

Construção de um Software Gerenciador de Solicitações de Reclamações e Sugestões para o Serviço Público

Renato Barbosa de Lima¹, Daniela Marques¹

¹Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Câmpus Hortolândia
Hortolândia – SP – Brasil

renatolimaeletrica@gmail.com, marquesdaniela@ifsp.edu.br

Abstract. *There is a population increase in Hortolândia city. Despite this growth, the city hall has only a service desk to register all complaints and suggestion of their population manually. The goal of this project is the development a web software that allows the citizens of Hortolândia to make their requests online providing agility to the process. This software is based on the laws that guide the information access. The population can register requests (suggestions or complaints) and verify the responses using the software. These responses will be available on the software reading a text messages or visualize the actions done by neighborhood and by request. The software will be responsive, to make it possible to use it in desktop or mobile device.*

Resumo. *Atualmente há um aumento populacional na cidade de Hortolândia. Apesar desse crescimento, a forma de realizar reclamações e sugestões na prefeitura da cidade é feito em um balcão de atendimento de maneira manual. O objetivo deste projeto é criar um software que permita que os munícipes façam suas solicitações via web, atendendo Leis de Transparência que norteiam a forma de acesso a informação pelos cidadãos e que agilizem esse processo. A população poderá registrar solicitações (sugestões ou reclamações) e verificar as respostas pelo software. Essas respostas serão disponibilizadas no software e poderão ser acessadas lendo uma mensagem de texto ou visualizando as ações divididas por bairro e por solicitante. O software será responsivo permitindo seu uso em desktops e dispositivos móveis.*

1. Introdução

As cidades têm se desenvolvido de acordo com as características de cada região. Hortolândia compõem a região metropolitana de Campinas e de acordo com as pesquisas demográficas efetuadas pelo Instituto Brasileiro de Geografia e Estatística (IBGE, 2017), o aumento populacional desta região é uma realidade. Hortolândia teve um crescimento de 13% comparando o censo realizado em 2010 e 2017, Campinas teve um crescimento de 10% neste mesmo período.

Apesar da cidade de Hortolândia ter tido um crescimento populacional nos últimos anos, a prefeitura adota um formato inadequado para acolhimento das sugestões ou reclamações da população, que é por meio de um balcão de atendimento.

O Governo do Estado de São Paulo, publicou a Lei nº 10.294 que rege o direito à informação, dispõe sobre a proteção e defesa do usuário do serviço público do Estado (SÃO PAULO, 1999). A União por meio da Lei de Transparência sugere a inovação de processos e dinamização dos serviços públicos e tem o propósito de regulamentar o direito constitucional de acesso dos cidadãos às informações públicas (BRASIL, 2011). A Lei nº 13.460 dispõe sobre participação, proteção e defesa dos direitos do usuário dos serviços públicos (BRASIL, 2017).

O Sistema de Informação ao Cidadão (SIC) é obrigatório em todos os órgãos das administrações públicas e é gerenciado pelo sistema e-SIC (Sistema Eletrônico do Serviço de Informações ao Cidadão) disponibilizado pelo governo federal em duas opções gratuitas de software.

Comentado [DM1]: tem isso mesmo? eu entendi que a comunidade solicita e não que dá soluções.

Comentado [r12R1]: A população fazendo sugestão pode apresentar soluções para problemas cotidianos ou pontuais ou ainda para apontar prováveis problemas futuros devido a falta ou má gestão de um determinado setor público.

Outro fator importante neste trabalho é o crescimento do uso de dispositivos móveis no Brasil. Conforme a Agência Nacional de Telecomunicações (ANATEL), o número de linhas móveis em operação já ultrapassa o número de habitantes do Brasil (ANATEL, 2017).

Considerando o crescimento populacional de Hortolândia, nas Leis que precisam ser cumpridas e no fato de que os dispositivos móveis possuem uma grande adesão atualmente, a Tecnologia da Informação agrega valor à comunicação entre o cidadão e o poder público.

O objetivo deste trabalho é o desenvolvimento de um software que permita a população fazer suas solicitações pela internet, criando assim um novo canal de comunicação. A população poderá registrar solicitações (sugestões ou reclamações) e receber suas respostas pelo software. Essas respostas poderão ser visualizadas por meio de mensagem de texto ou pela visualização das ações apresentadas pelo software que são divididas por bairro e por solicitante. A solução será responsiva, permitindo que o sistema seja acessível tanto em *desktops* quanto em dispositivos móveis, facilitando assim a comunicação entre o cidadão e o poder público.

A utilização do software, permitirá ao poder público armazenar informações a respeito das solicitações. Com ferramentas adequadas, a manipulação desses dados será mais fácil e poderá ajudar na gestão: identificando os bairros com mais problemas e quais departamentos são mais acionados, entre outras informações.

Este artigo está dividido da seguinte maneira: na Seção 1 é feita a introdução sobre o projeto, mostrando sua motivação; na Seção 2 são apresentadas soluções similares; a Seção 3 contém o referencial teórico; na Seção 4 são apresentados os recursos empregados no desenvolvimento; na Seção 5 é detalhado o estudo de caso e na Seção 6 conclui-se o artigo.

2. Sistemas similares

Com o objetivo de conhecer soluções similares, verificou-se que muitos órgãos públicos utilizam a opção Fale Conosco para relato de sugestões e reclamações ou Ouvidoria. Encontrou-se algumas ferramentas que tratam deste assunto. Abaixo são descritas essas soluções com um breve resumo:

- Colab.re: trata-se de um aplicativo privado disponível para realização de reclamações. Caso a prefeitura tenha um contrato, as reclamações feitas (por exemplo: um buraco na rua) pela comunidade são encaminhadas diretamente ao departamento responsável da prefeitura (COLAB, 2017).
- e-SIC: permite que qualquer pessoa, física ou jurídica, encaminhe pedidos de acesso à informação, acompanhe o prazo e receba a resposta da solicitação realizada para órgãos e entidades do Executivo Federal (e-SIC, 2017).
- e-OUV: este sistema é um canal integrado para encaminhamento de manifestações a órgãos e entidades do Poder Executivo Federal (e-Ouv, 2017).
- ReclameAqui Serviços Públicos: plataforma web disponível gratuitamente. A partir do momento em que uma reclamação é postada no site, o órgão responsável recebe a crítica. Fica a cargo dos órgãos públicos dar uma resposta aos cidadãos (ReclameAqui, 2017). A Figura 1 apresenta uma imagem desta plataforma com os dados referente a cidade de Hortolândia é possível observar que nenhuma das reclamações foram respondidas pela prefeitura.
- CidadãoOnline – serviço disponível para aplicativo móvel e web, contudo, o órgão público precisa pagar para instalar e configurar o sistema, por exemplo: e-mail para envio e recebimento de respostas e usuários e suas permissões no sistema. Para cada reclamação feita, é gerado um número de protocolo (CIDADÃOONLINE, 2017).

Comentado [DM3]: tem isso mesmo? eu entendi que a comunidade solicita e não que dá soluções.

Comentado [r14R3]: A população fazendo sugestão pode apresentar soluções para problemas cotidianos ou pontuais ou ainda para apontar prováveis problemas futuros devido a falta ou má gestão de um determinado setor público.

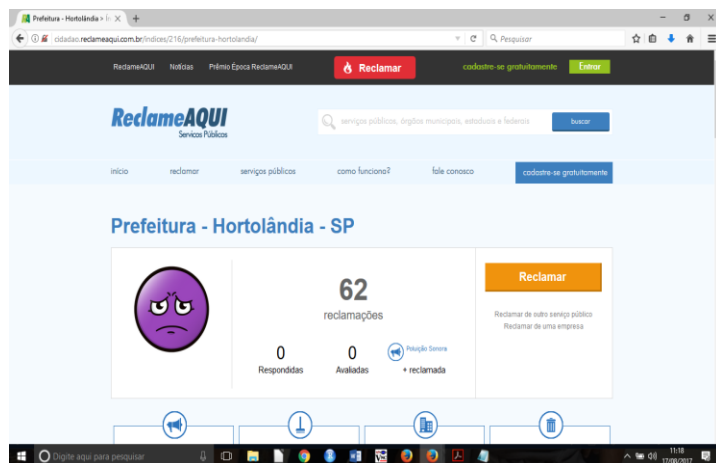


Figura 1. Reclama Aqui Serviços Públicos.

Estas plataformas são utilizadas em alguns órgãos públicos, entretanto, os mesmos não tem a obrigação de possuir nenhum tipo de convênio com as empresas. Conforme Lei nº 13.460 as reclamações do usuário devem ser encaminhadas à Ouvidoria do órgão diretamente, com prazo de resposta de 20 dias, prorrogáveis por mais 10 dias mediante justificativa (BRASIL, 2017).

Esta pesquisa possibilitou conhecer algumas soluções disponíveis e ampliar o conhecimento sobre o objeto de estudo deste trabalho. Percebeu-se que uma solução mais simples poderia auxiliar a prefeitura de Hortolândia sem gerar nenhum custo à prefeitura.

Comentado [DM5]: Só pode deixar isso se vc tiver referencia, senão é sua opinião e daí não pode existir em artigo científico. , dificultando o uso por parte dos interessados e desmotivando o uso por parte dos munícipes por não serem respondidas suas solicitações de pronto.

3. Referencial Teórico

Os conceitos e definições utilizados para este projeto serão descritos de maneira sucinta nas subseções a seguir.

3.1. Modelo de Desenvolvimento de Software

Software de computador é o produto que profissionais de software elaboram e, depois, mantêm ao longo do tempo. Esse produto abrange programas executáveis, código-fonte, documentação na forma impressa ou virtual, entre outros. O desenvolvimento de software não é uma tarefa trivial (PRESSMAN, 2011).

A Engenharia de Software afirma que o software não deve ser construído sem antes se fazer uma análise da solução. Portanto, sugere a adoção de um modelo de desenvolvimento de software. Segundo Soares (2004), existem vários processos de software definidos na literatura da Engenharia de Software. É comum ainda algumas organizações criarem seu próprio processo ou adaptem algum processo à sua realidade.

Dentre os vários processos existentes, existem as chamadas metodologias orientadas a plano, que possuem maior documentação e as metodologias ágeis, que focam no desenvolvimento e não na documentação e lidam com requisitos parcialmente definidos.

Adotar uma abordagem sistemática e organizada costuma ser a maneira mais eficiente de produzir software de alta qualidade. Para selecionar o método mais adequado para um desenvolvimento leva-se em consideração o tipo de software. Para sistemas web é melhor adotar

uma abordagem mais criativa e menos formal. Esse tipo de sistema requer habilidades distintas e com entregas iterativas (SOMMERVILLE, 2011).

3.2. Requisitos Funcionais e Não funcionais

Os requisitos de software nada mais são do que um conjunto de atividades que o software deve desempenhar, com suas limitações e restrições, além de características não ligadas diretamente às funções desempenhadas pelo software. Quanto mais compreensível, precisa e rigorosa for a descrição de um requisito de sistema, maior será a proporção quanto ao grau de qualidade do produto resultante (SOMMERVILLE, 2011).

Para representação dos requisitos funcionais pode-se usar a notação *Unified Modeling Language* (UML) elaborando o Diagrama de Caso de Uso. Por definição, um caso de uso é a especificação de uma sequência completa de interações entre um sistema e um ou mais agentes externos a esse sistema e é caracterizado por representar um relato de uma funcionalidade, sem revelar sua estrutura e o seu comportamento interno (BEZERRA, 2015).

Requisitos não funcionais são restrições que especificam os critérios que podem ser utilizados para avaliar o funcionamento de um sistema, por meio de comportamentos específicos (LARMAN, 2000). A segurança, a confiabilidade e o desempenho do sistema são exemplos de requisitos não funcionais.

3.3. Mineração de Dados

Podemos descrever mineração de dados, como um conjunto de técnicas que procura descobrir padrões de dados estruturados, ou não estruturados, visando gerar conhecimento e agregar valor à informação.

Segundo (Amo. Sandra, 2003) Mineração de Dados é uma área de pesquisa multidisciplinar, incluindo tecnologia de bancos de dados, inteligência artificial, aprendizado de máquina, redes neurais, estatística, reconhecimento de padrões, sistemas baseados em conhecimento, recuperação da informação, computação de alto desempenho e visualização de dados.

3.4. Software Livre

Os usuários possuem acesso livre ao código-fonte do software e podem alterar conforme suas necessidades. Um software livre envolve vários requisitos e o usuário além de poder instalar ou fazer distribuição, pode também alterar o código do software para estudo, cópias, modificações, sem a necessidade de permissão (PALMIERI, 2012, pág. 2).

3.5. HTML 5

A linguagem de marcação *HyperText Markup Language* (HTML) na versão 5 é um novo marco no desenvolvimento de páginas web, atendendo a norma W3C, trouxe um novo conceito a linguagem incorporando funcionalidades ao hipertexto, como por exemplo: maior organização e semântica aos marcadores. Possibilita o desenvolvedor modificar as características dos objetos de forma não intrusiva, fazendo com que isso fique transparente para o usuário final (FEITOSA, 2012).

3.6. CSS

Cascading Style Sheets (CSS) são folhas de estilos em cascata, que quando incorporado a página HTML traz recursos no controle da aparência na indentação e desenvolvimento de layouts responsivos tornando a visualização das páginas mais agradável e semanticamente corretas (ANDY, Budd).

3.7. JAVASCRIPT

O JavaScript é uma linguagem interpretada que é utilizada dentro de um ambiente maior, seja ele um script de um navegador, do lado do servidor, ou algo similar. JavaScript serve para dar dinamismo às páginas HTML. (MANZANO & TOLEDO,2010).

3.8. BOOTSTRAP

O Bootstrap é, atualmente, o principal framework CSS para construção do *front-end* de aplicações web. Um dos seus recursos é o sistema de grid responsivo, que permite desenvolver com facilidade páginas que se adaptam aos diferentes tamanhos de tela. Além disso, o Bootstrap oferece um amplo conjunto de recursos, que em muitos casos evita a necessidade de escrita de código CSS (GUIA DE LINGUAGEM: Bootstrap).

3.9. PHP

O nome PHP é uma sigla que significa “*PHP Hypertext Preprocessor*”. PHP é uma linguagem de *script* interpretada utilizada majoritariamente para desenvolver aplicações Web embutidas em HTML, mas também pode ser usada para o desenvolvimento de aplicações *desktop* e *scripts* de linha de comando. O PHP/FI foi criado por Rasmus Lerdorf por volta de 1995 com o intuito de controlar os acessos à sua *home page* (currículo *online*). O PHP é uma linguagem de *script Server-Side*, isto é, um conjunto de códigos que serão interpretados no servidor. (ZANDSTRA, 2009).

3.10. R Base

R é uma linguagem e ambiente para computação estatística e gráficos, o software R, é um software livre e possui diversas bibliotecas disponíveis para uso, o software pode ser utilizado acessando as bases de dados via linha de comando ou através de interface gráfica instalada posteriormente(AQUINO, 2014).

3.11. MVC

O padrão de arquitetura MVC (modelo, visão, controlador), é amplamente utilizado como base de gerenciamento de sistemas baseados em tecnologias WEB. Este padrão tem como característica organizar o sistema em camadas, separando suas funcionalidades gerando baixo acoplamento e apresenta sua arquitetura a partir de diferentes visões conseguindo separação e independência entre as camadas, proporcionando facilidade em manutenção além de facilitar o desenvolvimento de sistemas distribuídos em equipes. (SOMMERVILLE, 2011).

3.12. Star UML

Software utilizado para modelagem de diagramas utilizando a notação UML. Para a confecção dos modelos e diagramas, a ferramenta Star UML se mostrou muito prática, com um aprendizado rápido e de fácil usabilidade (STARUML, 2017).

3.13. SublimeText 3

O Sublime Text é um editor de texto sofisticado para código, com inúmeras funcionalidades e totalmente configurável de acordo com a necessidade do usuário (AMARAL, 2017).

3.14. Framework CodeIgniter

CodeIgniter utiliza linguagem PHP e está no mercado desde 2006. Foi desenvolvido sobre o paradigma da programação Orientada a Objetos sob o padrão de arquitetura de software MVC, mas também já dá suporte à metodologia *Hierarchical Model-View-Controller* (HMVC). O HMVC é uma evolução do MVC. No CodeIgniter, é como se a estrutura básica de funcionamento tivesse um nível hierárquico acima, tornando-a modular (CODEIGNITER, 2017).

Comentado [r16]: Referencia

3.15. Xampp

O Xampp é um conjunto de ferramentas distribuído sobre a licença de código livre ou *open-source*. Com ele, é possível rodar sistemas como *WordPress* e *Drupal* localmente, o que facilita e agiliza o desenvolvimento (XAMPP, 2017).

3.16. Servidor Apache

O servidor Apache é compatível com o protocolo HTTP versão 1.13. Suas funcionalidades são mantidas por meio de uma estrutura de módulos, o que permite que os usuários escrevam seus próprios módulos. Ele é multiplataforma, além de ser licenciado *General Public License* (GNU) (APACHE, 2017).

3.17. MYSQL

Segundo NAVATHE (2011) é um dispositivo que consegue fazer modificações e manipulações nos conjuntos de dados que estão armazenados em um Banco de dados, facilitando assim, seu processo de elaboração, controle de usuários, e gerenciamento.

4. Desenvolvimento do software

Neste projeto foi utilizada uma abordagem incremental, para cada incremento utilizou-se as fases contidas no modelo Cascata. Foram levantados os requisitos necessários no sistema, elaborado seu diagrama de caso de uso e depois feita a análise para dividir os requisitos em incrementos.

Antes do início do desenvolvimento, foram estudados os sistemas citados na Seção 2 para maior conhecimento sobre o assunto e para extrair ideias para o sistema.

Neste trabalho foi utilizada a linguagem PHP na versão 7.0 que oferece suporte nativo ao desenvolvimento orientado a objetos desde a versão 5.

4.1 Levantamento de Requisitos

A primeira atividade feita no processo de desenvolvimento foi o levantamento de requisitos, que foi dividida em duas tarefas: (i) identificar os requisitos do sistema e (ii) levantar quais informações seriam necessárias. Apesar das duas tarefas estarem interligadas, houve um processo distinto para cada uma.

Para o levantamento de requisitos foram levados em consideração os softwares estudados e a necessidade atual da prefeitura. A seguir são apresentados os requisitos identificados:

1. Armazenar dados de cadastro do usuário;
2. Registrar solicitações (sugestões ou reclamações);
3. Anexar mensagem;
4. Anexar imagem;
5. Salvar dados GPS;
6. Visualizar imagens na galeria de fotos;
7. Responder as solicitações;
8. Realizar login pelo *facebook*;
9. Criar um *chat* para atendimento.

Para o levantamento das informações que seriam úteis para o sistema, foi pesquisado quais eram os órgãos públicos que acolhiam as reclamações dos munícipes de Hortolândia com maior frequência. No sítio da Câmara Municipal de Hortolândia (CMH) há uma página que armazena as atas (em formato pdf) de todas as sessões realizadas para consulta (Figura 2). A partir desses arquivos e com ajuda de uma ferramenta de estatística foi possível extrair os dados utilizados nos processos de solicitações, selecionando os diversos pedidos feitos e relacionando os mesmos ao bairro que ocorreu a solicitação. Neste caso foi aplicado o uso do software R base e a interface RStudio utilizando os conceitos de mineração de dados.



Figura 2. Página de download das atas das sessões plenárias.

Com as informações obtidas conforme pesquisa feita nas atas das sessões da CMH durante o ano de 2016, foi definido quais informações deveriam ser coletadas pelo sistema. Essa tarefa definiu quais dados seriam recebidos dos munícipes, uma vez que a recepção e armazenamento de dados constantes nas solicitações seriam enviadas para aos setores envolvidos para análise. Consequentemente, os usuários do sistema teriam um acompanhamento da solicitação feita, assim como poderiam ver a solução encaminhada. Esses dados também poderiam auxiliar a gestão pública identificando os bairros que possuem mais problemas e dentro deste escopo, elencar quais os problemas mais recorrentes do município.

A partir dos dados coletados, lista de requisitos e informações que deveriam fazer parte do sistema, o diagrama de caso de uso foi elaborado. A Figura 3 apresenta o diagrama elaborado na ferramenta Star UML representando as funcionalidades do sistema que foi elaborado e os atores do sistema.

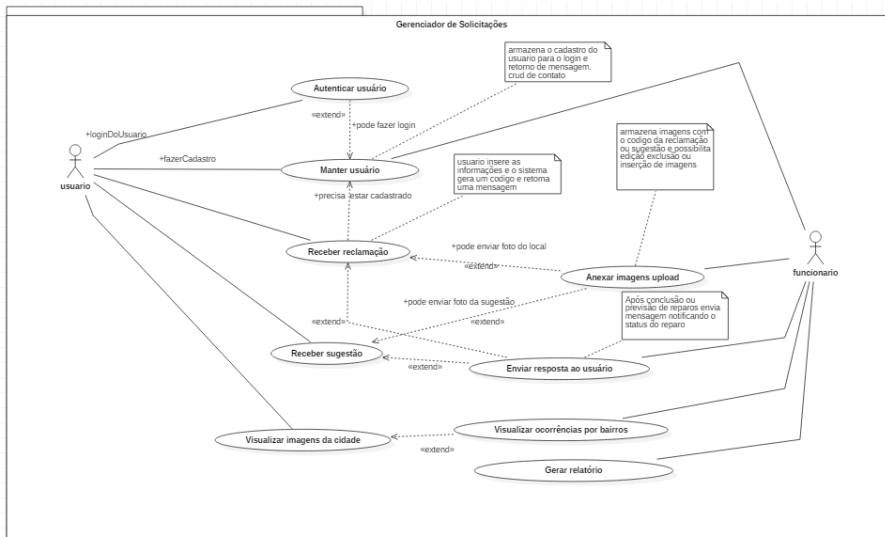


Figura 3. Diagrama de caso de uso.

No modelo incremental (Figura 4) adotado é necessário que os casos de uso sejam priorizados. A Tabela 1 mostra os casos de uso identificados por uma numeração. Os casos de uso foram divididos em incrementos em função do modelo adotado neste desenvolvimento. O primeiro incremento foi composto pelos Casos de Uso identificados de 1 a 4 (referente ao cadastro e login de usuário; *home* e Fale - Conosco); o segundo incremento composto pelos casos de uso de 5 a 9 (enviar solicitações; anexar imagens; visualizar ocorrências por bairro e gerar relatórios); e o último incremento (3) possui o caso de uso 10 que se refere à resposta do usuário.

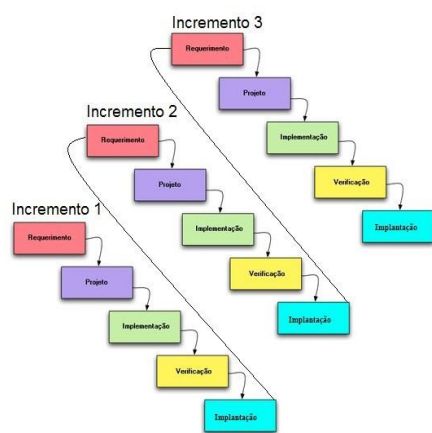


Figura 4. Modelo Incremental.

Tabela 1. Casos de Uso identificados e referido incremento

Identificador	Caso de Uso	Incremento
1	Manter usuário	1
2	Autenticar usuário	
3	Visualizar imagens da cidade (<i>Home</i>)	
4	Disponibilizar página Fale - Conosco	
5	Receber Reclamação	2
6	Receber Sugestão	
7	Anexar imagens (<i>upload</i>)	
8	Gerar relatório	
9	Visualizar ocorrências por bairro	
10	Enviar resposta ao usuário	3

Após o entendimento do negócio, análise dos requisitos e dados a serem armazenados, optou-se pelo desenvolvimento de uma aplicação web. Levou-se em consideração a necessidade

de atingir um número elevado de usuários acessando o sistema de qualquer local e a crescente utilização dos dispositivos móveis no Brasil.

4.2 Modelagem do Banco de Dados

Após a definição dos casos de uso foi possível conhecer as funcionalidades e dados que seriam necessários no sistema. Com a ajuda da ferramenta PhpMyadmin foi desenvolvido Diagrama Entidade-Relacionamento (DER) para um melhor entendimento do sistema. A Figura 5 apresenta o DER do domínio do sistema, modelando suas entidades e relacionamentos entre as mesmas.

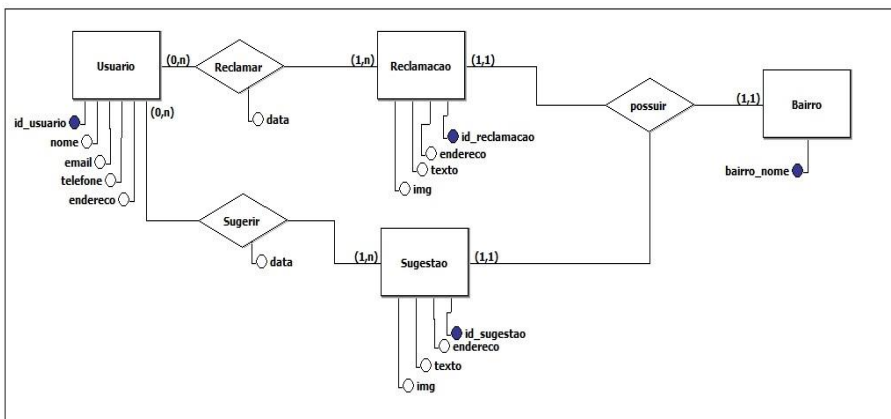


Figura 5. Diagrama Entidade-Relacionamento.

4.3 Implementação

Dentro de cada incremento, as seguintes atividades foram realizadas: elaboração da tela; codificação do caso de uso; realização dos testes e correção dos erros, quando necessário.

Para o primeiro incremento, a primeira atividade realizada foi a confecção do *layout* das telas do sistema contendo as funcionalidades previstas nos casos de uso identificados como 1, 2, 3 e 4 que representam o cadastro do usuário, a autenticação no sistema e a visualização de imagens da cidade e Fale-Comosco. Foi usado o Bootstrap onde seus componentes visuais são responsivos.

Após a construção das telas, foram efetuados os testes de responsividade e usabilidade adotando as heurísticas de usabilidade sendo frisada a importância da identificação da página, para que serve e onde posso ir, sem ter de pensar para executar a ação (KRUG, Steve, 2013). Foi desenvolvido um questionário de perguntas (Tabela 2) sobre a fluidez das páginas, facilidade de entendimento de itens e botões das telas, facilidade de preenchimento do cadastro de usuário, de login e de solicitações (reclamações e sugestões). As perguntas foram enviadas para 7 pessoas para avaliação das telas propostas.

Os testes também consideraram a verificação de mensagens de erros e acertos em relação às ações realizadas pelo usuário. Esta característica foi levada em consideração para proporcionar

Comentado [DM7]: Foi realmente feito esse questionário? Para ser válido é necessário que 5 pessoas respondam essa pesquisa e com perfis diferentes.

uma melhor experiência de usabilidade ao usuário, fazendo com que o mesmo não ficasse sem nenhuma resposta quando realizasse uma ação. A Figura 6 apresenta a tela inicial da aplicação para *desktop*.

Tabela 2. Questionário aplicado aos usuários para avaliação da *interface*

Número	Pergunta
1	As cores utilizadas na aplicação web são agradáveis?
2	O <i>layout</i> da aplicação é atraente?
3	A aplicação poderia ser útil na relação entre o cidadão e o responsável do setor público?
4	Com que frequência você costuma efetuar solicitações de serviços públicos.
5	O conteúdo da aplicação é relevante para o seu público alvo (pessoas acima de 16 anos)?
6	Foi fácil entender a aplicação e começar a utilizá-la?
7	As funcionalidades propostas nas telas da aplicação são adequadas?
8	Conseguiu chegar ao final da tarefa de efetuar o cadastro?
9	Conseguiu chegar ao final da tarefa de efetuar o <i>login</i> ?
10	Conseguiu efetuar uma solicitação ou reclamação?

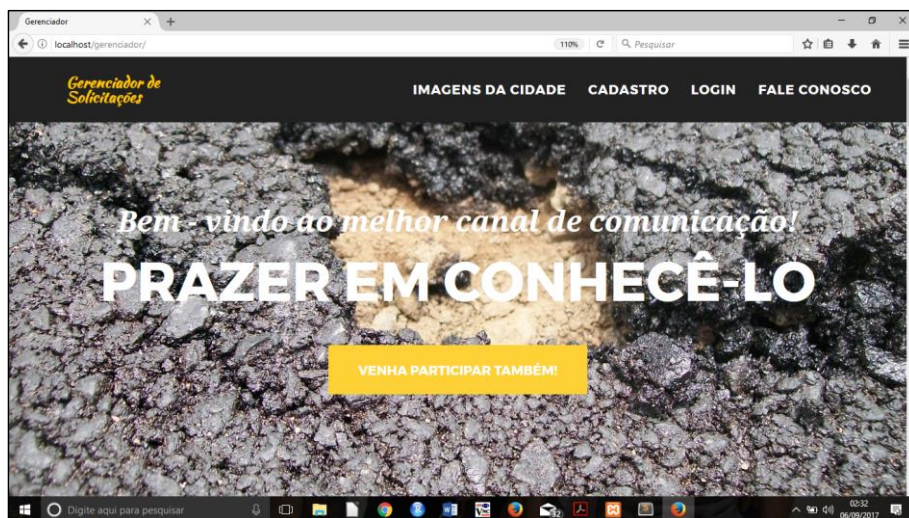


Figura 6. Tela inicial do sistema para *desktops*.

A segunda atividade dentro de cada incremento era a implementação das funcionalidades que seriam necessárias para o funcionamento dos casos de uso.

Nesta etapa do projeto, foi necessário o aprofundamento no uso da linguagem PHP em conjunto com o framework *CodeIgniter* para gerar as classes e objetos de cada tela e para utilizar a arquitetura MVC.

Neste projeto foi utilizado um servidor local apache. Sendo assim, a pasta ficará dentro do diretório `xampp/htdocs/meuprojeto`. A pasta “meuprojeto” contém 3 pastas (Figura 7):

- *application*: fica a aplicação que será desenvolvida;
- *system*: fica o núcleo do *framework* CodeIgniter; e,
- *user_guide*: fica toda a documentação do *framework*. Esta pasta deve ser retirada do projeto quando a aplicação for colocada em produção.



Figura 7. Estrutura de pastas utilizada pelo *framework* CodeIgniter.

A seguir são apresentadas alguns diretórios e arquivos da pasta *application*. Na pasta *config* estão as configurações relativas a todas as bibliotecas e arquivos que serão utilizadas no projeto, onde os principais arquivos são:

- *Autoload.php*: para inicializar as bibliotecas e configurações na inicialização da aplicação.
- *Database.php*: para definir as configurações do SGBD utilizado pela aplicação.
- *Routes.php*: para estabelecer a rota padrão da aplicação, e as rotas alternativas criadas para utilização de rotas amigáveis, pois o framework trabalha com o conceito de endereço composto por diretório da aplicação/controlador/método/seguimento.

Na pasta *controller*, ficam os controladores que na arquitetura MVC é responsável por receber requisições da *view* (tela) e quando necessário comunicar-se o *model*. O *model* irá retornar uma resposta a *view*.

Na pasta *model*, fica o modelo que recebe uma requisição do controlador e faz a interação com o banco de dados.

Na pasta *view*, encontra-se as telas e *templates* da aplicação onde o usuário acessa o sistema.

O arquivo *.htaccess* é responsável por tornar as URLs mais amigáveis aplicando regras às rotas para alterar o endereço na barra de tarefas e alterar o endereço de: `http://meuprojeto/index.php/controlador/login` para `http://meuprojeto/login`.

Dessa maneira, iniciou-se a fase de testes, verificando se as funcionalidades propostas nos casos de uso de 1 a 4 estavam corretas. Caso existissem erros ou inconsistências nas informações apresentadas em tela, os defeitos eram corrigidos.

Com a finalização dos testes do primeiro incremento, iniciou-se o 2º incremento, onde serão contemplados os casos de uso de 5 a 9. O segundo incremento seguiu o mesmo processo, elaboração de telas, codificação e correção de erros.

Esse incremento tinha uma característica específica pois todas as telas precisavam que o usuário estivesse logado. Neste caso, era criado um objeto de sessão por motivo de segurança. Um objeto de sessão é uma função nativa do framework *CodeIgniter*, que possui duas opções de armazenamento dos dados do usuário na sessão: (i) armazenada em forma de *array* num *cookie* temporário de acordo com a definição de tempo nas configurações internas e (ii) armazenando a sessão em um banco de dados.

Seguindo as instruções contidas na documentação (CODEIGNITER, 2017), é possível configurar uma chave de encriptação e utilizar o algoritmo MD5 para encriptar senhas, de modo que aumente a segurança ao salvar dados do usuário no banco de dados. A Figura 8 mostra a responsividade em dispositivos móveis das telas desenvolvidas. A Figura 9 apresenta a visão da página do Administrador logado no sistema para *desktops*.

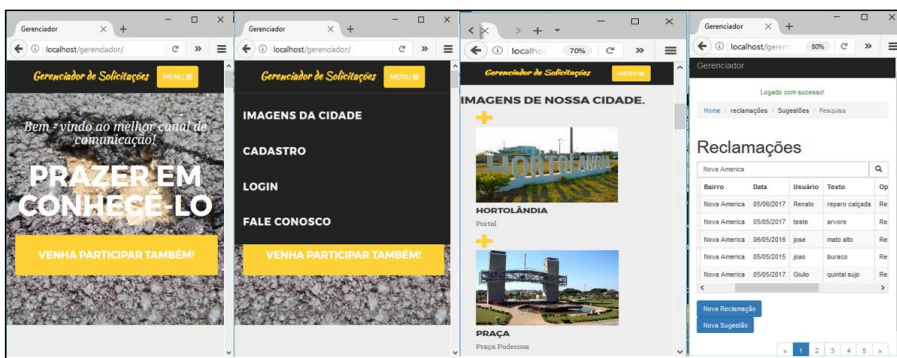


Figura 8. Sequência de telas em modo responsivo para dispositivos móveis.

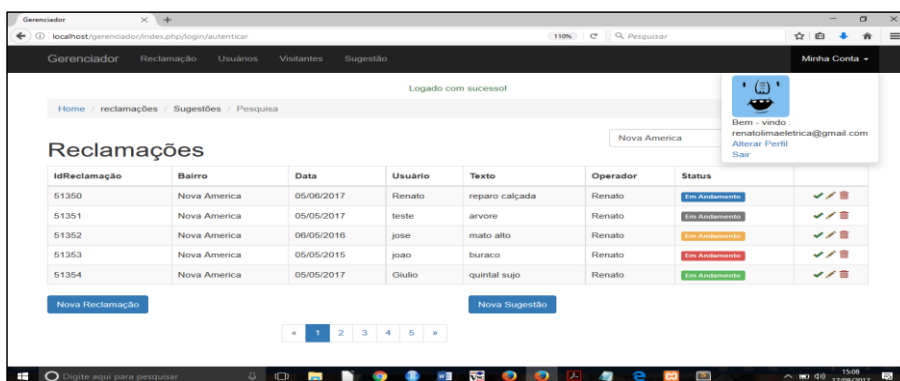


Figura 9. Visão da página Administrador logado (*desktop*).

Ao final do 2º incremento, o sistema foi considerado finalizado. O 3º incremento teve seus requisitos levantados, contudo foi considerado para um trabalho futuro. Atualmente seria necessária uma integração entre esse aplicativo e outros sistemas em produção da prefeitura para conseguir dar o retorno ao usuário.

5. Conclusão

O software desenvolvido permite que os munícipes registrem sugestões e reclamações *online*, usando *desktop* ou dispositivos móveis. O retorno ao usuário ainda não é feito de maneira automatizada pois teria que haver uma integração com o software da prefeitura.

O fato de poder realizar uma solicitação *online* facilitaria o acesso do cidadão à prefeitura, poupando o munícipe de ter que se deslocar até a prefeitura para registrar sua sugestão ou reclamação. Outro ponto a se destacar, é que a prefeitura teria informações das solicitações armazenadas adequadamente, podendo com isso gerar relatórios e melhorando o atendimento ao cidadão por meio de identificação dos bairros com mais problemas e quais os departamentos mais acionados da prefeitura.

Todos os casos de uso adicionados no incremento 1 e 2 tiveram suas funcionalidades desenvolvidas e testadas e todos os problemas encontrados foram corrigidos. Como trabalho futuro, seria necessário desenvolver o 3º incremento, integração com sistema da prefeitura para retorno ao usuário, isso completaria os casos de uso identificados. Outros requisitos foram mapeados e também poderiam ser desenvolvidos em futuras interações, são eles: proposta da utilização de coordenadas geográficas GPS, realizar login pelo *facebook* e criar um chat para atendimento.

Este projeto trouxe uma nova experiência de conhecimento, ao utilizar de forma integrada os conceitos aprendidos no curso de tecnologia em Análise e Desenvolvimento de Sistema e também trouxe vários desafios ao enfrentar situações onde não possuía conhecimento.

Destaca-se aqui que foi possível utilizar os conceitos aprendidos nas disciplinas de engenharia de software, análise orientada a objetos, programação orientada a objetos, banco de dados, estatística, tópicos especiais, programação web e gestão de projetos de maneira articulada para realização do projeto. Entendeu-se a necessidade de organizar e estabelecer metas e limites com duração nas atividades e a ordem em que essas atividades devem ser feitas. Foi possível compreender a importância de se gerar um artefato em cada fase do projeto e o porquê de sua utilização.

Dos vários desafios enfrentados no desenvolvimento do Trabalho de Conclusão de Curso, pode-se citar: elaboração de diagramas, desenvolvimento orientado a objetos utilizando o padrão MVC e principalmente a escrita e documentação do software e deste artigo. A utilização de conteúdos extras curriculares, como o estudo de um framework específico, proporcionou uma nova experiência, típicas de quando se descobre algo novo e que ajuda na formação do perfil profissional.

Referências Bibliográficas

ANATEL (2017). **Brasil encerra junho com 242,1 milhões de linhas móveis em operação.**

Disponível em: <<http://www.anatel.gov.br/dados/destaque-1/283-movel-acessos-maio>>.

Acesso em 31 ago. 2017.

- AMARAL, R. 2017. **Guia rápido do sublime text: 20 Dicas para quem não quer perder tempo:** nesse guia de consulta você encontrará o conteúdo que precisa para referenciar seus estudos sobre Editor de texto Sublime text. Disponível em: <<https://s3-sa-east-1.amazonaws.com/sublimetextdicas/ebook-guia-rapido-www.sublimetextdicas.com.br.pdf>>. Acesso em 15 ago. 2017.
- AMARAL S. de: **Curso de Data Mining, Programa de Mestrado em Ciência da Computação,** Universidade Federal de Uberlândia, 2003. <<http://www.deamo.prof.ufu.br/CursoDM.html>> Acesso em 20 ago. 2017.
- APACHE, 2017a. **Apache License.** Disponível em: <<http://httpd.apache.org/docs/2.2/pt-br/license.html>>. Acesso em 27 mar. 2017.
- APACHE, 2017b. **Servidor HTTP Apache.** Disponível em: <<http://httpd.apache.org/docs/2.2/pt-br/>>. Acesso em 27 mar. 2017.
- AQUINO, Jakson Alves de R para cientistas sociais/ Jakson Alves de Aquino. – Ilhéus, BA: EDITUS, 2014. 157 p.: il.
- BRASIL. **Lei nº12.527**, de 18 de Novembro de 2011. **Regulamenta o acesso a informações.** Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm>. Acesso em 31 ago. 2017.
- _____. **Lei nº 13.460**, de 26 de Junho de 2017. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2017/lei/L13460.htm>. Acesso em 31 ago. 2017.
- BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistema com UML.** Rio de Janeiro: Elsevier, 2015.
- CIDADÃO, 2017. **Cidadão.** Disponível em: <<http://www.cidadao.sp.gov.br/>>. Acesso em 31 ago. 2017.
- CIDADÃOONLINE, 2017. **Cidadão Online.** Disponível em: <<http://cidadaonline.sistemas4r.com.br/>>. Acesso em 31 ago. 2017.
- CODEIGNITER (2017). **CodeIgniter.** Disponível em: <https://codeigniter.com/>. Acesso em 31 ago. 2017.
- COLAB (2017). **Colab.re.** Disponível em: <<https://www.colab.re/>>. Acesso em 31 ago. 2017.
- DALL’OGLIO, Pablo. **PHP: programando com orientação a objetos.** São Paulo: Novatec, 2007.
- e-SIC. 2017. **Sistema Eletrônico de Serviço de Informação ao Cidadão.** Disponível em: <<https://esic.cgu.gov.br/sistema/site/index.aspx>>. Acesso em 31 ago. 2017.
- e-Ouv. 2017. **Sistema de Ouvidorias do Poder Executivo Federal.** Disponível em: <<http://www.ouvidorias.gov.br/>>. Acesso em 31 ago. 2017.
- FEITOSA, E. **O que é o HTML5.** 2012. Disponível em: <<http://www.devmedia.com.br/o-que-e-o-html5/25820>>. Acesso em 31 ago. 2017.

GUIA DE LINGUAGEM: **Bootstrap**: nesse guia de consulta você encontrará o conteúdo que precisa para referenciar seus estudos sobre framework Bootstrap. Disponível em: <<http://www.devmedia.com.br/guia/guia-de-referencia-bootstrap/38150>>. Acesso em 15 ago. 2017.

IBGE (2017). **Hortolândia**. Disponível em: <<https://cidades.ibge.gov.br/v4/brasil/sp/hortolandia/panorama/>>. Acesso em 20 ago. 2017.

KRUG, Steve. **Não me faça pensar**: Uma abordagem de bom senso à usabilidade na web. Trad. 2ª ed. Rio de Janeiro. Alta Books, 2013.

LAWCE, Bruce. **Introdução ao HTML5**. Rio de Janeiro: Alta Books, 2011.

MANZANO, José Augusto N. G. & Toledo, Suely Alves. **Guia de orientação e desenvolvimento de sites**: HTML, XHTML, CSS E JAVASCRIPT/JSCRIPT. São Paulo: Érica, 2010.

PHPMYADMIN. **Bringing MySQL to the web**. Disponível em: <<https://www.phpmyadmin.net/>>. Acesso em 27 ago. 2017.

PRESSMAN, Roger S. **Engenharia de Software**: uma abordagem profissional. 7. ed. Porto Alegre: AMGH, 2011.

R Development Core Team. **R**: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/> Acesso em 10 mar. 2017.

ReclameAqui, 2017. **Reclame Aqui Serviços Públicos**. Disponível em: <<http://cidadao.reclameaqui.com.br/>>. 31 ago. 2017.

SÃO PAULO (estado). Lei nº10.294, de 20 de abril de 1999. Disponível em: <http://www.corregedoria.sp.gov.br/adm/App_Cadastro/Uploads/Visualizar.aspx?id=75>. 31 ago. 2017.

SCHWABER, K, Beedle, M. **Agile Software Development with Scrum**. Prentice Hall, 2002.

SOMMERVILLE, I. **Engenharia de Software**. 9.ed. São Paulo: Addison-Wesley, 2011.

SOARES, Bruno. A. L. **Aprendendo a linguagem PHP**. Rio de Janeiro: Ciência Moderna Ltda, 2007.

SOARES, M. S. Comparação entre metodologias ágeis e tradicionais para o desenvolvimento de software. *Academia.edu*. Conselheiro Lafaiete, MG: Universidade Presidente Antônio Carlos, 2004. Disponível em: <http://www.academia.edu/9335641/Compara%C3%A7%C3%A3o_entre_Metodologias_%C3%81geis_e_Tradicionais_para_o_Desenvolvimento_de_Software>.

STAR UML, 2017. **StarUml**. Disponível em: <http://staruml.io/>. 31 ago. 2017. UML 2.0, 2011.

XAMPP, 2017. **Xampp**. Disponível em: <<https://www.apachefriends.org/pt.br>>. Acesso em: 27 mar. 2017.

ZANDSTRA, Matt. **Objetos PHP, padrões e prática**. 2. ed. Rio de Janeiro: Alta Books, 2009.