mais conhecido a nível mundial.

Conforme Neves e Ruas (2005, p. 21) comentam: “o MySQL foi desenvolvido e é disponibilizado pela empresa MySQL AB Limited Company, que atualmente vende um conjunto de serviços e produtos relacionados com a tecnologia MySQL. Na lista dos principais clientes deste SGBD constam nomes como: Alcatel, AOL, The Associated Press, Caterpillar, Cox Communications, DaimlerChrysler, Dow Jones, EarthLink, Enercon, Ericsson, Google, Hoover's Online, Hewlett-Packard, Lucent, Lufthansa, NASA, Nortel, NYSE, Omaha Steaks, Sabre Holdings, Siemens, Suzuki, Texas Instruments, Time Inc., UPS e US Census Bureau”.

1.7 Web Service

Web Service é um conjunto de padrões que possibilita a interoperabilidade entre sistemas heterogêneos. Esses padrões definem os protocolos utilizados para comunicação e formatos de interface para especificação e contratos de serviços (Josuttis, 2007).

A arquitetura dos Web Services é constituída na interação de três personagens distintos: Provedor de Serviços, Consumidor de Serviços e o Registro dos Serviços. A interação destes personagens envolve as operações de publicar, pesquisar e fazer a ligação dos serviços. Abaixo é definida cada função destes personagens:

- **Provedor de Serviços:** É a entidade (personagem) que cria os Web Services, que disponibiliza o serviço para que alguém possa utilizá-lo. A forma para que ocorra esse acesso é descrever o serviço em um formato padrão, compreensível para qualquer um que precise usar esse serviço e também publicar as características sobre o seu serviço em um registro central disponível.
- **Consumidor de serviços:** A utilização de um Web Services que foi criado por um provedor de serviços é chamada de consumidor de serviços. Este conhece a funcionalidade do serviço, que com a descrição disponibilizada pelo provedor de 17 serviços, através de uma pesquisa sobre o registro publicado obtém os detalhes, também pode obter os mecanismos para a ligação com este Web Services.

- **Registros dos Serviços:** Podemos dizer que um registro dos serviços é a localização central onde o provedor de serviços pode relacionar seus serviços onde os consumidores dos serviços podem pesquisá-los e encontrá-los. Ele contém as informações como os serviços que ela oferece a descrição técnica dos serviços e os detalhes da empresa.

O Sistema Android e o banco de dados MySQL não possuem propriedades que permitam a comunicação entre eles diretamente, fazendo uma analogia seria como duas pessoas de nacionalidades diferentes tentando se comunicar cada um no seu idioma nativo, nesse caso precisaríamos de um intérprete.

No nosso caso como ilustrado na figura 3, esse intérprete será o PHP, que irá fazer toda a comunicação. Por exemplo, se o usuário criar um novo pedido para o aplicativo dentro do dispositivo Android, o pedido necessitará de alguns campos de entrada, como nome de usuário, senha, para validar o usuário, depois ele vai ter três opções de tela, que irá preencher de primeiramente pegará qual tabela que é (1,2 ou 3) e qual a posição de cada item dentro dessa tabela. No nosso projeto o próprio código ficará encarregado de fazer isso cabendo ao usuário apenas escolher o item e selecionar o ícone adicionar, essa informação será passado para o nosso serviço web PHP. O serviço web irá se conectar ao banco de dados MySQL, e encontrar a tabela "tborder" e inserir uma nova linha no banco de dados MySQL com a informação que foi enviada para o nosso dispositivo Android. Uma vez que a nossa linha é criada, o Web Service mostrará alguns dados JSON que retornará se os dados foram adicionados com êxito ou não. Por último, o nosso dispositivo Android irá analisar esses dados JSON e exibir uma mensagem de "sucesso" ou uma mensagem "Falha, por favor tente novamente."

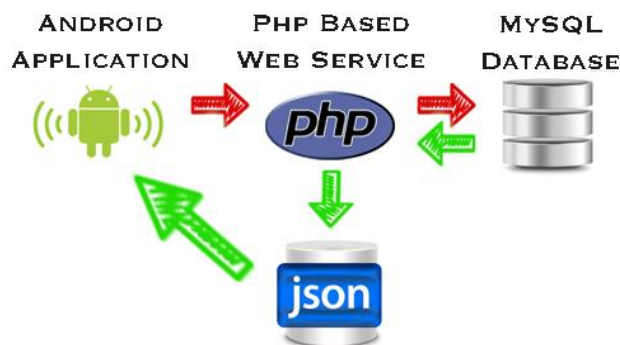


Figura 3: Conexão Remota Aplicativo Android – MySQL via Web Service

Fonte: <http://dchabmztumu0g.cloudfront.net/wp-content/uploads/2013/05/remotejson.png>

1.8 JSON - JavaScript Object Notation

JSON é um formato de texto leve e independente de plataforma, derivado dos literais de Java Script, usado para intercâmbio e serialização de dados. Seu modelo suporta quatro tipos primitivos de variáveis (inteiro, string, booleano e nulo), e dois tipos de variáveis estruturadas (arrays e objetos) (JSON, 1999). Simplicidade é a maior vantagem de JSON, que vêm se mostrando mais eficiente em aplicações Web Ajax do que XML (CROCKFORD, 2008).

A estrutura formada por um objeto JSON é similar a um dicionário, utilizando a estrutura chave e valor. A chave é sempre uma string, e precisa estar entre aspas duplas. Já o valor pode ser qualquer um dos tipos primitivos e de variáveis estruturadas e um sistema Cliente-Servidor no qual é utilizado o protocolo REST com JSON.

O REST é o acrônimo para Representational State Transfer (Transferência representacional de estado), é cada vez mais usado como alternativa ao “já antigo” SOAP onde que a principal crítica a este é a burocracia, algo que REST possui em uma escala muito menor.

REST é baseado no design do protocolo HTTP, que já possui diversos mecanismos embutidos para representar recursos como código de status, representação de tipos de conteúdo, cabeçalhos, etc.

O principal nesta arquitetura são as URLs do sistema e os resources (resource é um recurso, entidade). Ele aproveita os métodos HTTP para se comunicar, que são:

GET: Solicita a representação de um determinado recurso. É definido como um método seguro e não deve ser usado para disparar uma ação (remover um usuário, por exemplo);

POST: As informações enviadas no corpo (body) da requisição são utilizadas para criar um novo recurso. Também é responsável por fazer processamentos que não são diretamente relacionados a um recurso.

DELETE: Remove um recurso. Deve retornar o status 204 caso não exista nenhum recurso para a URI especificada.

PUT: Atualiza um recurso na URL especificada. Caso o recurso não exista, ele pode criar um. A principal diferença entre POST e PUT é que o primeiro pode lidar não somente com recursos, mas também pode fazer processamento de informações.

1.9 PHP

O PHP (um acrônimo recursivo para "PHP: Hypertext Preprocessor", originalmente Personal Home Page) é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na World Wide Web.

É importante ressaltar algumas características principais do PHP, tais como:

- Código fonte aberto;
- Linguagem interpretada;
- Multiplataforma;
- Comunidade de suporte e atualização constante;
- PHP é case sensitive, ou seja, as variáveis \$teste e \$TESTE são diferentes.

De acordo com Rocha (2007, p.19) o PHP também tem como uma das características mais importantes o suporte a um grande número de bancos de dados, como dBase, Interbase, mSQL, MySQL, Oracle, Sybase, PostgreSQL, MSSQL, Fontes ODBC e vários outros.

2. Materiais e Métodos

Neste trabalho foi realizada extensa revisão bibliográfica, tanto em revistas eletrônicas, livros e site do assunto na Internet. Foi implementado posteriormente ao artigo um protótipo (aplicativo) simplificado para evidenciar as características de como foi possível o compartilhamento e troca de dados em diferentes plataformas via Web Service.

O sistema escolhido para a criação desse aplicativo foi o Android. O Android é um sistema operacional baseado em um kernel Linux, o qual provê os mesmos serviços de segurança, gerenciamento de memória e de processos e todas outras características do kernel. Sua estrutura é formada pelas camadas: Linux Kernel, Bibliotecas, Ferramentas de Runtime do Android, Estrutura das Aplicações e Aplicações (DEV GUIDE, 2011).

Para desenvolvedores já habituados a programação utilizando a linguagem Java, o processo de criação de um aplicativo utilizando a plataforma Android se dá de forma até mesmo simplificada, bastando entender alguns conceitos de como o mesmo funciona.

Utilizando Java é possível, através do conceito de orientação a objetos referente ao reuso de código, transportar uma aplicação totalmente desktop para uma aplicação móvel com pequenos ajustes desde que respeitadas as limitações de hardware. A plataforma

Android tem potencial para derrubar barreiras em direção ao sucesso em desenvolvimento e vendas de aplicativos de telefonia móvel devido sua portabilidade e facilidade de uso e implementação dos produtos para o mesmo (ROGERS et al., 2009).

Nesse projeto também se utilizará um banco de dados externo MySQL, que é um banco de dados relacional gratuito, eficiente e otimizado para aplicações. O Android não suporta MySQL, mas é possível utilizar um banco de dados online, desde que se possua: Webservice e Cliente que é a parte que acessa o serviço, no nosso caso o aplicativo.

Web Service é a mais recente evolução nos padrões de desenvolvimento de aplicações distribuídas permitindo que aplicações cooperem facilmente e compartilhem informações e dados umas com as outras. Espera-se que esta evolução altere a forma como as aplicações são construídas e desenvolvidas, como a informação é apresentada e compartilhada e como software é comprado e vendido. O Web Service deverá fazer a requisição no banco de dados e retornar uma resposta para ser processada, no nosso projeto feito pelo JSON (JavaScript Object Notation) que é um formato de intercâmbio de dados leve. Atualmente muito utilizado, por que além de sua leveza é muito simples e compacto, trazendo assim uma solução prática para a conversa entre o aplicativo e o webservice, que nesse projeto está sendo feito através de um método webservice escrito em PHP (um acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML.

Para fins de testes foi instalado um servidor local chamado WAMP, é acrônimo para o Windows, Apache, MySQL e PHP, Perl, Python. É um pequeno servidor que é executado no sistema operacional Windows. O WAMP inclui um administrador de banco de dados e de dados phpMyAdmin. Usando isto, pode-se criar um novo banco de dados, tabelas e inserir dados. O WAMP Server oferece a possibilidade de desenvolvedores para fazer a aplicação Web localmente com o sistema operacional. WAMP Server está disponível gratuitamente e é open source.

3.8 Estudo de caso

A aplicação desenvolvida teve como base inicial ser parte integrante do sistema ERP SOLID, sistema esse criado para gestão de pequenos e médios negócios, na área de bares e alimentação (restaurante e fast-food) onde seja necessário a disponibilização de um cardápio para os clientes. Mas com a evolução do projeto do aplicativo e a descontinuação do sistema

ERP SOLID, decidiu-se preparar o aplicativo para interagir com qualquer sistema ERP que use um banco de dados para guardar informações do cardápio oferecido, ampliando assim o alcance do aplicativo.

Primeiramente o aplicativo terá a função de se integrar com o as tabelas cardápio e clientes, a primeira é para que o aplicativo possa listar todos os produtos oferecidos pelo estabelecimento, e a segunda para que usuário seja identificado especificando o responsável por determinado pedido feito via celular, evitando assim a espera do cliente pelo atendimento de um garçon para anotar o pedido, diminuindo o tempo de espera para que esse pedido esteja pronto.

O aplicativo possui um banco de dados interno, o SQLite nativo do sistema Android onde no nosso projeto para fins de testes foi armazenado um cardápio ilustrativo, também possui acesso externo com o banco de dados do estabelecimento, que poderá substituir o cardápio do banco de dados interno, via Web Service, que ficará responsável por fazer as requisições para o banco de dados central de forma segura, mantendo a integridade do mesmo.

Esse acesso ao banco de dados externo será necessário para o fechamento do pedido, login, cálculo do tempo estimado para o preparo do pedido, consequentemente baixa no estoque, e futuramente pagamento via o próprio aplicativo.

Durante o processo de desenvolvimento, iniciou-se nas abstrações dos dados a partir de um levantamento de requisitos para analisar as funcionalidades e como a mesma iria gerir esse processo de ocorrências de exceções na aplicação. Para isto desenvolveu-se dois diagramas UML (Unified Modeling Language), um de Caso de Uso (Figura 4) e outro de diagrama de classes (Figura 5).

O Caso de Uso aponta as principais funcionalidades que a aplicação irá disponibilizar ao ator (cliente) durante seu uso que é a inserção de itens dentro de uma lista de pedidos e também do ator (administrador), que poderá manipular o banco de dados externo.

O diagrama de classes é considerado por muitos autores como o mais importante e o mais utilizado diagrama da UML. Seu principal enfoque está em permitir a visualização das classes que irão compor o sistema com seus respectivos atributos e métodos, bem como em demonstrar como as classes do sistema se relacionam, se complementam e transmitem informações entre si.

Diagrama de Caso de Uso

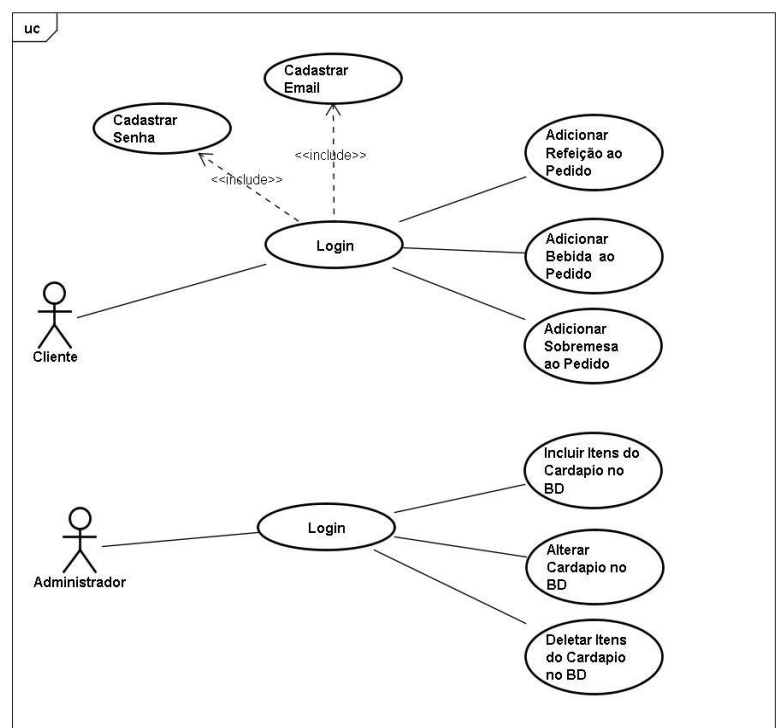


Figura 4: Caso de Uso – Usuário no Sistema

Fonte: O Autor

Diagrama de Classe

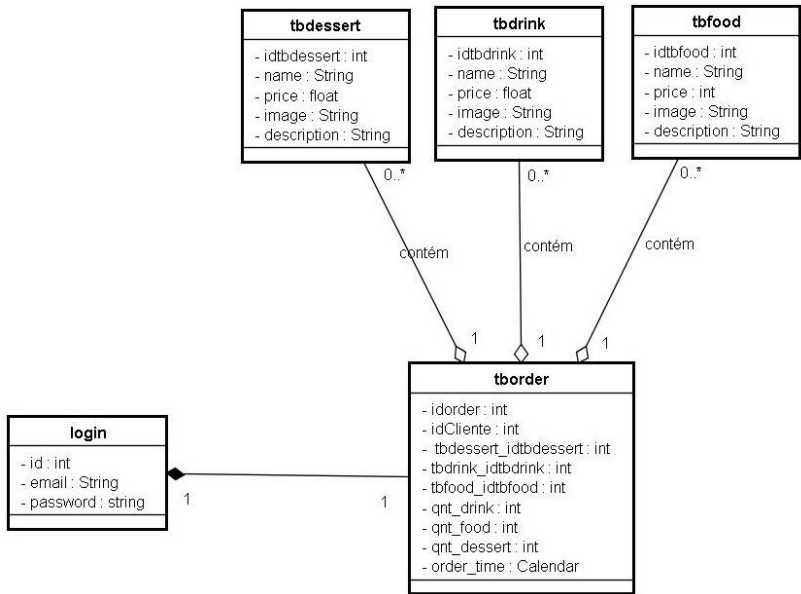


Figura 5: Caso de Uso – Usuário no Sistema

Fonte: O Autor

Também foi criado o Modelo Entidade Relacionamento (Figura 6) também chamado Modelo ER, ou simplesmente MER, como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).

Modelo Entidade Relacionamento (MER)

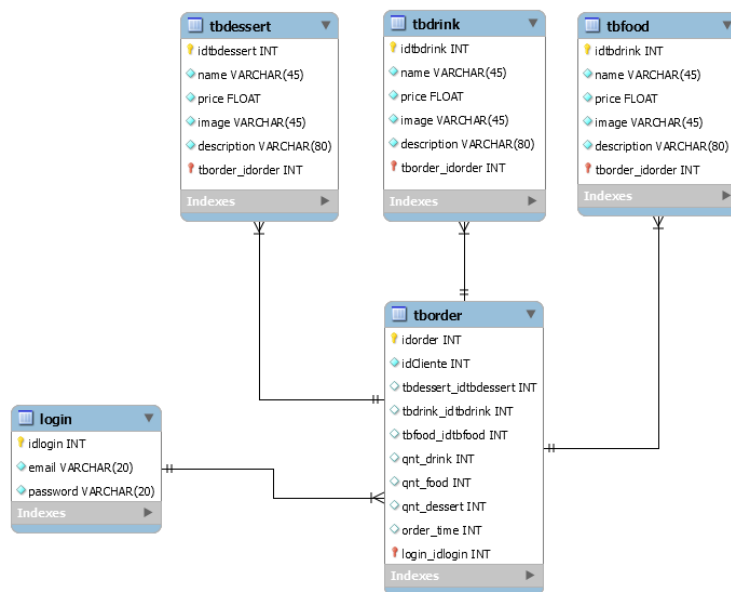


Figura 6: Modelo Entidade Relacionamento – Banco de Dados

Fonte: O Autor

4. Discussão dos Resultados

Primeiramente foi pensado em um logotipo que representasse a ideia do propósito do aplicativo desenvolvido. A figura 7, a seguir, traz a representação da logomarca, enfatizando o conceito do projeto em si:



Figura 7: Logomarca

Fonte: Autor

A Figura 8, a seguir, demonstra o ícone do aplicativo já inserido no sistema Android, já instalado em um celular com Android, seguido da Figura 9 que demonstra tela inicial de login do aplicativo.

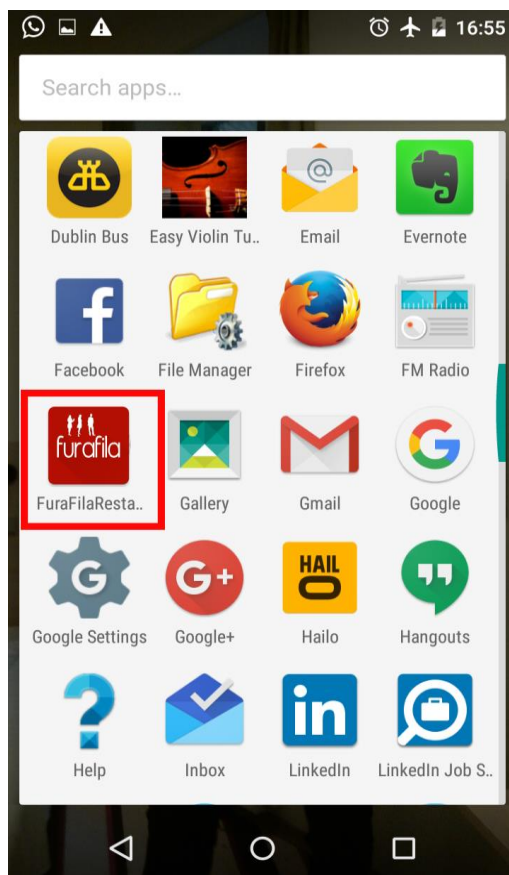


Figura 8: Ícone Fura-Fila

Fonte: Autor

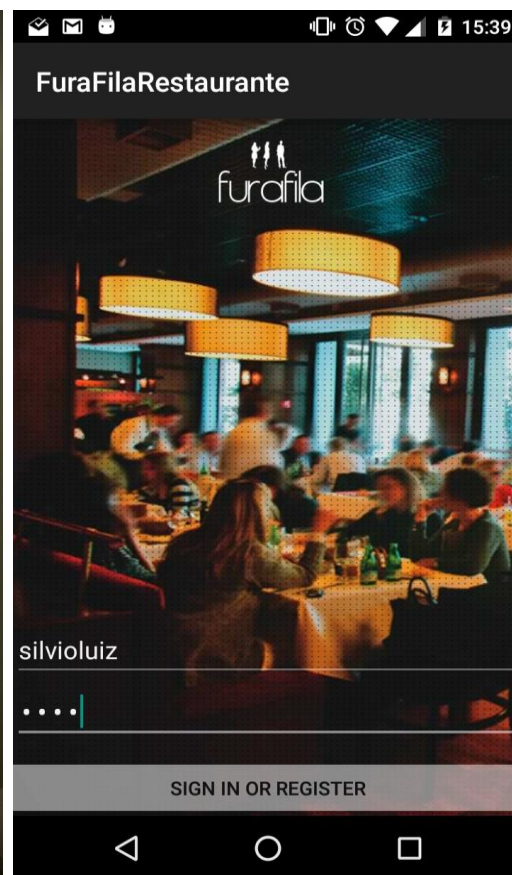


Figura 9: Tela Inicial de Login

Fonte: Autor

O resultado principal é o desenvolvimento de um aplicativo que ofereça um serviço de cardápio onde o usuário interaja com o mesmo selecionando itens e efetuando pedidos em um grupo de abas que exibem uma lista de itens de rolagem, conforme ilustrado nas figuras.

As Figuras 10, 11 e 12 a seguir, listam todas as Refeições, Bebidas e Sobremesas disponíveis, a ideia da List View é possibilitar ao usuário a lista completa dos itens disponíveis no cardápio:

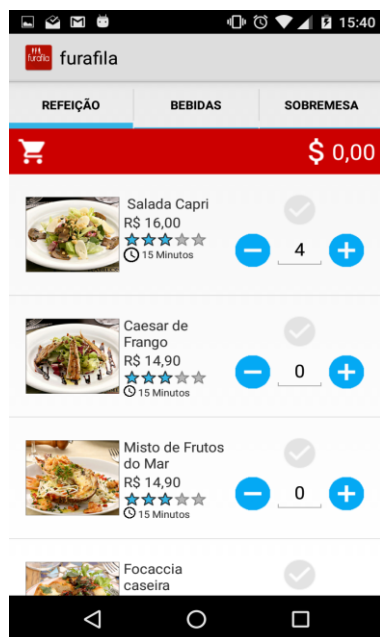


Figura 10 – Aba Refeição

Fonte: Autor

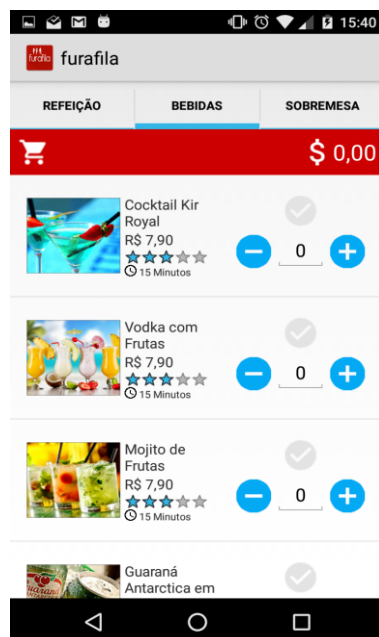


Figura 11 – Aba Bebidas

Fonte: Autor

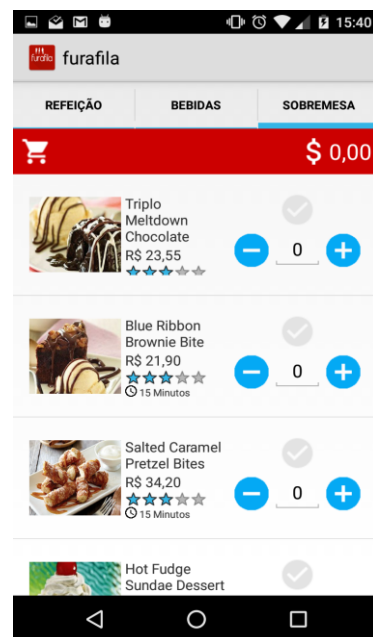


Figura 12 – Aba Sobremesa

Fonte: Autor

As figuras 13, 14, e 15 detalham os itens do cardápio:



Figura 13: Detalhe Refeição

Fonte: Autor



Figura 14: Detalhe Bebidas

Fonte: Autor



Figura 15: Detalhe Sobremesa

Fonte: Autor

No smartphone e no emulador testado, o aplicativo apresentou uma utilização fluindo normalmente, respondendo rápido aos comandos do usuário, sem travamentos ou crash inesperados.

O fator negativo foi que a maioria de aplicações Android existentes assumem que o usuário quer criar e preencher seu banco de dados em tempo de execução e não usará um banco de dados independente com sua aplicação. Mas inicialmente para fim de testes no aplicativo desenvolvido o usuário não necessita modificar o cardápio, ou inserir quaisquer informações no aplicativo, sendo necessário que o banco de dados denominado no projeto como “CardapioDB”, como mostra a Figura 16, esteja preenchido. Nesse caso a manipulação do banco de dados do Android apresentou limitações de “Root”.

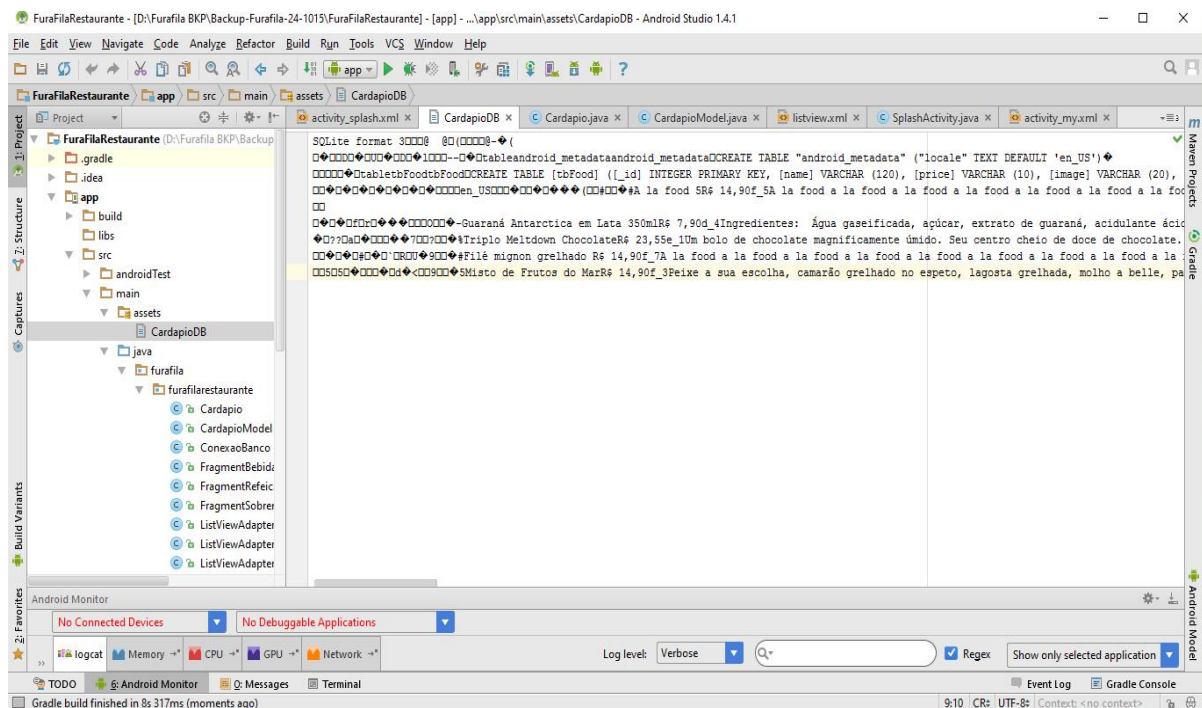


Figura 16: Pasta Assets com CardapioDB

Fonte: Autor

Os sistemas operacionais modernos, que é o caso do Android, possui a herança de sistemas Unix da década de 1970, que por sua vez tem níveis de permissões para acessos controlados, que impedem que pessoas não autorizadas de acessar documentos de outros usuários, ou mesmo acesso de documentos entre usuários. Isso também impede que qualquer um modifique ou apague os arquivos que constituem o sistema operacional (os arquivos do sistema).

O “Root” no Android é o processo que permite com que os usuários de smartphones, tablets e outros dispositivos que possuam o sistema operacional móvel Android tenham controle privilegiado (conhecido como “acesso root”) dentro do subsistema Android. O “Root” é frequentemente realizado com o objetivo de superar as limitações que as operadoras

e fabricantes de hardware colocam em alguns dispositivos, resultando na capacidade de alterar ou substituir aplicativos e configurações do sistema, executar aplicativos especializados que requerem permissões de administrador ou realizar outras operações que são de algumas maneiras inacessíveis para um usuário comum do Android.

No Android, o “Root” também pode facilitar a remoção completa e substituição do sistema operacional do dispositivo, geralmente com uma versão mais recente de seu sistema operacional atual. Portanto foi necessário a criação de um método que usa um banco de dados SQLite de sua pasta “assets” e cópia em um banco de dados no caminho da aplicação, de forma que a API SQLiteDatabase possa abrir e acessar o banco normalmente. Devido a necessidade da manipulação do banco de dados via a pasta “assets”, esta pasta conterá recursos externos utilizáveis pela aplicação, porém, estes recursos serão acessíveis somente programaticamente, no caso do aplicativo em desenvolvimento atuará checando se o banco de dados denominado “CardapioDB” existe, se o mesmo não existir ele criará esse banco, uma vez que ele checou que o banco existe, o próximo passo será comparar se houve qualquer modificação nesse banco de modo a atualizar o mesmo.

Como resultado final do projeto foi criado a integração com o banco de dados externo dentro de um servidor local, preparando desse modo a o aplicativo para interagir com qualquer ERP que use o um banco de dados SQL e que permita a interação através de um Web Service, que define uma arquitetura de comunicação entre softwares independentemente da plataforma que eles foram desenvolvidos. A característica marcante dessa arquitetura é que a comunicação sempre é realizada em rede e deve estar sempre disponível. Essa integração se faz necessário para que seja possível realizar os pedidos a partir do aplicativo e que o mesmo seja registrado no sistema.

Essa integração permitirá a ordem do preparo do pedido com o pedido fechado pretende-se que um painel seja disponível na cozinha, como mostra a Figura 17. Essa tela terá mostrará a lista de pedidos obedecendo a ordem de chegada sendo essa lista atualizada a cada 2 (dois) segundos.



Lista de Pedidos

ID	Cliente	Bebida	Quantidade de Bebida	Sobremesa	Quantidade de Sobremesa	Prato Principal	Quantidade do Prato	Tempo de Preparo
866	1	null		Triplo Meltdown Chocolate	4	Filé Mignon Chateaubriand	0	0
875	1	null		Triplo Meltdown Chocolate	4	Filé Mignon Chateaubriand	0	0
871	1	null		Blue Ribbon Brownie Bite	5	Filé Mignon Chateaubriand	0	0
868	1	null		Salted Caramel Pretzel Bites	3	Filé Mignon Chateaubriand	0	0
872	1	null		Salted Caramel Pretzel Bites	3	Filé Mignon Chateaubriand	0	0
874	1	null		Salted Caramel Pretzel Bites	4	Filé Mignon Chateaubriand	0	0
869	1	null		Hot Fudge Sundae Dessert Shooter	6	Filé Mignon Chateaubriand	0	0
877	1	null		Hot Fudge Sundae Dessert Shooter	7	Filé Mignon Chateaubriand	0	0
865	1	null		Mousse De Chocolate	0	Misto de Frutos do Mar	8	0
867	1	null		Mousse De Chocolate	1	Filé Mignon Chateaubriand	0	0
870	1	null		Mousse De Chocolate	3	Filé Mignon Chateaubriand	0	0
873	1	null		Mousse De Chocolate	0	Misto de Frutos do Mar	5	0
876	1	null		Mousse De Chocolate	0	Caesar de Frango	6	0
878	1	null		Mousse De Chocolate	0	Salada Capri	4	0
879	1	null		Mousse De Chocolate	0	Salada Capri	5	0
880	1	null		Mousse De Chocolate	0	Caesar de Frango	4	0
881	1	null		Mousse De Chocolate	0	Focaccia caseira	4	0
882	1	null		Mousse De Chocolate	0	Caesar de Frango	3	0

Figura 17: Lista de Pedidos - Resultado Web Service

Fonte: Autor

Considerações Finais

Após a implementação do protótipo, foram obtidos os resultados esperados de um aplicativo mobile integrado com o banco de dados via Web Services. A integração de sistemas heterogêneos, a interoperabilidade e interação com outros sistemas desenvolvidos em plataformas diferentes. Através do JSON foi possível essa comunicação, a única ressalva é que se faz necessário um prévio conhecimento da estrutura do arquivo JSON a ser processado para que o programa fosse criado adequadamente. Através da tecnologia Java foi possível o desenvolvimento do aplicativo que permite o usuário, junto a ferramentas de apoio como o servidor local WAMP, também o uso da IDE Android Studio foi extremamente eficiente para o desenvolvimento do aplicativo já que a Google não só o tornou estável, como também o está chamando de IDE oficial. O Google recomenda fortemente que os desenvolvedores usem o Android Studio a fim de se beneficiar das últimas atualizações da IDE.

Os Web Services têm sido muito utilizados para permitir a integração de ferramentas corporativas. Eles tornam possível o desenvolvimento de soluções desacopladas que facilitam

sua integração com sistemas externos, com a comunicação padronizada que possibilita a independência entre diferentes plataformas e linguagens de programação.

O conceito de Web Service solucionou dois grandes problemas no mundo do desenvolvimento de software. Primeiro, permitiu que aplicações independentemente de plataforma trocassem informações. Segundo, mudou o conceito que temos sobre reutilização.

Várias modificações foram realizadas ao longo do desenvolvimento do projeto, revelando uma estratégia que inicialmente era a integração com um ERP específico, para um solução de um problema que surgiu ao longo do mesmo, descobrir uma solução eficiente que permitisse que uma aplicação android que possui um banco de dados nativo estabelecesse uma comunicação eficaz com um banco de dados externo.

No trabalho, tecnologia como o REST é simples de entender e pode ser adotado em praticamente qualquer cliente ou servidor com suporte a HTTP/HTTPS, as principais vantagens foram a facilidade no desenvolvimento, o aproveitamento da infraestrutura web existente e um esforço de aprendizado pequeno.

A necessidade constante de investir em tecnologia é resultado de um mercado competitivo e globalizado que exige qualidade, eficiência, melhores produtos, serviços e processos. Com isso, a inovação tecnológica se tornou fundamental para qualquer organização que deseja ganhar relevância no mercado em que atua.

Inovar diferencia uma empresa de sua concorrente, aumenta a participação no mercado, a lucratividade e o nível de emprego e renda; agrega valor aos produtos e serviços; amplia a relação com novos mercados, abre portas para novas parcerias e novos conhecimentos e ainda melhora o posicionamento no mercado.

Com o intuito de inovar e já considerando a importância que a tecnologia da informação possui dentro de qualquer segmento, a ideia do desenvolvimento do aplicativo se mostrou relevante, possuindo uma contribuição significativa para o segmento apresentado, trazendo uma vantagem competitiva para qualquer estabelecimento que queira implementar o sistema desenvolvido. Uma vez que todas as ferramentas apresentadas são open source, a viabilidade do projeto é total, e eficaz.

Futuros Trabalhos

Alguns possíveis trabalhos futuros relacionados ao trabalho de conclusão de curso são apresentados a seguir:

1. Criar uma tabela Cardápio para o banco de dados externo, uma vez que para fins de prototipação foi disponibilizado apenas uma tabela cardápio interna, deixando assim o aplicativo preparado para qualquer estabelecimento, podendo assim o mesmo fazer a manutenção em seu próprio cardápio.
2. Integrar no fechamento do pedido uma Interface de Programação de Aplicativos (API) para iniciar uma requisição de pagamento pelo próprio aplicativo.
3. Redesenhar o layout do aplicativo baseado no comportamento do usuário, concentrando as funcionalidades para entregar uma experiência de usuário com auto desempenho e interface intuitiva, a fim de facilitar a experiência do usuário com o aplicativo.

Referências

Android Developers. Disponível em: <<http://developer.android.com>>. Acesso em: 10 nov. 2015.

Android Studio - Intelligent Code Editor 2015. Disponível em: <<http://developer.android.com/sdk/installing/studio.html>>. Acesso em: 15 nov. 2015.

BORNSTEIN, DAN (2008). "**Presentation of Dalvik VM Internals**". Google. p. 22. Retrieved 2010-08-16.

DE SORDI, J. O. **Tecnologia da Informação Aplicada aos Negócios**. São Paulo: Atlas. 2003.

eWeek. "**Mobile Industry Revenues to Reach \$1.9 Trillion in 2015: GSMA**." <<http://www.eweek.com/c/a/Mobile-and-Wireless/Mobile-Industry-Revenues-to-Reach-19-Trillion-in-2015-GSMA-284164>>. Acesso em: 17 nov. 2015.

Introdução ao Java JDK. Disponível em: <<http://www.devmedia.com.br/introducao-ao-java-jdk/28896>> Acesso em: 12 jun. 2016.

JOSUTTIS, N. M., SOA in Practice – **The Art of Distributed System Design**, Beijing; Cambridge; Farnham; Köln; Paris; Sebastopol; Taipei; Tokyo: O'Reilly, 2007, 324 p

LECHETA, R. R. (2009). **Google Android - Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 1ª Edição, São Paulo: Editora Novatec, 2009.

Mobile Time. **Apps de compras e utilitários puxam o crescimento no uso de mobilidade em 2014**. Disponível em: <<http://www.mobiletime.com.br/06/01/2015/apps-de-compras-e-utilitarios-puxam-o-crescimento-no-uso-de-mobilidade-em-2014/400802/news.aspx>> Acesso em: 14 nov. 2015.

NEIL SMYTH. **Android Studio Development Essentials** **Android Studio Development Essentials**. Second Edition © 2015 Neil Smyth.

NEVES, Pedro M. C., RUAS, Rui P. F., **O Guia Prático do MySQL**, Vila Nova de Famalicão, Centro Atlântico, 2005.

ROCHA, Cerli Antônio. **PHP – ASP – JSP**: Desenvolvendo web sites dinâmicos. 2007.

ROGERS, R.; LOMBARDO, J.; MEDNIEKS, Z.; MEIKE, B. **Android Application Development**. Sebastopol: O'Reilly Media, 2009.

TANENBAUM, Andrew S. **Sistemas Operacionais Modernos**, 2 ed. São Paulo: Prentice-Hall, 2003.

TURBAN, E.; MCLEAN, E.; WETHERBE, J. **Tecnologia da Informação para Gestão: Transformando os Negócios na Economia Digital**. Porto Alegre: Bookman, 2004.