

Desenvolvimento de um Sistema de Gestão e Monitoramento para Consultórios Veterinários e *Pet Shops*

Caroline Rodrigues Dorneles, Ana Paula Canal, Alexandre de Oliveira Zamberlan

Sistemas de Informação – Centro Universitário Franciscano
CEP 97.010-032 – Santa Maria – RS – Brasil

caroll.dorneles@gmail.com, apc@unifra.br, alexz@unifra.br

Abstract. *This paper presents the design and development of a management system for veterinary office and shops for pets. The system manages customers and their pets, keeping registration information services performed. The system also maintains information about employees, their tasks and their performances. Thus, the system helps in monitoring fulfilled procedures, helping to control spending. The project was built by FDD methodology, using Astah and Bizagi modeling tools, and technologies as PHP, jQuery, PHPMailer and PHPQRCode.*

Resumo. *Este trabalho apresenta o projeto e o desenvolvimento de um sistema de gestão para consultórios veterinários e lojas para animais de estimação (Pet Shops). O sistema faz a gestão de clientes e de seus animais, mantendo informações cadastrais (em geral) de serviços realizados. O sistema também mantém informações sobre os funcionários, suas tarefas e seus desempenhos. Dessa forma, o sistema contribui no monitoramento dos procedimentos realizados, auxiliando no controle de gastos. O projeto foi construído por meio da metodologia Feature Driven Development (FDD), com o uso das ferramentas de modelagem Astah e Bizagi, e das tecnologias PHP, jQuery, PHPMailer e PHPQRCode.*

1. Introdução

Os animais, principalmente cães e gatos, estão inseridos no âmbito familiar, onde são considerados membros da família. Segundo dados do Instituto Brasileiro de Geografia e Estatística (IBGE), coletados no ano de 2013, existem mais cães nos lares brasileiros do que crianças. De cada cem famílias, quarenta e quatro criam cachorros e somente trinta e seis têm crianças de até doze anos de idade [Arias 2015].

Com isso, houve a expansão do mercado de *Pet Shops*, com inúmeras variedades de produtos, desde os mais diversos tipos de rações, até acessórios e brinquedos para os animais, proporcionando conforto e qualidade de vida aos animais de estimação. O mercado de *Pet Shops* tem ganhado espaço na economia do país, bem como, os Consultórios Veterinários devido aos avanços da medicina veterinária [Elizeire 2013].

Portanto, esse aumento do mercado de estabelecimentos especializados, sugere-se a utilização de Sistema de Gestão e Monitoramento para Consultórios Veterinários e *Pet Shops*. Um sistema informatizado auxilia, de forma prática e ágil, também na prestação do atendimento, dispensando processos manuais, muitas vezes redundantes e falhos, como por exemplo o uso de fichas em papel [Elizeire 2013]. Com o *software* de gestão, a busca de informações de cada animal é mais rápida, precisa e auxilia em certas decisões, uma vez que é possível descobrir determinados padrões de atendimento

[Santos 2013]. Por exemplo, qual a raça de cachorro permanece mais tempo no estabelecimento, qual o tipo de pelagem que fica menos tempo na tosa, entre outros indicadores.

Nesta pesquisa, assume-se que Consultório Veterinário é o local para atendimento clínico, como curativos e vacinação de animais, não sendo possível ter internação e realização de cirurgias. Já a *Pet Shop* realiza a venda de produtos e serviços, como banhos e tosas.

Para identificação de padrões, que ajudam as empresas na tomada de decisões, surge o conceito de gestão do conhecimento (representar, manipular, compartilhar e aplicar). Para Rezende e Abreu (2011), o conhecimento é a capacidade de interpretar. Quando a informação é trabalhada por pessoas e por recursos computacionais pode gerar cenários, simulações, oportunidades e competitividade. O conhecimento deriva da informação, e cada vez mais os administradores ou gestores usam os sistemas de maneira mais voltada para a gestão do conhecimento [Zamberlan 2010].

Este trabalho está inserido no contexto de Sistemas de Informação do tipo Sistemas de Informação Gerenciais (SIG). Segundo Laudon e Laudon (2014), esse tipo de sistema é utilizado para monitorar e controlar a empresa, podendo prever seu desempenho futuro, além de gerar relatórios que auxiliam o gerente da empresa na tomada de decisões.

Pelo motivo da expansão de lojas para animais de estimação e de consultórios veterinários, o foco deste trabalho foi projetar e desenvolver um sistema de gestão e monitoramento, em que oferece cadastro, consulta e alteração de dados de clientes (proprietários), animais e de funcionários. O sistema permite emissão de relatórios gerais, identificação de padrões e relações de serviços. E como objetivos específicos: i) pesquisar e escrever sobre Sistemas de Informação neste contexto; ii) pesquisar e escrever sobre tecnologias usadas; iii) pesquisar e modelar um modelo de negócio; iv) aplicar uma modelagem de desenvolvimento de *software* para esse tipo de sistema; v) identificar e aplicar *frameworks* ou padrões de projeto para esse tipo de sistema; vi) identificar trabalhos relacionados; vii) projetar e modelar o sistema; viii) implementar o sistema; ix) avaliar.

Finalmente, para auxiliar o entendimento, o artigo está dividido em 5 seções. A Seção 2 apresenta conceitos de Sistemas de Informação, tecnologias utilizadas e trabalhos relacionados. A Seção 3 trata do projeto e da aplicação da metodologia de desenvolvimento escolhida. Na Seção 4, as considerações finais do trabalho são apresentadas.

2. Referencial Teórico

Nesta seção, são abordados conceitos e os principais tipos de Sistemas de Informação, as tecnologias usadas para projetar e implementar o *software*, bem como trabalhos correlatados.

2.1. Sistemas de Informação

Segundo Laudon e Laudon (2014), Sistema de Informação (SI) é um conjunto de componentes inter-relacionados que coletam (entrada), processam, armazenam

(processo) e distribuem (saída) informações destinadas a apoiar a tomada de decisões, a coordenação e controle em uma organização, fornecendo um mecanismo de *feedback*.

Os Sistemas de Informação podem ser classificados de várias maneiras, porém como existem diferentes interesses, especializações e níveis dentro de uma organização, também existem diversos tipos de sistemas. Os principais tipos de sistemas são: Sistemas de Informação Gerenciais (SIG), Sistemas de Apoio à Decisão (SAD), Sistemas de Processamento de Transações (SPT), Sistemas de Informação para Executivos (SIE), Sistemas de Gestão da Cadeia de Suprimentos (SGCS), Sistemas de Gestão do Relacionamento com o Cliente (SGRC) e Sistemas de Gestão do Conhecimento (SGC) [Laudon e Laudon 2014].

Desta forma, acredita-se que o SIG é o que mais se encaixa no projeto deste artigo, pois segundo Laudon e Laudon (2014), este tipo de sistema tem foco na eficiência operacional (monitoramento e controle), fornecendo aos administradores e tomadores de decisões as informações para ajudá-los a atingir metas da empresa. Tais sistemas transformam dados em informações úteis e geram relatórios sobre o desempenho atual da organização.

2.2. Tecnologias Utilizadas no Projeto

O *Hypertext Preprocessor* (PHP) é uma linguagem de *script* estruturada voltada para a criação de páginas dinâmicas, tendo recursos, como suporte a diversos servidores Web e seções. Outros recursos foram adicionados com o intuito de transformá-la em uma linguagem orientada a objetos [Dall'Oglio 2013].

O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a Linguagem de Consulta Estruturada - SQL (*Structured Query Language*) com interface simples. Ele executa em vários sistemas operacionais (Windows, Linux, Mac, Solaris, entre outros). É compatível com diversas linguagens de programação (Java, PHP, Python, Delphi, Ruby, C/C++) e é pouco exigente quanto aos recursos de *hardware* [Pacievitch 2011].

Quick Response Code (QR Code) é um código de barras 2D, conforme apresentado na Figura 1, composto por caixas e pontos, formando um padrão similar à de uma matriz. Esse tipo de codificação permite que possam ser armazenadas centenas de vezes mais dados do que código de barras tradicionais [Turban e Volonino 2013]. Quando um leitor de códigos faz a leitura, o aplicativo pode carregar um *site*, fazer o *download* de um arquivo, enviar uma mensagem, por exemplo, todas essas informações gravadas nessa matriz de dados [Turban e Volonino 2013].



Figura 1: Código gerado em QR Code com a *String* "Peludinho".

2.3. Desenvolvimento de Sistemas

O termo sistema é usado universalmente, existindo diferentes tipos, como sistemas de computadores, sistemas de pagamentos, sistemas de gestão, sistema educacional, sistema de monitoramento, entre outros. Um sistema é um conjunto de *softwares* que interagem para atingir um objetivo em comum [Sommerville 2011].

Para autores como Pressman (2010) e Sommerville (2011), os processos de desenvolvimento de sistemas de informação estão em constante melhoria. O processo de desenvolvimento de *software* pode incluir um método de trabalho estruturado, em etapas gerenciáveis individual e coletivamente, que têm como objetivo produzir, de forma coordenada, *software* para qualquer aplicação em geral. Existe uma diversidade de processos de *software*, não existindo um processo ideal a ser seguido. As organizações desenvolvem abordagens diferentes para o processo de seus *softwares* [Sommerville 2011].

Pressman (2010) e Sommerville (2011) enfatizam que as atividades podem variar de acordo com cada modelo de desenvolvimento de *software*, mas independentemente do modelo de processo utilizado, algumas atividades podem ser utilizadas de forma genérica como: comunicação: envolve comunicação e colaboração com o cliente e outros interessados, em que abrange o levantamento de requisitos e outras atividades relacionadas; planejamento: estabelece um plano para o trabalho de desenvolvimento. Nessa fase descrevem as tarefas, os riscos do projeto, os artefatos a serem produzidos e o cronograma de trabalho; modelagem: responsável pelos modelos que permitem entender melhor os requisitos do *software*; construção: geração do código fonte a partir das funcionalidades especificadas e os testes necessários para descobrir erros do *software*; implantação: entrega do *software* ao cliente, que avalia o produto e fornece *feedback* com base na avaliação.

O ciclo de desenvolvimento de sistemas é utilizado pelas organizações para grandes projetos de Tecnologia da Informação (TI). Esse ciclo de vida é um *framework* estruturado, que consiste em processos sequenciais pelos quais os sistemas de informação são desenvolvidos. Este ciclo tem oito fases: i) Investigação de sistemas; ii) Análise de sistemas; iii) Projeto de sistemas; iv) Programação; v) Teste; vi) Implementação; vii) Operação; viii) Manutenção. Projetos de desenvolvimento menores podem exigir apenas um subconjunto de tarefas [Turban e Volonino 2013].

O uso de uma metodologia para o desenvolvimento de sistemas ajuda i) na qualidade do *software*; ii) os clientes ficam mais satisfeitos; iii) possibilita maior controle quanto à entrega do sistema no prazo definido; iv) exige uma boa documentação; v) possibilita o acompanhamento e gerenciamento dos processos passo-a-passo, entre outros benefícios.

2.4. Metodologia de Desenvolvimento de Softwares

Para Pressman (2010), as metodologias tradicionais surgiram em um cenário de construção de *software* muito diferente do atual. As tarefas para desenvolvimento de *softwares* eram realizadas em terminais ditos “burros” e *mainframes*, onde o custo para correção de erros era consideravelmente elevado. Então, para minimizar os problemas durante o desenvolvimento, na época, exigia-se planejamento e boa documentação, antes do início do desenvolvimento. Exemplos de metodologias tradicionais são Cascata, Prototipação, Espiral, RUP (*Rational Unified Process*), Linear, entre outras.

Por outro lado, as metodologias mais modernas, conhecidas como ágeis, são usadas em projetos em que se tem um conjunto mínimo de funcionalidades e são úteis para entregar valor ao cliente. Caracterizam-se por serem adaptativas e orientadas a pessoas [Pressman 2010]. A engenharia de *software* ágil combina filosofia com um conjunto de princípios de desenvolvimento. A filosofia defende a satisfação do cliente e a entrega prévia. O princípio de desenvolvimento prioriza a entrega mais que a análise e projeto. Também priorizam a comunicação ativa e contínua entre clientes e desenvolvedores. Exemplo de metodologias ágeis são XP (*eXtreme Programming*), Scrum, FDD (*Feature-Driven Development*) [Pressman 2010].

FDD é uma metodologia ágil para o processo de engenharia de *software* que tem foco na entrega frequente do sistema funcionando para o cliente, em ciclos rápidos de no máximo duas semanas e na utilização de boas práticas durante o ciclo do seu desenvolvimento [Pressman 2010]. O ciclo de vida da FDD é composto por duas fases (Concepção e Planejamento, Construção) e cinco processos. O processo começa com a modelagem abrangente (análise orientada por objetos), seguindo pela construção da lista de funcionalidades e seu planejamento, logo após é iniciada a fase de construção que está dividida em dois processos, planejar por funcionalidade e desenvolver por funcionalidade (programação e testes orientado a objetos) [Pressman 2010]. Sendo assim, a FDD foi selecionada para realização deste trabalho.

2.5. Trabalhos Relacionados

No trabalho, proposto por Santos (2013), é apresentado um sistema de controle de identificação e de atividades realizadas em um *Pet Shop* por rádio frequência (*RFID*). Esse sistema é integrado a um micro controlador *Arduino R3 Uno*, permitindo que os dados do animal e as atividades possam ser armazenados em um banco de dados Oracle. O projeto usou a linguagem PL/SQL (Oracle) e Java. O sistema controla as atividades realizadas com o animal, seus donos e as atividades realizadas por meio de etiquetas eletrônicas.

Já o objetivo do trabalho proposto por Junior, et al. (2012) foi apresentar um estudo para a implementação de rotinas, na linguagem de Programação Java, para a criação de códigos de barra *QR Code*, utilizando conexão com o banco de dados para Rastreabilidade de Grãos. Com a utilização de um *framework*, o processo foi facilitado, permitindo ao consumidor visualizar as características do produto, por meio de um celular. Dessa forma, o produtor tem em mãos uma ferramenta para registrar todas as informações relevantes do seu processo produtivo.

Por último, o trabalho proposto por Souza e Aires (2011) descreveu o desenvolvimento de programas implementados nas linguagens de programação C e C#. O primeiro programa desenvolvido foi o Gerador de *QR Code*, em que o usuário coloca as informações e o programa gera um *QR Code* no formato PNG. O segundo programa desenvolvido foi o Decodificador de *QR Code*, que é usado em conjunto com uma câmera. O *software* processa a imagem de um *QR Code* através de uma câmera e analisa as informações. O terceiro programa desenvolvido foi um sistema que controla e gerencia o banco de dados.

Por meio dos trabalhos relacionados, foi possível nortear-se para a definição de algumas tecnologias. Em comum com o primeiro trabalho correlato, tem a ideia de todo o controle de todas as atividades realizadas com o animal em um *Pet Shop*. Porém para

isso, Santos (2013) usa *RFID* para o controle das atividades, o que torna o projeto mais caro, enquanto que neste trabalho foi usado *QR Code*, de custo mais baixo. O segundo e terceiro trabalhos serviram para melhor entendimento do funcionamento do *QR Code* e foram de ajuda para decidir as demais tecnologias usadas neste projeto.

3. Projeto e Aplicação da Metodologia

Nesta seção, é descrito o projeto desenvolvido conforme a metodologia FDD.

3.1. Projeto

O sistema de gestão para Consultórios Veterinários e *Pet Shops* oferece cadastro, consulta e alteração de dados de seus clientes (proprietários), dos seus animais e de funcionários. O *software* permite a emissão de relatórios gerais, identificação de padrões e relações de serviços como a raça e tamanho do animal. Por exemplo, qual a raça de cachorro permanece mais tempo no estabelecimento, qual o tipo de pelagem que fica menos tempo na tosa, etc. Assim, a busca pela informação de cada animal ou serviço prestado torna-se mais rápida e precisa (e com menos gastos caso fichas em papel sejam utilizadas, uma prática comum em Consultórios Veterinários e/ou *Pet Shop*).

Uma vez realizado o cadastro do animal, é possível gerar e imprimir o *QR Code*, que através da sua leitura pode ler diversos dados, como os dados do cliente ou em que parte do *Pet Shop* o animal encontra-se. Assim, ao final do cadastro do serviço a ser realizado, será impresso o *QR Code* e, simultaneamente, será gerado uma sequência de números randômicos que é enviada para o *e-mail* do cliente, para ser usada como senha de acesso e para que possa acompanhar o andamento do serviço de seu animal na Web por meio do trabalho proposto por Machado (2016).

Depois de impresso o *QR Code*, pode-se anexá-lo na coleira, na gaiola ou onde o animal ficar durante o serviço realizado no estabelecimento. Com isso, é possível monitorar o animal por meio da leitura desse código, sistema implementado. A cada procedimento a ser realizado com o animal, o código é lido pelo leitor de *QR Code* e atualizado o *status* no banco de dados, sendo este processo realizado pelo sistema do Machado (2016). O leitor pode ser sem fio ou também pode ser usado um dispositivo móvel capaz de ler *QR Codes* e de conectar-se à Internet para atualização do banco de dados.

Assim, a cada procedimento realizado, é possível saber a localização do animal (seu *status* durante a prestação de serviço), por exemplo, se está na gaiola, se está no banho, ou se já pronto. Isso possibilita, além do acompanhamento do proprietário, que o estabelecimento encontre alguns padrões, como qual a raça fica mais tempo no banho, qual o empregado é mais ágil na tosa, etc.

Finalmente, para o código *QR Code* ser gerado, foi usada a biblioteca *PHPQRCode*, que oferece uma interface de programação (API) para criar imagens de código de barras.

3.2. Análise de Requisitos e Aplicação das Fases da FDD

A metodologia escolhida para este projeto, como já mencionado, é a *Feature Driven Development* ou Desenvolvimento Dirigido por Funcionalidades. Essa metodologia

pode ser aplicada em projetos de software de tamanho médio e grande, sendo usada como uma ferramenta para projetos orientados a objetos [Pressman 2010].

Os Requisitos de *Software* são funções, objetivos, propriedades, restrições que o sistema deve possuir para satisfazer padrões ou especificações de acordo com o usuário [Sommerville 2011]. Requisitos Funcionais descrevem as funcionalidades do sistema. Eles são especificações dos serviços que o sistema deve prover e como os sistemas devem reagir a certas entradas [Rezende e Abreu 2011]. Os requisitos funcionais levantados foram: RF01- Efetuar *Login*; RF02- Cadastrar Funcionário; RF03- Alterar Funcionário; RF04- Excluir Funcionário; RF05- Cadastrar Cliente; RF06- Alterar Cliente; RF07- Excluir Cliente; RF08- Gerenciar Serviços; RF09- Cadastrar Produto; RF10- Alterar Produto; RF11- Excluir Produto; RF12- Realizar Leitura *QR Code*; RF13- Gerenciar Relatórios.

Os requisitos não funcionais são aqueles não relacionados diretamente com às funções específicas do sistema. Neste trabalho, assumiu-se: RNF01- O usuário deve digitar *login* e senha válidos para acessar o sistema; RNF02- A base de dados deve ser protegida para acesso de apenas usuários autorizados; RNF03- O sistema será desenvolvido na linguagem PHP, que é uma ferramenta que pode ser usada na maioria dos sistemas operacionais; RNF04- O sistema utilizará SGBD MySQL, que é um banco de dados gratuito.

3.3. Aplicação da Metodologia FDD

Nesta fase, foram feitos os diagramas de classes UML, que descrevem os objetos relevantes dentro do domínio do problema. Também foram desenvolvidos diagramas de sequência, que descrevem como os objetos interagem para cumprir responsabilidades. Por fim, foi realizada a identificação das funcionalidades do sistema, que servirão para guiar o desenvolvimento no FDD.

3.3.1. Desenvolvimento de um Modelo Geral

É uma atividade inicial que abrange todo o projeto. Deve-se adquirir conhecimento do domínio de negócio em questão e desenvolver um modelo geral. Geralmente, é realizada por membros do domínio de negócio e por desenvolvedores, sob a orientação de um modelador de objetivos experiente [Retamal 2008].

Nesta etapa é gerado um modelo abrangente, o diagrama de domínio (*Object Model*), mas para melhor visualização do projeto, substituiu-se esse diagrama. A Figura 2 apresenta o diagrama geral do projeto, que foi modelado usando-se a notação BPMN (Modelagem de Processo de Negócio). Inicia-se o sistema por *login*, na sequência, há a possibilidade de efetuar cadastros. Em seguida, caso cliente esteja na base, é possível o cadastro do atendimento (definição de um serviço). Esse processo gera o *QR Code* como relatório de entrada do animal do sistema.

3.3.2. Construção de uma Lista de Funcionalidades

Atividade que abrange todo projeto. Para identificar todas as funcionalidades que satisfaçam os requisitos, a lista de funcionalidades é construída com base no modelo global [Retamal 2008]. É formada para decompor funcionalmente o domínio em áreas e atividades de negócio. Embora o FDD não exija o Diagrama de Casos de Uso, com a

Figura 3, é possível visualizar as funcionalidades do sistema, que é de gerenciar clientes e funcionários, realizar atendimento e gerenciar relatórios.

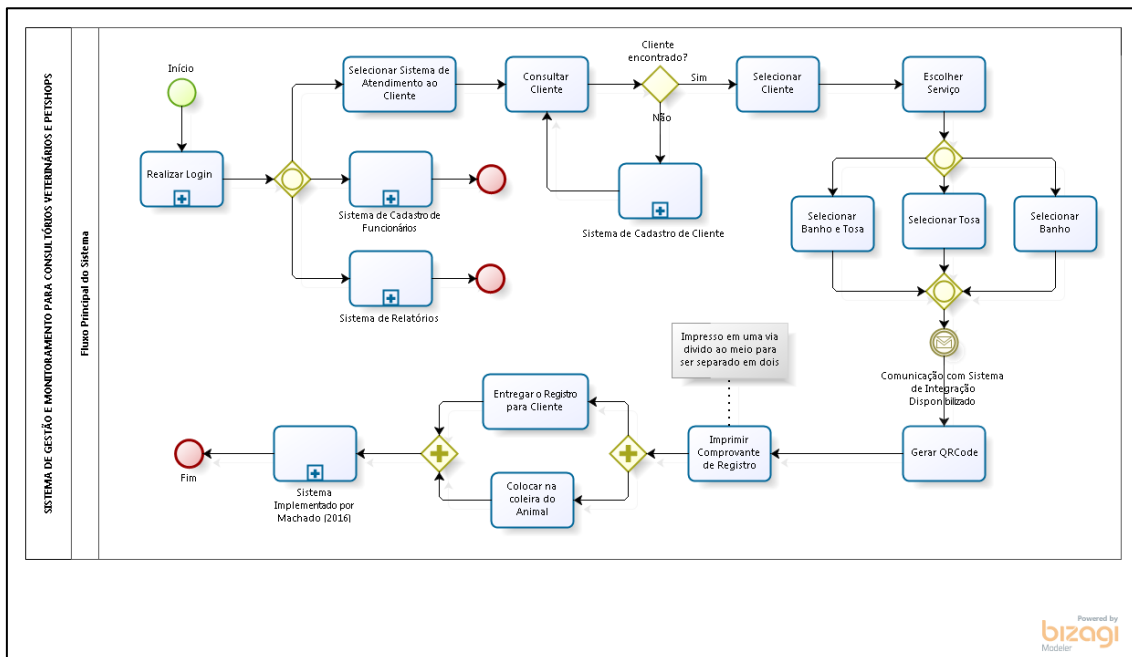


Figura 2: Modelagem do Fluxograma do Sistema.

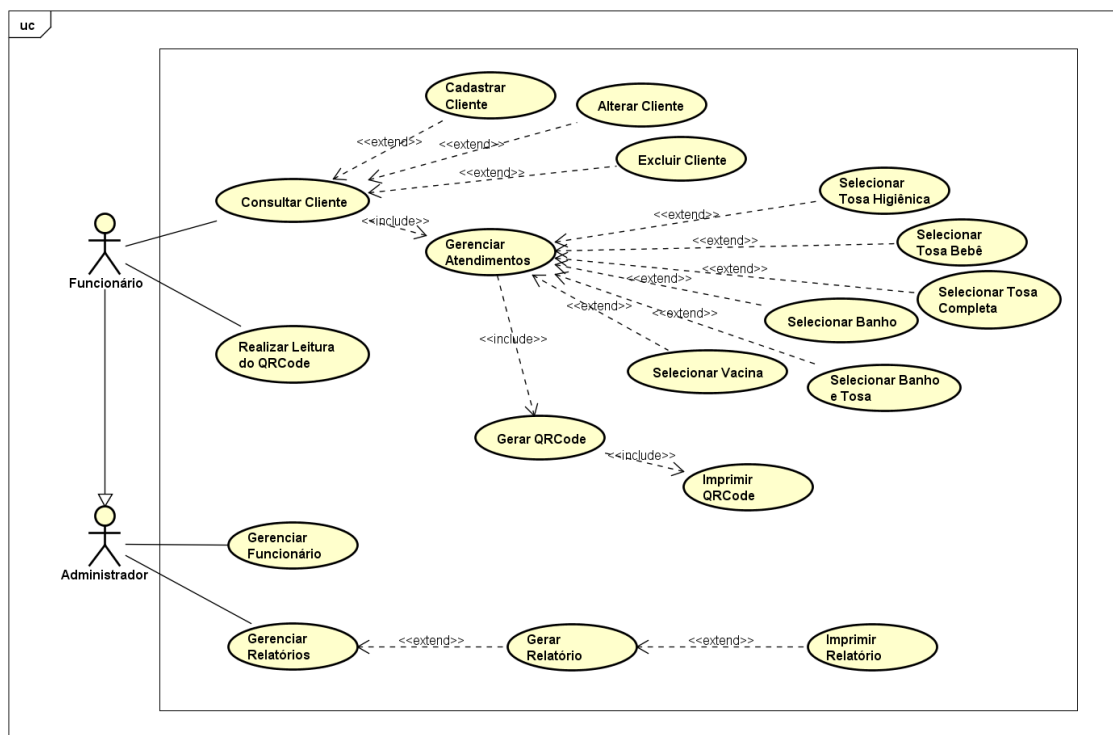


Figura 3: Diagrama de Caso de Uso.

A partir dos diagramas produzidos, foi possível a construção do diagrama de classes para o projeto, conforme a Figura 4, onde mostra um conjunto de classes e seus relacionamentos. A classe Atendimento é considerada a principal, porque a partir dela gera-se o fluxo de atendimento de serviços.

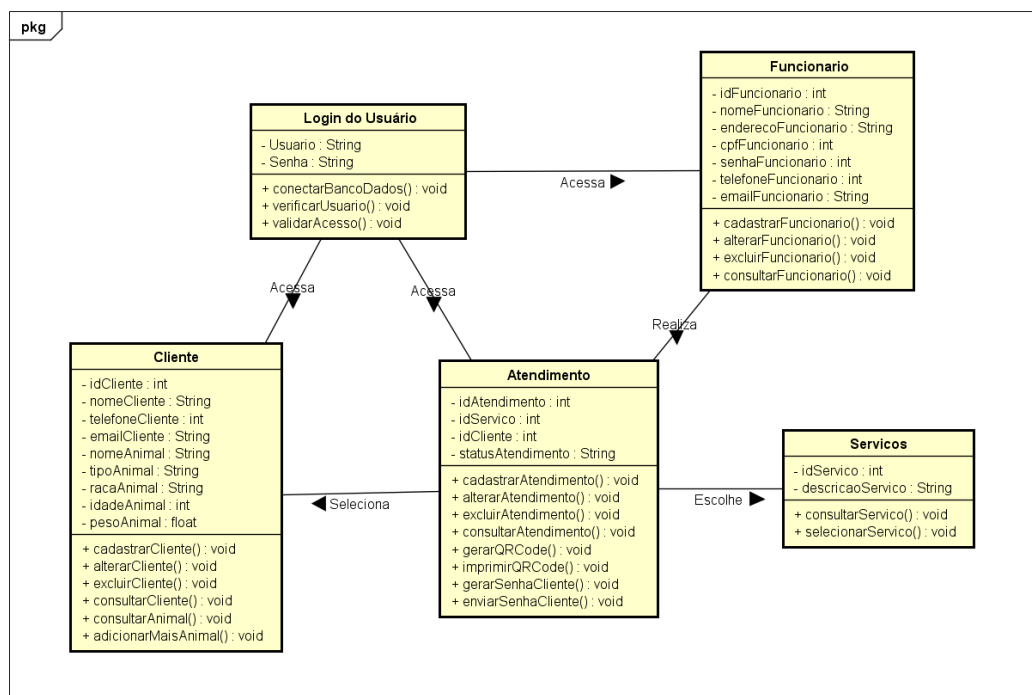


Figura 4: Diagrama de Classes.

3.3.3. Planejar por Funcionalidade

Atividade que abrange todo o projeto. Para produzir o plano de desenvolvimento, os gerentes e programadores chefes planejam a ordem na qual as funcionalidades serão implementadas [Retamal 2008]: i) implementar as classes geradas na fase de listagem das funcionalidades; ii) escrever os métodos em suas respectivas classes conforme o caso de uso; iii) testar e avaliar; iv) validar.

3.3.4. Detalhar por Funcionalidade

Dentro de uma iteração de construção, a equipe detalha os requisitos e outros artefatos para a codificação de cada funcionalidade, identificando as classes (incluindo os testes). Registra-se, porém, que o sistema deste trabalho foi desenvolvido por apenas uma pessoa. Nesta fase, então, são gerados os diagramas de sequência das funcionalidades, conforme mostra a Figura 5 [Retamal 2008].

Além disso, na Figura 6 apresenta-se a arquitetura de todo o sistema, incluindo o trabalho desenvolvido por Machado (2016), que se encontra do lado B da figura. No lado A, está o desenvolvido deste trabalho, onde encontra-se a página *e-clientes*, que trata do gerenciamento de clientes. Em seguida, foi implementado a página *e-atendimentos*, que realiza o controle de atendimentos realizados, bem como a geração de QR Codes.

O *e-funcionarios* faz todo o gerenciamento de funcionários, após foi implementado o *e-relatorios* onde podem ser visualizados vários relatórios. Por fim, foram implementados a página principal *e-index* e a página *index* que contém o formulário de *login* para acesso ao sistema.

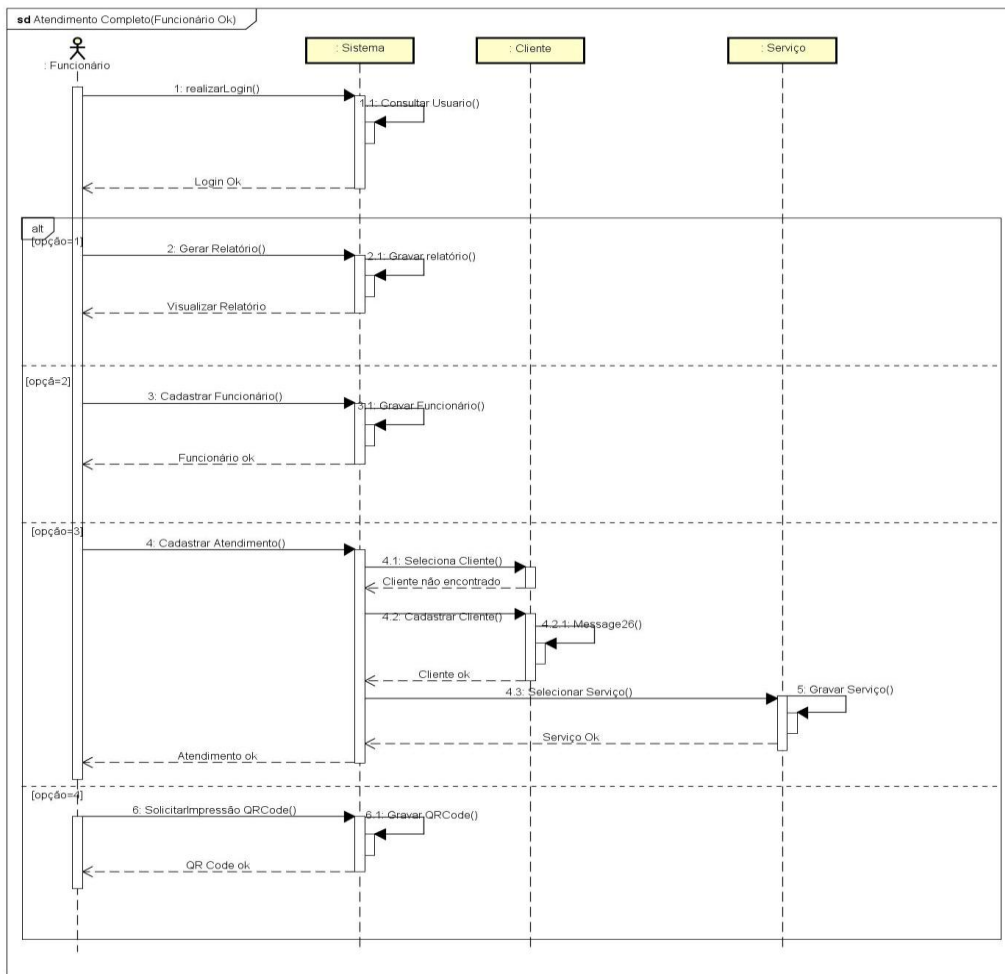


Figura 5: Diagrama de Sequência.

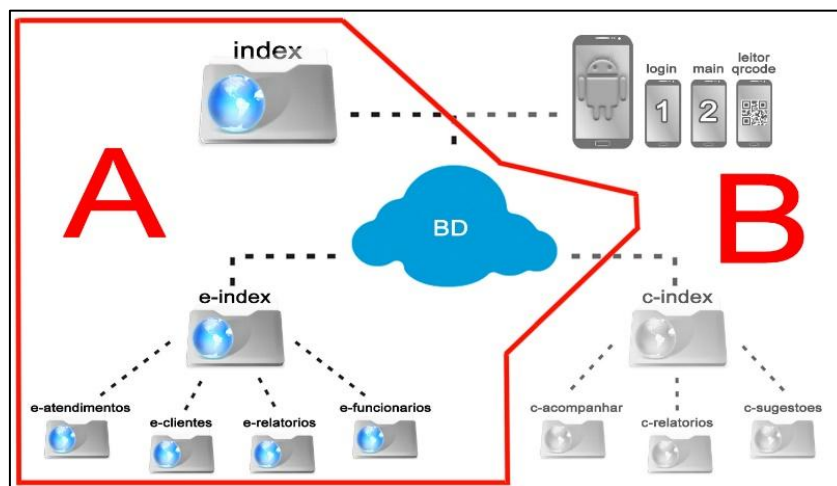


Figura 6: Implementação do Sistema.

Para visualizar a implementação do trabalho, a Figura 7 ilustra a página principal do sistema em diferentes dispositivos, demonstrando a responsividade (adaptabilidade a diferentes resoluções) do sistema.



Figura 7: Interface da Tela Inicial do Sistema em Diferentes Dispositivos.

Neste processo, é detalhado o que vai ser desenvolvido, como a tela de: Registrar Atendimentos, Cadastrar Clientes, Cadastrar Funcionários e Visualizar Relatórios, como pode ser visualizado na Figura 8.



Figura 8: Tela com *Link* para as Funcionalidades do Sistema.

A Figura 9 mostra a tela de visualização de relatórios, onde se pode ter acesso a diversos tipos desses bem como imprimi-los. Alguns dos principais relatórios implementados são: tempo médio gasto por determinado funcionário enquanto procedimento de banho; qual raça mais tomou banho.

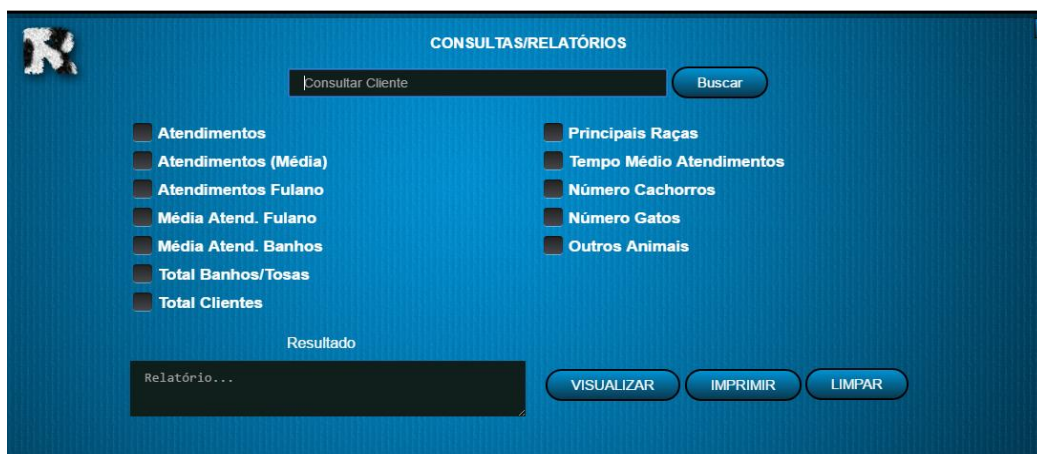


Figura 9: Tela de Visualizar Relatórios.

3.3.5. Desenvolver por Funcionalidade

É uma atividade para cada funcionalidade, a fim de produzir uma função com valor para o cliente. São implementados classes e métodos para satisfazer as funcionalidades anteriormente geradas. Logo depois passa pela fase de testes e depois é gerada uma versão [Retamal 2008].

Neste processo são implementados todas as funcionalidades descritas no processo anterior. Para não haver uma repetição na descrição do sistema, foi escolhido descrever a página de atendimento, pois ela é utilizada para o gerenciamento de todo o atendimento no estabelecimento, já que é por meio dela que o principal fluxo de execução do sistema acontece. A implementação utilizada nos *links* de clientes e de funcionários é semelhante ao de atendimentos, sendo que a diferença está apenas nas particularidades mínimas de cada subsistema.

Na Figura 10, inicialmente é realizado a condição usado para manter o usuário logado no sistema. Logo a seguir, está a instancia do arquivo *conexao.php*, que contém os dados de conexão com o banco de dados. Após, a lista usada para mostrar o resultado do item consulta é inicializada com vazio para que não mostre mensagem de erro, já que os dados só estarão disponíveis na tela após a consulta. A variável *\$q* receberá a imagem do QR Code depois de cadastrado. Assim, foi passada uma imagem sem fundo apenas para identificar o local em que o QR Code aparecerá.

```
<?php
if (!isset($_SESSION)) session_start();
if (!isset($_SESSION['UsuarioID'])) {
    session_destroy();
    header("Location: ../index.php"); exit;
}
?>
<?php
include "conexao.php";

$array[0] = ''; $array[1] = ''; $array[2] = ''; $array[4] = ''; $array[5] = ''; $array[6] = '';
$senha = '';
$senhaCliente = '';
$qrcode = '';
$q = ''; //fundo da imagem qr so com borda
$resposta = '';
```

Figura 10: Conexão com o Banco de Dados e Inicialização das Variáveis.

A Figura 11 contém uma condição usando o parâmetro *isset(REQUEST[''])* para quando o botão de consulta for clicado. Esse botão captura o valor contido no campo busca que será consultado na tabela clientes do banco de dados. O resultado é mostrado por meio de uma lista, diretamente no formulário HTML.

```
//Condicao usado para busca de clientes. Quando clicar em 'buscar', faz tal coisa.
if (isset($_REQUEST['buscar'])) {
    $busca = $_POST['busca'];
    $sql = "SELECT * FROM clientes WHERE nome='$busca'";
    $query = mysql_query($sql);
    while($row = mysql_fetch_assoc($query)) {
        $array[0] = $row['id'];
        $array[1] = $row['nome'];
        $array[2] = $row['fone'];
        $array[4] = $row['email'];
        $array[5] = $row['animal'];
        $array[6] = $row['especie'];
    }
}
```

Figura 11: Código Utilizado para a Busca dos Clientes, por meio do botão “buscar”.

Na Figura 12, é ilustrado a implementação da condição que recebe os valores contidos no formulário HTML de cadastro de atendimentos, sendo gerado o QR Code. Inicialmente, é incluído a biblioteca PHP QR Code, que possui a implementação da construção da imagem com QR Code. Após, é recebido os valores do formulário HTML e inicializado a variável *\$status* com valor “Gaiola”, já que este será sempre o *status* inicial do atendimento.

Na Figura 13, é mostrado a inserção dos dados no banco de dados. Na Figura 14, está a condição usada para gerar o arquivo no formato PDF, utilizando a biblioteca mPDF. A impressão gerada serve para ser anexado junto ao animal e para entregar ao cliente, caso deseje. Para finalizar a implementação, a conexão com o banco de dados é encerrada.

Na Figura 15, é desenvolvida a validação de acesso ao sistema com usuário e senha dispostos no banco de dados em uma coluna de acesso, pois para acessar o sistema de gestão somente o acesso do proprietário é criado.

```
} elseif (isset($_REQUEST['cadastrar'])) {
    require("phpqrcode/qrlib.php");
    $nome = $_POST['nome'];
    $fone = $_POST['fone'];
    $email = $_POST['email'];
    $animal = $_POST['animal'];
    $especie = $_POST['especie'];
    $servico = $_POST['servico'];
    $status = 'Gaiola';

    $nomeCortado = substr("$nome", 0, 5); //pegar somente as 5 primeiras letras do nome

    for($num=0; $num<5; $num++){
        //sorteia 1 numero de 0 a 9 5 vezes concatenando cada uma por meio do ponto apos a variavel
        $senha .= rand(0,9);
        $senhaCliente .= rand(0,9);
    }

    //pega data do cadastro e coloca no banco
    date_default_timezone_set('America/Sao_Paulo');
    $dataIni = date('Y-m-d H:i');

    $qrcode = $nomeCortado.$senha.'.'.$animal;
        //nome da imagem, qualidade da imagem, tamanho
    QrCode::png($qrcode, "Imagem_QRCODE.png", QR_ECLEVEL_H, 10);

    $qr = '';
}
```

Figura 12: Dados Usados para o Cadastramento no Banco de Dados.

```
$sql = mysql_query(
"INSERT INTO atendimentos
(id, nome, fone, email, animal, especie, servico, qrcode, status, dataIni, dataFim, funcionario, senhaCliente)
VALUES
('','$nome', '$fone', '$email', '$animal', '$especie', '$servico', '$qrcode', '$status', '$dataIni', '', '', '$senhaCliente')"
)or die(mysql_error());

$resposta = "Atendimento cadastrado com sucesso";
```

Figura 13: Insert usado para Cadastrar o Atendimento no Banco de Dados.

```

//Se sql ok, gera .pdf
if($sql){
    // PEGA O ARQUIVO MODELO
    $pdf = file_get_contents("modelopdf.html");

    date_default_timezone_set('America/Sao_Paulo');
    $datahora = date('d-m-Y H:i');

    // SUBSTITUI COM OS DADOS FORNECIDOS
    $pdf = str_replace("#cliente", "$nome", $pdf);
    $pdf = str_replace("#animal", "$animal", $pdf);
    $pdf = str_replace("#datahora", "$datahora", $pdf);
    $pdf = str_replace("#servico", "$servico", $pdf);
    $pdf = str_replace("#senhaCliente", "$senhaCliente", $pdf);
    $pdf = str_replace("#qr", "$qr", $pdf);

    // SOLICITA A CLASS MPDF
    require_once("mpdf/mpdf.php");

    // INSTANCIA A CLASS MPDF
    $mpdf = new mPDF();

    // ESCRIVE O PDF
    $mpdf->WriteHTML($pdf);

    // SALVA PDF
    $mpdf->Output('atendimento.pdf');
}
mysql_close($conexao);
?>

```

Figura 14: Geração do Cadastro de Atendimento em PDF.

```

<?php
if (!empty($_POST) AND (empty($_POST['usuario']) OR empty($_POST['senha']))) {
    header("Location: index.php"); exit;
}
include "c2rsistemas.com.br/conexao.php";

$usuario = mysql_real_escape_string($_POST['usuario']);
$senha = mysql_real_escape_string($_POST['senha']);

$sql = "SELECT `id`, `usuario` FROM `acesso` WHERE (`usuario` = '". $usuario ."'
        AND `senha` = '". $senha ."' ) LIMIT 1";
$query = mysql_query($sql);
if (mysql_num_rows($query) == 1) {
    $resultado = mysql_fetch_assoc($query);
    if (!isset($_SESSION)) session_start();
    $_SESSION['UsuarioID'] = $resultado['id'];
    $_SESSION['UsuarioNome'] = $resultado['usuario'];
    header("Location: e-index.php"); exit;
} else {
    echo '<script type="text/javascript">alert("Usuario inexistente ou senha incorreta.")</script>';
    exit ('<meta http-equiv="refresh" content="1; url=index.php" />');
}
?>

```

Figura 15: Validação de acesso ao sistema.

4. Avaliação do Sistema

Com processo de avaliação do sistema, aplicou-se um teste de usabilidade, em que usuários testadores identificaram algumas melhorias, como: textos usados nos botões não estavam concisos com a funcionalidade; falta de botão fechar no topo das janelas das funcionalidades; responsividade do sistema não está eficiente; formulários não estão validando certos tipos de entrada de dados; necessidade de cores mais claras por causa de certos monitores.

Também para avaliação do sistema do lado da empresa, pode-se obedecer o seguinte fluxo: i) definir empresa e implantar o sistema desenvolvido para avaliação; ii)

realizar capacitação básica (funcionário e proprietário) de uso do sistema; iii) incentivar a utilização do sistema, pela empresa durante uma semana, realizando os processos disponibilizados; iv) aplicar questionário qualitativo para verificar: CRUD (C- *create*, R- *retrieve*, U- *update*, D- *delete*), relatórios, QR Code como processo.

Todavia, para que se obtenha uma avaliação mais cuidadosa, sugere-se um processo que envolva a análise de todas as funcionalidades do sistema pelos funcionários de Consultórios Veterinários e *Pet Shops*.

5. Conclusão

O trabalho proposto teve como foco Sistemas de Informação Gerencial. Nesse contexto, buscou-se revisar conceitos básicos, os processos e ferramentas de construção de sistemas. O trabalho teve como objetivo o projeto e a implementação de um sistema de gestão e monitoramento das atividades de consultórios veterinários e *pet shops*, em que a modelagem e alguns detalhes da implementação foram exibidos.

Registra-se, também, que este estudo teve integração com o trabalho realizado por Machado (2016), que auxilia no processo de leitura de QR Code, atualizando o *status* do atendimento permitindo o monitoramento dos serviços, bem como do tempo gasto por cada funcionário por um determinado procedimento. Ademais, todas as ferramentas apontadas no projeto do sistema foram utilizadas e combinadas adequadamente.

O sistema construído possui as seguintes funcionalidades previstas na modelagem: gerenciar cliente; gerenciar funcionário; registrar atendimento e visualizar relatórios. Sendo que as funcionalidades de gerenciar produtos, controle de caixa, estoque e fornecedores não foram implementadas ainda.

Com o *software* desenvolvido, o proprietário do estabelecimento, além de ter todo o gerenciamento de cadastro de clientes e funcionários, terá relatórios de quantos banhos foram feitos por dia, raça de cada animal, avaliação de qual tamanho e raça tomam mais banhos, quanto tempo um funcionário leva para dar banho em uma determinada raça, entre outros.

Todo o sistema foi implementado conforme o projeto inicial, contudo há várias melhorias que poderão ser feitas futuramente, desde novas funcionalidades no sistema e também alterações como: o uso do *RFID* (identificação por rádio frequência) ao invés de QR Codes; utilização de um *framework* para realizar a persistência de dados, facilitando no gerenciamento de banco de dados; implementação do sistema para gestão de produtos do estabelecimento, com intuito de trabalhar o controle de entrada e saída de produtos, estoque e fornecedores.

Referências

- Arias, Juan. (2015) “Lares Brasileiros já tem mais Animais do que Crianças”, http://brasil.elpais.com/brasil/2015/06/09/opinion/1433885904_043289.html.
Novembro.
- Dall’Oglio, Pablo. (2013), PHP Programando com Orientação a Objetos, Editora Novatec.

- Elizeire, B. Mariane, Waquil D. Pauloe Gerardi Daniel. (2013) “Expansão do Mercado Pet e a Importância do Marketing na Medicina Veterinária”, Universidade Federal do Rio Grande do Sul -Trabalho de Conclusão de Curso – UFRGS.
- Junior, M. L. Heder, Verner, V. Lucas, Vaz, S. C. Monica e Vaz, G. M. S. Maria. (2012) “Estudo de Integração da Tecnologia QR Code com Banco de Dados do Framework RastroGrão”, Universidade Estadual de Ponta Grossa – UEPG.
- Laudon, Jane e Laudon, Kenneth. (2014), Sistemas de Informação Gerenciais, 11ª Edição, Editora Pearson.
- Machado, Robson P. (2016) “Sistema Online para Controle Integrado de Atividades para Consultórios Veterinários e Pet Shops”, Trabalho Final de Graduação do Curso de Sistemas de Informação, Centro Universitário Franciscano- Santa Maria – RS - Brasil.
- O’Brien, James A. (2011), Sistemas de Informação e as Decisões Gerenciais na Era da Internet, 3ª Edição, Editora Saraiva.
- Pacievitch, Yuri. (2011) “MySQL”, <http://www.infoescola.com/informatica/mysql/>. Outubro.
- Pressman, Roger S. (2010), Software Engineering a Practitioner’s Approach, Seventh Edition, McGraw-Hill Education.
- Retamal, Adail Muniz. (2008) “Feature-Driven Development: Descrição dos Processos”, <http://www.heptagon.com.br/files/FDD-Processos.pdf>. Outubro.
- Rezende, Denis A. e Abreu, Aline França De. (2011), Tecnologia da Informação Aplicada a Sistemas de Informação Empresariais, 8ª Edição, Editora Atlas.
- Santos, João Victor Marques Dos. (2013) “Controle de Atividades para Petshop Utilizando RFID”, Curso Engenharia de Computação, Centro Universitário de Brasília Trabalho de Conclusão de Curso – UniCEUB.
- Sommerville, Ian. (2011), Engenharia de Software, 9ª Edição, Editora Pearson.
- Souza, Hermes A. Dias. (2004) “Criando, Inserindo e Exibindo PHP + MySQL”, http://www.macwebhost.com.br/manuais/phpMyAdmin_Tutorial.pdf. Outubro.
- Souza, Marcos de Castro e Aires, K. R. Texeira. (2011) “Desenvolvimento de um Sistema de Controle de Acesso baseado em QR Code”, Departamento de Informática e Estatística, Universidade Federal do Piauí – UFPI.
- Turban, Efraim e Volonino, Linda. (2013), Tecnologia da Informação para Gestão: Em Busca do Melhor Desempenho Estratégico e Operacional, 8ª Edição, Editora Bookman.
- Zamberlan, Alexandre. (2010). Informática Educacional. Sistemas de Informação ou Sistemas de Conhecimento?, página19–20. Revista da ANEC.