



FACULDADE ÚNICA DE IPATINGA
SISTEMAS DE INFORMAÇÃO

MARQUESCHARLON RODRIGUES SANTOS

**DESENVOLVIMENTO DE SISTEMA PARA REALIZAÇÃO DE RESERVA
DE UMA MESA EM ESTABELECIMENTOS E CHECK-IN,
COM BASE NA METODOLOGIA ÁGIL.**

IPATINGA - MG

2016

MARQUESCHARLON RODRIGUES SANTOS

**DESENVOLVIMENTO DE SISTEMA PARA REALIZAÇÃO DE RESERVA
DE UMA MESA EM ESTABELECIMENTOS E CHECK-IN,
COM BASE NA METODOLOGIA ÁGIL.**

Monografia apresentado ao Curso de Sistemas de Informação da Faculdade Única de Ipatinga, como requisito parcial para obtenção do título de bacharel em Sistemas de Informação.

Orientador: Thiago de Medeiros Gualberto

IPATINGA - MG

2016

Marquescharlon Rodrigues Santos

**DESENVOLVIMENTO DE SISTEMA PARA REALIZAÇÃO DE RESERVA
DE UMA MESA EM ESTABELECIMENTOS E CHECK-IN,
COM BASE NA METODOLOGIA ÁGIL.**

Monografia apresentado ao Curso de Sistemas de Informação da Faculdade Única de Ipatinga, como requisito parcial para obtenção do título de bacharel em Sistemas de Informação.

Aprovada em 13/12/2016

BANCA EXAMINADORA

Thiago de Medeiros Gualberto
Faculdade Única de Ipatinga

Jutson Ribeiro Machado
Faculdade Única de Ipatinga

Filipe Costa Fernandes
Faculdade Única de Ipatinga

Dedico aos meus pais e parentes, por me
fazerem acreditar que, embora a distância, as
dificuldades que possam ser inúmeras, ainda é
possível conquistar.

AGRADECIMENTOS

Agradeço aos meus colegas, meus professores e minha família por terem ajudado na construção desse trabalho.

Agradeço ao Prof. Orientador Thiago Medeiros pela paciente e dedicada orientação, pela competência e amizade.

Aos professores Felipe Fernandes e Jutson Machado que fizeram parte da banca examinadora, pelas importantes observações apresentadas.

LISTA DE ILUSTRAÇÕES

GRÁFICO 1 – Nível de aceitação da tecnologia em estabelecimentos	17
GRÁFICO 2 – Níveis de dificuldades encontradas hoje em estabelecimentos.....	18
GRÁFICO 3 – Opinião dos entrevistados sobre o relacionamento entre ambas as partes..	18
GRÁFICO 4 – Avaliação dos estabelecimentos na visão dos próprios clientes	19
FIG.1: Exemplo de testes de validação	27
FIG.2: Frente de um Cartão de História	29
FIG.3: MVC em Ruby On Rails.....	31
FIG.4: Sistema sendo acessado por vários dispositivos	32
FIG.5: Realizar Login.....	34
FIG.6: Cadastro de usuário	34
FIG.7: Recuperar senha	34
FIG.8: Visualizar produtos	35
FIG.9: Acessar Painel Administrativo.....	35
FIG.10: Administração do sistema	36
FIG.11: Cadastrar produtos	36
FIG.12: Adicionar novo produto	37
FIG.13: Deixar o produto visível para os clientes	37
FIG.14: Status Mesa	38
FIG.15: Obter mais detalhes da mesa.....	39
FIG.16: Realizar a reserva da mesa	40
FIG.17: Reserva realizada	40
FIG.18: Comprovante da reserva	42
FIG.19: Cartão de História – Página inicial	43
FIG.20: Verso do Cartão – Página inicial.....	44
FIG.21: Cartão de História – Cadastrar usuário	45
FIG.22: Verso do Cartão – Cadastrar usuário	46
FIG.23: Cartão de História – Após realizar <i>login</i>	47
FIG.24: Verso do Cartão – Após realizar <i>login</i>	48
FIG.25: Criar a tabela no banco de dados	49
FIG.26: Cartão de História – Mesas	50
FIG.27: Verso do Cartão – Mesas	51
FIG.28: Cartão de História – Página Mesas alterada.....	52

FIG.29: Cartão de História – Produtos	53
FIG.30: Estrutura deste projeto	54
FIG.31: Tela inicial de uma nova aplicação	57
FIG.32: Erros e soluções	58
FIG.33: Cronograma.....	60

LISTA DE ABREVIATURAS E SIGLAS

C3 - Chrysler Comprehensive Compensation System

CSS - Cascading Style Sheet

DRY - Don't Repeat Yourself

GB - Gigabyte

HD - Hard Disk

HTML - Hypertext Markup Language

IDE - Integrated Development Environment

MVC - Model-View-Controller

PHP - Hypertext Preprocessor

QR Code - Código Quick Response

RAM - Random Access Memory

ROM - Read-Only Memory

SO - Operational Systems

SSD - Solid-State Drive

XP - Extreme Programming

YARV – Yet Another Ruby VM

RESUMO

A cada dia aumentam mais a necessidade de tornar algum procedimento, uma determinada produção ou tarefa mais ágil. Para isso é necessário entender que em muitos casos devemos abrir mão de seus atuais conceitos e ideias que ficam arraigadas, atrapalhando na tomada de decisão, atrasando o projeto com procedimentos que irão apenas gastar mais tempo e ideias conhecidas também como paradigmas mentais. Para alcançar tal agilidade é necessário concentrar a força da equipe apenas nas tarefas que exigem maior urgência. A metodologia Extreme Programming (XP) traz como principal função em um projeto a sua própria construção, isso é, valoriza mais os códigos, considerado também a melhor documentação, para isso, é necessário um código limpo, bem estruturado, intuitivo e rápido. Assim, este projeto visa solucionar o problema da comunicação entre os estabelecimentos e seus clientes, com objetivo de tornar a interação entre ambas as partes mais segura e intuitiva, podendo contar também com as recomendações por meio de check-in de quem já esteve no local. Tal proposta oferece um sistema capaz de auxiliar o estabelecimento em sua organização e, além disso, oferecer um serviço de qualidade aos seus clientes, padronizando seu atendimento e oferecendo novas formas de pagamento.

Palavras-chave: Métodos ágeis. Metodologia Extreme Programming. Código limpo e bem estruturado.

ABSTRACT

Every day increases more the need to make some procedure, a production or faster task. For this it is necessary to understand that in many cases it is necessary to give up your current concepts and ideas that are rooted, hindering decision making, delaying the project with procedures that will just spend more time known ideas as well as mental paradigms. To achieve this agility is necessary focus the strength of the team only in tasks that require greater urgency. The methodology Extreme Programming (XP) has as its main function in a design your own building, that is, value more the codes, also considered the best documentation for this, clean code is necessary, well-structured, intuitive and fast. Thus, this project aims to solve one of the problems found in the relationship between institutions and their customers, in order to make the communication between both safer and more intuitive parts. This proposal provides a system to assist the establishment in your organization and, in addition, offer a quality service to its customers by standardizing their service and offering new forms of payment.

Keywords: Agile methods. Extreme Programming methodology. Clean and well structured code.

SUMÁRIO

1. INTRODUÇÃO.....	13
1.1. TEMA	14
1.2. PROBLEMA	14
1.3. HIPÓTESE.....	14
1.4. JUSTIFICATIVA	15
1.5. OBJETIVOS	15
1.5.1. GERAL	15
1.5.2. ESPECÍFICOS	15
2. METODOLOGIA.....	16
2.1. PESQUISA QUALITATIVA	16
2.2. FERRAMENTAS DE PESQUISA.....	16
2.3. MÉTODOS DE PESQUISA	20
3. REFERENCIAL TEÓRICO	21
3.1. METODOLOGIAS ÁGEIS.....	21
3.2. EXTREME PROGRAMMING (XP)	21
3.3. METODOLOGIA DE DESENVOLVIMENTO	24
3.3.1. CARTÃO DE HISTÓRIA	26
4. DESENVOLVIMENTO	32
4.1. FUNCIONALIDADES DO SISTEMA	33
4.1.1. LOGIN, CADASTRO DE USUÁRIO E RECUPERAÇÃO DE SENHA ...	33
4.1.2. CADASTRO E VISUALIZAÇÃO DOS PRODUTOS	34
4.1.3. RESERVADA MESA.....	38
4.2. PRINCIPAIS CARTÕES DE HISTÓRIA DO SISTEMA.....	43
4.3. INSTALAÇÕES E CONFIGURAÇÕES	53

4.4. DIFICULDADES NO DECORRER DO PROJETO.....	57
5. CRONOGRAMA	60
6. CONCLUSÃO.....	61
ANEXO 1.....	65

1. INTRODUÇÃO

Os inúmeros afazeres do dia-a-dia nos forçam a buscar por soluções que colaborem com o tempo e dinheiro, permitindo que determinada ação seja ágil, alcançando maior comodidade e retorno naquilo que se investe.

Uma forma de se alcançar tal conforto é a intervenção da tecnologia, pois a cada dia surgem novas tendências a fim de colaborarem com o ser humano em diversos aspectos da vida. Em muitos casos, o foco se encontra nos dados que são coletados do próprio usuário. Assim, o tratamento desses dados, a consulta das informações, podem oferecer relatórios diários, semanais, mensais e anuais. Não só uma visão macro do negócio, mas oferecer caminhos que contribuem para alcançar o objetivo desejado, evitando gastos. Segundo TELES (2005), a elaboração do relatório se dá mais para o final do desenvolvimento do sistema.

Controlar as atividades e obter relatórios não resolverão todos os problemas. Para isso, é apresentado neste trabalho uma proposta em que tanto o cliente quanto o estabelecimento ganharão em agilidade, credibilidade, segurança e organização. Tal proposta consiste no desenvolvimento de um sistema em que, a partir de sua utilização facilitará a gestão do estabelecimento e também contribuirá para oferecer um serviço de qualidade e uma melhor interação com o cliente.

Com a utilização do sistema, o cliente poderá reservar sua mesa online e através do reconhecimento de um QR Code (Código Quick Response) realizar o check-in na entrada do estabelecimento, permitindo ao cliente fazer sua reserva/pré-cadastro ainda no conforto de sua casa, como por exemplo, a reserva de uma passagem aérea, tal qual, é realizada online e tendo em vista seu check-in antes do embarque. Após o seu acesso ao estabelecimento será possível realizar seus pedidos com base no cardápio disponibilizado, e ainda acumular promoções na medida que se compra, por exemplo, a cada compra de um lanche você ganha um ponto e depois pode trocá-lo por uma bebida ou lanche. Assim, ao invés de acumular pontos apenas no cartão de crédito, como muitos fazem, poderá também receber do estabelecimento, incentivando o cliente a voltar. Isso faz com que o estabelecimento tenha um feedback mais assertivo daquilo que não só o cliente consumiu, mas o que seus amigos ou colegas consomem. Na realização do pedido de cada membro da mesa irá exibir opções como, mãe, pai, namorado(a), esposo(a), amigo(a), colega de trabalho, etc. Com essas informações, o estabelecimento poderá oferecer convites, promoções como dia das mães ou dos pais e com

base em seu histórico conseguir oferecer um atendimento de excelência. Este processo não pode prever tudo que o cliente deseja, mas sugerir algo com base em suas últimas compras.

Este trabalho está distribuído da seguinte maneira: No capítulo 1 é explanado sobre a introdução deste projeto e suas informações básicas, além de expor o tema e o problema da comunicação entre o cliente e o estabelecimento no qual esse projeto pretende solucionar. Também, apresenta a hipótese do que se espera desse projeto, a justificativa pela qual o tema foi escolhido e os reais objetivos deste projeto. O capítulo 2 descreve o tipo de metodologia escolhida para esse projeto, suas ferramentas e a forma que está sendo buscada as informações que são a base para esse projeto. O capítulo 3 apresenta o referencial teórico, tudo aquilo que é usado nesse projeto, as ferramentas e as tecnologias. O capítulo 4 apresenta o desenvolvimento, ilustrações, informações que descrevem o comportamento do sistema, seus recursos, suas telas e as principais funções do sistema. O capítulo 5 apresenta o cronograma utilizado para o desenvolvimento deste projeto, portanto, nele é encontrado exatamente as semanas na qual cada tarefa foi executada. Por fim, o capítulo 6 apresenta as referências utilizadas como base para o desenvolvimento deste trabalho.

1.1. TEMA

Desenvolvimento de sistema para realização de reserva de uma mesa em estabelecimentos e check-in, com base na metodologia ágil.

1.2. PROBLEMA

Como proporcionar maior comodidade ao cliente, evitando fila, constrangimento e atraso em seus pedidos e pagamentos, fazendo uso da tecnologia?

1.3. HIPÓTESE

Oferecer um cardápio online, atualizado pelo próprio estabelecimento e ainda com a condição de reservar sua mesa, pagar sua conta de uma forma ágil, permitindo que o cliente escolha a melhor forma de pagamento. Além de tornar a comunicação mais clara e objetiva entre o estabelecimento e o cliente, permitindo fazer uso da tecnologia sem a necessidade de altos investimentos.

1.4. JUSTIFICATIVA

Esse tema foi escolhido com o objetivo de melhorar os serviços oferecidos ao cliente pelo estabelecimento, possibilitando obter uma gama de informações através de seu uso como: o que mais foi vendido, o que as pessoas consomem no que diz respeito a bebidas, comidas, entre outros, e com isso, oferecer serviços ainda melhores. Há restaurantes, pizzarias e outros estabelecimentos que não conseguem oferecer uma total atenção. Na maioria das vezes atrasam o pedido, passam um pedido na frente do outro e para piorar a situação ainda forçam o cliente a enfrentar uma enorme fila. Por este motivo, investir no relacionamento entre o cliente e o estabelecimento é uma boa solução, sem contar nos dados históricos que o estabelecimento poderá armazenar de seus clientes.

1.5. OBJETIVOS

1.5.1. GERAL

Promover um atendimento ágil por parte dos estabelecimentos, evitando filas e constrangimentos.

1.5.2. ESPECÍFICOS

Desenvolver um sistema capaz de auxiliar na comunicação entre o cliente e o estabelecimento, permitindo que seja feita de uma maneira mais intuitiva.

Oferecer comodidade e condições especiais, específicas para cada perfil de usuário cadastrado.

Permitir que o cliente faça uma breve avaliação do estabelecimento, a fim de eliminar boa parte com gastos com impressão, como aqueles formulários impressos que ficam perto do caixa e que nem todos preenchem.

2. METODOLOGIA

2.1. PESQUISA QUALITATIVA

Para a realização deste trabalho foi utilizado o tipo de Pesquisa Qualitativa, que mais se relaciona com o objetivo deste projeto. Isso se dá pelo fato dela considerar a familiaridade que há entre o universo real com o que está sendo estudado.

Além disso, permite que possamos manter o foco na qualidade, uma vez que, só é possível saber se um software é bom e aceitável se lhe for agregado algum valor TELES (2005). Com isso, pode-se atender as expectativas dos clientes e providenciar as modificações conforme a realização dos testes no decorrer do desenvolvimento.

2.2. FERRAMENTAS DE PESQUISA

Observação do participante: Conta com a experiência pessoal do desenvolvedor e também do orientador deste trabalho, informações de clientes que reclamam da falta de organização em alguns estabelecimentos, dificuldade para reservar uma mesa, principalmente, em datas comemorativas.

Entrevista: Gerentes e clientes de vários estabelecimentos relataram em entrevista que gostariam de ter uma nova forma de interagir entre ambas as partes (GRAF.1), cerca de 70% disseram que se faz necessário a intervenção, porém, 30% disseram que é desejável, isso porque alguns acreditam que manter a forma como está não causará prejuízo porque vem dando certo, mas será questão de tempo até que os clientes comecem a escolher o estabelecimento que lhe possa oferecer maior comodidade. Nenhum dos entrevistados acharam que a proposta não seria relevante.

GRÁFICO 1 – Nível de aceitação da tecnologia em estabelecimentos



Fonte: Próprio autor (2016).

Com base nas entrevistas, dar a possibilidade dos próprios clientes registrarem suas mesas, realizarem seus pedidos e pagarem suas contas de forma independente do estabelecimento pode ser uma boa solução (GRAF.2). Entre 10 pessoas 2 dizem ter alto grau de dificuldade para reservar uma mesa, 2 dizem ter dificuldades mas ainda é tolerável, outras 4 pessoas dizem que as vezes possuem dificuldades.

Entre 10 pessoas, 5 acreditam que, a forma como se realiza o pedido hoje é com dificuldade, mas ainda é tolerável, 3 dizem que as vezes possuem dificuldades, 1 diz ter alto grau de dificuldade, 1 diz não ter dificuldade alguma.

Agora, entre as pessoas que dizem, o fato de ter que pegar fila para realizar o pagamento torna o processo mais cansativo. Entre 10 pessoas 3 concordam, 4 dizem que nem sempre precisam pegar fila, 2 pessoas dizem que as vezes possuem dificuldade mas é tolerável, apenas uma pessoa não quis responder essa questão. Dessa forma, permitir aproveitar o tempo ao máximo sem se preocupar em enfrentar filas enormes e fazer com que o cliente tenha o prazer de visitar o estabelecimento e levar sua família e seus amigos.

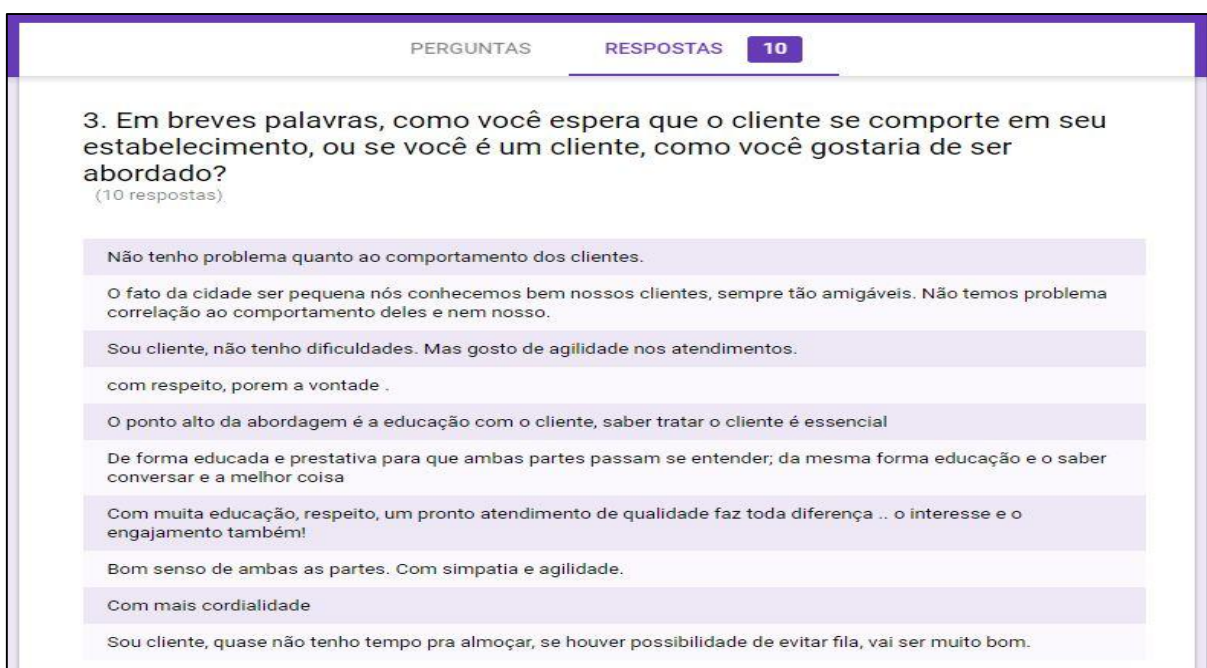
GRÁFICO 2 – Níveis de dificuldades encontradas hoje em estabelecimentos.



Fonte: Próprio autor (2016).

Os entrevistados expuseram uma breve opinião sobre como a empresa espera que o cliente se comporte e como o cliente espera ser abordado pelos atendentes (GRAF.3).

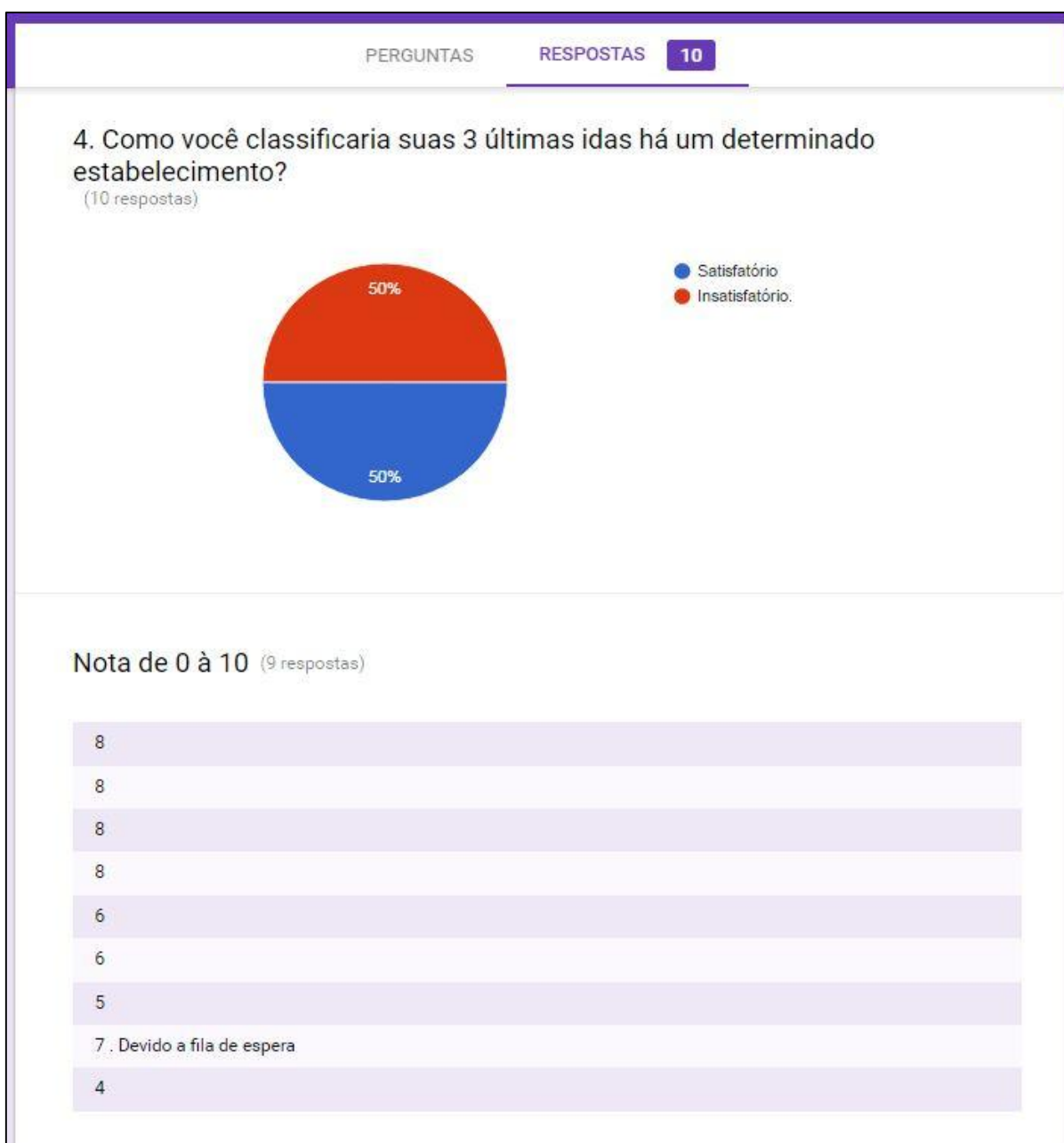
GRÁFICO 3 – Opinião dos entrevistados sobre o relacionamento entre ambas as partes.



Fonte: Próprio autor (2016).

Foi solicitado aos entrevistados, independentemente de serem gerentes ou funcionários, uma nota sobre a experiência obtida em um estabelecimento. Tendo em vista a esta visita, 50% disseram que estão satisfeitos com o atendimento das três últimas vezes que compareceram há um determinado estabelecimento, outros 50% disseram ter ficado insatisfeitos (GRAF.4). No final do formulário foi solicitado ao entrevistado que atribuisse uma nota de 0 à 10 levando em consideração toda a comunicação que há hoje entre o estabelecimento e o cliente. Entre os 10 entrevistados 9 se manifestaram, apenas uma pessoa não quis se manifestar. A média de satisfação é de 6,66 pontos.

GRÁFICO 4– Avaliação dos estabelecimentos na visão dos próprios clientes



Fonte: Próprio autor (2016).

Questionário: O formulário utilizado na entrevista se encontra no ANEXO 1. Foi realizado com 10 (dez) clientes que frequentam vários estabelecimentos na qual este trabalho visa propor soluções, estes responderam o formulário por e-mail, por telefone e presencial. Em tal questionário, eles expuseram a opinião sobre a dificuldade que é economizar tempo quando se está em um determinado estabelecimento e o tempo que se gasta enfrentando fila para pagar a conta.

2.3. MÉTODOS DE PESQUISA

Pesquisa Descritiva: Foi escolhido esse tipo de pesquisa pelo fato deste trabalho estar se tratando não só da solução para o tempo a ser economizado em todo o processo mas, por estar ligado diretamente com a relação entre o estabelecimento e seus clientes. Além disso, pelo fato dessas informações serem obtidas através de questionários e observação.

No próximo capítulo será apresentado um conceito sobre as tecnologias utilizadas neste projeto e a melhor forma encontrada de usá-las.

3. REFERENCIAL TEÓRICO

A literatura científica que deu suporte a esta pesquisa foi embasada na apresentação de novos métodos considerados rápidos e vantajosos para o desenvolvimento de softwares TELES (2005). Tais práticas, trazem consigo um conjunto de valores, princípios e práticas que permitem uma melhoria contínua SATO (2007).

Para isso, se fez necessário a quebra de alguns paradigmas mentais, conceitos que antes eram tidos como obrigatórios e que hoje vem se tornando opcionais pelo fato do objetivo ser o desenvolvimento do sistema, ao invés de documentações longas e confusas.

3.1. METODOLOGIAS ÁGEIS

A expressão “Metodologias Ágeis” ficou conhecida em 2001 DEVMEDIA (2016), quando vários especialistas estudaram os métodos Scrum e Extreme Programming (XP) e com isso definiram alguns princípios TELES (2005).

Tais princípios existem para utilizar-se como canal entre valores e práticas, não sendo os únicos a serem utilizados no desenvolvimento de software MANIFESTO AGIL (2016). Por exemplo, o princípio do humanismo, que sugere a comunicação entre as pessoas como uma melhor forma de comunicar-se. Diferente da comunicação escrita que em quase todos os casos são ignoradas, não incentivando o aprimoramento na comunicação. O princípio da simplicidade, com o objetivo de desenvolver o código utilizando o menor número de métodos e classes. O princípio do *feedback*, com o objetivo de buscar informações para uma melhoria. E por fim, o princípio da coragem, em que a equipe precisa de coragem para exigir e conseguir diariamente um *feedback* do cliente DESENVOLVIMENTO MÉTODO ÁGIL (2016b).

Se todos os envolvidos no projeto estiverem concentrados naquilo que é importante para o projeto, então, conseguirão obter o resultado esperado DESENVOLVIMENTO MÉTODO ÁGIL (2016^a). Para isso, a XP defende cinco valores relacionados diretamente com os princípios citados anteriormente.

3.2. EXTREME PROGRAMMING (XP)

É uma metodologia cujo o objetivo é realizar os processos de uma maneira mais ágil, de certa forma, define métodos, dos quais, dão prioridade ao trabalho a ser realizado, ao invés, de dedicar grande parte do tempo gerando relatórios.

No entanto, para alcançar o objetivo esperado fazendo uso dessa metodologia é necessário entender os cinco valores que a XP prega. Tais quais:

- **Comunicação:** Existem diversas formas de se realizar a comunicação, porém, a mais eficaz é quando duas pessoas realizam um diálogo presencial. Permite que sejam analisados os gestos, a postura, as expressões faciais, o tom de voz, as emoções, entre outros.
- **Coragem:** Ouvir atentamente o cliente e não impedir a criatividade dele é algo muito importante. Porém, equipes XP consideram este comportamento impossível de ser evitada e procura realizar as mudanças com segurança e coragem. É necessário acreditar em seus meios de proteção, como desenvolvimento orientado a testes, programação realizada por duas pessoas alternando apenas o teclado e assegurar que o código permaneça limpo e bem estruturado.
- **Feedback:** Defende que a busca pelo feedback permite tornar o projeto mais econômico, pelo fato dos erros poderem serem detectados e corrigidos rapidamente.
- **Respeito:** Importar-se uns com os outros dentro da equipe é a melhor forma de construir e manter este valor e isso representa um ganho muito grande para o desenvolvido do projeto. Assim, cada um saberá ouvir, compreender e respeitar o ponto de vista do outro que é considerado importante para que o software seja construído com sucesso.
- **Simplicidade:** Traz consigo o conceito *Just in Time* DESENVOLVIMENTO MÉTODO ÁGIL (2016c), permitir que a equipe se preocupe apenas com o que é necessário no momento.

Segundo SCHOEPPING (2012), as práticas que o XP apresenta não são inéditas e nem originais, mas são responsáveis por fazerem os valores e princípios funcionarem. Não tem como dizer qual será a melhor, porque dependem diretamente da forma como é visto a oportunidade de melhoria no projeto. Por isso, BECK (1999) cita que, a franqueza de uma

prática é compensada pela força de outra, ambas trabalham em conjunto. Tais praticas se dividem em duas:

- **Práticas primárias:** Se dá na organização e pode ser aplicada com segurança no início do projeto. Em determinada situação, a pessoa se vê perdida ao encontrar diversos erros, então, o objetivo dessa parte é auxiliar nessa tomada de decisão, cujo foco está na organização das ideias.
- **Práticas corolárias:** Esta etapa é considerada a mais difícil de ser implementada se principalmente for realizada sem antes ter passado pelas práticas primárias BECK (2004). Desenvolver o software de qualquer maneira não irá garantir uma boa estruturação e sim a possibilidade de encontrar erros dos quais irão prejudicar o funcionamento do sistema e com isso torná-lo inviável. Esta prática é a que terá o real envolvimento do cliente, realizando testes e obtendo respostas do mesmo.

Algumas atitudes devem ser tomadas a fim de tornar o processo mais conciso, de forma a economizar tempo com palavras e investir mais na prática. Segundo POPPENDIECK & POPPENDIECK (2003), quando a entrega do serviço acontece em períodos mais curtos, duas semanas por exemplo, mudanças de rumo podem ser tomadas mais rapidamente, permitindo adiar decisões sem que isso gere consequências indesejáveis. A XP, assim como SCHOEPPING (2012) traz exemplos de algumas atitudes:

1. Reunir todos os envolvidos no projeto em uma sala para realizar uma breve reunião, onde o objetivo será determinar as atividades e discutir sobre os resultados esperados.
2. Ter uma equipe completa, contratando pessoas que tenham as habilidades necessárias. Da mesma forma, a pessoa cuja as habilidades não forem mais úteis ao projeto deve deixar a equipe.
3. Tornar o ambiente de trabalho mais comunicativo, utilizar diagramas ou até mesmo colocar breves cartões de história BLOG DO ABU (2016) de usuários na parede.
4. Não fazer hora-extra, porque ela pode trazer benefícios a curto prazo, porém, isso vai desgastando as pessoas e influenciando de forma negativa a produção dela e até mesmo de toda a equipe.

A maioria dos conceitos da XP causa desentendimento entre os que defendem os princípios que cercam a forma clássica, muitas vezes esses conceitos não fazem sentido se aplicados separadamente. Entre tantos pontos da XP que diferenciam da forma do desenvolvimento tradicional, um que mais se destaca é o *feedback* a todo o momento, incentivando mais a comunicação entre as pessoas, tornando o processo mais simples e ágil.

3.3. METODOLOGIA DE DESENVOLVIMENTO

Todo esse projeto foi desenvolvido usando o Ruby além de utilizar o Ruby On Rails que é o *framework*. RUBY ON RAILS (2016), criado por Yukihiro Matsumoto em 1993 que teve início ao público apenas a partir de 1995 RUBY (2016). Essa linguagem traz várias vantagens como ser considerada orientada a objetos, fortemente tipada e dinâmica. Tais vantagens serão destacadas neste trabalho por serem consideradas mais relevantes. De forma resumida, Ruby foi criada para ser uma linguagem muito simples de ler e ser entendida para economizar não só no desenvolvimento mas na manutenção. Para fazermos uso de tal linguagem será necessário a instalação de um interpretador, já que é uma linguagem interpretada.

Existia um interpretador criado pelo próprio criador do Ruby, simples, sem nenhum gerenciador de memória complexo, muito menos a compilação em tempo de execução. No entanto, a versão 2.0 é a mais utilizada, também conhecida como *Yet Another Ruby VM* (YARV) SASADA (2016), que já possui uma máquina virtual e outros recursos mais avançados.

Este projeto usa como base Extreme Programming (XP) cujo criador deste método se chama Kent Beck, tendo dado início a este conceito na década de 90 quando um projeto chamado *Chrysler – Comprehensive Compensation System (C3)* estava acontecendo SCHOEPPING (2012). BECK(1999) publicou o livro *Extreme Programming (XP)* cujo objetivo é definir métodos de desenvolvimento de *software* focado em projetos realizados por equipes menores e onde as informações como os requisitos ainda se tornam vagas. Uma metodologia ágil cuja prioridade não é uma documentação detalhada, e sim, o *software* em funcionamento, no entanto, não descarta o uso da documentação mas prioriza o tempo.

Conforme TELES (2005), o tempo é como o inimigo de todos os projetos, uma vez que o escopo impacta na duração do projeto, ambos estão associados. Minimizando o escopo, o tempo é reduzido e portanto as chances de sucesso crescem.

A Metodologia XP detalha alguns conceitos, tais como, a interação entre as pessoas, ao contrário de uso de processos e ferramentas; o *software* executando, ao invés de uma documentação longa; respostas rápidas para as mudanças, diferente da forma clássica que é seguir planos antes estabelecidos.

Essa metodologia vive em um universo oposto aos conceitos pregados pelos grandes desenvolvedores e analistas tradicionais, substituindo as muitas papeladas pelo próprio código-fonte do sistema, o que exige um código simplesmente limpo e de fácil entendimento. Todos estão envolvidos no projeto e tem como o principal objetivo, fazer com que a pessoa trabalhe naquilo que ela seja boa, o que garante um melhor desempenho e produtividade.

De maneira a entendermos este contexto, temos o exemplo de um dos maiores desafios para a empresa Toyota que deixou de seguir o conceito apresentado por Taylor, partindo da ideia de que a melhor forma de realizar uma tarefa é ela não ser estática, mas sim dinâmica TELES (2005).

Decidiu investir nesse método de desenvolvimento e isso gerou grande polêmica a pouco mais de 50 anos atrás. Desenvolvedores, analistas e profissionais que antes estavam habituados a enxergarem apenas a sua forma tradicional não quiseram acreditar, pois o conceito *Just in Time* não era tão conhecido e poucos conseguiam ver o seu valor. Pois bem, hoje a Toyota é uma das indústrias automobilísticas que mais lucram no mundo, ficou entre as 10 maiores marcas que mais cresceram no primeiro trimestre de 2016G1 - AUTO ESPORTE (2016), pois acredita que produzir algo que ninguém irá utilizar é desperdício.

Para TELES (2005) desperdício é tudo aquilo que não adiciona valor ao produto, valor tal como percebido pelo cliente. E para POPPENDIECK & POPPENDIECK (2003), tradução TELES (2005), o ideal é descobrir o que o cliente deseja e então fazer ou desenvolver isso e entregar exatamente o que ele quer, virtualmente de imediato. O que quer que atrapalhe a rápida satisfação da necessidade do cliente é um desperdício.

A Toyota percebeu aquilo que hoje é também pregado pela XP, não produzir mais que o necessário, saber o momento de cada fase para que a equipe consiga obter o sucesso. Para BABA (2008), o Diagrama de Pareto utilizado para dados qualitativos é a melhor opção quando se quer definir priorização de temas, isso é, concentrar esforços naquilo que se pode alcançar maior ganho. E para XP, o Princípio de Pareto também se aplica ao desenvolvimento de sistemas, onde 20% das funcionalidades costumam alcançar 80% ou mais do benefício esperado DESENVOLVIMENTO MÉTODO ÁGIL (2016d).

Como dito anteriormente, o XP não abre mão da documentação, apenas entende de uma forma diferente, por isso exige um código mais limpo e de fácil entendimento. O código

é considerado a melhor documentação que possa ter em um projeto, pois o foco deve estar no desenvolvimento o que traz novos conceitos e entende que nada pode ser feito se não for com base na história escrita pelo cliente. História em XP substitui o que tradicionalmente é conhecido como requisitos ou casos de usos.

3.3.1. CARTÃO DE HISTÓRIA

Para BECK (1999), uma forma de expressar os requisitos do sistema do ponto de vista do cliente é coletar informações na presença do desenvolvedor e formular o “Cartão de História” TELES (2005) que contém as informações mais importantes “do que será feito” sem se preocupar “como”, podendo ser aplicada em qualquer metodologia ágil. Ainda, segundo TELES (2005), cabe aos membros do projeto assegurar que as funcionalidades que têm que ser feitas sejam implementadas em primeiro lugar.

Para obter um Cartão de História completo é preciso possuir três itens:

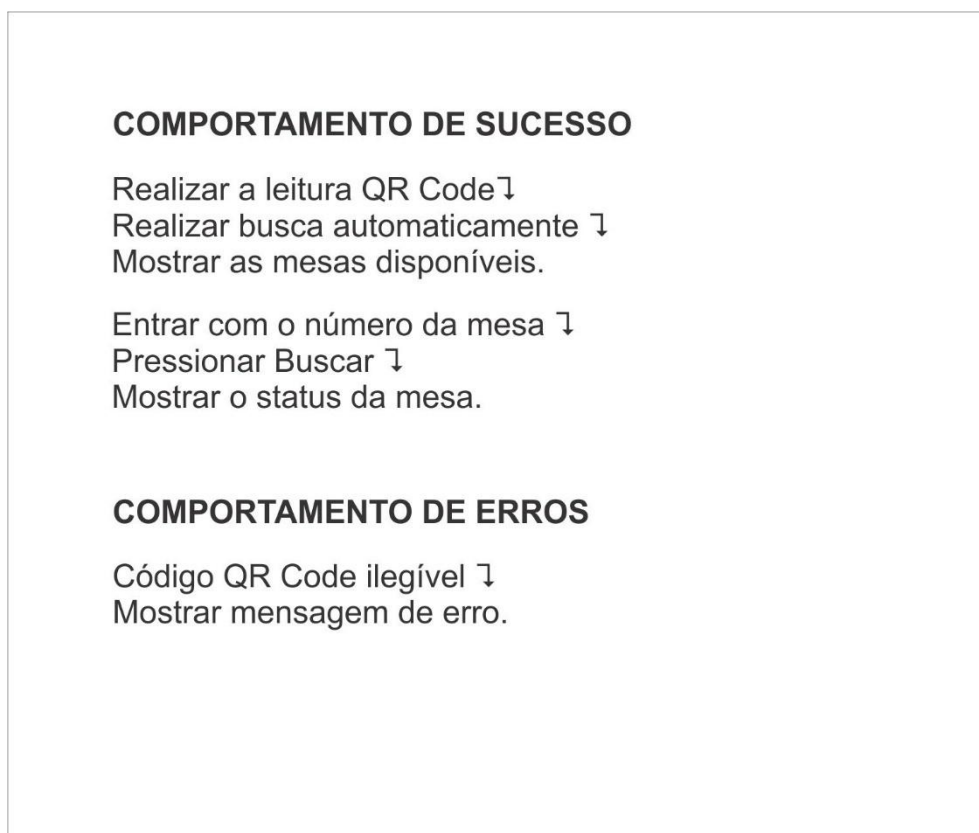
- **Informações do cartão:** Nesta primeira parte é importante que esteja em destaque o nome, a descrição da história, um número como referência, um título, etc.
- **Dados de conversa com o usuário:** Nesta etapa, deverá constar a história contada pelo próprio usuário e nada de informações técnicas, apenas um linguajar claro, de forma que todos os envolvidos no projeto consigam ler e entender. Essas informações serão base para obtenção de maiores detalhes no decorrer do desenvolvimento.
- **Testes de validação:** Pra saber se um cartão de história está correto, ele precisa dar condições de testá-lo e a forma de realizarmos estes testes se dá em analisar o layout construído a partir da história, testar cada ação e no final separar em duas colunas, comportamento de sucesso e erros (FIG.1).

Para realizar o teste de validação é preciso simular uma determinada ação e imaginar o comportamento do sistema. Para dar este exemplo é preciso ficar claro que, o cliente obterá o código QR Code no momento que realizar a reserva da mesa, podendo levar impresso ou apenas acessar o sistema pelo celular e apresenta-lo para leitura, caso não consiga apresentar o

código, basta apenas informar o CPF para a consulta de seu cadastro e reserva. Levando em consideração que o cliente tenha esse código impresso, tende os seguintes testes:

- **Primeiro passo:** Realizar a leitura do QR Code.
- **Segundo passo:** Realizar a busca com base nesse código obtido, tal qual, será realizada de uma maneira automática após a obtenção dessa referência.
- **Terceiro e último passo:** O sistema irá mostrar o número de mesas disponíveis. Da mesma forma será testado o comportamento de erros. Por exemplo: Ao tentar realizar a leitura do código QR Code, posicionando a câmera do celular frente ao código o sistema não conseguiu identificá-lo, neste caso é apresentado uma mensagem de erro informando ao cliente que, o sistema não foi capaz de realizar a leitura do código. Outra alternativa é realizar a pesquisa do cadastro do cliente pelo CPF.

FIG. 1 Exemplo de testes de validação.



Fonte: Próprio autor (2016).

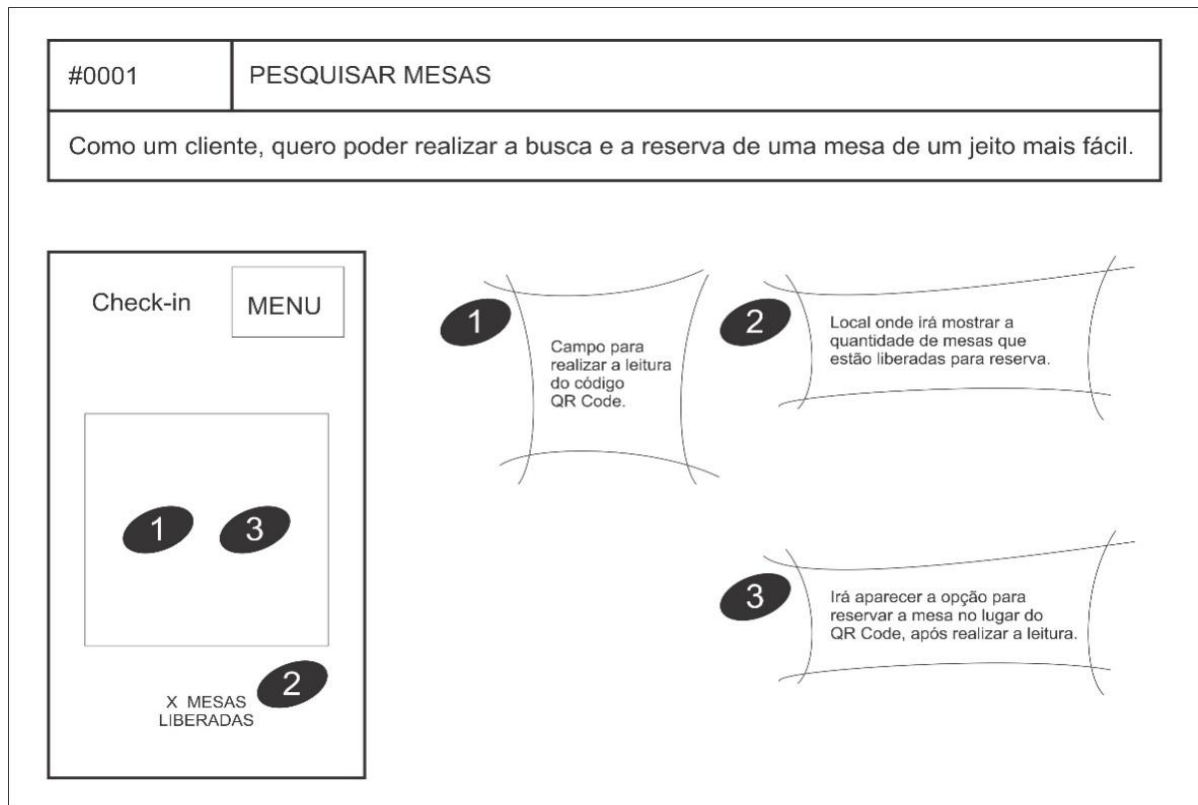
Não se pode esquecer que, na frente do cartão deverá constar o layout básico com base na história(FIG.2) e no verso do cartão apenas os testes de validação.

É necessário informar um número no qual possa ser a referência do cartão de história, ao lado direito desse número informar um nome curto e em baixo descrever uma pequena descrição com base na história contada pelo cliente. Como mostra a figura (FIG.2), pode ser observado que essa descrição da história começa com as seguintes palavras: Como um cliente, quero.... Depois de criar esse cabeçalho do cartão com as devidas informações o restante do espaço em branco é livre para a criação do layout e suas devidas anotações. Como mostra a figura abaixo, sem informações técnicas, tudo deve ser desenvolvido com o intuito de que todos da equipe consiga entender.

Além da referência do cartão, existem as referências para o entendimento do layout, com isso, economizar espaço e realizar um trabalho mais organizado. Como, por exemplo, a descrição para as referências 1, 2 e 3. Na figura 2, a referência ao número 1 é relacionada ao campo para realizar a leitura do código QR Code. O número 2 trata do local onde irá mostrar a quantidade de mesas que estão liberadas para reserva e no 3 irá aparecer a opção para reservar a mesa no lugar do QR Code, após realizar a leitura os identificadores 1, 2 e 3 contidos no layout é uma rápida descrição, que está ligado ao modelo desenhado. Isso permite aproveitar mais o espaço e tornar o cartão mais organizado. Por exemplo:

- 1) Campo para realizar a leitura do código QR Code, podendo também digitá-lo;
- 2) Local onde irá mostrar a quantidade de mesas que estão liberadas para reserva.
- 3) Irá aparecer a opção para reservar a mesa no lugar do QR Code, após realizar a leitura.

FIG. 2 Frente de um cartão de história.



Fonte: Próprio autor (2016).

Para o desenvolvimento desse sistema será utilizado algumas ferramentas:

- Ruby On Rails v4.2.3: Framework de código aberto (*open source*) que promete uma maior facilidade e agilidade no desenvolvimento de sistemas Web RUBY ON RAILS (2016). Ele traz uma estrutura pré-definida, moldada com base no padrão de arquitetura *Model-View-Controller (MVC)* DUARTE (2011).
- Sublime Text v3: Ferramenta de desenvolvimento *Integrated Development Environment (IDE)* utilizado para desenvolvimento não só da linguagem Ruby, e sim, para várias SUBLIME TEXT (2016).
- Twitter Bootstrap Rails v3: Framework desenvolvido em *Cascading Style Sheet (CSS)* e Javascript, com o foco no design, fazendo uso da linguagem *Hypertext Markup Language (HTML)* e que cumpre com a mesma função do Bootstrap, também um framework simples e prático BOOTSTRAP (2016).

- MySQL v5.7.12.0: De acordo com JOBSTRAIBIZER (2009), MySQL é um dos bancos de dados mais utilizados no mundo e que possui o código aberto (*open source*).

É necessário conhecer as ferramentas, portanto, segue abaixo uma breve descrição:

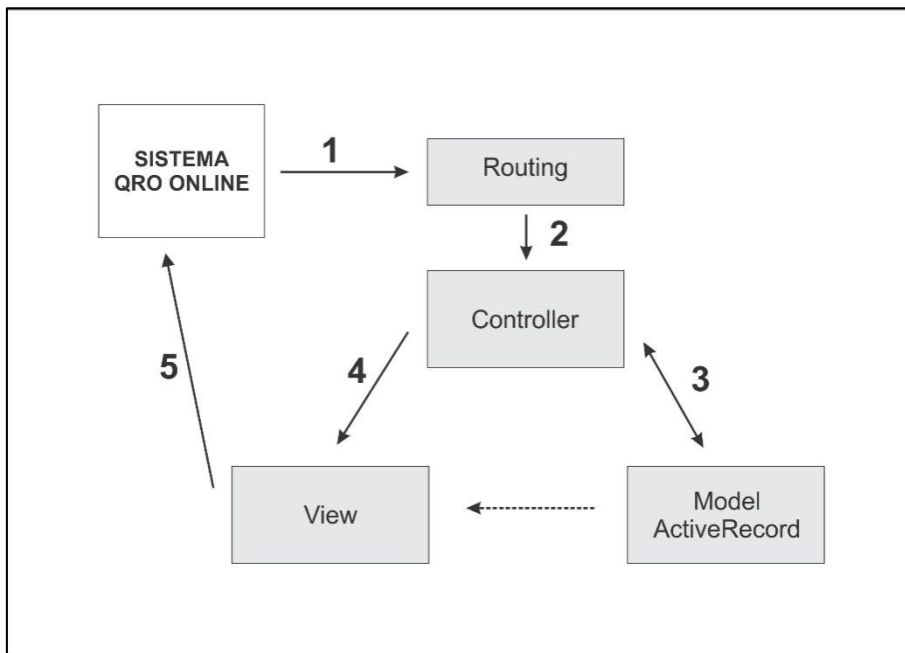
Ruby - Criada em 1993 no Japão por Yukiro Matsumoto, vista pelo mundo apenas em 1995. O que lhe fez ser tão popular é o seu poderoso *framework* Ruby On Rails. É uma linguagem de programação interpretada, isso permite um *feedback* da máquina ao desenvolvedor de uma maneira mais rápida, um bom exemplo de como isso funciona é quem já trabalha com *Hypertext Preprocessor (Php)* e *JavaScript*. Além de contar com recursos para processar arquivos como textos, ela é simples e direta. Tudo que for possível de ser manipulável em Ruby é objeto. Alguns recursos em destaque são:

- Sintaxe simples;
- Tratamento de exceções;
- É uma linguagem completa, orientada a objetos;
- Possível ser utilizada em diversos *Operational Systems (SO)*.

Ruby On Rails – Utiliza-se da linguagem Ruby, bem desenvolvido vem se tornando uma ferramenta muito prática, ajudando a construir sistemas *Web* de uma maneira mais fácil e rápido. Oferece também uma camada poderosíssima para a manipulação do banco de dados, conhecida como *Active Record*. Defende também o conceito de *don't repeat yourself (DRY)*, que é reutilização de códigos.

MVC – Arquitetura que tem por objetivo propor um padrão, conhecida também por *Model-View-Controller* ou simplesmente MVC. O (*Model*) representa o banco de dados e todo o modelo da aplicação, o (*View*) representa a visão, o tão conhecido HTML, sobrou o (*Controller*), responsável por realizar a comunicação entre os dois (FIG.3).

FIG. 3 MVC em Ruby On Rails.



Fonte: Próprio autor (2016).

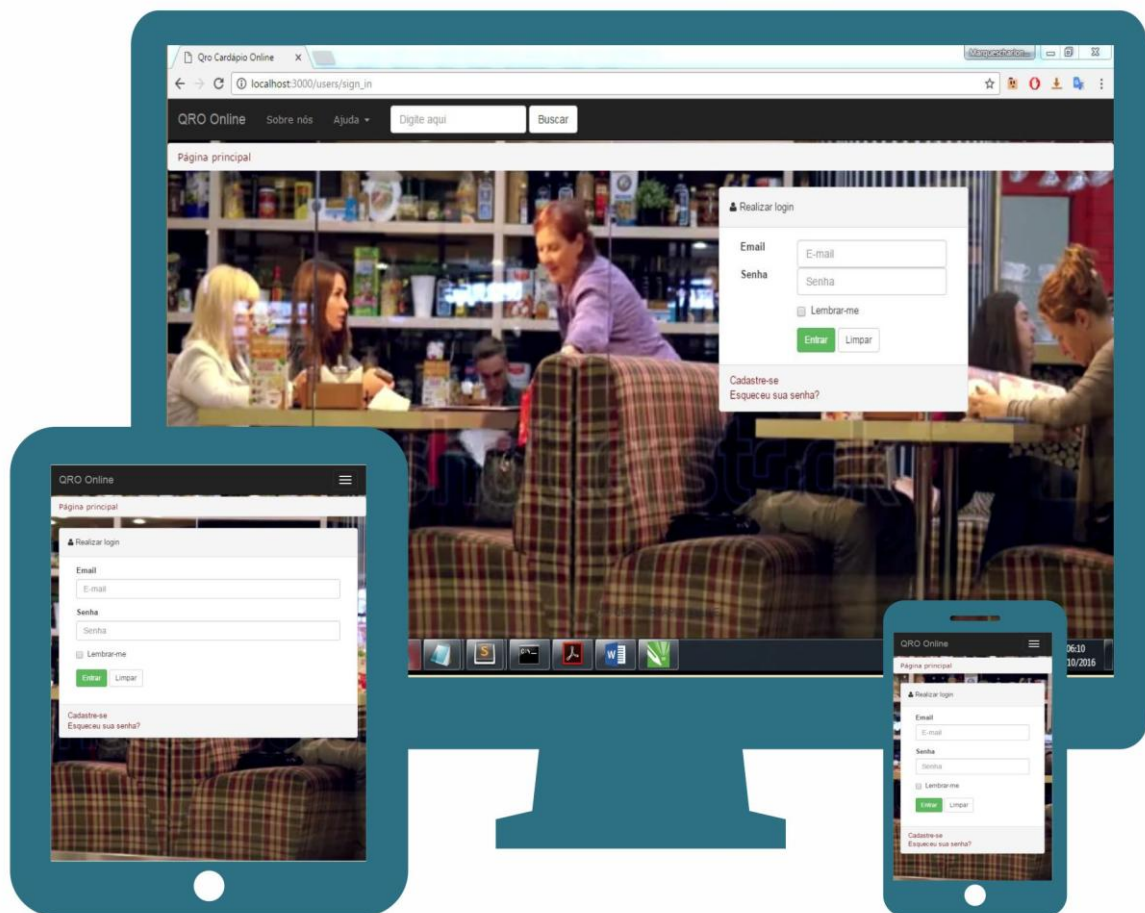
Rails possui diversas facilidades, uma delas em destaque é o (*generators*), onde realmente acontece toda a mágica. Para gerar uma aplicação necessário acessar o *prompt* de comando e digitar o seguinte comando *rails new <nome_do_projeto>*, será criada toda uma estrutura básica em MVC.

No próximo capítulo será explanado sobre o desenvolvimento do trabalho proposto neste documento.

4. DESENVOLVIMENTO

Neste trabalho, foi desenvolvido um sistema WEB responsivo, que segundo FERREIRA (2010) evita inúmeras dificuldades como intervenção técnica, a fim, de realizar instalação em cada máquina. O intuito é integrar o cliente com o estabelecimento oferecendo comodidade e agilidade para aqueles que apreciam ir a um restaurante, pizzaria, etc. O acesso ao sistema pode ocorrer por diversos aparelhos como, computador, tablet e smartphone (FIG.4). A diferença deste sistema para outros existentes no mercado é que este não oferecerá apenas opções do cliente usá-lo para pedir o seu lanche estando em casa, mas realizar o mesmo procedimento no local não dependendo mais do atendente chegar até a mesa e anotar o pedido, a fim, de tornar a comunicação ainda mais rápida e o tempo de espera mais curto.

FIG. 4 Sistema sendo acessado por vários dispositivos.



Fonte: Próprio autor (2016).

Imagine que hoje seja o dia dos namorados, você já visitou o estabelecimento algumas vezes e possui o cadastro no sistema. Por este motivo, você poderá ser informado, por e-mail ou por mensagem antecipadamente, sobre essa data especial, ficar por dentro das promoções e o mais importante, ser informado apenas sobre o que possa lhe ser mais relevante com base em suas últimas compras. Caso seja sua primeira visita ao estabelecimento e tenha o acesso ao sistema, será sugerido produtos que mais se consomem no dia e em períodos diferentes como, por exemplo, na semana ou durante o mês. Até o momento está sendo buscado apenas o conforto, mas o que será essencial para o cliente? Será necessário ter um Computador, Smartphone ou um Tablet cujo browser seja compatível, além, do acesso a internet, depois, será preciso apenas reservar sua mesa e solicitar o pedido, independentemente se está sozinho ou acompanhado de seus amigos ou familiares.

4.1. FUNCIONALIDADES DO SISTEMA

Até o momento foi descrito sobre as funções básicas do sistema, no entanto, é dado continuidade detalhando as funcionalidades principais do sistema, a começar então com o cadastro de usuário.

O sistema será em nuvem e o pré-requisito mínimo para a contratação é que, o estabelecimento tenha um computador, leitor de código QR Code, todavia, será possível utilizar o sistema por um tablet, smartphone até como plano de contingência, caso o computador dê algum problema. E se por algum motivo faltar a internet, é de responsabilidade do estabelecimento entrar em contato com a empresa prestadora de serviços de internet. A vantagem de seu funcionamento ser em nuvem é que, se a máquina estragar, poderá utilizar até um tablet para realizar a reserva ou então o Check-in, até que o leitor de QR Code seja implementado no sistema, para uso da câmera dos aparelhos, será necessário apenas realizar a pesquisa da reserva pelo CPF.

4.1.1. LOGIN, CADASTRO DE USUÁRIO E RECUPERAÇÃO DE SENHA

Pode ser observado que na página principal (FIG.5) é solicitado o usuário e a senha, podendo ser acessado pelo e-mail cadastrado. Caso ainda não tenha o cadastro é possível realizá-lo clicando no link “Cadastre-se” que fica logo abaixo do botão “Entrar”. Ao clicar no link cadastrar, é direcionado para outra página, na qual, será solicitado o e-mail e a senha a ser cadastrada (FIG.6). O sistema irá enviar uma solicitação de confirmação que após ser

confirmada pelo usuário, o sistema será liberado para o uso. Caso o usuário perca a senha, não será possível acessar o sistema, porém, terá a possibilidade de redefini-la, para isso, necessário apenas clicar no link “Esqueceu sua senha?”, logo é redirecionado para uma outra página (FIG.7), o sistema pede que o usuário informe o e-mail cadastrado, depois disso é só clicar no botão “Enviar instruções para redefinição da senha”. Então, o usuário recebe um e-mail com um link para a redefinição desta senha.

FIG. 5 Realizar Login.

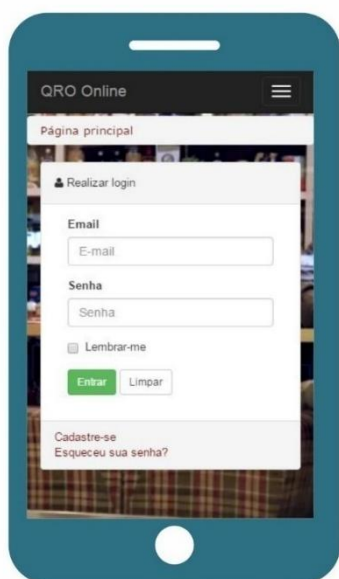


FIG. 6 Cadastro de usuário.

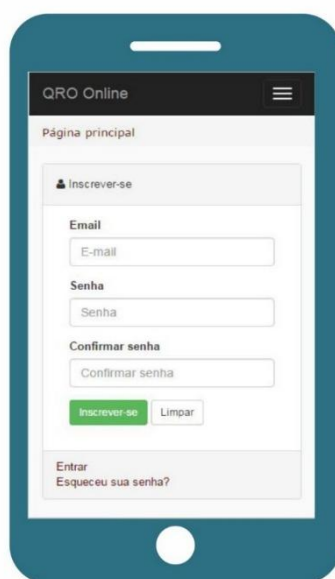
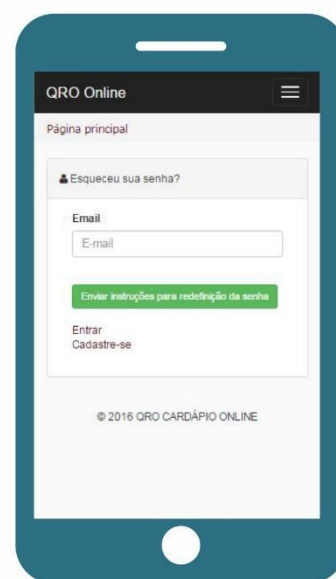


FIG. 7 Recuperar senha.



Fonte: Próprio autor (2016).

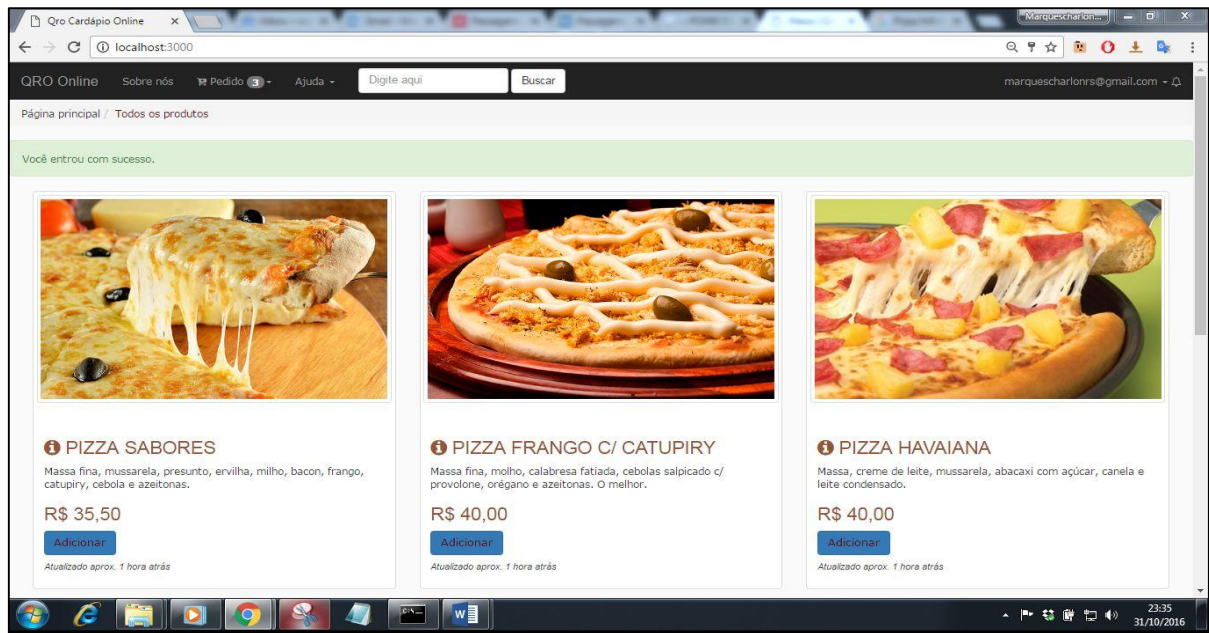
Fonte: Próprio autor (2016).

Fonte: Próprio autor (2016).

4.1.2. CADASTRO E VISUALIZAÇÃO DOS PRODUTOS

Após realizar o *login*, o sistema redireciona automaticamente para a tela de produtos (FIG.8), utilizar o cardápio online traz vantagem como a comodidade de poder realizar o pagamento através do próprio sistema, sem ter a necessidade de pegar fila, além de contar com a agilidade na entrega de seu pedido.

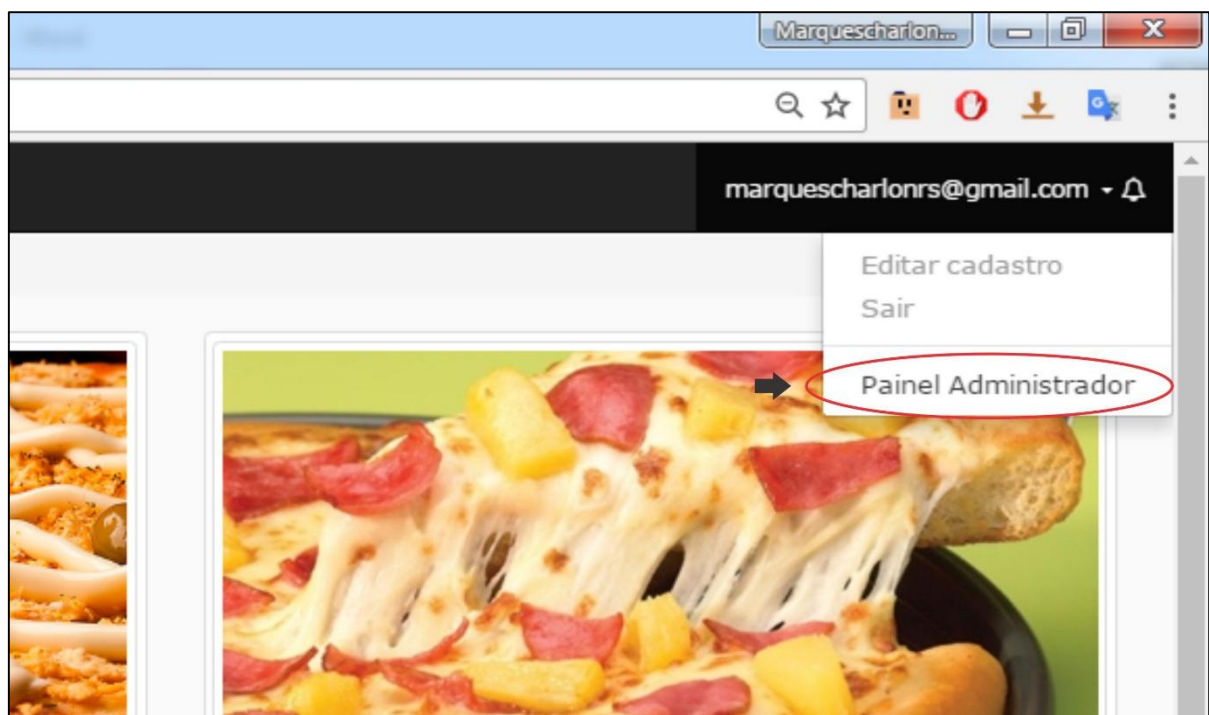
FIG. 8 Visualizar produtos.



Fonte: Próprio autor (2016).

Para o estabelecimento realizar o cadastro dos produtos é necessário acessar o “Painel Administrativo”, para isso basta clicar em “Painel Administrador” (FIG.9).

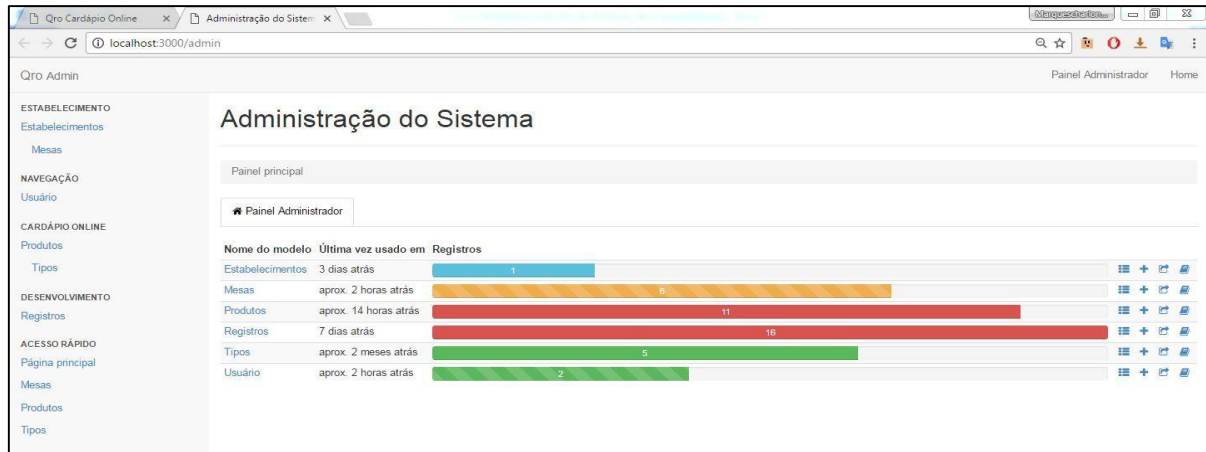
FIG. 9 Acessar Painel Administrativo



Fonte: Próprio autor (2016).

Após acessar o “Painel Administrativo” (FIG.10), então, basta clicar em “Produtos”, é listado os produtos já adicionados (FIG.11). Nesta janela possui uma barra de menus e para adicionar um novo produto é só clicar no menu “Criar novo” (FIG.12).

FIG. 10 Administração do sistema



Fonte: Próprio autor (2016).

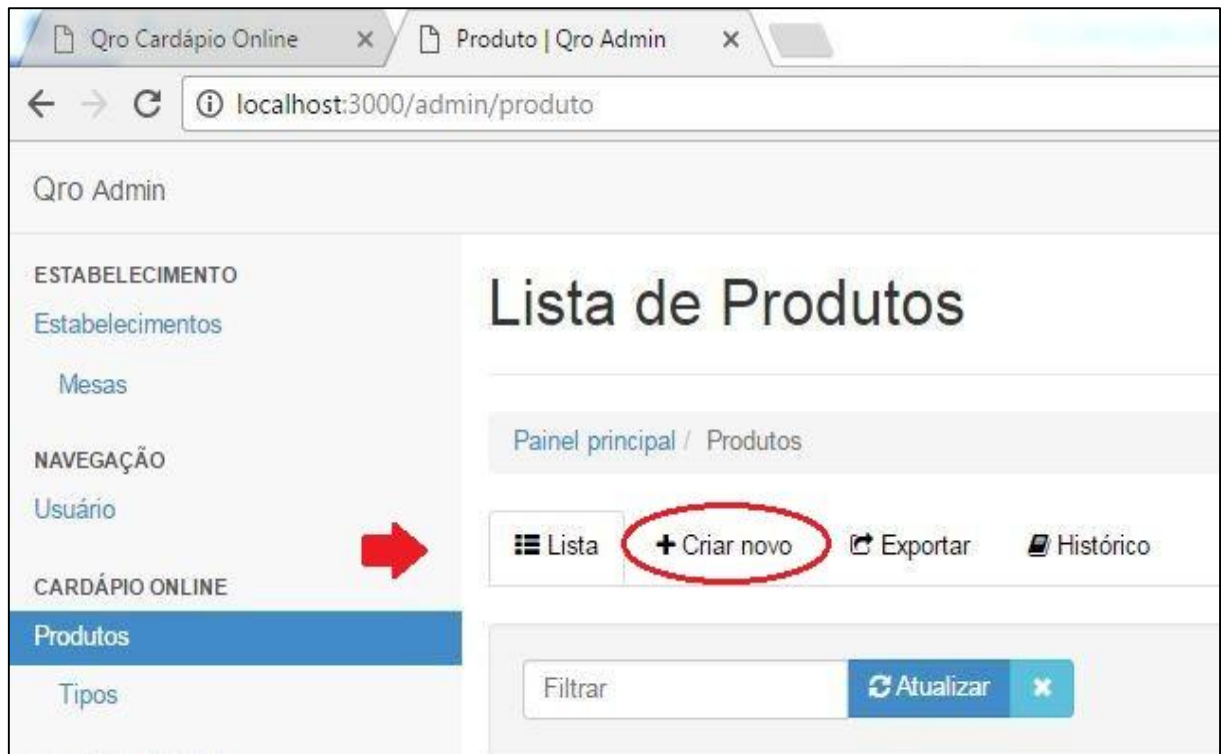
FIG. 11 Cadastrar produtos.

Id	Nome	Descrição	Tipo	Valor	Cupom
12	PIZZA ALHO	Massa, molho, mussarela, orégano, alho, tomate...	Tipo #3	45.0	-
11	PIZZA FRANGO C/ CATUPIRY	Massa fina, molho, calabresa fatiada, cebolas sa...	Tipo #3	40.0	-
10	PIZZA SABORES	Massa fina, mussarela, presunto, ervilha, milho, ...	Tipo #4	35.5	-
9	PIZZA ATUM	Massa, molho, mussarela, orégano, azeitona, at...	Tipo #4	20.0	-
8	PIZZA BANANA	Massa, creme de leite, mussarela, banana c/ aç...	Tipo #4	30.0	-
6	PIZZA HAWAIANA	Massa, creme de leite, mussarela, abacaxi com ...	Tipo #2	40.0	-
4	MODA DA CASA	Massa fina, mussarela, presunto, ervilha, milho, ...	Tipo #2	35.0	-
3	CALABRESA	Massa fina, molho, calabresa fatiada, cebolas sa...	Tipo #4	35.0	-
2	BACON	Massa fina, molho, mussarela, bacon, tomate, or...	Tipo #2	35.0	-
1	PIZZA 4 QUEIJOS	Molho, Mussarela, Provolone, Parmesão, Catupir...	Tipo #2	30.0	-

Fonte: Próprio autor (2016).

Ao clicar no menu “Criar novo” (FIG.12) o usuário poderá adicionar informações nos seguintes campos: Nome do produto, a descrição, o tipo, valor, imagem do produto e para deixa-lo visível para os clientes basta apenas marcar o produto como “Publicado” (FIG.13).

FIG. 12 Adicionar novo produto.



Fonte: Próprio autor (2016).

FIG. 13 Deixar o produto visível para os clientes.



Fonte: Próprio autor (2016).

4.1.3. RESERVADA MESA

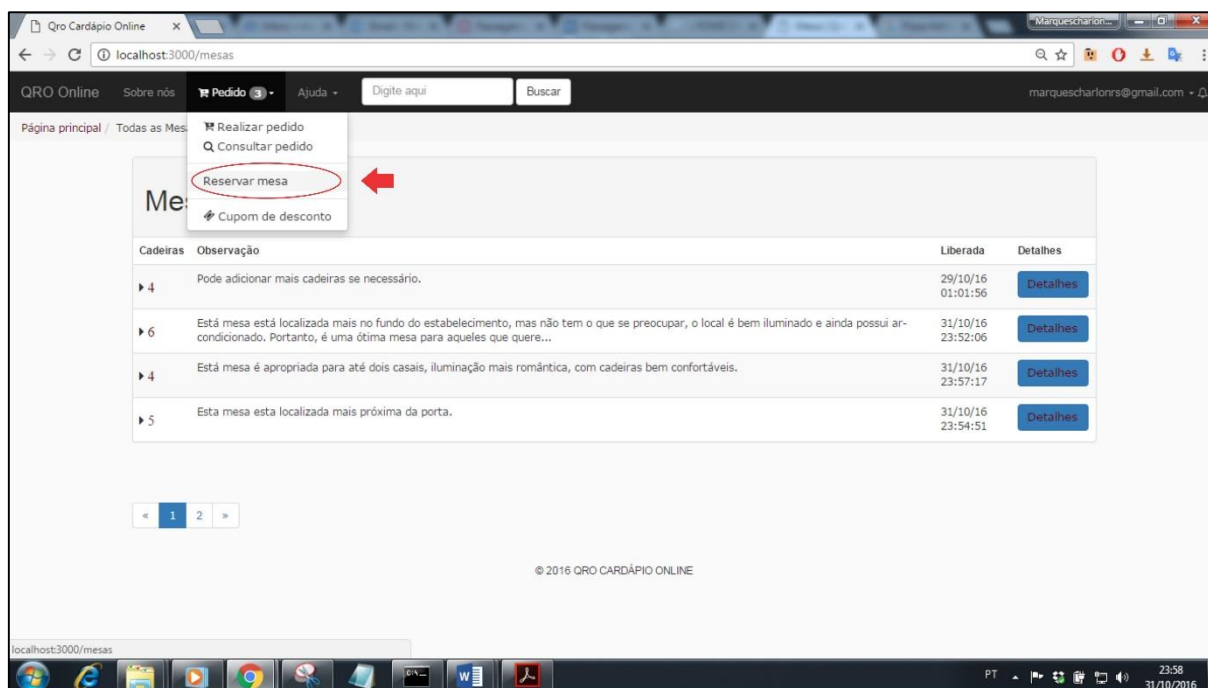
Se o objetivo do cliente é reservar uma mesa no restaurante e levar a família, pode realizar a reserva online. Ao chegar no estabelecimento o atendente realiza apenas um check-in, para isso, o cliente apresenta o QR Code em seu smartphone ou então impresso que é gerado ao realizar a reserva. Deixar para reservar a mesa quando chegar no restaurante estará sujeito a atrasos e até mesmo correr o risco de não haver mais mesas disponíveis.

Para realizar a reserva da mesa basta clicar no botão “Pedido”, em seguida “Reservar mesa (FIG.14). Em sua primeira tela é possível visualizar a quantidade de cadeiras disponíveis junto as mesas, as observações, a data que as mesas foram liberadas e o botão “Detalhes” que encaminha para o registro da mesa. Também, é possível pesquisar a mesa pelas observações, digitando no campo da caixa de busca.

A página Mesas possui também a paginação, isso é, limite de mesas a serem visualizadas por página, é automático, se o número de mesas cadastradas e liberadas foram maiores que quatro, então, é criada uma nova página, para visualizar é só se orientar pela barra criada pela própria paginação que fica no canto inferior da tela.

Com as mesas já listadas, o usuário pode escolher e clicar no botão “Detalhes”. Então, é redirecionado para uma página exclusivamente para a única mesa escolhida (FIG.15).

FIG.14 Status Mesa.

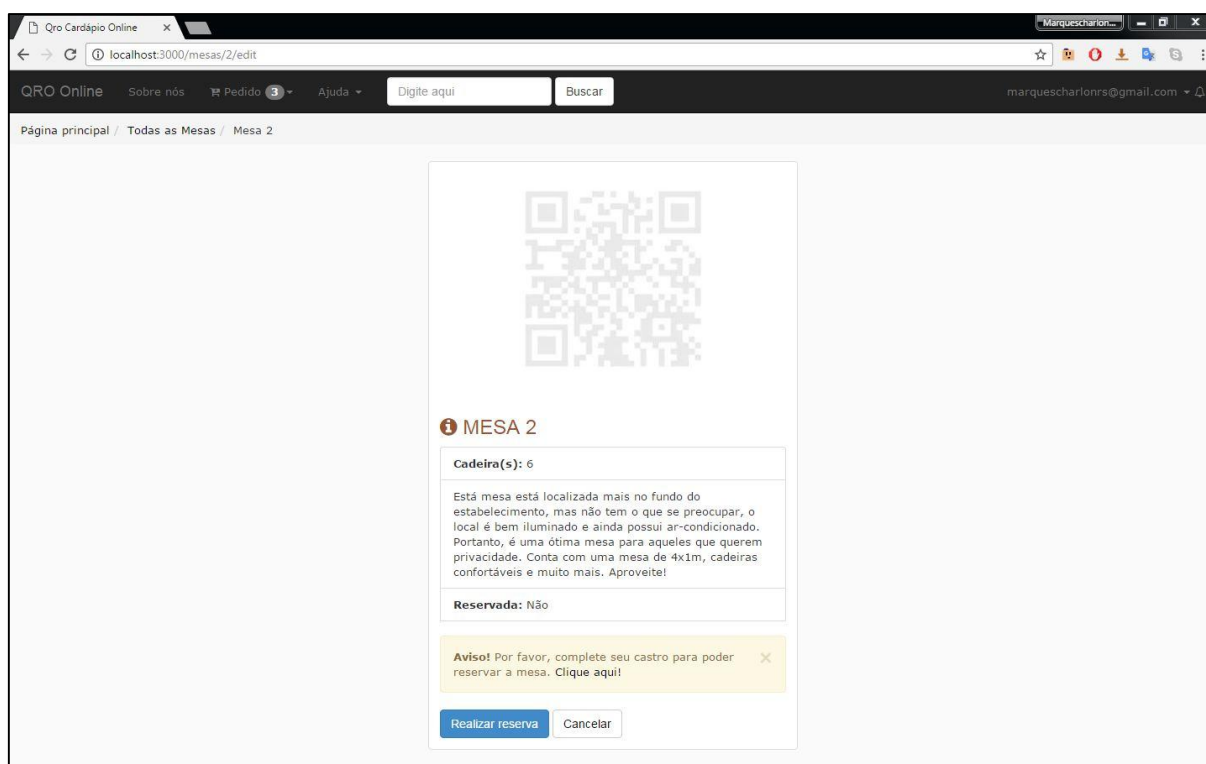


Fonte: Próprio autor (2016)

Agora é possível ter uma melhor visão das informações. Observa também que, o QR Code não está totalmente visível, isso porque só quando a mesa está reservada é que o usuário que realizou a reserva é que irá obter o código, enquanto isso, o que está na tela não é legível e também é uma imagem fictícia (FIG.15).

Outro detalhe que pode ser observado é a tarja amarela, ela pede para o usuário completar o seu cadastro, informando seus dados pessoais para transmitir uma melhor segurança para o estabelecimento. Enquanto o usuário não completar o cadastro ele não conseguirá realizar a reserva.

FIG. 15 Obter mais detalhes da mesa.



Fonte: Próprio autor (2016)

Após completar os dados a tela já muda, aparece novas informações.

Além de contar com a agilidade que será o atendimento, o estabelecimento fica resguardado sobre algumas questões jurídicas, pois, submeterá ao cliente o termo de compromisso¹. Ao realizar a reserva da mesa o cliente estará concordando com os termos estabelecidos pelo estabelecimento (FIG.16). Após marcar a opção “Marque esta opção se deseja reservá-la.” é só clicar no botão “Realizar reserva”. Reserva realizada (FIG.17).

¹ Termo no qual define os direitos e deveres, tanto para o cliente, quanto para o estabelecimento. Tal qual, é proposto pelo estabelecimento.

FIG. 16 Realizar a reserva da mesa.

QRO Online Sobre nós Pedido 3 Ajuda Digite aqui Buscar marquescharlonns@gmail.com

Página principal / Todas as Mesas / Mesa 2

MESA 2

Cadeira(s): 6

Está mesa está localizada mais no fundo do estabelecimento, mas não tem o que se preocupar, o local é bem iluminado e ainda possui ar-condicionado. Portanto, é uma ótima mesa para aqueles que querem privacidade. Conta com uma mesa de 4x1m, cadeiras confortáveis e muito mais. Aproveite!

Reservada: Não

☐ Marque esta opção se deseja reservá-la.

Reservando a mesa você estará concordando com os [termos](#).

Realizar reserva Cancelar

Fonte: Próprio autor (2016)

FIG. 17 Reserva realizada.

QRO Online Sobre nós Pedido 3 Ajuda Digite aqui Buscar marquescharlonns@gmail.com

Página principal / Todas as Mesas / Mesa 2

Mesa atualizada com sucesso.

MESA 2

Cadeira(s): 6

Está mesa está localizada mais no fundo do estabelecimento, mas não tem o que se preocupar, o local é bem iluminado e ainda possui ar-condicionado. Portanto, é uma ótima mesa para aqueles que querem privacidade. Conta com uma mesa de 4x1m, cadeiras confortáveis e muito mais. Aproveite!

Reservada: Sim - 2016-11-27 19:33:15 -0200

Máximo de acompanhantes: 6

Cliente: Marquescharlon Santos

CPF: 99988877700

CEP: 35160000

Rua: Rua X

Fonte: Próprio autor (2016)

Este é o comprovante da reserva a ser apresentado no estabelecimento (FIG.18), observa que antes era preciso chegar no estabelecimento, entrar na fila e aguardar alguma

mesa ser liberada. No entanto, o atendente precisa apenas confirmar o cadastro, esse sistema favorece ainda mais aqueles que por ventura estejam levando pessoas da terceira idade. Isso irá não só favorecer aos clientes, mas ajudar a aumentar a clientela, uma vez que, dar atenção às necessidades dos clientes é também agregar valor ao produto. Essa reserva online proporciona não só agilidade no processo mas a garantia de que, ao chegar no estabelecimento o cliente não ficará frustrado tendo que esperar por uma mesa, imagina se for uma família que quer comemorar o dia das mães fora de casa, levar a vó que possivelmente esteja visitando, então, força-la a enfrentar isso pode ser desgastante. Portanto, a melhor solução é propor ao cliente que realize a reserva pelo sistema, estando ainda no conforto de sua casa.

No entanto, se esquecerem de levar o comprovante de sua reserva ou o código impresso não estiver legível, então, será possível realizar uma pesquisa de sua reserva informando apenas o CPF. Cada estabelecimento determinará sua regra, deixando visível a informação de que, caso o cliente não compareça no estabelecimento no horário previsto, após X minutos e inúmeras tentativas de contato, então, a mesa será liberada, tendo então, sua reserva cancelada.

FIG. 18 Comprovante da reserva



MESA 2

Cadeira(s): 6

Esta mesa está localizada mais no fundo do estabelecimento, mas não tem o que se preocupar, o local é bem iluminado e ainda possui ar-condicionado. Portanto, é uma ótima mesa para aqueles que querem privacidade. Conta com uma mesa de 4x1m, cadeiras confortáveis e muito mais. Aproveite!

Reservada: Sim - 2016-11-27 19:33:15 -0200

Máximo de acompanhantes: 6

Cliente: Marquescharlon Santos

CPF: 99988877700

CEP: 35160000

Rua: Rua X

Numero: 0

Cidade: Ipatinga

Bairro: Horto

Complemento: Próximo ao Hotel SanDiego

Tel Fixo: (31)9999-9999

Celular: (31)98888-8888

Aviso! Por favor, verique se seus dados estão atualizados. Caso não estejam Clique aqui! (/users/edit) para atualizá-los.

[Voltar \(/mesas\)](#) [Imprimir](#)

Fonte: Próprio autor (2016).

Caso há algum pane de energia no momento que estiver realizando a reserva, não concluindo-a, será necessário acessar o sistema novamente e realizar a reserva. Caso seja necessário, o gerente do estabelecimento poderá adotar em imprimir um relatório dos clientes e suas respectivas mesas já registradas, caso ocorra vários problemas como internet lenta e equipamento com mal funcionamento, lembrando que o mesmo deverá atualizar no sistema o mais breve possível. Para uma melhor segurança, o backup é realizado pela própria empresa que fornece o sistema, e pode ser solicitado a qualquer momento. Por padrão, todo o final do dia é realizado o backup de cada estabelecimento.

A fim de desenvolver um sistema para complementar este projeto, foi dado a atenção há algumas das principais funcionalidades, como:

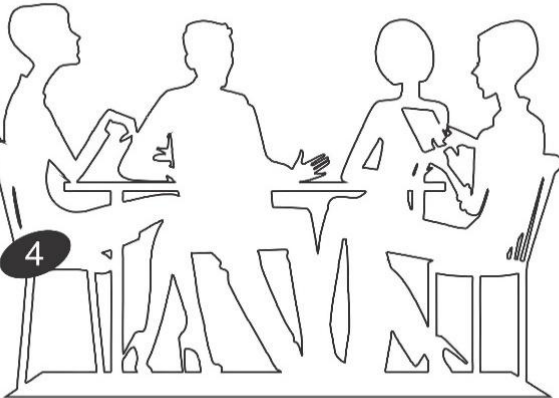
- Login, cadastro de usuários e recuperação da senha;
- Cadastro e visualização dos produtos;
- Reserva da mesa;

4.2. PRINCIPAIS CARTÕES DE HISTÓRIA DO SISTEMA

Os cartões de história são desenvolvidos junto ao cliente². Segue abaixo alguns dos principais cartões desenvolvidos junto a este projeto. Na página principal ficou definido que, seria uma página simples contendo apenas um vídeo no fundo e um formulário para acesso ao sistema (FIG.19).

FIG. 19 Cartão de História – Página inicial.

#0001	PÁGINA INICIAL
Como um cliente, quero me deparar com uma página simples, com um vídeo no fundo, com os campos login e senha para o usuário poder realizar o cadastro.	

QRO Online	Sobre nós	Ajuda▼	Digite aqui	Buscar	1
Página principal					
				<div>3</div> <div> Login </div> <div> E-mail <input type="text"/> </div> <div> Senha <input type="password"/> </div> <div> <input type="checkbox"/> Lembrar-me </div> <div> Entrar Limpar </div> <div> Cadastre-se Esqueceu sua senha? </div>	
4				1 - Barra de menus padrão em todas as páginas; 2 - Campo de busca; 3 - Formulário para ser realizar o acesso/cadastro; 4 - No background deverá passar um vídeo simples de pessoas em um estabelecimento se comunicando e felizes;	

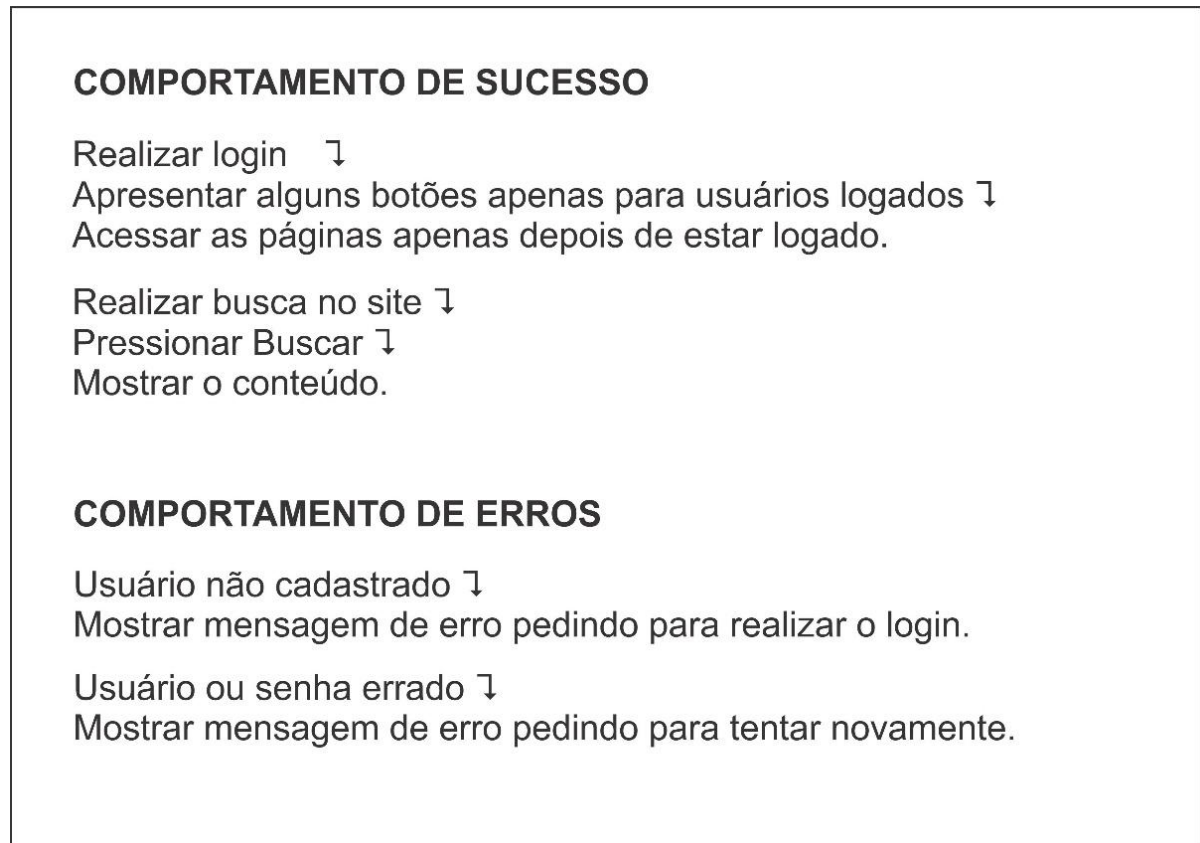
Fonte: Próprio autor (2016).

No verso de cada cartão de história possui suas validações, isso é, os testes dos quais o sistema precisa passar sem nenhum problema. Porém, caso ocorra algum erro o sistema

² Neste caso o cliente é o estabelecimento.

realizará esse tratamento, informando alguma mensagem, no qual, está aconselhando o usuário a realizar algum procedimento. Por exemplo, se o cliente tentar forçar o acesso as outras páginas, logo será impresso uma mensagem na tela pedindo que ele realize o *login*, caso não o tenha, então, realizar o cadastro (FG.20).

FIG. 20 Verso do Cartão – Página inicial.



Fonte: Próprio autor (2016).

Cada cartão possui sua referência, o ID, que será único para especificações detalhadas. No cartão cadastrar usuário (FIG.21), a mensagem é clara, o cliente quer que o usuário possa realizar o cadastro no sistema, de maneira simples. Porém, neste layout, ele especifica se no cadastro inicial o cliente irá precisar de muitos campos ou permitir que seja um simples cadastro e depois ele possa completa-lo. Isso para que o cliente possa acessar as páginas mais rápido e não ficar frustrado em ter que gastar tempo realizando um enorme cadastro, uma vez que esse tempo poderá ser gasto navegando no sistema.

FIG. 21 Cartão de História – Cadastrar usuário.

#0002	CADASTRAR USUÁRIO
Como um cliente, quero poder realizar o cadastro sem depender de terceiros ou aprovação.	

QRO Online	Sobre nós	Ajuda ▾	<input style="width: 90%;" type="text" value="Digite aqui"/>	<input style="width: 80%;" type="button" value="Buscar"/>	
Página principal					

Inscrever-se

E-mail

Senha

Confirmar senha

Entrar
 Esqueceu sua senha?

© 2016 QRO CARDÁPIO ONLINE

1 - Barra de menus padrão em todas as páginas;

2 - Campo de busca;

3 - Formulário para ser realizar o cadastro;

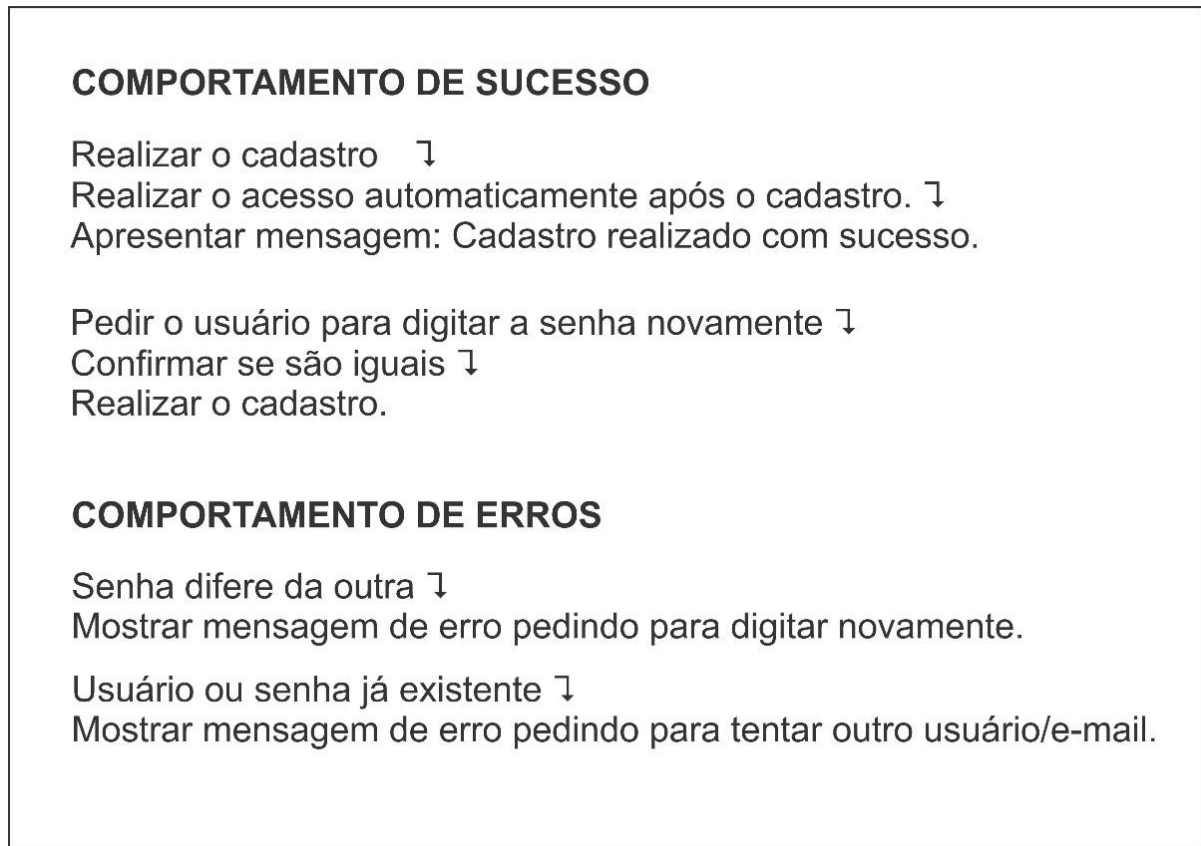
4 - No background deverá ser limpo, sem imagens ou vídeo.

Fonte: Próprio autor (2016).

O verso do cartão cadastrar usuário (FIG.22) possui os testes realizados e o comportamento de erros. Permite que o sistema interaja com o cliente, ele não acusa de nenhum mal procedimento, mas aconselha com mensagens simples, a fim, de que o usuário leia e entenda o seu propósito.

A idéia do cartão de história é justamente economizar tempo com algumas informações técnicas. Pode-se dizer que o seu maior objetivo é o próprio layout BLOG DO ABU (2016), no entanto, nele precisa constar as informações necessárias para a realização deste projeto ou página em questão.

FIG. 22 Verso do Cartão – Cadastrar usuário.



Fonte: Próprio autor (2016).

O próximo cartão de história especifica questões como, algumas limitações, isso é, visualizações de menus apenas para os usuários cadastrados. Por exemplo, no menu PEDIDOS, quem não possui o cadastro é certo que também não tenha feito nenhum pedido (FIG.23). Caso tenha o interesse em apenas realizar o cadastro, o sistema consegue construir o histórico de cada usuário, principalmente por questões de auditoria, no registro das modificações, registro dos IPs das máquinas e outras informações que o framework Ruby On Rails proporciona com agilidade ANTHONY LEWIS (2015).

FIG. 23 Cartão de História – Após realizar o *login*.

#0003	APÓS REALIZAR O LOGIN
Como cliente, eu quero ter visão dos produtos, pedidos e configurações após realizar o login.	

QRO Online	Sobre nós	Pedido ①	Ajuda	<input type="text" value="Digite aqui"/>	Buscar	1	usuario@email.com
Página principal / Todos os produtos							
4	<p>1 - Barra de menus padrão em todas as páginas. Agora deverá apresentar os menus Pedido e usuário.</p> <p>2 - Campo de busca agora deverá realizar a busca correspondente a página acessada atualmente.</p> <p>3 - Configurações do usuário.</p> <p>4 - Breadcrumb para auxiliar o usuário.</p> <p>5 - Junto ao menu PEDIDO, mostrar a quantidade de pedidos abertos.</p>						
© 2016 QRO CARDÁPIO ONLINE							

Fonte: Próprio autor (2016).

Ao definir o que será realizado após o *login*, foram efetuadas algumas validações que constam no verso do cartão #0003 (FIG.24). O mesmo mostra sugestões para que o usuário possa levar em consideração, aponta caminhos para uma possível solução, e além disso, permite também algumas regras para o sistema. Por exemplo, se o usuário tentar acessar com uma senha inválida, o sistema deverá informá-lo, a realizar o *login* com uma nova senha.

É utilizado recursos do Ruby e seu framework Ruby On Rails para construir um sistema de gestão multi-estabelecimento, no qual, não será necessário nenhuma instalação na máquina, e sim, o cadastro online em um único sistema, tendo seu banco de dados separado.

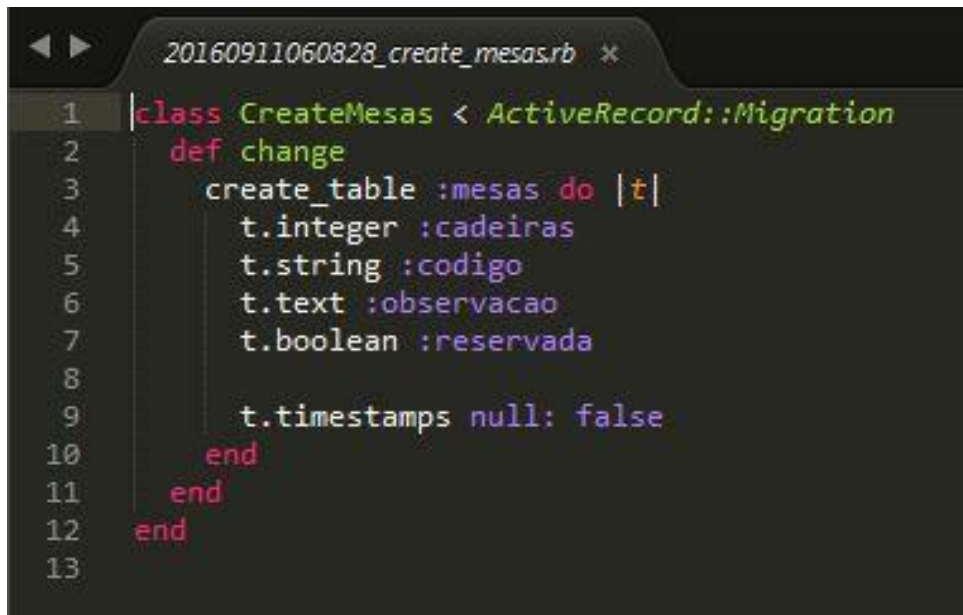
FIG. 24 Verso do Cartão – Após realizar o login.

<p>COMPORTAMENTO DE SUCESSO</p> <p>Poder editar as configurações do usuário ↴ Registrar as atualizações do usuário junto o ip da máquina ↴ Apresentar mensagem: Atualização realizada com sucesso.</p> <p>Permitir a busca no controller de acordo com a página atual ↴ Apenas digitar e clicar no botão “Buscar” ↴ Limpar o campo e clicar em “Buscar” novamente para limpar o filtro.</p> <p>COMPORTAMENTO DE ERROS</p> <p>Senha atual errada ↴ Mostrar mensagem de erro pedindo para digitar novamente.</p>
--

Fonte: Próprio autor (2016).

O cartão de história referente as mesas define o seu layout básico, apresenta partes do menu de pedidos, além, de apresentar os campos de sua tabela, campos que são definidos através das *migrations*, utilizadas para controle do banco de dados, permitir que a equipe de desenvolvimento possa modifica-lo facilmente. Portanto, para criar a tabela mesas, por exemplo, foi necessário apenas adicionar em sua *migrations* comandos assim como mostra a figura (FIG.25).

FIG. 25 Criar a tabela no banco de dados.



```
20160911060828_create_mesas.rb x
1 class CreateMesas < ActiveRecord::Migration
2   def change
3     create_table :mesas do |t|
4       t.integer :cadeiras
5       t.string :codigo
6       t.text :observacao
7       t.boolean :reservada
8
9       t.timestamps null: false
10    end
11  end
12 end
13
```

Fonte: Próprio autor (2016).

O XP não exige que seja contratado uma equipe só para o banco de dados DESENVOLVIMENTO MÉTODO ÁGIL (2016a), uma vez que o mesmo precisa corresponder ao layout, sem contar com a versatilidade proporcionada pela linguagem Ruby e seu framework Ruby On Rails, a possibilidade de criar procedimentos, dos quais, são realizados no banco de dados, isso tudo fica junto ao projeto, conhecido também por *migrate* ANTHONY LEWIS (2015).

Como mostra o cartão de história Mesas(FIG.26) o cliente poderá escolher sua mesa. No campo observação poderá constar informações como ambiente, próximo a porta, mais nos fundos e etc. Isso dependerá exclusivamente do estabelecimento, a forma como irá proporcionar essas informações.

FIG. 26 Cartão de História – Mesas.

#0004

MESAS

Como cliente, eu quero ter visão das mesas junto ao menu pedido.

QRO Online

Sobre nós

Pedido1

Ajuda

Buscar

usuario@email.com

Página principal / Todas as mesas

Realizar pedido

Consultar pedido

Reservar mesa2

Cupom de desconto

Mesas

1 - Menu Pedido deverá ter submenus
Realizar pedido > Consultar pedido > Reservar mesa > Cupom de desconto.
2 - Página após clicar no submenu "Reservar mesa".

ID	CADEIRAS	CÓDIGO	OBSERVAÇÃO	RESERVADA	CRIADO EM:	BOTÕES
2	6		XXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXX	Não	00/00/00 00:00:00	<input type="text"/>
2	6		XXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXX	Não	00/00/00 00:00:00	<input type="text"/>
2	6		XXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXX	Não	00/00/00 00:00:00	<input type="text"/>

© 2016 QRO CARDÁPIO ONLINE

Fonte: Próprio autor (2016).

No verso do cartão de história referente as mesas(FIG.27) define apenas algumas obrigatoriedades, onde o campo deverá constar algum valor/informação. Como mensagens amigáveis, instruir o usuário o que deve fazer ou até mesmo informa-lo se uma determinada ação foi realizada com sucesso.

FIG. 27 Verso do Cartão – Mesas.

<p>COMPORTAMENTO DE SUCESSO</p> <p>Poder realizar a leitura do código QRCode, porém, não deixá-lo muito grande. ↴ Informar o status da mesa ↴ Informar o número de cadeiras.</p> <p>COMPORTAMENTO DE ERROS</p> <p>Dificuldade para realizar a leitura do código no monitor. ↴ Mostrar mensagem de erro pedindo para tentar novamente.</p>
--

Fonte: Próprio autor (2016).

A vantagem de trabalhar com o cartão de história é o quanto ele diminui o custo do tempo em pequenas ou grandes modificações, por exemplo, a princípio o cliente³ queria a página das mesas conforme visto na figura (FIG.26), depois solicitou algumas modificações, como tirar o campo ID, trocar os nomes das colunas e permitir que o cliente possa obter o código QR Code apenas clicando no botão “Detalhes”, então, para realizar essas modificações foi necessário apenas substituir o cartão de história por outro, com uma nova referência #0006 (FIG.28).

³ Neste caso o cliente é o estabelecimento, no qual, solicitou o desenvolvimento do sistema.

FIG. 28 Cartão de História – Página Mesas alterada.

#0006

PÁGINA MESAS ALTERADA

Como cliente, eu quero ter visão das mesas junto ao menu pedido.

QRO Online

Sobre nós

Pedido 1

Ajuda

Digite aqui

Buscar

usuario@email.com

Página principal / Todas as mesas

Realizar pedido

Consultar pedido

Reservar mesa 2

Cupom de desconto

1 - Menu Pedido deverá ter submenus

Realizar pedido > Consultar pedido > Reservar mesa > Cupom de desconto.

2 - Página após clicar no submenu "Reservar mesa".

Mesas

Cadeiras	Observação	Liberada	Detalhes
6	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	00/00/00 00:00:00	<input type="text"/>
6	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	00/00/00 00:00:00	<input type="text"/>
6	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	00/00/00 00:00:00	<input type="text"/>

© 2016 QRO CARDÁPIO ONLINE

Fonte: Próprio autor (2016).

O desenvolvimento da janela produtos foi baseado no layout do cartão de história – produtos (FIG.29), obedecendo as regras para paginação, disposição das informações e etc. Tal janela consiste apenas na apresentação dos produtos e seu valor, porém, a proposta deste trabalho é que futuramente seja desenvolvido em uma única janela a possibilidade de registrar a mesa, escolher o produto e realizar o pagamento, fazer isso utilizando um componente do *bootstrap* chamado modal. Com isso a economia de tempo e a diminuição do número de cliques até o pedido ser concluído diminuirá consideravelmente.

FIG. 29 Cartão de História – Produtos.

#0005	PRODUTOS
Como cliente, eu não quero ver todos os produtos de uma vez, se possível uma paginação.	

QRO Online	Sobre nós	Pedido	Ajuda	<input style="width: 80%;" type="text" value="Digite aqui"/> <input style="width: 10%;" type="button" value="Buscar"/>	usuario@email.com
Página principal / Todos os produtos					

<div style="border: 1px solid #ccc; height: 100px; margin-bottom: 10px;"></div> <div style="text-align: center;"> <p>NOME DO PRODUTO</p> <p>Descrição do produto, teste, teste, teste, teste, teste, teste, teste.</p> <p>R\$00,00</p> <p>Atualizado aprox. 2 horas</p> </div>	<div style="border: 1px solid #ccc; height: 100px; margin-bottom: 10px;"></div> <div style="text-align: center;"> <p>NOME DO PRODUTO</p> <p>Descrição do produto, teste, teste, teste, teste, teste, teste, teste.</p> <p>R\$00,00</p> <p>Atualizado aprox. 2 horas</p> </div>	<div style="border: 1px solid #ccc; height: 100px; margin-bottom: 10px;"></div> <div style="text-align: center;"> <p>NOME DO PRODUTO</p> <p>Descrição do produto, teste, teste, teste, teste, teste, teste, teste.</p> <p>R\$00,00</p> <p>Atualizado aprox. 2 horas</p> </div>
<div style="display: flex; align-items: center;"> 1 <div style="display: flex; align-items: center;"> << <div style="display: flex; gap: 5px;"> 1 2 3 4 </div> >> </div> </div> <div style="margin-top: 10px; font-size: 0.8em;"> <p>1 - Modelo de como deverá ficar o produto.</p> <p>2 - O sistema deve adicionar a paginação automaticamente a cada 3 produtos.</p> </div>		

Fonte: Próprio autor (2016).

4.3. INSTALAÇÕES E CONFIGURAÇÕES

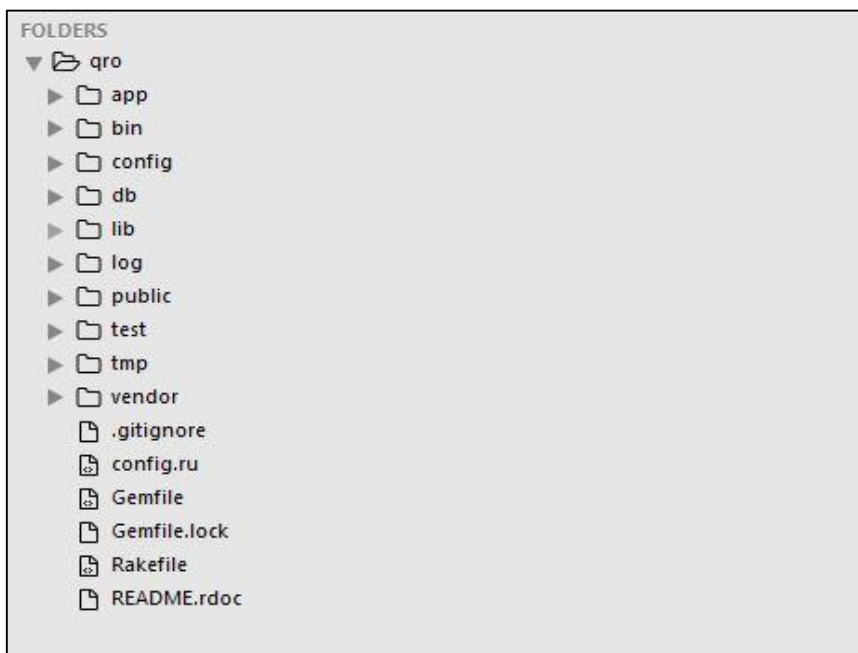
Este projeto foi desenvolvido em uma máquina cujo processador é i5, 8 *Gigabyte (GB)* de memória *Random Access Memory (RAM)* e *Hard Disk (HD) Solid-State Drive (SSD)* 240GB de memória *Read-Only Memory (ROM)*, Windows 7 Ultimate 64bit.

A versão escolhida do Ruby foi a 2.2.4-x64, não é a versão mais recente, porém, o objetivo é permanecer com a versão instável para depois realizar o upgrade. Para realizar a instalação basta fazer o download do arquivo *setup* conforme orientações em GitHub RUBY ON RAILS (2016).

O objetivo desse sistema é também servir como referência para o aprendizado das ferramentas utilizadas em seu desenvolvimento, então, encontra neste trabalho orientações e sugestões, a fim, de apontar o melhor caminho para quem estiver começando a trabalhar com a linguagem Ruby e seu *framework* Ruby On Rails, conhecer a solução antes mesmo que o problema aconteça pode ser o melhor meio de se economizar tempo POPPENDIECK & POPPENDIECK (2003).

Segue abaixo a estrutura deste projeto (FIG.30), criado pelo Ruby On Rails após executar o comando *rails new qro -d mysql*. Onde *rails* é o gerador da estrutura, o comando *new* define um novo projeto, *qro* é o nome dado ao projeto e *-d mysql* define o banco de dados utilizado, caso não seja especificado o banco, por padrão é instalado o *SQLite*. Após ter criado o projeto acesse o seu diretório através do *prompt* de comando e digite o seguinte comando para inicializar o servidor *rails server* ou simplesmente *rails s*.

FIG. 30 Estrutura deste projeto.



Fonte: Próprio autor (2016).

Abaixo se encontra uma pequena descrição da estrutura:

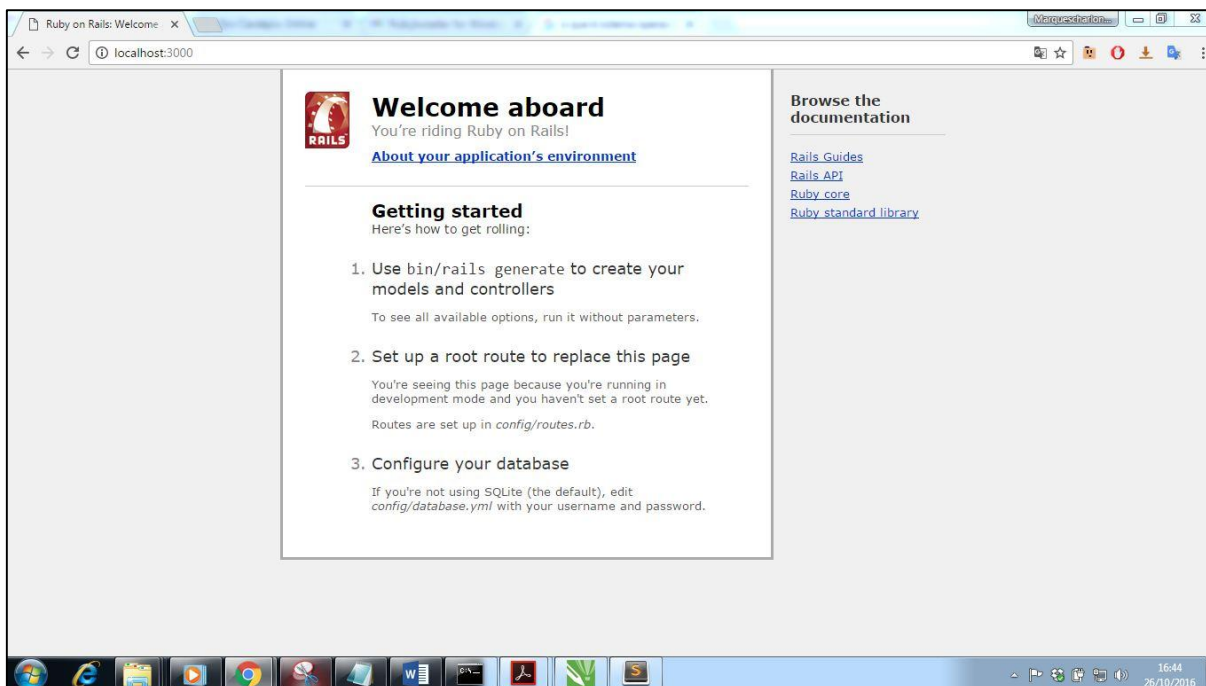
- **app/** - Contém todos os arquivos necessários para manipular o banco de dados, como:
 - **assets:**
 - imagens/
 - javascripts/
 - stylesheets/
 - **controllers:**
 - application_controller.rb;
 - estabelecimentos_controller.rb;
 - mesas_controller.rb;

- produtos_controller.rb;
- registros_controller.rb;
- tipos_controller.rb.
- **helpers:**
 - application_helper.rb;
 - estabelecimentos_helper.rb;
 - mesas_helper.rb;
 - produtos_helper.rb;
 - registros_helper.rb;
 - tipos_helper.rb.
- **mailers.**
- **models:**
 - estabelecimento.rb;
 - mesa.rb;
 - produto.rb;
 - registro.rb;
 - tipo.rb;
 - user.rb.
- **views:**
 - devise/
 - estabelecimentos/
 - layouts/
 - mesas/
 - pages/
 - produtos/
 - registros/
 - tipos/
- **bin/** - Possui os scripts responsável por inicializar a aplicação;
- **config/** - Contém os arquivos de routes, database e demais arquivos;
- **config.ru** - Configuração para servidores em Rack;
- **db/** - Contém as relações das migrações do banco;
 - 20160911032018_create_tipos.rb;
 - 20160911060828_create_mesas.rb;

- 20160911061514_create_produtos.rb;
- 20160912020816_remove_imagem_produto.rb;
- 20160912021402_add_attachment_imagem_to_produtos.rb;
- 20160928192439_create_registros.rb;
- 20161001024825_alter_type_valor_produtos.rb;
- 20161001025522_add_default_valor_produtos.rb;
- 20161001030137_add_default_cadeiras_mesas.rb;
- 20161006020827_alter_default_valor_produtos.rb;
- 20161006021529_alter_precision_valor_produtos.rb;
- 20161024222649_devise_create_users.rb;
- 20161025181811_add_column_mesa_id_to_users.rb;
- 20161025192427_remove_mesa_id_from_users.rb;
- 20161026001059_add_column_mesa_codigo_to_users.rb;
- 20161028201638_add_column_mesa_id_to_user.rb;
- 20161028213719_add_colum_cpf_nome_cep_cidade_logradouro_bairro_numero_complemento_telfixo_telcelular.rb;
- 20161029070626_create_estabelecimentos.rb.
- **Gemfile, Gemfile.lock** – Esses arquivos armazenam a lista das pendências;
- **lib/** - Arquivos estendido para a aplicação;
- **log/** - Arquivos de log;
- **public/** - Contém arquivos públicos, estáticos e compilados;
- **Rakefile** – Responsável por localizar as tarefas que podem ser executadas pelo prompt;
- **README.rdoc** – Manual básico de instrução para a aplicação;
- **README.md** – Também manual básico para a aplicação;
- **test/** - Arquivos de testes;
- **temp/** - Arquivos temporários;
- **vendor/** - Lugar para armazenar todos os códigos de terceiros;

Após iniciar o servidor pelo comando *rails server* o WEBrick será inicializado, este é um servidor web padrão para o Ruby. Após inicializa-lo basta apenas abrir a janela do navegador e digitar o endereço <http://localhost:3000>(FIG.31).

FIG. 31 Tela inicial de uma nova aplicação.



Fonte: Próprio autor (2016).

Diferente das documentações tradicionais, o Ruby On Rails permite que seja gerado uma documentação referente a cada RubyGems especificada no arquivo *Gemfile*, conhecidas também por *gems*, são as bibliotecas/dependências que trazem consigo informações para a realização de uma determinada função no projeto, sua nomenclatura apresenta o nome e sua versão, *gem* '*<nome>*', '*<versão>*'.

Para gerar essa documentação é necessário apenas acessar o diretório deste projeto utilizando o *prompt* e digitar o comando *rakedoc:rails*. Será construído a documentação a partir do próprio sistema, detalhando todas as *gems*, especificando como elas funcionam, onde obter mais referências, como utilizá-las e como foram aplicadas no projeto. Este conteúdo pode ser encontrado dentro do próprio projeto, o caminho é: *nome_do_projeto\doc\api*.

4.4. DIFICULDADES NO DECORRER DO PROJETO

A principal dificuldade está relaciona a linguagem Ruby juntamente com seu framework Ruby On Rails. Na internet se encontra poucos materiais sobre Ruby On Rails em português. Porém, a vantagem é que em inglês há tudo que é preciso. Portanto, basta pesquisar o erro no Google.

Tendo em vista as dificuldades encontradas para o desenvolvimento do sistema que tem como objetivo apoiar este projeto, fazendo uso do comando *scaffolds*⁴ foi gerado um conjunto de arquivos que compõem a arquitetura MVC DUARTE (2011) do projeto, para que todos os erros, todas as dificuldades, todas as soluções fossem registradas juntamente ao ambiente no decorrer do desenvolvimento, além de servir como uma material de apoio para quem queira aprender Ruby On Rails (FIG.32), portanto, é uma excelente base de dados.

FIG. 32 Erros e soluções.

Id	Referencia	Descricao	Observacao	Concluido	Created at
16	Gemfile / gem 'devise'	1) gem 'devise' 2) bundle ...	Documentação: https://git...	✗	24 de Outubro de 2016, 2...
15	Error: Server certificate B...	Descrição completa do er...	Documentação Devkit: htt...	✓	11 de Outubro de 2016, 1...
14	Formatar :decimal (R\$ 00...	Para apresentar os valore...	É só acessar a document...	✓	06 de Outubro de 2016, 0...
13	Erro: Rake aborted!	C:\Users\CHARLON\Drop...	Foi necessário apenas ex...	✓	05 de Outubro de 2016, 2...
12	Gemfile / gem "font-awes...	1) Adicionar gem "font-aw...	Para utilizar no index.html...	✓	05 de Outubro de 2016, 2...
11	Definir Porta Padrão para...	1) Em config/boot.rb, adic...	Onde está (:Port => 3000...	✓	04 de Outubro de 2016, 0...
10	Adicionando Login na pá...	1) Adicionar os códigos a...	Ainda falta adicionar a GE...	✓	03 de Outubro de 2016, 2...
9	Video na página principal	1) Adicionar a estrutura n...	Alguns videos não funcio...	✓	03 de Outubro de 2016, 2...

Fonte: Próprio autor (2016).

Como se trata de muitas informações, o melhor a se fazer é organizar o trabalho a ser realizado. Na tabela Registro, mostrada acima, possui todos os erros e o passo-a-passo para as correções, desde a instalação até configurações avançadas, orientações sobre o arquivo *Gmfile* e entre outros.

Após a instalação e configuração do *App*⁵ recém criado em Ruby algumas funções não estavam funcionando como a *Destroy*⁶, foi preciso instalar o *Node-v4.4.5-x86.exe*, uma plataforma para desenvolvimento de aplicações baseadas em rede utilizando *JavaScript*, pois faltavam algumas bibliotecas para o funcionamento do *App*. Tanto a função quanto o upload de imagens foram corrigidos e problemas que apareceriam futuramente quando tentarem

⁴ Comando que cria uma estrutura pronta para poder criar, ler, alterar e excluir em uma tabela.

⁵ Se refere ao sistema inteiro, também podendo ser chamado de *App*.

⁶ É uma função responsável por realizar a exclusão de um registro dentro de uma determinada tabela.

realizar o upload de qualquer imagem, tal erro é possível ser visto logo na instalação da *gem paperclip*⁷.

Por medida de segurança este projeto foi desenvolvido e armazenado no *dropbox*, caso o Windows desse problema, ao menos não teria perdido o projeto, sem dizer que, fica muito mais fácil para compartilhar o projeto com outros computadores.

⁷ Componente responsável por realizar todo o processamento das imagens, como corte, upload e etc.

5. CRONOGRAMA

Para o desenvolvimento deste trabalho foi preciso elaborar um cronograma especificando as tarefas em cada etapa (FIG.33).

FIG.33 Cronograma.

Etapas do TCC 1		CRONOGRAMA TCC 1 / 2016																											
		Meses - Semanas																											
		Março				Abril				Maio				Junho				Julho				Agosto				Setembro			
		1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª
Escolha do tema																													
Justificativa																													
Hipóteses ou questões norteadoras																													
Objetivos																													
Referencial teórico																													
Metodologia e pesquisa																													
Encontros com o orientador																													
Revisão Bibliográfica																													
Criação dos instrumentos para a coleta de dados																													
Coleta e análise dos dados.																													
Discussão dos resultados com orientador																													
Desenvolvimento do Sistema																													
Redação final do TCC																													
Encaminhamento à correção linguística																													
Entrega do TCC																													
Apresentação à Banca Examinadora																													
Correções indicadas pela Banca Examinadora																													
Entrega final do TCC (encadernado e CD)																													

Fonte: Próprio autor (2016)

6. CONCLUSÃO

Conclui-se que, este projeto atende uma das necessidades encontradas em um estabelecimento, que é a reserva de uma mesa. Em alguns casos submetendo o cliente a uma fila absurda e obrigando-o a esperar sem nenhum conforto até que alguma mesa esteja desocupada. Para o desenvolvimento deste sistema foi necessário enfrentar diversos desafios, entre eles está a própria linguagem de programação utilizada (Ruby), ainda sendo nova no mercado e pouca discutida em ambientes acadêmicos. Outros métodos que estão adquirindo espaço no mercado aos poucos são os métodos ágeis, utilizados neste trabalho para proporcionar o desenvolvimento em um tempo hábil. O sistema apresentado neste trabalho oferece a possibilidade de realizar a reserva da mesa online. Há a necessidade apenas de realizar o check-in na entrada do estabelecimento, pois, um pré-cadastro já foi realizado pelo próprio cliente. Assim como é feita a reserva de uma passagem aérea, tal qual, é realizada online e tendo em vista seu check-in antes do embarque, também será a reserva e o check-in da mesa. Isso proporciona não só conforto, mas a segurança de que ao chegar no estabelecimento será atendido sem a necessidade de enfrentar filas, pois, o seu cadastro já foi feito e a mesa já está reservada.

REFERÊNCIAS

ANTHONY LEWIS. **Rails Crash Course: a no-nonsense guide to Rails development.** San Francisco, 2015.

BABA, Vinicius Akira. **Diagnóstico e análise de oportunidade de melhoria em um restaurante universitário por meio da Filosofia Seis Sigma.** Ribeirão Preto, 2008. Dissertação (Mestrado em Administração de Organizações) –Universidade de São Paulo – USP.

BECK, Kent. **Extreme Programming Explained: Embrace change, second edition.** Addison-Wesley, 2004.

BECK, Kent. **Extreme Programming Explained: Embrace Change.** Addison-Wesley, 1999.

BLOG DO ABU. **Cartão de História.** Disponível em: <<http://blogdoabu.blogspot.com.br/2010/02/cartao-de-historia.html>>. Acesso em: 26 de Maio de 2016.

BOOTSTRAP. **Gettingstarted.** Disponível em: <<http://getbootstrap.com/getting-started/>>. Acesso em: 26 de Outubro de 2016.

CAELUM. **Apostila Desenvolvimento Ágil para Web com Ruby On Rails.** Disponível em: <<https://www.caelum.com.br/apostila-ruby-on-rails/a-linguagem-ruby/>>. Acesso em: 13 de Março de 2016.

DESENVOLVIMENTO MÉTODO ÁGIL. **Extreme Programming.** Disponível em: <<http://www.desenvolvimentoagil.com.br/xp/>>. Acesso em: 13 de Março de 2016a.

DESENVOLVIMENTO MÉTODO ÁGIL. **Princípios do XP.** Disponível em: <<http://www.desenvolvimentoagil.com.br/xp/principios/>>. Acesso em: 13 de Março de 2016b.

DESENVOLVIMENTO MÉTODO ÁGIL. **Simplicidade.** Disponível em: <<http://www.desenvolvimentoagil.com.br/xp/valores/simplicidade>>. Acesso em: 13 de Março de 2016c.

DESENVOLVIMENTO MÉTODO ÁGIL. **Desenvolvimento Tradicional**. Disponível em: <http://www.desenvolvimentoagil.com.br/xp/desenvolvimento_tradicional>. Acesso em: 13 de Março de 2016d.

DEVMEDIA. **Conceitos básicos sobre Metodologias Ágeis para Desenvolvimento de Software (Metodologias Clássicas x Extreme Programming)**. Disponível em: <<http://www.devmedia.com.br/conceitos-basicos-sobre-metodologias-ageis-para-desenvolvimento-de-software-metodologias-classicas-x-extreme-programming/10596>>. Acesso em: 06 de Maio de 2016.

DEVMEDIA. **Planejando seu projeto com eXtreme Programming**. Disponível em: <<http://www.devmedia.com.br/planejando-seu-projeto-com-extreme-programming-parte-i/4273#>>. Acesso em: 26 de Maio de 2016.

DUARTE, Aldrey Rocha. **Metodologia Rails: Análise da arquitetura Model View Controller Aplicada**. Belo Horizonte, 2011. Monografia (Análise de Sistemas) - Universidade Federal de Minas Gerais – UFMG.

FERREIRA, Rodrigo Talles. **Migração de um sistema desktop para um sistema WEB dinâmico: Pizzaria delivery**. Belo Horizonte, 2010. Monografia (Análise de Sistemas) - Universidade Federal de Minas Gerais – UFMG.

G1 - AUTO ESPORTE. **Só Toyota e Jeep crescem em vendas entre 10 maiores marcas no 1º tri**. Disponível em: <<http://g1.globo.com/carros/noticia/2016/04/so-toyota-e-jeep-crescem-em-vendas-entre-10-maiores-marcas-no-1-tri.html>>. Acesso em: 09 de Junho de 2016.

JOBSTRAIBIZER, Flávia. **SQL para Profissionais**. São Paulo: Digerati Books, 2009.

MANIFESTO AGIL. **Manifesto para o desenvolvimento ágil de software**. Disponível em: <<http://www.manifestoagil.com.br/>>. Acesso em: 16 de Dezembro de 2016.

POPPENDIECK, Mary; POPPENDIECK, Tom. **Lean software development: an agile toolkit**. 1. ed. Upper Saddle River, NJ: Addison-Wesley, 2003.

RUBY ON RAILS. **Contribute on GitHub**. Disponível em: <<https://github.com/rails/rails>>. Acesso em: 09 de Junho de 2016.

RUBY. **Sobre o Ruby**. Disponível em: <<https://www.ruby-lang.org/pt/about/>>. Acesso em: 09 de Junho de 2016.

SASADA, Koichi. **YARV: Yet Another RubyVM**. Tokyo, 2016.

SATO, Danilo Toskiaki. **Uso eficaz de métricas em métodos ágeis de desenvolvimento de software**. São Paulo, 2007. Dissertação (Obtenção do Grau de Mestre em Ciências) – Ciência da Computação, Instituto de Matemática e Estatística da Universidade de São Paulo.

SCHOEPPING, Guilherme. **Um estudo exploratório a partir de um framework para seleção de práticas ágeis**. Florianópolis, 2012. Dissertação (Programa de Pós-Graduação em Ciência da Computação) – Centro Tecnológico, Universidade Federal de Santa Catarina.

SUBLIME TEXT. **Download**. Disponível em: <<https://www.sublimetext.com/3>>. Acesso em: 13 de Março de 2016.

TELES, Vinícius Manhães. **Um estudo de caso da adoção das práticas e valores do extreme programming**. Rio de Janeiro, 2005. Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro – UFRJ, IM DCC, 2005.

ANEXO 1

FORMULÁRIO PARA LEVANTAMENTO DE DADOS

Desenvolvimento de sistema para Check-in com base na metodologia Ágil

1. Com base no relacionamento atual que existe entre o estabelecimento e os clientes, qual a sua consideração quanto intervenção da tecnologia, a fim, de criar uma maior aproximação?

- a) ☐ Desejável
- b) ☐ Necessário
- c) ☐ Não relevante

2. Ao julgar essa questão com base em suas experiências, qual o grau de dificuldade encontrada ao realizar cada procedimento? Em uma escala de 0 a 5 marque:

0 – Nenhuma dificuldade

3 – Com dificuldade, mas ainda tolerável

5 –Alto grau de dificuldade

a) Reservar a mesa	0	1	2	3	4	5
b) Realizar o pedido	0	1	2	3	4	5
c) Realizar o pagamento	0	1	2	3	4	5

3. Em breves palavras, como você espera que o cliente se comporte em seu estabelecimento, ou se você é um cliente, como você gostaria de se comportar?

4. Como você classificaria suas 3 últimas idas há um determinado estabelecimento?

- a. ☐ Satisfatório
- b. ☐ Insatisfatório. Nota de 0 à 10 _____.