

# **SISTEMA DE GESTÃO DE ENTREGAS PARA PIZZARIAS UTILIZANDO AS PLATAFORMAS *WEB* E *MOBILE* EM RIO VERDE – GO**

*Júlio César Almeida Nery*<sup>1</sup>

*Daniela Cabral de Oliveira*<sup>2</sup>

*Marcio Rubens Sousa Santos*<sup>3</sup>

## **RESUMO**

No cenário atual de Rio Verde – Goiás, entrar em contato com uma pizzaria para realizar um pedido pode ser complexo, dificultoso e mais demorado do que se imagina. Na prática, o cliente encontra problemas, uma vez que não se tem informações sobre os cardápios disponíveis na pizzaria, os atendentes quase sempre não estão disponíveis para o atendimento, não são informados sobre o andamento do pedido. Esses problemas foram solucionados através do desenvolvimento do sistema *web* e aplicativo *mobile* híbrido que permite aos usuários verem os cardápios disponíveis, realizar e acompanhar as informações dos pedidos, serem notificados sobre as alterações realizadas, o que, consequentemente, melhorou a experiência do usuário. Com o sistema, as pizzarias conservam o relacionamento com os clientes, agregando valor ao negócio, de forma a conseguir mantê-los fiéis. Para o desenvolvimento da aplicação *web* foram utilizadas linguagens HTML, CSS, JavaScript e PHP, o banco de dados utilizado foi o MySQL; visando facilitar e agilizar o desenvolvimento do sistema *web*, foram utilizados os *frameworks*, Bootstrap e Laravel e, para o desenvolvimento do aplicativo *mobile* híbrido, foram utilizados os *frameworks* Ionic, AngularJS, Apache Cordova. Foi aplicado o ciclo de vida para desenvolvimento ágil Kanban, juntamente com a ferramenta Trello para acompanhar o desenvolvimento do software. Pensando em manutenções futuras foram implementados padrões de projetos como *Active Record*, *Factory* e *Repository*. No sistema foram elaborados, o acompanhamento em tempo real do entregador, envio de notificações *push* sobre a alteração de status do pedido e a leitura de cupons de desconto por QR-Code. Após a elaboração do desenvolvimento realizou-se a verificação e validação dos sistemas através dos testes: análise estática automatizada com a ferramenta RIPS, teste funcional com a ferramenta Badboy, teste unitário com o PHPUnit e teste de estresse com a ferramenta Apache JMeter. Foi realizada a hospedagem do site em um servidor, publicação do software no Google Play e pesquisa qualitativa com estudo de caso em uma pizzaria, comprovou-se a satisfação dos usuários em relação ao software. Todas as premissas foram completadas com sucesso, e os resultados foram satisfatórios.

Palavras-chave: Gestão. Tecnologia. Pizzaria. Entrega. Pedidos.

---

<sup>1</sup> Graduando em Engenharia de Software pela Universidade de Rio Verde, Rio Verde, GO.

<sup>2</sup> Orientadora, especialista em Gestão Estratégica de Negócios.

<sup>3</sup> Coorientador, mestre em Engenharia Elétrica.

## 1 INTRODUÇÃO

A sociedade atual é muito competitiva, sendo marcada pela utilização em volume cada vez maior de tecnologia, principalmente para o melhor desempenho das empresas e na busca de destaque dessas no mercado onde está inserida. Assim, para o empreendimento que será destaque neste estudo, uma pizzaria, o foco principal foi a prestação de um serviço inovador e bastante interessante, em função da praticidade que se deseja apresentar aos clientes que buscam pelo serviço delivery de entrega de pizza nas residências, com facilidade de acesso e rapidez na prestação do serviço.

Hoje os desafios que se apresentam no mercado são inúmeros, e criar estratégias para se manter no mercado é necessária. Assim, realizou-se um estudo com o objetivo de buscar alternativas que pudessem estabelecer ações que favorecessem a implantação do negócio.

Ao idealizar um sistema de informação que envolve serviços e produtos para o atendimento em uma pizzaria, observou-se que o desenvolvimento de um aplicativo que possibilite melhorias no atendimento das pizzarias de Rio Verde, principalmente quanto aos pedidos para entregas a domicílio, de forma a promover o crescimento do segmento, mesmo porque o comportamento do consumidor tem alterado, o que favorece um negócio inovador e promissor.

No intuito de obter sucesso e gerar facilidade no atendimento do cliente, buscou-se uma estratégia para expandir o negócio, por meio de um sistema que viesse a facilitar e agilizar atendimento ao cliente via telefone. Nesse sentido, far-se-á uma abordagem que envolva uma atitude positiva entre os serviços e produtos oferecidos pela pizzaria, utilizando-se uma plataforma que venha auxiliar no atendimento, por meio de um ambiente virtual que disponibilize um contato direto com o cliente.

Ao utilizar uma comunicação clara e eficiente, por meio de um aplicativo proporcionar-se-á um relacionamento com clientes e, também, com os colaboradores, em que o fluxo de informações no ambiente corporativo favorecerá a melhoria no atendimento ao cliente e a divulgação do negócio.

O sistema de contato para entregas de pizzas atualmente tem sido via telefone, contudo isso cria um momento de espera ao precisar realizar cadastro, ser informado sobre o valor a ser pago, escolher o tipo de pizza e, em seguida, aguardar a entrega na residência. No entanto, essa maneira de atendimento tem gerado problemas, ocasionando falhas na comunicação, limitação

do atendimento simultâneo e falta de informação do cliente sobre produtos e pedidos, o que acaba gerando custo, tempo e insatisfação dos clientes.

A constância de problemas dessa natureza acaba por gerar insatisfação dos clientes em relação ao estabelecimento, os quais, em determinadas circunstâncias, deixam de adquirir o produto junto àquela empresa, o que acaba gerando prejuízos ao negócio.

Assim, surge o intuito de desenvolver um projeto que vise melhorar a necessidade do empreendedor em atender os clientes, com a finalidade de manter o relacionamento e proporcionar o controle do sistema de entrega via plataformas *web* e *mobile*, com a intenção de facilitar as atividades corriqueiras de uma empresa deste ramo e melhorar a imagem que ela tem diante de seus clientes.

O desenvolvimento de um sistema específico para pizzarias, a ser utilizado pelas empresas no município de Rio Verde, que possa gerenciar os pedidos de entrega à domicílio veio de encontro com as necessidades verificadas no mercado local. Por meio desse aplicativo são armazenadas informações sobre consumidores, a fim de possibilitar a manutenção de um relacionamento, proporcionar praticidade, agilidade e comodidade na realização de pedidos, além de melhorar a qualidade dos serviços prestados, desde a solicitação via telefone, até a entrega do produto no local indicado pelo cliente.

Dessa forma, este projeto propõe desenvolver um sistema de gestão de entregas para pizzarias nas plataformas *web* e *mobile*, que auxilie o empreendedor no gerenciamento das entregas e que garanta maior comodidade, qualidade e praticidade para o cliente, com propósito de tornar o relacionamento entre ambos duradouro.

Com o objetivo de apresentar a necessidade de adotar estratégias de marketing de relacionamento por meio digital, com ênfase na prestação dos serviços e produtos oferecidos; dividiu-se o estudo em capítulos.

## **2 MATERIAL E MÉTODOS**

Para o desenvolvimento do projeto serão apresentadas a seguir as ferramentas utilizadas para elucidar e especificar os requisitos que promoveram a validação e verificação do aplicativo que se almejava consolidar no mercado de pizzarias no município de Rio Verde – Goiás.

Inicialmente, para o desenvolvimento do projeto proposto, criou-se o anteprojeto, que teve como objetivo definir as diretrizes e aspectos relacionados ao projeto desenvolvido, que consiste na identificação do problema abordado, tais como: objetivo geral, objetivo específico, justificativa e hipótese. Por intermédio do documento, foi possível demonstrar o funcionamento do cenário atual, verificar a importância do mesmo e definir as metas a serem realizadas.

Em seguida, foi elaborado o Documento de Especificação de Requisitos de *Software* (DERS), com o intuito de esclarecer o propósito do projeto, interessados, restrições obrigatórias, nomeando convenções e definições, fatos, regras de negócios, estimativas de prazos para a elaboração (cronograma), escopo do trabalho, banco de dados, requisitos funcionais e não funcionais, diagramas, ferramentas, linguagens de programações e frameworks para desenvolvimento do software.

Para a elaboração dos diagramas, foi utilizada a ferramenta gratuita *Astah Community*. Foram elaborados os diagramas de caso de uso, que conforme Sommerville (2003, p.113), “um use-case identifica os agentes envolvidos em uma interação e especifica o tipo de interação”.

Com a conclusão da elaboração dos diagramas de caso de uso, foram identificadas 25 funcionalidades, as quais estão presentes nas aplicações, que 18 requisitos pertencem à aplicação *web*, os requisitos 3 referem-se às duas aplicações e 4 pertencem à aplicação *mobile*.

A próxima etapa do desenvolvimento do projeto foi a elaboração do diagrama de classe, que relata um conjunto de classes, interfaces e relacionamentos.

Baseado no diagrama de classe, foi elaborado o Modelo de Entidade e Relacionamento (MER), utilizando a versão gratuita da ferramenta *MySQL Workbench*. Esse também foi escolhido para manipular as informações contidas no banco de dados da aplicação. O MER representa a estrutura do banco de dados da aplicação, contendo entidades, atributos e relacionamentos.

Após a finalização do MER, foi construído o dicionário de dados, que contém as características lógicas do banco de dados da aplicação, que são utilizadas no desenvolvimento do *software*.

O MySQL é o Sistema Gerenciador de Banco de Dados (SGBD), foi escolhido entre vários sistemas gerenciadores por ser uma ferramenta gratuita desenvolvida sob a GPL (Licença Pública Geral) e, também, o mais apropriado para o gerenciamento do banco de dados.

Foi elaborado o diagrama de componentes para compreender melhor a estrutura geral da aplicação, a fim de demonstrar os componentes na posição dos *frameworks* e suas determinadas funções no sistema.

Para facilitar a gestão da informação e a tomada de decisão no desenvolvimento, foi implementado o método de desenvolvimento ágil Kanban.

Segundo Mariotti (2016), “o Kanban, com seu mecanismo de sinalização, tem como objetivo apresentar uma atividade de trabalho em processo, ou seja, o número de atividades ou cartões em circulação é equivalente à capacidade do sistema”.

Foi implantado a ferramenta Trello para realizar os procedimentos do Kanban, a fim de facilitar as informações armazenadas e, também, ser uma ferramenta usual e não possuir custo para sua utilização básica.

Com a finalidade de mensurar a qualidade da prestação do serviço de entrega das pizzarias de Rio Verde – Goiás, foi realizado uma pesquisa de campo. Foi escolhido os estabelecimentos de pizzarias devido à praticidade de atendimento de entrega aos clientes.

A pesquisa qualitativa contou com a participação de 54 voluntários que responderam às perguntas para avaliação de 844 estabelecimentos, conforme dados da Prefeitura Municipal de Rio Verde, entre eles lanchonetes, bares, restaurantes, pizzarias, sorveterias e alimentação são cadastrados como estabelecimentos de pizzarias.

Foi escolhido para realização do estudo de caso da pesquisa a Pizzaria Cristo Rei devido à localização do estabelecimento, situado em um bairro com ênfase comercial e com grande público alvo, além disso, o proprietário se dispôs prontamente em utilizar como teste o software para prestação do serviço da entrega.

As instruções apresentadas no Anteprojeto e DERS, orientaram a criação da aplicação. A partir do pensamento claro sobre o sistema, o primeiro passo foi instalar os recursos e ferramentas necessárias para a implementação do sistema de forma estruturada.

Para codificação do sistema utilizou-se o *Notepad++*<sup>4</sup> e as linguagens utilizadas para o desenvolvimento da aplicação *web* foram: HTML, CSS, JavaScript e PHP.

---

<sup>4</sup> <https://notepad-plus-plus.org>

O *HyperText Markup Language* (HTML) é uma linguagem de marcação de texto para publicação de conteúdos para *web*. No desenvolvimento foi utilizado a sua versão mais atualizada disponível atualmente, o HTML5.

A linguagem *Cascading Style Sheets* (CSS) é uma folha de estilos composta por camadas, que formata a informação que é entregue pelo HTML e prepara as informações para serem utilizadas para estilização da aplicação.

JavaScript é uma linguagem de programação interpretada, utilizada para gerenciar e controlar comportamentos na página *web*.

Para o gerenciamento da lógica da aplicação do lado do servidor, foi utilizado a linguagem de programação *HyperText Preprocessor* (PHP).

Conforme afirma Welling (2005), ao se desenvolver uma página HTML é possível embutir o código PHP, que será executado e interpretado no servidor *web* e pode gerar HTML ou qualquer outra saída que o visitante verá.

Para facilitar e agilizar o processo de desenvolvimento, foi empregado alguns *frameworks*, na aplicação *web*: Bootstrap e Laravel e na aplicação *mobile* híbrida foram utilizados AngularJS, Ionic e Cordova.

Silva (2015) explica que *Frameworks* são conjuntos de componentes utilizados para solucionar problemas com funcionalidades nas aplicações. No desenvolvimento de *softwares* oferecem uma estrutura padrão, que agilizam o processo de desenvolvimento.

Silva (2015) aponta que o *framework* Bootstrap<sup>5</sup> é o *framework front-end* que facilita o trabalho do desenvolvedor, trabalha na usabilidade e possibilita o desenvolvimento de projetos responsivos, dando foco aos dispositivos móveis na *web*.

O tema utilizado no sistema é o Material Admin Responsive + Dark Skin 1.6.1 adquirido em {wrap}bootstrap<sup>6</sup> cujo código do pedido é WB039R4L5 e o ID da licença é *Single Application License* = 02835145-f1d4-4143-9dc8-fdb8daee43e9, a aquisição do tema facilitou no desenvolvimento do portal, contendo configurações pré-configuradas do *front-end*.

---

<sup>5</sup> <http://getbootstrap.com/>

<sup>6</sup> <https://wrapbootstrap.com/>

O Laravel foi escolhido como *framework* PHP devido a simplicidade, flexibilidade, sintaxe e rica documentação<sup>7</sup>. O *framework* utiliza a arquitetura *Model-View-Controller* (MVC) na qual é possível desenvolver aplicações seguras, modernas, de forma rápida.

O Laravel auxiliou a linguagem PHP na aplicação *web*, manipulando as informações do sistema. Esse vem pré-configurado com componentes para facilitar e agilizar o desenvolvimento, além de conter diversos modelos com padrões de projeto já definidos.

Foram elaborados os modelos (*models*) a primeira etapa do MVC, e para facilitar em manutenções futuras, foi utilizado o *design pattern Repository*, o qual foi criado uma nova camada intitulada *Repository*, que é a classe entre a aplicação e a parte de dados.

Iniciou-se o processo de elaboração das telas (*views*), a segunda parte da arquitetura MVC, que foi utilizado o tema Material Admin adaptado para o padrão a ser usado no Laravel. O tema foi construído com os componentes *Bootstrap* e *jQuery*<sup>8</sup> que facilitou e agilizou no desenvolvimento das telas.

Para gerenciar o recebimento das informações das telas, foram criados os controladores (*controllers*), terceira etapa da arquitetura MVC. Nos mesmos foram realizados o CRUD (*create, read, update, delete*) da aplicação, validações, e também algumas regras de negócio.

Foi implementado também o envio de e-mail na aplicação, que após a realização de cada pedido é enviado por e-mail para a empresa, com intuito de alertar o administrador da empresa sobre o recebimento dos pedidos.

Foi iniciado o processo de desenvolvimento da API (*Application Programming Interface*) em português Interface de Programação de Aplicações. Para realizar o processo de autorização e autenticação do usuário, foi utilizado a biblioteca do framework de comunicação OAuth2<sup>9</sup> e ajustado os middlewares na aplicação. A API é responsável por fornecer recursos e serviços para a aplicação *mobile* híbrida, e seu acesso é realizado por meio de *tokens*.

Após a conclusão da API foi iniciado o desenvolvimento *mobile*, com a combinação dos frameworks Ionic, Cordova e AngularJS para aplicação *mobile* híbrida.

Na arquitetura *mobile* híbrida, os sistemas operacionais móveis possuem um componente chamado *Web View* que é um navegador executado dentro do aplicativo móvel,

---

<sup>7</sup> <https://laravel.com/docs/> e <http://laravel-recipes.com/>

<sup>8</sup> <https://jquery.com/> - framework JavaScript

<sup>9</sup> <https://tools.ietf.org/html/rfc6749>

que é executado códigos HTML, CSS e JavaScript, sendo possível criar uma página *web* com tecnologias anteriores e executar dentro do aplicativo (RAVULAVARU, 2015).

Com o Ionic foram inclusos os elementos visuais como *abas*, botões e barras de navegação com interface quase nativa dentro do aplicativo híbrido. Também foi possível enviar notificações *push* sobre cada alteração realizada no andamento do pedido.

Juntamente com o Ionic foi utilizado o AngularJS como *framework* para gerenciamento da lógica da aplicação e o Cordova para a construção da parte nativa do aplicativo.

O framework AngularJS foi o responsável por fazer o roteamento, controle das funções e dados contidos no aplicativo *mobile*.

O Cordova foi utilizado para realizar a comunicação com a parte nativa dos *smartphones*. Apache Cordova é um *framework* de código livre para a criação de aplicações nativas usando HTML5. Os criadores do Apache Cordova queriam construir aplicações móveis multi-plataforma e, então, decidiram implementar uma combinação de tecnologias nativas e aplicações *web*. Esse tipo de aplicação móvel é chamado de aplicação híbrida (WARGO, 2014).

O Apache Cordova foi responsável pela comunicação do aplicativo com o dispositivo, sendo possível acessar a câmera para leitura do QR-Code e, assim, proporciona descontos nos pedidos, e é possível também acessar recursos como a geolocalização do *smartphone* para acompanhar em tempo real a localização do entregador.

Portanto, o Bootstrap foi responsável pelo *front-end* e o Laravel do *back-end* da aplicação *web*. No aplicativo *mobile* híbrido o Ionic é responsável pela parte visual, o AngularJS gerenciou os dados e o Cordova acessou a parte nativa do dispositivo.

No processo de desenvolvimento outros padrões de projeto também foram utilizados além do *Repository*, tais como, *Factory* e *Active Record*. Esses já vieram devidamente configurado no *framework* Laravel, sendo assim foi necessário apenas adquirir conhecimento para a utilização.

Os processos de desenvolvimento realizados em ambas plataformas foram cuidadosamente estruturados, de forma que a localização de qualquer função do código esteja fácil e passiva às atualizações e correções quando necessário.

Após a conclusão do desenvolvimento, iniciou-se o processo de realização dos testes no software desenvolvido. Os testes têm o intuito de verificar se o software atingiu as especificações projetadas e se está funcionando corretamente no seu ambiente projetado,



revelando falhas para que possam ser corrigidas antes da versão final. Foram realizados os testes de análise estática, unitário, funcional e estresse.

Para o teste de análise estática foi utilizado a RIPS, essa detecta automaticamente as vulnerabilidades em aplicações PHP, analisando o código fonte para detectar possíveis funções vulneráveis que podem ser manipuladas por um usuário mal-intencionado (RIPS, 2016).

Os testes unitários foram realizados a fim de manter a integridade das classes, esses mantêm o funcionamento adequado do software. Foi utilizado a ferramenta PHPUnit por vir com o ambiente devidamente configurado no Laravel para a elaboração das classes de testes.

Foi realizado o teste funcional com a ferramenta Badboy na aplicação web, a fim de validar todos os requisitos funcionais, para garantir que a aplicação para que atenda as funcionalidades projetadas no documento de especificação de requisitos.

Portanto, para finalizar a etapa dos testes, foi realizado o teste de estresse na aplicação *mobile* híbrida, utilizando a ferramenta Apache JMeter, com o objetivo de identificar o comportamento da aplicação, mediante uma situação de uso crescente, chegando ao limite em termos de volume de dados transacionados e número de usuários para o atual cenário.

Foi iniciada a preparação dos ambientes para a realização do estudo de caso, juntamente com a segunda etapa da pesquisa qualitativa. Para a aplicação web foi realizada a publicação dos dados de um servidor hospedeiro da *DigitalOcean* e adquirido o domínio *osprintf.com* para facilitar na localização do site pela pizzaria.

O aplicativo *mobile* híbrido foi publicado na Google Play, para a realização da segunda etapa da pesquisa qualitativa. Encontrou-se algumas dificuldades como a criação do certificado para a realização da publicação, mas todas foram superadas.

## 4 RESULTADOS E DISCUSSÕES

Nesta seção é demonstrado o resultado final do sistema e os resultados obtidos nos testes. Ao iniciar o projeto, foram especificadas as funcionalidades correspondentes aos requisitos funcionais do *software*, que são cruciais para a execução do sistema conforme pretendido.

A Figura 01 representa a tela padrão do sistema *web*, todas as telas dos componentes *Create, Read, Update, Delete (CRUD)* contem modelos parecidos como a representada na figura a seguir.

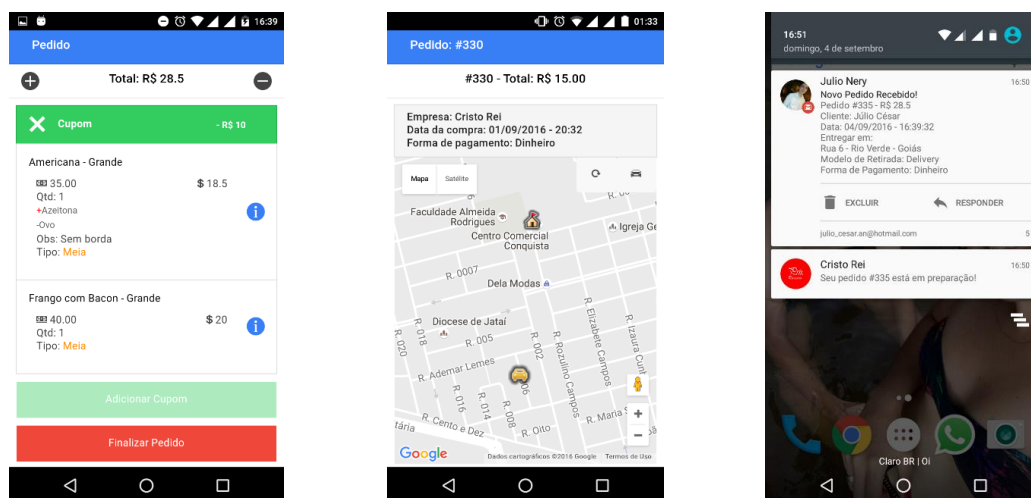
**FIGURA 011** – Representação da tela de pedidos do sistema

ID	TOTAL	DATA	ITENS	ENTREGADOR	STATUS	AÇÕES
380	R\$ 43.00	18/11/2016 - 22...	Pizza Tradicional;	—	Pendente	[Edit]
379	R\$ 67.97	07/11/2016 - 21...	Rúcula e Tomate ...	Entregador	Entregue	[Edit]
375	R\$ 165.96	04/11/2016 - 20...	A Moda Cristo Re...	Entregador	Entregue	[Edit]
373	R\$ 62.00	02/11/2016 - 18...	Açaí com Nutela; ...	Entregador	Entregue	[Edit]
372	R\$ 70.00	02/11/2016 - 17...	Pizza de Bacon; ...	Entregador	A caminho	[Edit]
371	R\$ 48.00	29/10/2016 - 21...	Pizza de Calabre...	Entregador	Entregue	[Edit]
367	R\$ 143.97	28/10/2016 - 20...	A Moda Cristo Re...	Entregador	Entregue	[Edit]

Fonte: Júlio César Almeida Nery (2016).

Já a Figura 02 representa algumas das funcionalidades existentes no aplicativo *mobile*, como a adição de cupom de desconto via QR-Code, acompanhamento em tempo real do entregador e o recebimento de notificações push sobre a alteração de status realizada no sistema *web*.

**FIGURA 02** – Representação dos componentes do aplicativo *mobile*



Fonte: Júlio César Almeida Nery (2016).

Após a conclusão do sistema iniciou-se a etapa dos testes, para realizar a verificação e validação do software desenvolvido.

No primeiro teste de análise estática automatizada realizado no RIPS, demonstrando uma contaminação realizada pelo usuário. Foram identificadas 21 vulnerabilidades classificando-as em diferentes módulos.

Porém grande parte dessas vulnerabilidades são falsos positivos, que foram apontadas funções maliciosas, no entanto se tratam de funções válidas e sem problemas, algumas delas para manter o funcionamento adequado do *framework*.

O resultado do teste funcional obteve um total de testes gravados e realizado de 40, constando sucesso em todos os testes, garantindo que todas as funcionalidades do sistema *web* estavam devidamente implementadas e funcionando corretamente.

Com o resultado do teste de estresse foi possível medir a estabilidade, capacidade e confiabilidade do ambiente em que a aplicação foi hospedada, permitindo a prevenção de potenciais problemas na capacidade de resposta. Contudo, é possível orientar uma tomada de decisão sobre melhorias na aplicação, arquitetura e infraestrutura de acordo com a popularidade da aplicação.

Portanto com base nos resultados obtidos nos testes realizados, confirma-se concluído o desenvolvimento do software, satisfazendo a qualidade e obtenção dos requisitos almejados, e possíveis melhorias a serem implementadas.

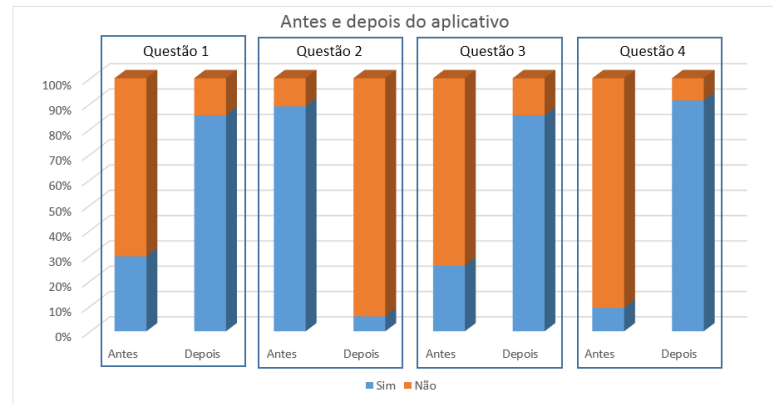
Ao desenvolver um estudo de campo, continha o objetivo de obter informações sobre a realização dos pedidos de pizza por telefone. Com os resultados desse estudo, verificou-se a relevância prática do desenvolvimento de um aplicativo para o auxílio dos pedidos *delivery*, porque a maioria dos participantes do estudo realizam seus pedidos por telefone, os quais não apresentaram ser eficiente os serviços oferecidos. Observou-se que os pedidos não são repassados devidamente e que não há o repasse fiel das informações constantes do pedido. Enfim, o nível de satisfação dos serviços oferecidos foi classificado como regular.

Com o objetivo de obter informações sobre a viabilidade da realização dos pedidos de pizza por meio do aplicativo, a fim de defender a aplicabilidade desse no auxílio de pedidos *delivery* foi realizado uma segunda etapa da pesquisa qualitativa.

O Gráfico 12 demonstra a comparação dos questionários na realização e acompanhamento dos pedidos antes e depois do aplicativo. Observou-se que os participantes

ficaram satisfeitos quanto aos serviços oferecidos pelo aplicativo (Questão 1) e que não existe atraso com os pedidos feitos (Questão 2), as informações são repassadas com eficiência (Questão 3) e as informações dos pedidos mantêm-se fieis (Questão 4).

**GRÁFICO 1** – Comparação do antes e depois aplicativo



## 5 CONCLUSÃO

À princípio, foram identificados alguns problemas, os quais justificam a elaboração do projeto proposto, tais como atraso e falhas no atendimento, falta de informações sobre os produtos e pedidos, que ocasiona insatisfações dos clientes. Visando a solucionar essas dificuldades, foi levantado a hipótese para o desenvolvimento de uma aplicação *web* que realiza todo o controle das informações e uma aplicação *mobile* híbrida que confere as informações dos produtos e pedidos e realiza pedidos de forma ágil e prática.

Realizar cadastros com as informações dos produtos e geração de relatórios, são de responsabilidade da aplicação *web*. Após conclusão dos cadastros, foi desenvolvido a aplicação *mobile* híbrida para proporcionar aos clientes um mecanismo para a realização de pedidos com praticidade e agilidade, que busca as informações presentes na aplicação *web* que recebe as requisições passadas da aplicação *mobile* híbrida. Para a realização de todos esses procedimentos foi necessário empenho e muito esforço, devido ao cenário conter um ambiente com diversas tecnologias distintas, comunicando-se entre si, em prol da usabilidade do usuário.

Dessa forma, para trazer comodidade aos clientes foi desenvolvido o acompanhamento em tempo real da localização do entregador, implementado envio de notificações *push* para informá-los sobre cada alteração no status do pedido e leitura de cupons de desconto via QR-Code. Para praticidade das pizzarias, foi implementado o envio de e-mail, informando sobre novos pedidos recebidos e liberação para recebimento de pedidos por horário determinado.

A plataforma web foi desenvolvida mantendo uma estrutura harmônica entre as páginas, desse modo os formulários disponíveis para o usuário são intuitivos, facilitando sua utilização. Mantendo a usabilidade do usuário todo o projeto é responsivo, dessa forma, o mesmo pode acessar de diferentes dispositivos, com dimensões de telas diferentes, que a aplicação se adapta e redimensiona-se nos diferentes tipos de tela. O processo de desenvolvimento exigiu muita pesquisa, a fim de manter as usabilidades e funcionalidades para o usuário.

A aplicação *mobile* híbrida está totalmente voltada para o cliente e entregador, por meio dela que os mesmos realizam e acompanham os pedidos com agilidade e praticidade. Estão disponíveis o acompanhamento do entregador em tempo real através do Google Maps integrado ao aplicativo, ler cupom de desconto via QR-Code e receber as notificações sobre o pedido. Essa etapa foi muito desgastante, devido à complexidade para realizar a manipulação das informações entre as plataformas, realização dos tratamentos de exceções e regras de negócio demandarem esforço e tempo.

Visando implementações futuras e de forma a enriquecer o projeto, foi implementado e utilizado padrões de projeto, tais como *Active Record*, *Factory* e *Repository*, que demandou uma busca por novos conhecimentos, devido à sua complexidade, sendo mais uma conquista alcançada. Após todo o desenvolvimento, foram realizados os testes comprovando que o sistema está funcional nas plataformas que foram desenvolvidas.

Com o estudo realizado na Cristo Rei, foi possível verificar que, mesmo havendo uma cultura de pedidos das pizzarias por meio do telefone, há uma grande mudança para o uso de aplicativos nesse empreendimento. Essa nova possibilidade auxilia os empreendedores, bem como os clientes, ofertando-lhes praticidade e agilidade em seus pedidos. Contudo, muito ainda deverá ser realizado para que essa nova possibilidade venha a ser utilizada com maior frequência e passar a ser rotina no comércio.

A tendência é que haja uma maior efetivação de serviços em pizzeria por meio do uso de aplicativo para favorecer o negócio, principalmente nos casos de entrega a domicílio, porque essa nova modalidade de negociação se ajusta perfeitamente às tendências de utilização de novas tecnologias.

Portanto, as premissas abordadas no objetivo final foram contempladas com sucesso e os resultados obtidos apontaram que a hipótese levantada foi confirmada de forma satisfatória, devido ao fato dos clientes terem auxílio na realização e acompanhamento de pedidos e, com isso, consegue-se reduzir o tempo e custo, tornando o processo mais prático e ágil para as pizzarias e consumidores de Rio Verde – Goiás.

## REFERÊNCIAS

- MARIOTTI, F.S. *Kanban: o ágil adaptativo*. s./d. Não paginado. Disponível em: <<http://www.garcia.pro.br/EngenhariadeSW/artigosMA/A6%20-%2045-6-%20Kanbam.pdf>>. Acesso em: 15 out. 2016.
- RAVULAVARU, A. *Learning Ionic*. Birmingham: Packt, 2015.
- RIPS. *RIPS free PHP security scanner using static code analysis*. 2016. Não paginado. Disponível em: <<http://rips-scanner.sourceforge.net/>>. Acesso em: 09 out. 2016.
- SILVA, M. S. *Bootstrap 3.3.5: Aprenda a usar o framework Bootstrap para criar layouts CSS complexos e responsivos*. São Paulo: Novatec, 2015.
- SOMMERVILLE, I. *Engenharia de Software*. 6. ed. São Paulo: Addison Wesley, 2003.
- WELLING, L. *PHP e MySQL desenvolvimento Web*. Rio de Janeiro: Elsevier, 2005.
- WARGO, J. M. *Apache Cordova 3 Programming*. New Jersey: Pearson, 2014.

*DELIVERY MANAGEMENT SYSTEM FOR PIZZERIAS USING AS WEB  
AND MOBILE PLATFORMS IN RIO VERDE - GO*

**ABSTRACT**

In the current scenario of Rio Verde - Goiás, to get in touch with a pizzeria to place an order can be complex, difficult and take longer than you think. In practice, the customer finds problems, because there is no information on menus available at the pizzeria, the support employees are not always available for the service, they are not informed about the progress of the request. These problems have been solved through the web system development and mobile hybrid application that allows customers to see the menus available, to realize and monitor information requests, be notified of the changes made, which consequently improved the customers experience. With the system, the pizzerias maintain the relationship with customers, adding value to the business in order to be able to keep them loyal. For the development of web application were used HTML, CSS, JavaScript e PHP, the database used was MySQL to facilitate and optimize the web system development, it was used frameworks such as Bootstrap and Laravel and to the development of mobile hybrid application, it was used the Ionic frameworks, with the screens and AngularJS interactions with control the logic of the application and Apache Cordova, with access to native parts of the devices. It was used the life cycle for Kanban agile development, in conjunction with Trello tool to accompany the development of the software. Thinking about future maintenance, project standards such as Active Record, Factory and Repository have been implemented. It was implemented in the application the monitoring of the pizza delivery, sending push notifications about the status change request and reading discount coupons for QR-Code. After the implementation of development took place the verification and validation of systems through the tests: automated static analysis, focused on the vulnerabilities with the RIPS tool, functional testing with the Badboy tool, unit testing with PHPUnit and stress test with the Apache JMeter tool. It was carried out the website hosting on a server, publication of the software on Google Play and qualitative research with case study in a pizzeria. It has been shown that the customers satisfaction in relation to the software. All premises have been completed successfully, and the results indicate that the hypothesis was confirmed satisfactorily.

Keywords: Management. Technology. Pizzeria. Delivery. Requests.