

DESENVOLVIMENTO DE UM SISTEMA WEB PARA TROCA DE LIVROS

Marcos Vinicius Silva Ortiz¹

Rodrigo Malara²

RESUMO

Este trabalho tem como objetivo demonstrar as fases do processo de um sistema web voltado para o compartilhamento de publicações impressas, tais como livros e revistas sem a utilização de valor monetário, utilizando-se de créditos disponibilizados através de cada troca efetuada no próprio sistema. Para o desenvolvimento do sistema foi utilizada a linguagem de programação PHP orientada a objeto, através do framework CodeIgniter, com suporte ao modelo MVC, junto do gerenciador de banco de dados MySQL, utilizando a ferramenta Atom para a fase de programação. Foi utilizada ferramentas da UML 2.0 para auxiliar todo o processo de desenvolvimento, desde a fase de levantamento de requisitos até a implantação das telas do sistema. O processo contou também com o auxílio do GIT como software de gestão de configuração. O processo atingiu o objetivo ao final do processo, tendo como resultado um sistema web funcional que permite a inclusão de usuários, cadastro de publicações disponíveis e lista de desejos, busca por publicações cadastradas, solicitações de publicações de outros usuários, e apresentação organizada por solicitações enviadas e recebidas.

PALAVRAS-CHAVES

Sistema Web PHP. Orientação a Objeto. MVC.

¹ Acadêmico do Curso de Sistemas de Informação no Centro Universitário de Araraquara (UNIARA). E-mail: visortiz@gmail.com

² Professor M.Sc. Orientador e Coordenador do Curso de Sistemas de Informação no Centro Universitário de Araraquara (UNIARA). E-mail: rmalara@uniara.com.br

DEVELOPMENT OF A WEB SYSTEM FOR BOOKS EXCHANGE

ABSTRACT

This work aims to demonstrate the phases of the process of a web system designed for sharing printed publications, such as books and magazines without the use of monetary value, using the available credits through of each change made in the system itself. To develop the system we used the PHP programming language object-oriented, through the CodeIgniter framework with MVC model support, with the MySQL database manager, using Atom tool for the programming phase. It was used UML 2.0 tools to support the entire development process, from gathering requirements to the development of the system's screens. The process also had the assistance of GIT as configuration management software. The process reached its goal at the end, resulting in a functional web system that allows the inclusion of users, registration of available publications and wishlist, search for registered publications, requests for other users publications, and visually organized by sent and received requests.

KEYWORDS

PHP Web System. Object-Oriented. MVC.

1 INTRODUÇÃO

A evolução da tecnologia tem permitido uma aproximação cada vez maior e mais rápida entre pessoas através da Internet. A Web tem se estabelecido como uma ferramenta favorável, possibilitando uma interação quase imediata entre seus usuários, resultando em uma ausência de distância geográfica, enquanto obstáculo, que proporcionou grande facilidade na contribuição de novas ideias e concepções.

Com o aumento constante em número de pessoas com acesso à Internet, uma grande quantidade de setores tem investindo fortemente em disponibilizar seus serviços na Web, ganhando espaço rapidamente na rede, como por exemplo o setor cultural, o comercial, e o de entretenimento. Muitos destes setores, inclusive, possuem representantes criados exclusivamente para a plataforma Web, como é o caso da Amazon.com, loja que realiza vendas de seus produtos exclusivamente através da plataforma Web.

Outros tipos de serviço que vem ganhando força na Internet são aqueles voltados para o compartilhamento de bens e serviços entre seus usuários, onde cada vez mais pessoas optam por realizar um determinado serviço ou distribuir algum bem material sem o auxílio de dinheiro, recebendo em troca bens ou serviços de outras pessoas. Um exemplo deste tipo de sistema é o site www.temacucar.com.br que permite aos seus usuários realizarem empréstimos ou doações de bens materiais entre pessoas que vivem em localidades próximas umas às outras, dispensando por completo o uso de dinheiro como moeda de troca.

Inspirado neste tipo de sistema, onde bens e serviços são compartilhados sem o uso de capital, este trabalho propõe o desenvolvimento de um serviço que tem como objetivo possibilitar o compartilhamento de publicações impressas entre seus usuários. O sistema em questão, desenvolvido para Web, permite o anúncio de livros ou qualquer outro tipo de publicação por parte de seus usuários para que sejam transferidas a outros usuários do site. As trocas são realizadas através de créditos dentro do sistema, os quais são adquiridos através de cada envio de publicação realizado, o qual tem seu valor estipulado em um crédito por cada transação efetuada. Além disso o sistema também permite a criação de uma lista de

publicações desejadas pelo usuário e também um sistema de ranking que permite a avaliação entre seus usuários, garantindo assim uma maior segurança e melhorias em cada troca efetuada.

2 DESENVOLVIMENTO

2.1 REVISÃO BIBLIOGRÁFICA

2.1.1 UML

A UML, Linguagem de Modelagem Unificada, é a linguagem visual padrão usada no mundo todo para engenharia de software, baseada no conceito de orientação a objeto, disse Guedes (2011).

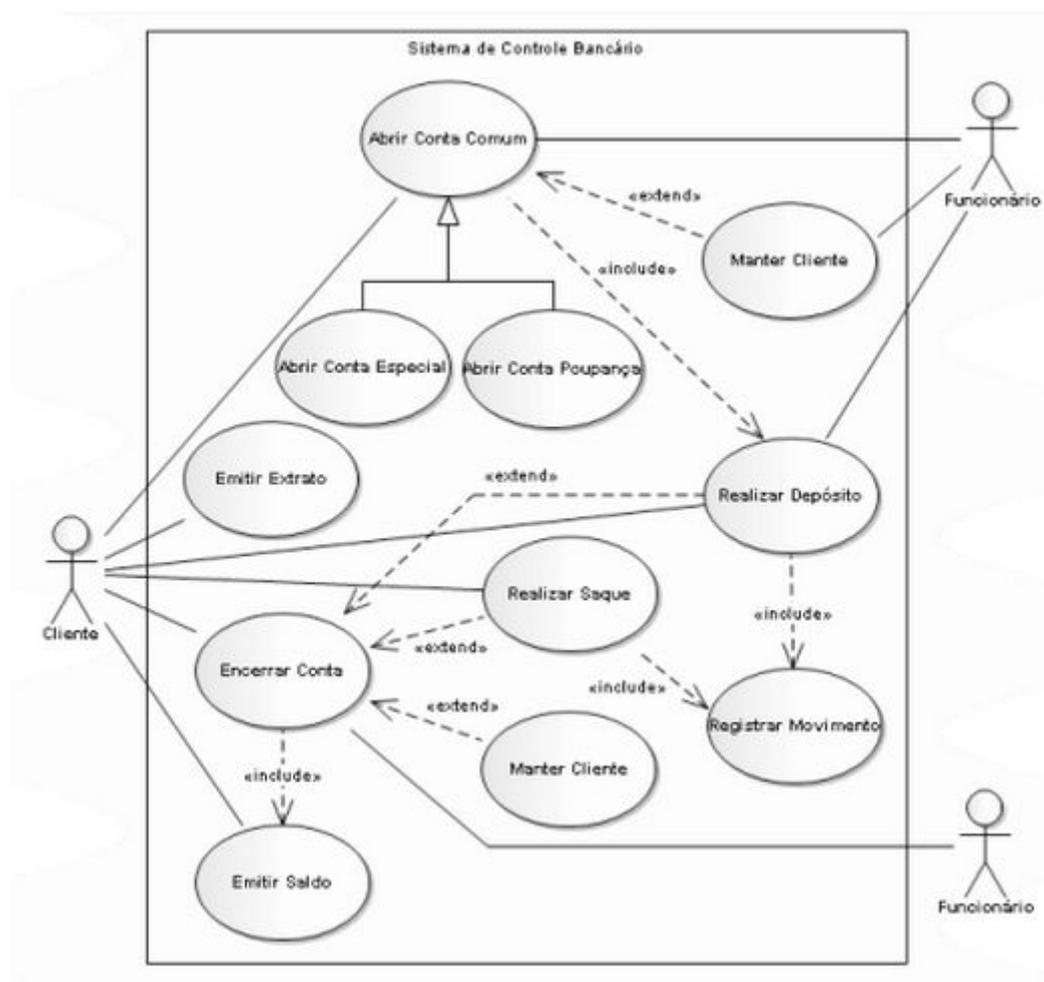
O objetivo da UML, segundo Pressman (2011) é descrever projetos de software para facilitar a especificação, construção e documentação dos artefatos utilizados no desenvolvimento do produto, auxiliando este processo fornecendo suporte à identificação dos requisitos e características do produto. A linguagem fornece vários diagramas que facilitam o levantamento de requisitos do produto, tal qual seu comportamento, lógica, estrutura física necessária, entre outros. Vale ressaltar que por ser uma linguagem de auxílio, dificilmente um produto exigirá que se aplique todas as ferramentas da UML, pois deve-se usar apenas aquelas que cabem em seu escopo de forma específica.

Segundo Guedes (2011), os sistemas de informação estão em constante mudança, podendo ser comparados a sistemas vivos que nunca são completamente finalizados, e sim atualizados, a partir de mudanças requisitadas por clientes, adoção de novas estratégias de mercado ou até mesmo alterações nas leis de um determinado local que acarreta em alterações do escopo do sistema. Um sistema deve ser projetado com a prévia condição de ser escalonável e uma documentação

bem elaborada e atualizada facilita o trabalho dos desenvolvedores em construir e manter o produto.

A primeira fase do desenvolvimento de um software é o levantamento de requisitos onde deve-se compreender, de acordo com Guedes (2011), todas as necessidades do usuário e o que este espera que seja realizado pelo produto, divididas em requisitos funcionais e não-funcionais, sendo que requisitos funcionais são aqueles que ditam as ações que o software ou componente deve realizar, como por exemplo o cadastro de usuários ou então a transferência de dados entre contas quando requisitado pelo usuário, enquanto os requisitos não-funcionais se referem à como o software irá realizar suas funções, podendo se referir ao desempenho do mesmo, interface externa do sistema, restrições do produto ou a qualidade do software.

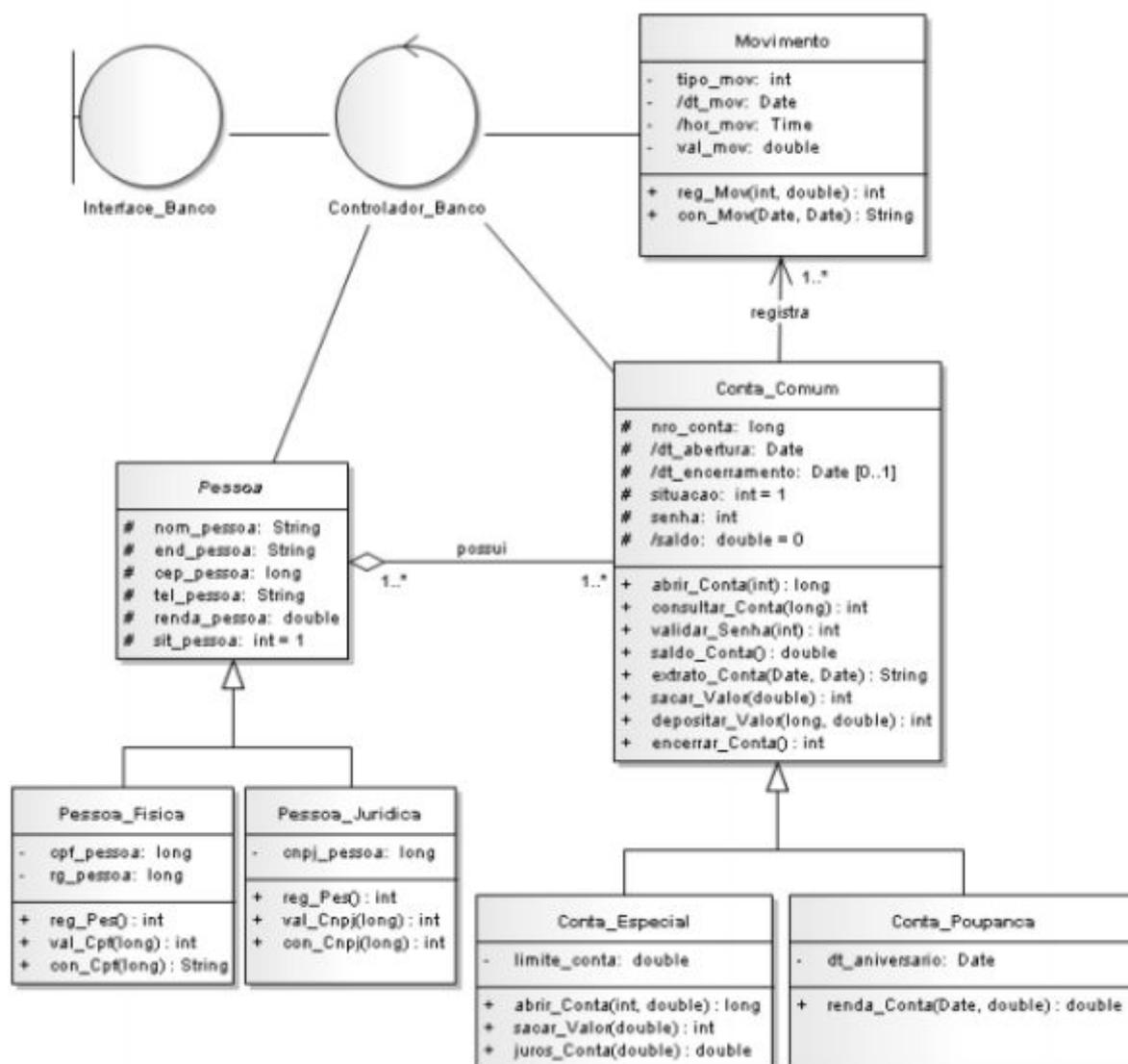
Figura 1 - Exemplo de Caso de Uso



Fonte: Guedes, 2001

O diagrama de caso de uso, outra ferramenta da UML, é uma ferramenta de linguagem abstrata, como diz Pressman (2011), que possibilita visualizar o comportamento do sistema e como usuários específicos podem interagir com ele. É representado por atores, que definem cada tipo ou nível de usuário do sistema, e casos de uso, que definem as ações que o sistema deve executar a partir de uma interação do usuário ou de outro caso de uso, como mostra a Figura 1.

Figura 2 - Exemplo Diagrama de Classes



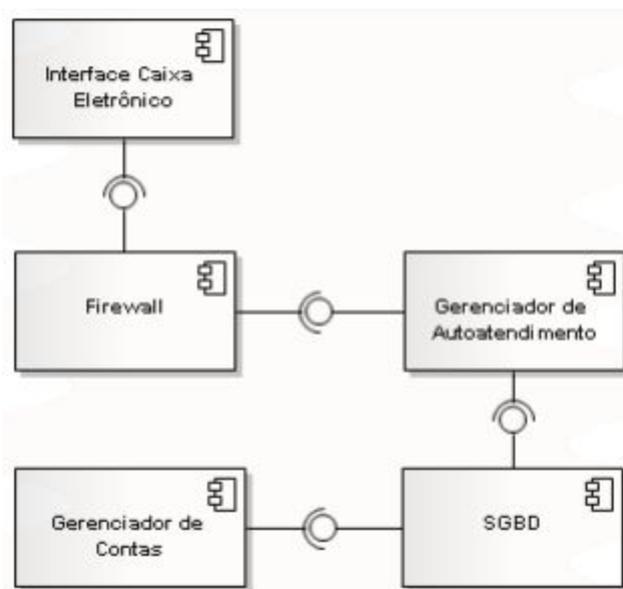
Fonte: Guedes, 2001

Outra ferramenta da UML considerada das mais importantes por muitos autores, o diagrama de classes, define, de acordo com Guedes (2011), as classes

de um sistema, incluindo seus atributos e métodos, junto com os relacionamentos que essas tem entre si o diagrama de classes tem por objetivo demonstrar as associações entre as classes de um produto e definir como esses relacionamentos interagem para a realização das tarefas do sistema, como mostra a Figura 2.

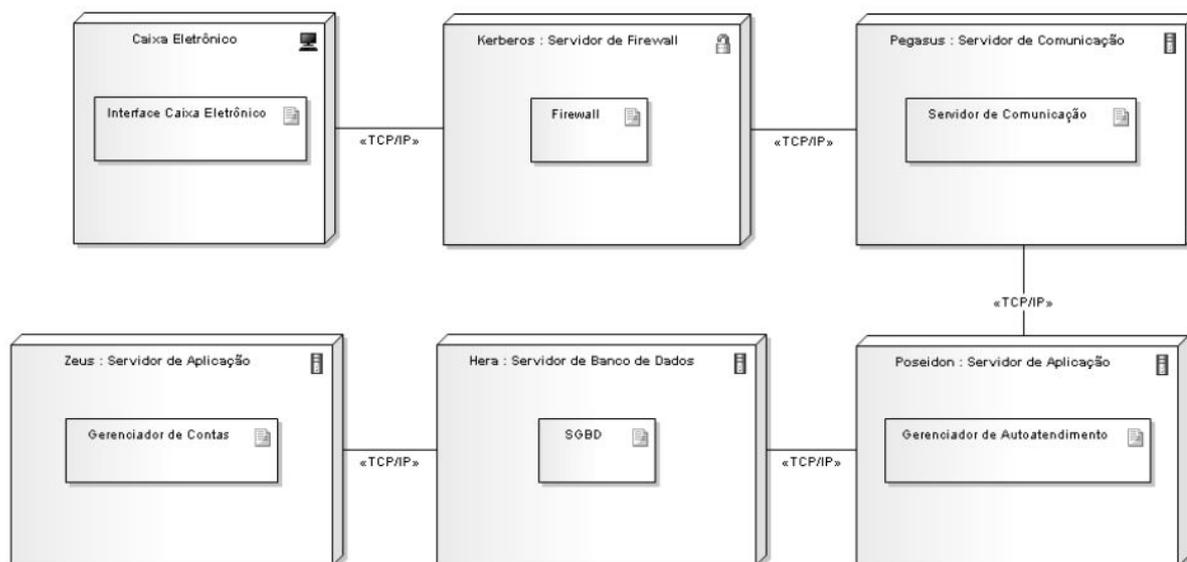
Existem também o diagrama de componentes, exibido na Figura 3, e o diagrama de implantação, apresentado na Figura 4. O primeiro “representa os componentes do sistema quando o mesmo for ser implementado em termos de módulos de código-fonte, bibliotecas, formulários, arquivos de ajuda, módulos executáveis etc.” (GUEDES, 2001, p. 38) enquanto o segundo “determina as necessidades de hardware do sistema, as características físicas como servidores, estações, topologias e protocolos de comunicação, ou seja, todo o aparato físico sobre o qual o sistema deverá ser executado”. (GUEDES, 2001, p. 39).

Figura 3 - Exemplo Diagrama de Componentes



Fonte: Guedes, 2001

Figura 4 - Exemplo Diagrama de Implantação



Fonte: Guedes, 2001

2.1.2 PHP

O PHP (Hypertext Preprocessor) é uma linguagem de script de código aberto normalmente executada no servidor. Segundo o Manual do PHP (Cowburn, 2015), ao invés de necessitar de comandos de linguagem para execução de código HTML (Hypertext Markup Language), o PHP pode conter código HTML embutido, separando comandos e suas instruções próprias da linguagem dentro de marcações “<?php ?>” permitindo assim uma enorme facilidade em alternar o código entre HTML e PHP. Por esse motivo o PHP se torna uma linguagem simples para desenvolvedores iniciantes como também uma ferramenta poderosa para desenvolvedores mais avançados.

O Manual do PHP (Cowburn, 2015) também diz que a linguagem, por ser executada no servidor, pode manipular o HTML antes de enviá-lo para o cliente, que receberá apenas o HTML já processado, sem acesso ao código fonte do processamento da página. Porém o PHP não está limitado apenas a gerar páginas Web, fornecendo uma imensa variedade de possibilidades como por exemplo

geração de imagens, arquivos pdf, coletar dados de formulários, gerar conteúdo dinâmico, dentre muitos outros, podendo ser trabalhado de modo estrutural, orientado a objeto ou então ambos.

De acordo com sua documentação oficial o PHP pode ser instalado na maioria dos sistemas operacionais, e é suportado por diversos servidores como por exemplo o Apache, IIS, dentre outros. Segundo W3Techs (2015), um montante de 81,5% dos 10 milhões de websites mais acessados na Web utilizam PHP como linguagem de servidor.

Com o intuito de agilizar o desenvolvimento em PHP foram criadas bibliotecas que fornecem funções prontas que são mais comumente utilizadas no desenvolvimento, com uma abordagem simples. O CodeIgniter é um exemplo de biblioteca PHP, que segundo sua documentação oficial (2015), é uma biblioteca gratuita, leve e rápida que permite o carregamento de suas funções de modo dinâmico, importando-as apenas se forem necessárias, utiliza-se do modelo MVC, que organiza o ambiente de desenvolvimento, separando a lógica da apresentação do produto, possui métodos prontos para conexão com banco de dados, envio de e-mail, validação de formulários, manipulação de imagens de forma simples, dentre muitas outras funções.

2.1.3 JavaScript

De acordo com a MDN (2015), o JavaScript é uma linguagem de programação dinâmica desenvolvida inicialmente para ser usada em páginas da Web. Por ser uma linguagem baseada em script, ou seja, interpretada por navegadores, o JavaScript permite que suas instruções sejam executadas sem a necessidade do seu processamento ser realizado no servidor, promovendo a manipulação de dados e componentes de páginas Web a partir do lado do cliente. A linguagem possui conceitos de orientação de objetos, assim como herança e encadeamento de protótipos, possuindo suporte por todos os navegadores

modernos desde 2012 e recentemente tem ganhado notoriedade em ambientes de servidor, como por exemplo o Node.js.

Atualmente existe um vasto número de bibliotecas JavaScript que tem por finalidade simplificar e incrementar as funções nativas do JavaScript. Segundo a W3Techs (2015), a biblioteca mais usada entre os 10 milhões de sites mais acessados na Web é a jQuery, utilizada em 66.4% dos sites monitorados. Seu criador, John Resig diz em sua página pessoal (2015) que o desenvolvimento do jQuery se deu pela necessidade em tornar o JavaScript mais poderoso, extendendo seus seletores para funcionar de maneira parecida com o padrão utilizado em códigos CSS (Cascading Style Sheets), por meio de identificadores, classes ou meta tags, tornando-os inclusive hierárquicos.

Segundo o site da ferramenta (2015), a biblioteca jQuery foi desenvolvida em código aberto, permitindo à sua comunidade desenvolvê-la de forma colaborativa, e hoje oferece ampla possibilidade no desenvolvimento de código JavaScript possuindo não só uma vasta biblioteca de métodos em seu projeto nativo como também muitos plugins de terceiros que oferecem uma grande variedade de opções no desenvolvimento de páginas Web.

Dois exemplos de plugins jQuery que estendem ainda mais sua biblioteca são o Cycle (<http://malsup.com/jquery/cycle>) e o jQuery Modal (<https://github.com/kylefox/jquery-modal>). O primeiro facilita a desenvolvimento de animações para apresentação de imagens e conteúdo enquanto o segundo incrementa a funcionalidade de criação de lightboxes pelo JavaScript.

2.1.4 MySQL

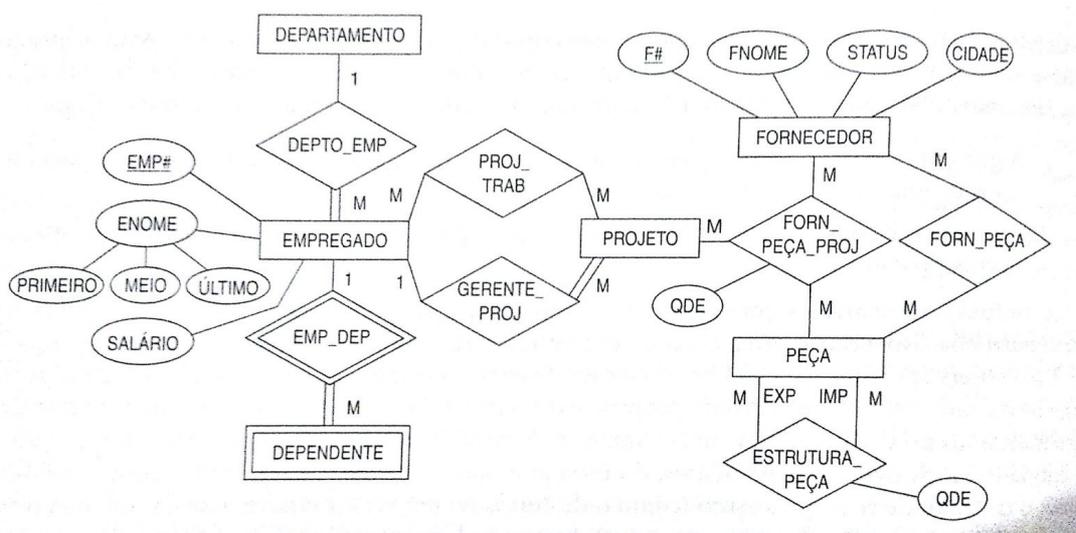
Com a ascensão da tecnologia e a migração de vários negócios para a Internet, um importante fator a ser levado em conta é a capacidade de armazenamento de dados para cada tipo de negócio. Atualmente, para se manter uma base de dados utiliza-se os chamados Sistemas de Gerenciamento de Banco de Dados. Esses sistemas, de acordo com Date (2003), são softwares que tem por

objetivo armazenar informações e fornecer maneiras de alterá-las, buscá-las, recuperá-las ou removê-las quando for solicitado.

Segundo Heuser (2009) pode-se utilizar linguagem gráfica ou textual para se construir um modelo de dados, denominado esquema de banco de dados. O esquema não deve definir os dados que serão apresentados, mas sim sua estrutura, contendo os campos necessários para a inserção dos dados do dado que irá ser armazenado.

A modelagem é realizada geralmente em duas etapas, com o desenvolvimento do modelo conceitual e em seguida o modelo lógico, de acordo com Heuser (2009), que define o modelo conceitual como a forma abstrata de se retratar a estrutura da base de dados, independente do sistema que irá ser utilizado para gerenciá-lo, utilizando-se do Diagrama Entidade-Relacionamento (DER).

Figura 3 - Diagrama Entidade-Relacionamento



Fonte: Date, 2003

A Figura 3 representa um DER que informa uma base de dados que contém Empregados que possuem atributos de nome e salário, além de um código atribuído a ele, sendo cada empregado cadastrado associado a um departamento e podendo estar associado a vários dependentes. Estes empregados são associados a Projetos, que por sua vez possuem relacionamentos com Fornecedores e Peças.

O modelo lógico, segundo Heuser (2009) é a representação do banco de dados voltado para o sistema gerenciador que será utilizado para a modelagem. No caso de ser usado o modelo relacional, os dados serão apresentados em forma de tabela, possuindo o nome de cada campo, acrescido do tipo de dado que irá ser armazenado no mesmo, como mostra a Figura 4.

Figura 4 - Exemplo de tabelas de banco de dados relacional

CodTipoProd	DescrTipoProd
1	Computador
2	Impressora

CodProd	DescrProd	PrecoProd	CodTipoProd
1	PC desktop modelo X	2.500,00	1
2	PC notebook ABC	3.500,00	1
3	Impressora jato de tinta XX	500,00	2
4	Impressora laser XX	1.500,00	2

Fonte: Heuser, 2009

O MySQL é um sistema gerenciador de banco de dados relacional que de acordo com Milani (2007) possui mais vantagens que a maioria de seus concorrentes, pois além de possuir todas as ferramentas e características de um sistema gerenciador de grande porte, consegue ser extremamente rápido, independente do tamanho da aplicação, e tem sido, de acordo com o autor, o banco de dados mais utilizado em aplicações Web

2.1.5 Hospedagem Web

Segundo a página do serviço de hospedagem UOL Host (2015), para que um website possa ser visualizado através da Internet, este precisa estar hospedado em um servidor que comporte todos os recursos que tenham sido implementados e

também possua espaço de armazenamento suficiente para hospedar todos os arquivos do site. Existe uma grande quantidade de opções, que vão das gratuitas às comercializadas, oferecendo os mais diversos tipos de suporte de tecnologias como por exemplo Tomcat, Apache, entre outros.

Além do servidor, que irá armazenar todas as dependências do site, também é necessário a contratação de um domínio, que irá facilitar a busca e o acesso ao site, uma vez que sem ele o site não possui um endereço amigável na rede e é apenas referenciado pelo seu número de protocolo (IP).

De acordo com estatísticas da W3Techs (2015), o Apache é o software servidor mais utilizado entre os dez milhões de websites mais acessados no mundo. Por ser um software de código livre, tem seus código-fonte constantemente melhorado, o que tem contribuído para se tornar um software com excelente performance, confiável, com muitos recursos nativos ou instalados a partir de módulos e altamente compatível com diversas plataformas.

2.1.6 Software de Gestão de Configuração

Durante o desenvolvimento do software, como dito em capítulos anteriores, ocorrem várias mudanças no escopo do produto e conseqüentemente precisamos monitorar e gerenciar tais mudanças de forma eficaz. Pressman (2011) nos diz que com o decorrer das alterações no produto, caso esses processos não sejam monitorados, o software pode sofrer com baixas em qualidade ou em prazo de entrega, o que dificulta o processo de desenvolvimento como um todo.

A Gestão de Configuração de Software (Software Configuration Management - SCM), segundo Pressman (2011), é um aglomerado de processos voltados à identificação de alterações ocorridas a qualquer momento no produto em desenvolvimento, bem como o controle dessas alterações, garantindo que as modificações estejam ocorrendo de maneira correta e relatando os envolvidos sobre todas as mudanças que venham a ser aplicadas durante todo o período de atividade do sistema.

Entre as características e funções mais notáveis de SCMs citadas por Pressman (2011) pode-se lista gerenciamento de versões, um recurso que possibilita o registro de todas as mudanças ocorridas em componentes individuais do sistema, permitindo o retrocesso para versões antigas durante testes e depurações do sistema, e também acompanhamento de dependências, usado para relacionar componentes ou artefatos do sistema entre si, garantindo a integridade do sistema ao reportar ao desenvolvedor mudanças com níveis de relação associativas ou dependentes.

2.1.7 GIT

Segundo pesquisas feitas pela Eclipse Community Survey (2014), a ferramenta Git tem se constituído como a mais promissora atualmente para a gestão de configuração de software, usada por 33,3% dos desenvolvedores de sistemas, contra 30,7% de uso do SVN, a segunda ferramenta mais usada.

A vantagem de se usar o Git, de acordo com Chacon e Straub (2014), é a agilidade de todo o processo, ao usar snapshots (fotos do código) para realizar comparações entre versões do sistema, ao invés de armazenar as próprias alterações como fazem outros softwares de versionamento. Outro diferencial apontado por Chacon e Straub (2014) é o suporte para o desenvolvimento não linear, permitindo repositórios paralelos ao repositório principal do sistema, facilitando sua manutenção e implementação de novas funcionalidades e as testando antes mesmos de alterar o repositório principal do projeto.

3 METODOLOGIA E ANÁLISE

Para que o Sistema Web pudesse ser desenvolvido, o primeiro passo tomado foi a elaboração do diagrama de caso de uso, apresentado no apêndice “A”,

o qual auxiliou a visualização do sistema pelo ponto de vista do usuário final, nos fornecendo uma descrição das principais funcionalidades do sistema e as interações que essas teriam com os atores do sistema.

A próxima etapa no desenvolvimento do sistema foi o levantamento de requisitos para que todas as características e funcionalidades do sistema fossem identificadas. Foram levantados requisitos funcionais do sistema, como telas de cadastro de usuários, cadastros de livros desejados, cadastro de livros disponíveis para troca, tela de gerenciamento de solicitações de troca de livros efetuadas ou recebidas. Também foram identificados requisitos de qualidade do sistema. Alguns desses requisitos estão demonstrados no documento de requisitos no apêndice “B”. Este documento foi baseado no diagrama anterior, de caso de uso, usando-o como guia para a identificação de requisitos.

Após o término da elaboração do documento de requisitos teve-se início a elaboração do projeto de protótipos de telas que o sistema iria conter. A ferramenta escolhida para a criação dessas telas (protótipos) foi o Adobe Fireworks, por conter ferramentas que facilitam o desenho de páginas Web. Alguns protótipos podem ser vistos no apêndice “C”

O próximo passo efetuado foi a criação do documento de casos de uso textual, onde foi-se estabelecido fluxos principais e alternativos, incluindo a identificação de atores e precondições, para cada funcionalidade do sistema, descrevendo cenários como por exemplo a adição de livros disponíveis ou o envio de solicitação de troca, o qual pode ser visualizado no documento de casos de uso no apêndice “D”.

Durante este processo foram identificados alguns pontos importantes do sistema, como a linguagem a ser utilizada em seu desenvolvimento e o sistema gerencial de banco de dados a ser implementado. O PHP foi escolhido como linguagem de programação, sendo utilizado através do Framework CodeIgniter, por fornecer uma maior facilidade no uso da abordagem de Orientação a Objeto, com métodos de uso simplificado, e também por trabalhar com o modelo MVC (Model View Controller), facilitando a organização estrutural do projeto.

Tendo o diagrama de caso de uso sido elaborado, teve início a modelagem de diagramas da UML, todos a seguir utilizando-se a ferramenta Astah Professional,

pelo motivo de possuir modelos para todos os diagramas de modelagem seguindo o padrão da UML 2.0 e outros vastamente utilizados na área da Tecnologia da Informação. O primeiro diagrama a ser modelado foi o Diagrama de Entidade Relacionamento Conceitual, apresentado no apêndice “E”, que permitiu uma identificação de alto nível da estrutura que nosso banco de dados deveria ter, contendo as propriedades que cada entidade teria e a forma em que essas entidades se relacionariam, facilitando assim a modelagem do Diagrama de Entidade Relacionamento Lógico, que pode ser visto no apêndice “F”, que identifica a modelagem completa do nosso banco de dados, apresentando as tabelas do banco, seus campos e seus respectivos tipos, chaves primárias, estrangeiras e também sua cardinalidade.

Em seguida ocorreu a modelagem do Diagrama de Classes, o Diagrama de Componentes e o Diagrama de Implantação, podendo ser visualizados respectivamente nos apêndices “G”, “H” e “I”.

Levando em conta os resultados obtidos até esta etapa, deu-se início à fase do desenvolvimento. Para tal, foi primeiramente configurado um reservatório Git para ser feita a monitoração e versionamento das alterações do sistema. Após o cadastro no website da ferramenta, um repositório vazio foi criado e depois clonado localmente para então ter início o desenvolvimento do sistema utilizando-se o editor Atom, por ser distribuído de forma gratuita e possuir uma imensa variedade de plugins que agilizam a programação. Dentro deste repositório vazio foi instalado localmente o CodeIgniter, o qual gerou uma estrutura de pastas nos moldes do modelo MVC, das quais destacam-se as pastas *config*, *controllers*, *models* e *views*, cada uma com sua função. A pasta *config* armazena arquivos que fazem as configurações do projeto como por exemplo as configurações de conexão com banco de dados, a pasta *controllers* possui as classes principais do sistema como as que realizam as manipulações dos dados, a *models* possui classes que interagem com o banco de dados e a *views* possui arquivos de apresentação. Foi-se criada também uma pasta *assets* para o armazenamento de imagens, arquivos css e javascript.

Durante o processo de desenvolvimento do produto foi possível adquirir novos conhecimentos como por exemplo a manipulação de imagens com o CodeIgniter como mostra a Figura 5.

Figura 5 - Código utilizado para salvar imagem enviada pelo usuário

```
41 function inserir() {
42     $data['id_usuario'] = $this->input->post('idu');
43     $data['tipo'] = 1;
44     $data['titulo'] = $this->input->post('titulo');
45     $data['autor'] = $this->input->post('autor');
46     $data['editora'] = $this->input->post('editora');
47     $data['ano_pub'] = $this->input->post('ano_pub');
48     $data['descricao'] = $this->input->post('descricao');
49
50     //Gera nome unico + extensao para o arquivo
51     $ext = substr(strrchr($_FILES['foto']['name'],'.'),1);
52     $rand = md5(uniqid(rand(), true)) .'.' . $ext;
53
54     //Salva arquivo na pasta
55     $sourcePath = $_FILES['foto']['tmp_name'];
56     $targetPath = "assets/imgs/book_covers/" . $rand;
57     move_uploaded_file($sourcePath,$targetPath);
58
59     //Salva nome do arquivo no banco
60     $data['foto'] = $rand;
61
62     $this->load->model('estante_model');
63
64     if ($this->estante_model->inserir($data)) {
65         redirect('estante');
66     } else {
67         log_message('error', 'Erro ao inserir o livro.');
```

Fonte: Autor

O método acima recebe todos os dados vindos do formulário de cadastro de livros disponíveis da view inclusive uma imagem. A extensão desse arquivo será armazenado em uma variável, em seguida uma nova variável recebe uma string

gerada randomicamente e em seguida é concatenada com a extensão previamente armazenada, para então o arquivo ser movido para a pasta reservada. O nome gerado randomicamente impede de se ter duas imagens com o mesmo nome na pasta onde as imagens serão armazenadas, evitando conflitos durante o salvamento. O nome do arquivo é então armazenado no banco de dados e o retorno de sucesso ou erro da operação é retornada para a view.

Também teve-se a oportunidade de conhecer alguns plugins fornecidos para a biblioteca jQuery que auxiliam na forma de exibição de animações para o usuário. O Cycle, por exemplo permite a exibição de listas animadas, que são exibidas em forma carrossel, bastando apenas a importação do plugin na view onde será utilizada e uma pequena instrução com as configurações desejadas para o funcionamento, como mostra a Figura 6.

Figura 6 - Código de configuração do plugin Cycle para jQuery

```
1  $(function(){
2
3      //SLIDESHOW (PAGINA INICIAL)
4      if($('.slideshow').length) {
5          $('.slideshow').cycle({
6              fx: 'carousel',
7              speed: 1000,
8              timeout: 1000,
9              width: 1000
10         });
11     }
12
13 });
```

Fonte: Autor

O código acima mostra uma lista de itens recebendo configurações de animação, assim como a velocidade da animação e tamanho da vitrine de exibição dos itens.

Após a conclusão do desenvolvimento foi necessário hospedar o website em algum servidor e contratar um domínio para torná-lo disponível na Internet. Alguns serviços atualmente oferecem simultaneamente os dois recursos (hospedagem e domínio) como é o caso do Hostinger (<http://www.hostinger.com.br>) que fornece

planos contratados e também gratuitos, e se destaca por possuir serviços completos para o armazenamento de websites baseados em PHP, com banco de dados MySQL. O grande diferencial do serviço, de acordo com a página da empresa (24/10/2015), é sua alta capacidade de armazenamento de arquivos, com 2Gb de espaço em seu plano gratuito, e a não inclusão automática de propagandas nos websites hospedados, prática muito comum em serviços gratuitos do mesmo tipo.

O produto foi hospedado e teve seu registro efetuado no endereço www.bibliotroca.xyz, domínio este fornecido gratuitamente pelo próprio servidor Hostinger, estando acessível na data do presente trabalho.

4 CONCLUSÃO

Através do desenvolvimento do Sistema Web apresentado neste trabalho foi possível utilizar e melhorar os conhecimentos adquiridos no decorrer do curso, além da possibilidade de se adquirir novos conhecimentos como por exemplo o uso do framework PHP CodeIgniter para facilitar o desenvolvimento orientado a objeto e como utilizar o modelo MVC em um projeto. Também destaca-se a abordagem de requisições AJAX em JavaScript, tornando o website mais dinâmico.

Sobre a UML podemos citar o entendimento da importância do uso de seus diagramas durante o desenvolvimento de um software, auxiliando no processo como um todo desde o levantamento de requisitos até a implementação do sistema.

Pôde se notar também a utilidade e importância do uso de um software de gestão de configuração como o GIT, que auxiliou todo o processo de desenvolvimento, tornando-o mais simples em situações que, sem seu uso, poderiam ter comprometido o planejamento do projeto.

O presente trabalho resultou em um Sistema Web funcional em PHP que atende a todos os requisitos levantados no início do projeto e está representado no apêndice "I".

Futuramente pretende-se incluir algumas funcionalidades ao sistema como por exemplo a implementação de uma API que contenha informações de

publicações, como é o caso da Google Books API, para tornar o processo de cadastro de publicações no site mais simples. Também existe a pretensão de se relacionar as publicações desejadas com as publicações disponíveis por outros usuários, criando um sistema inteligente de sugestões de trocas para facilitar o uso do sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

CHACON, Scott; STRAUB, Ben. Git Pro. 2 ed. California: Apress, 2014. 574p.

CODEIGNITER, Codelgniter User Guide. Disponível em: <http://www.codeigniter.com/user_guide>. Acesso em: 29 out. 2015.

DATE, C. J. Introdução a Sistemas de Banco de Dados. 8. ed. Rio de Janeiro: Elsevier Editora, 2003. 865p.

ECLIPSE COMMUNITY SURVEY, Eclipse Community Survey 2014 Results. Disponível em: <<http://eclipse.dzone.com/articles/eclipse-community-survey-2014>>. Acesso em 24 out. 2010.

GUEDES, Gilleanes T. A. UML 2.0 - Uma Abordagem Prática. 2. ed. São Paulo: Novatec Editora, 2011. 488p.

HEUSER, Carlos Alberto. Projeto de Banco de Dados. 6. ed. Porto Alegre: Bookman Editora, 2009. 282p.

JQUERY FOUNDATION, Our Projects. Disponível em: <<https://jquery.org/projects/>>. Acesso em 10/11/2015.

MILANI, André. MySQL - Guia do Programador. 1. ed. Curitiba: Novatec Editora, 2007. 400p.

MOZILLA DEVELOPER NETWORK, Javascript. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em 22 out. 2015.

PETER, Cowburn. Manual do PHP. Disponível em: <http://php.net/manual/pt_BR/>. Acesso em: 29 out. 2015.

PRESSMAN, Roger S. Engenharia de Software: Uma Abordagem Profissional. 7. ed. Porto Alegre: AMGH Editora, 2011. 780p.

RESIG, John. Selectors in Javascript. Disponível em: <<http://ejohn.org/blog/selectors-in-javascript>>. Acesso em 22 out. 2015).

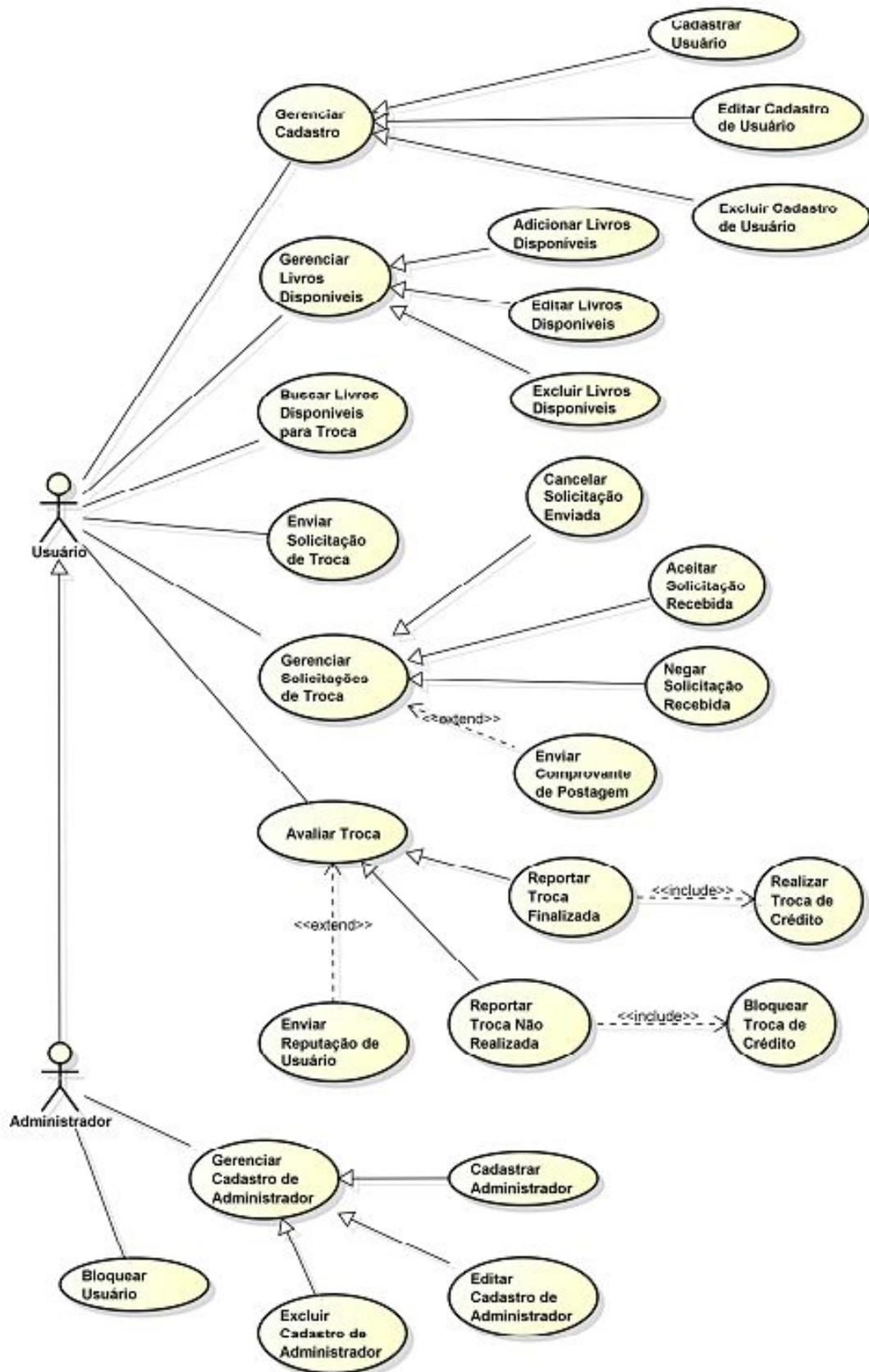
UOL HOST, FAQ Hospedagem. Disponível em <<http://www.uolhost.uol.com.br/faq/hospedagem/site-hospedagem-dominio-etc-entenda-melhor-o-que-sao.html>>. Acesso em 28 out. 2015.

W3TECHS, Usage of server-side programming languages for websites. Disponível em: <http://w3techs.com/technologies/overview/programming_language/all>. Acesso em: 29 out. 2015.

W3TECHS, Usage of JavaScript libraries for websites. Disponível em: <http://w3techs.com/technologies/overview/javascript_library/all>. Acesso em 22 out. 2015.

W3TECHS, Usage of web servers for websites. Disponível em: <http://w3techs.com/technologies/overview/web_server/all>. Acesso em 28 out. 2015.

APÊNDICE A - DIAGRAMA DE CASO DE USO



APÊNDICE B - DOCUMENTO DE REQUISITOS

1 Visão Geral do Sistema

O sistema tem como principal objetivo oferecer o auxílio a usuários que desejem tanto oferecer como adquirir livros na base da troca entre utilizadores do serviço, oferecendo organização e agilidade.

Os utilizadores deste sistema serão usuários cadastrados que tenham interesse em realizar trocas de livros e também administradores do sistema. Os usuários comuns utilizarão o mesmo para gerenciar livros que queiram disponibilizar para troca ou então livros que tenham interesse. Já administradores do sistema o usarão para o manuseio de dados na realização de cadastros diversos e gerenciamento das atividades do sistema.

2 Requisitos Funcionais

2.1 Requisitos Funcionais de Entrada

1. O sistema deve permitir a inserção, alteração e exclusão de administradores do sistema.
2. O sistema deve realizar o cadastro de novos administradores, solicitando ao mesmo as informações para sua realização. Os itens do cadastro são nome, e-mail e senha.
3. O sistema deve gerar automaticamente um código único a cada novo administrador cadastrado.
4. O sistema deve exibir uma mensagem caso o usuário tente cadastrar um administrador cujo e-mail já seja cadastrado no sistema, informando-o do ocorrido e impedindo o término do cadastro.
5. O sistema deve permitir a alteração de todo e qualquer dado de administradores, com exceção do seu código de identificação.
6. O sistema deve permitir a exclusão de qualquer administrador desde que o

cadastro do mesmo exista no sistema.

7. O sistema deve permitir a inserção, alteração e exclusão de usuários.

8. O sistema deve realizar o cadastro de novos usuários, solicitando ao mesmo as informações para sua realização. Os itens do cadastro são nome, e-mail, senha, data de nascimento, cep, endereço, número, bairro, cidade e estado.

9. O sistema deve gerar automaticamente um código único a cada novo usuário cadastrado.

10. O sistema deve guardar automaticamente a data e hora em que o cadastro foi realizado.

11. O sistema deve exibir uma mensagem caso o usuário tente cadastrar um e-mail já cadastrado no sistema, informando-o do ocorrido e impedindo o término do cadastro.

12. O sistema deve permitir a alteração de todo e qualquer dado de usuários, com exceção do seu código de identificação.

13. O sistema deve permitir a exclusão de qualquer usuário desde que o cadastro do mesmo exista no sistema.

14. O sistema deve permitir a inserção, alteração e exclusão de livros disponíveis para cada usuário.

15. O sistema deve realizar o cadastro de novos livros disponíveis para cada usuário, solicitando ao mesmo as informações para sua realização. Os itens do cadastro são título, autor, editora, ano de publicação e descrição.

16. O sistema deve gerar automaticamente um código único a cada novo livro disponível cadastrado.

17. O sistema deve permitir a inserção de foto no cadastro de livros disponíveis desde que o formato do arquivo seja suportado pelo mesmo (png ou jpg).

18. O sistema deve permitir a alteração de todo e qualquer dado de livros disponíveis, com exceção do seu código de identificação.

19. O sistema deve permitir a exclusão de qualquer livro disponível desde que o cadastro do mesmo exista no sistema e não esteja com status de solicitações de troca ativa.

20. O sistema deve permitir a inserção, alteração e exclusão de livros desejados para casa usuário..

21. O sistema deve realizar o cadastro de novos livros desejados para cada usuário, solicitando ao mesmo as informações para sua realização. Os itens do cadastro são título, autor, editora, ano de publicação e descrição.
22. O sistema deve gerar automaticamente um código único a cada novo livro desejado cadastrado.
23. O sistema deve permitir a alteração de todo e qualquer dado de livros desejados, com exceção do seu código de identificação.
24. O sistema deve permitir a exclusão de qualquer livro desejado desde que o cadastro do mesmo exista no sistema.

2.2 Requisitos Funcionais de Processamento

25. O sistema deve permitir a consulta de usuários existentes no banco de dados. A consulta poderá ser realizada a partir do nome.
26. O sistema deve permitir a consulta de livros disponíveis existentes no banco de dados. A consulta poderá ser realizada a partir do título, autor ou editora.
27. O sistema deve permitir o envio de solicitação de troca de livros entre um usuário e outro, desde que o usuário solicitante contenha créditos positivos.
28. O sistema deve permitir o cancelamento de solicitação de troca de livros desde que a solicitação ainda esteja pendente.
29. O sistema deve bloquear interações, por usuários que não estejam envolvidos, em livros que estejam com solicitações de troca já aceitas.
30. O sistema deve voltar a permitir interações, por usuários que não estejam envolvidos, em livros que tenham sua solicitação de troca negada.
31. O sistema deve permitir ao usuário solicitante reportar uma troca como realizada, desde que a troca tenha sido aceita previamente.
32. O sistema deve realizar a troca de créditos, retirando um crédito do usuário solicitante e incrementando um crédito ao usuário doador, caso a troca seja reportada como realizada.
33. O sistema deve permitir ao usuário solicitante reportar uma troca como não realizada, desde que a troca tenha sido aceita previamente.
34. O sistema deve bloquear a troca de créditos entre os usuários no caso do

usuário solicitante reportar a troca como não realizada.

35. O sistema deve permitir o envio de comprovante de postagem a partir do usuário ofertante caso uma troca seja reportada como não realizada.

36. O sistema deve permitir a classificação da reputação de um usuário ofertante a partir de um usuário solicitante após a troca ter sido finalizada com sucesso.

2.3 Requisitos Funcionais de Saída

37. O sistema deverá enviar um e-mail ao usuário caso seja realizada uma solicitação de troca de algum livro disponível do mesmo.

38. O sistema deverá enviar um e-mail ao usuário solicitante caso uma solicitação de troca seja respondida.

39. O sistema deverá enviar e-mails aos usuários quando uma troca for reportada como realizada, informando-os de seus créditos atualizados.

40. O sistema deverá enviar e-mails aos usuários quando uma troca for reportada como não realizada, informando-os do bloqueio da troca de créditos.

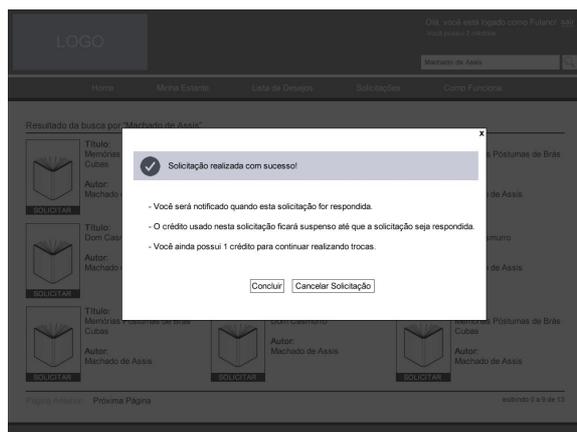
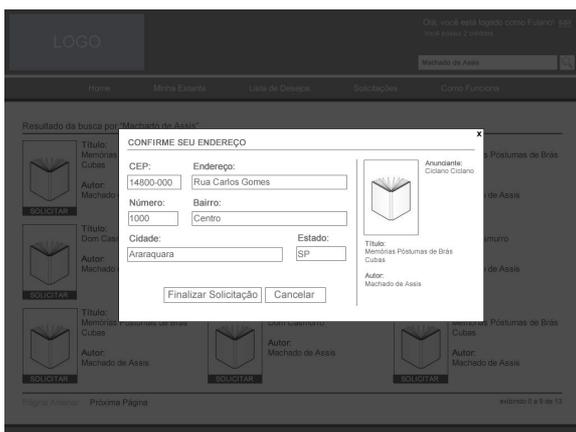
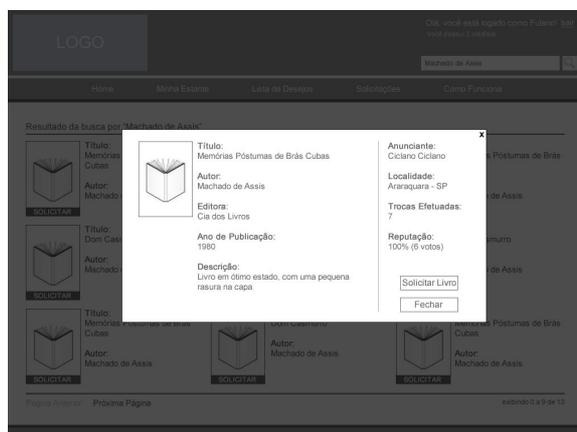
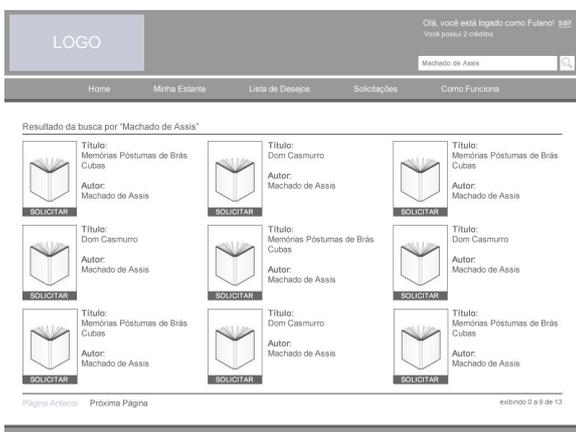
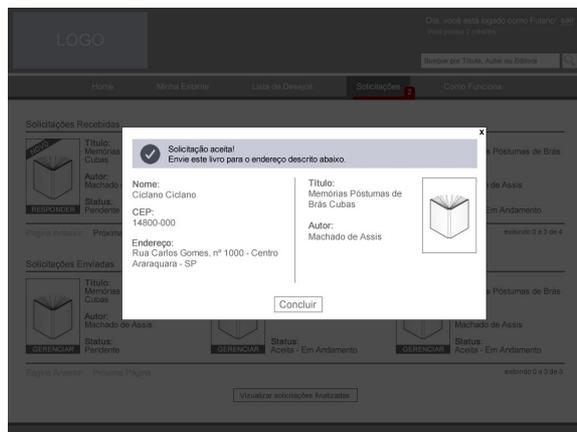
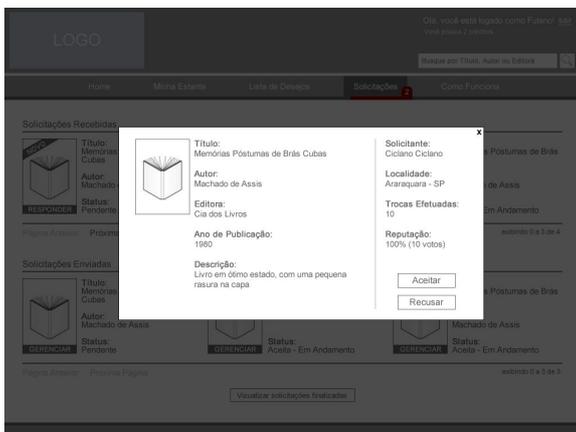
3 Requisitos de Qualidade

3.1 Confiabilidade

41. O sistema deve gerar um backup de todas as trocas realizadas com os respectivos dados dos usuários.

3.2 Portabilidade

42. O sistema deverá funcionar corretamente nos principais navegadores disponíveis no mercado (ex.: Google Chrome, Internet Explorer, Mozilla Firefox, Safari, etc.)



APÊNDICE D - CASO DE USO TEXTUAL

Caso de Uso: Cadastras Usuário

Descrição: Usuário realiza seu cadastro para ter acesso às funcionalidades do sistema.

Ator: Usuário.

Pré-condições: Usuário não deve estar logado no sistema.

Cenário Principal:

1. Usuário clica em “Cadastrar” na página inicial do site.
2. O sistema abre a página de cadastro de usuários.
3. Usuário preenche o formulário com seus respectivos dados (Nome completo, E-mail, Senha, Data de nascimento, Endereço, Numero, Cidade e Estado).
4. O sistema valida se todos os campos foram preenchidos.
5. O sistema valida data de nascimento e formatação de e-mail.
6. O sistema checa se o e-mail informado não está cadastrado no banco de dados.
7. Usuário clica em Enviar Cadastro.
8. O sistema trava campos de inserção de dados.
9. O sistema salva os dados informados no banco de dados.
10. O sistema carrega os dados salvos, efetuando o login do usuário.
11. O sistema exibe a mensagem “Seu cadastro foi realizado com sucesso”.
12. O sistema exibe links para alteração e exclusão do cadastro realizado.

Cenário Alternativo:

- 6.1. O sistema exibe uma mensagem de e-mail já cadastrado e solicita novo e-mail para prosseguir o cadastro.

Caso de Uso: Adicionar Livros Disponíveis

Descrição: Usuário adiciona livros que deseja deixar disponíveis para troca.

Ator: Usuário.

Pré-condições: Usuário deve estar logado no sistema.

Cenário Principal:

1. Usuário clica no link para adicionar novos livros disponíveis.
2. O sistema carrega formulário de inserção de novos livros disponíveis.
3. Usuário preenche o formulário com os respectivos dados do livro a ser adicionado (Título, Autor, Editora, Ano de Publicação e Descrição).
4. Usuário insere foto para ilustrar capa do livro.
5. O sistema valida se arquivo selecionado corresponde a um arquivo suportado (png ou jpg).
6. Usuário clica em “Adicionar Livro”.
7. O sistema verifica se todos os campos foram preenchidos.
8. O sistema salva os dados informados no banco de dados.
9. O sistema salva o arquivo de imagem enviado em uma pasta com nome gerado randômicamente.
10. O sistema fecha o formulário.
11. O sistema exibe a mensagem “Livro adicionado com sucesso”.
12. O sistema exibe link para adicionar novo livro.

Cenário Alternativo:

- 5.1. O sistema exibe uma mensagem de arquivo inválido ou não suportado e solicita novo arquivo.

Caso de uso: Enviar Solicitação de Troca

Descrição: Usuário realiza solicitação para adquirir um livro que outro usuário esteja oferecendo.

Ator: Usuário.

Pré-condições: Usuário deve estar logado no sistema, usuário deve possuir créditos positivos.

Cenário Principal:

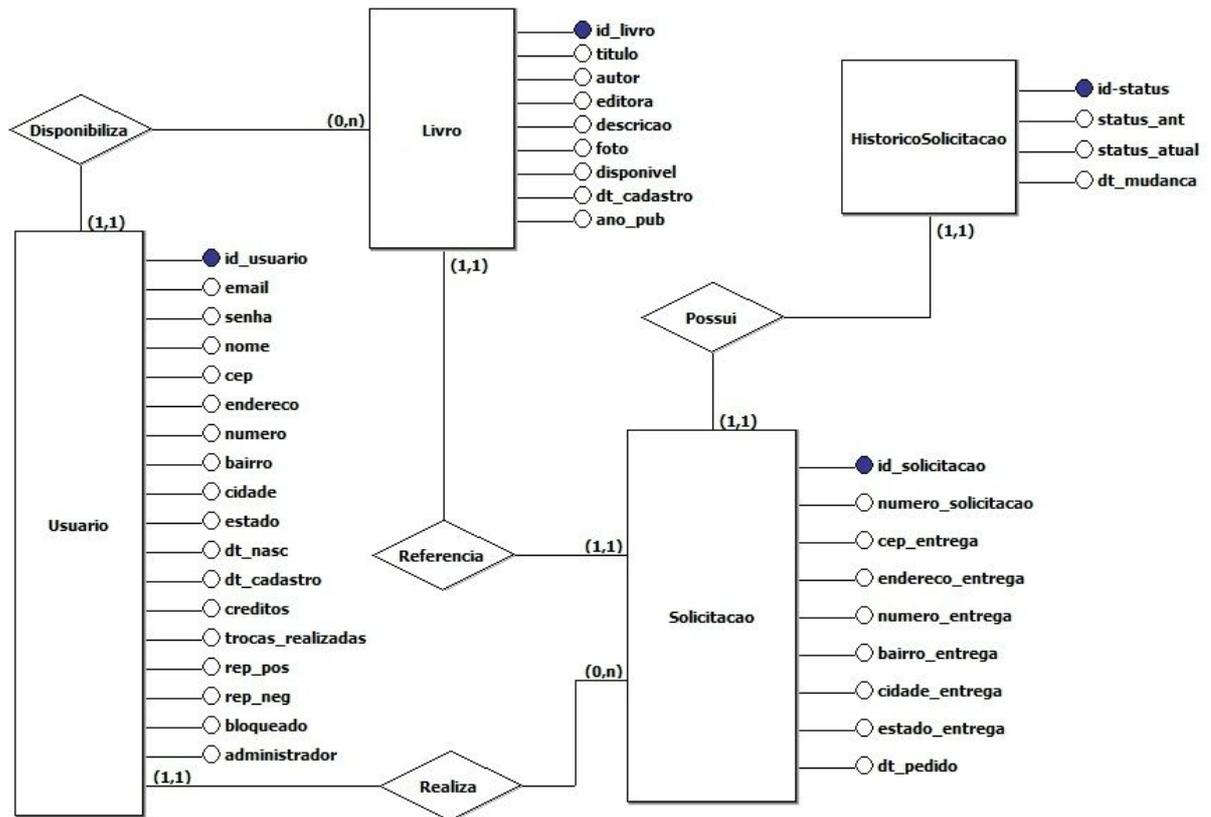
1. Usuário clica na foto do livro que deseja solicitar.

2. O sistema carrega uma lightbox com dados completos do livro em questão (Título, Autor, Editora, Ano de Publicação e Descrição) e do usuário ofertante (Nome Completo, Cidade, Estado, Número de trocas efetuadas e Reputação).
3. Usuário clica em “Solicitar Livro”.
4. O sistema carrega dados de endereço do usuário e os exibe dentro de um formulário para que o usuário confirme seus dados.
5. Usuário clica em “Finalizar Solicitação”.
6. O sistema bloqueia 1 crédito do usuário.
7. O sistema exibe a mensagem “Solicitação realizada com sucesso”.
8. O sistema envia um e-mail para o usuário ofertante avisando-lhe que há uma nova solicitação pendente.

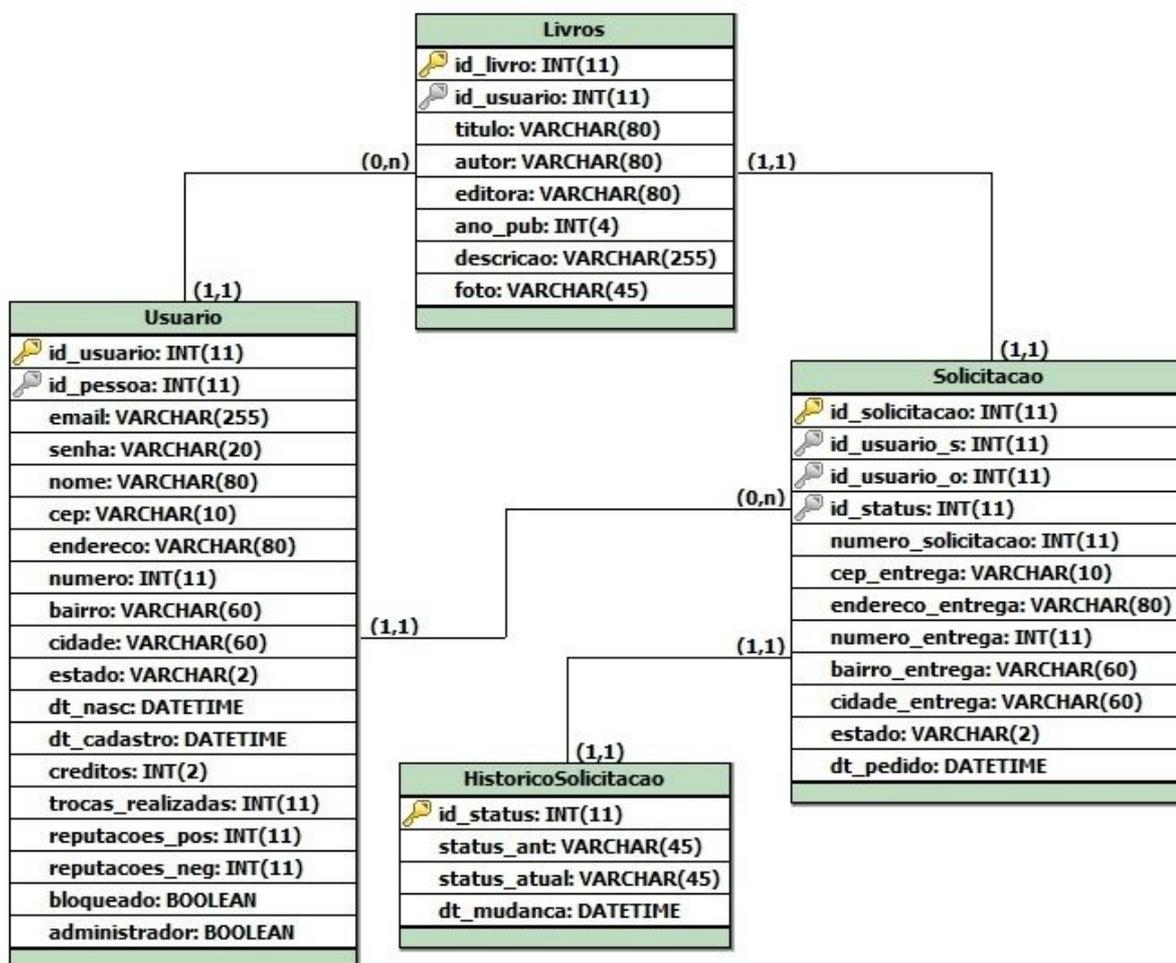
Cenário Alternativo:

- 4.1. Usuário altera dados e clica em “Finalizar Solicitação”.
- 4.2. O sistema exibe uma mensagem de dados inválidos e solicita o preenchimento correto dos dados.

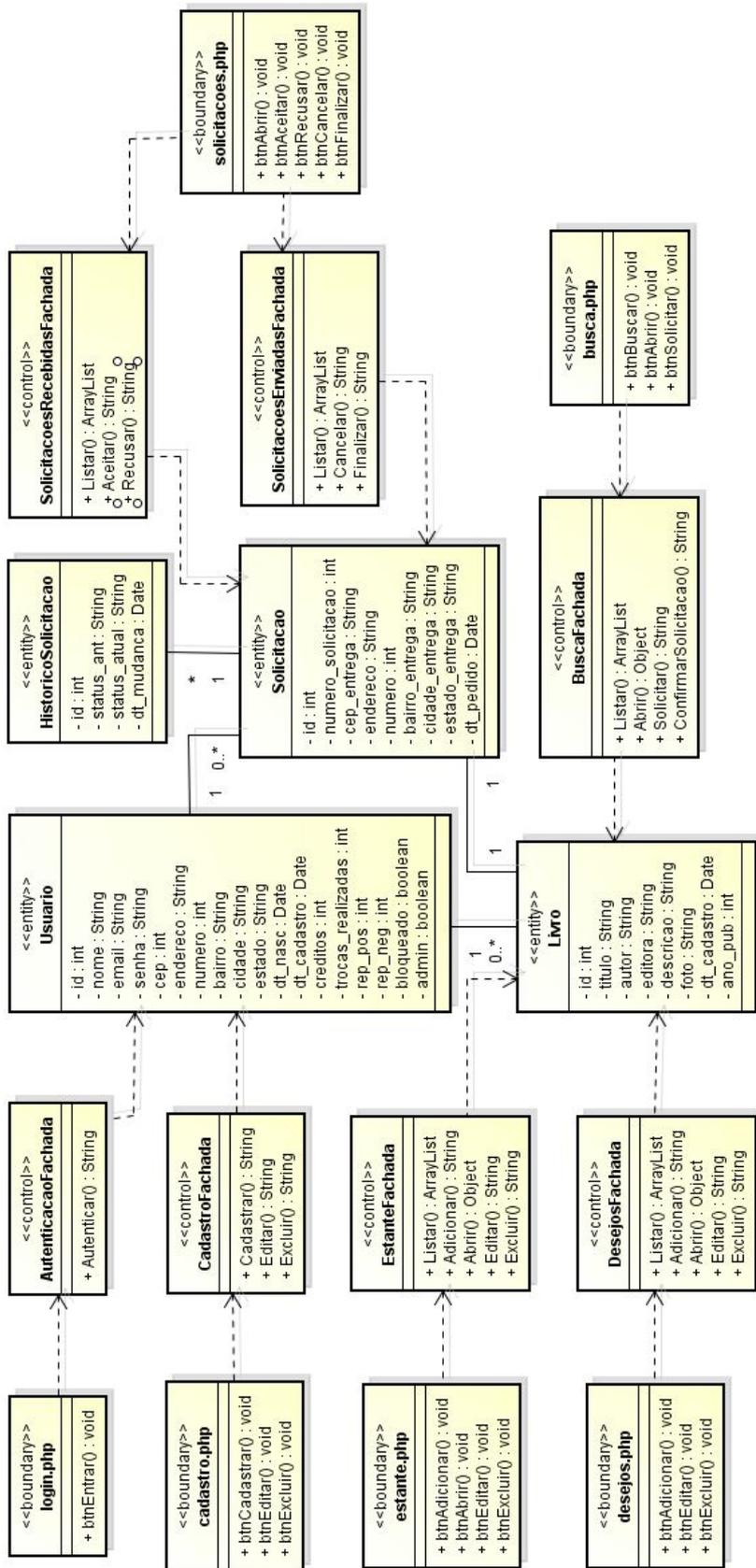
APÊNDICE D - DIAGRAMA ENTIDADE RELACIONAMENTO CONCEITUAL



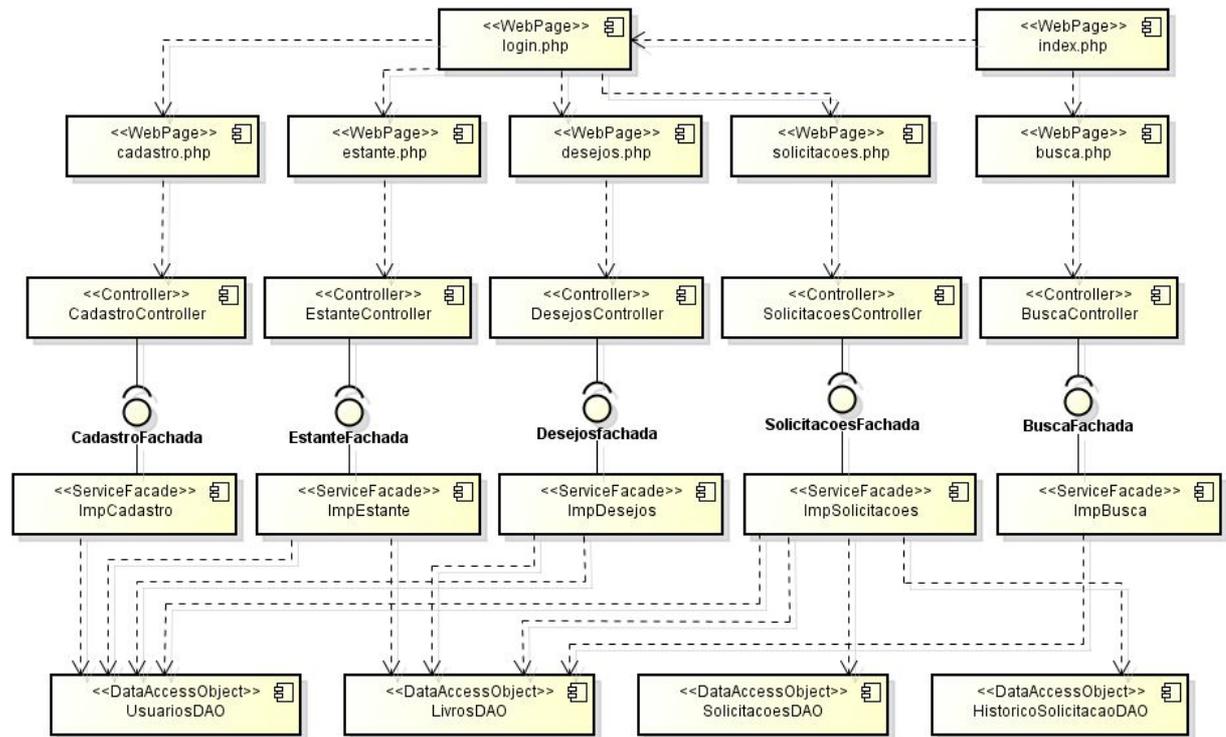
APÊNDICE E - DIAGRAMA ENTIDADE RELACIONAMENTO LÓGICO



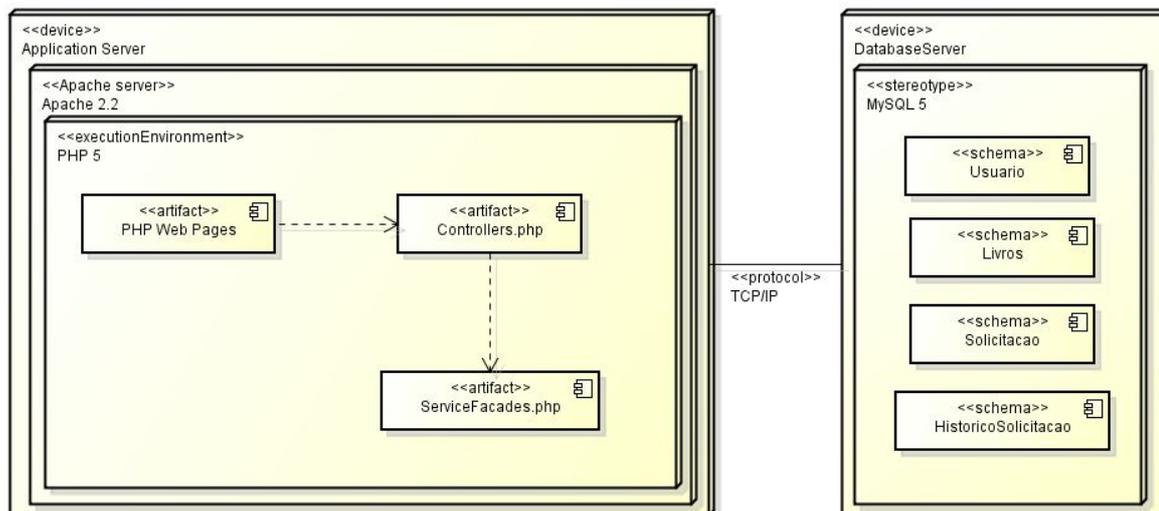
APÊNDICE F - DIAGRAMA DE CLASSES



APÊNDICE G - DIAGRAMA DE COMPONENTES



APÊNDICE H - DIAGRAMA DE IMPLANTAÇÃO



APÊNDICE I - REPRESENTAÇÃO DO SISTEMA



BIBLIOTROCA

Busque por Título, Autor ou Editora

E-mail Senha **ENVIAR** **CADASTRAR**

Livro novo é aquele que você nunca leu.

Passe um livro adiante e adquira outro, de sua escolha, de qualquer pessoa.

Sem dinheiro envolvido, apenas livros!

Livros mais recentes:

WALTER KIRN
MARKUS ZUSAK
O FETICO DAS TREVAS
A CULPA É DAS ESTRELAS
Gabriel Garcia



BIBLIOTROCA

Olá, você está logado como Vinicius! [sair](#)
Você possui 3 créditos

Busque por Título, Autor ou Editora

Home Minha Estante Lista de Desejos Solicitações Como Funciona

Livros recentemente desejados

Humberto Wiesneck
O PAI DOS BURROS
— Diccionario de palavras comuns e frases feitas —

ROSAMUNDE PILCHER
Um Encontro Suspeito

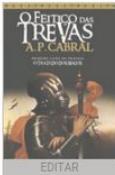
MICHEL FOUCAULT
e o Desejo

OSHO
O livro das mulheres

O sol, o genoma e a internet
Freeman Dyson

Livros recentemente cadastrados

A CULPA É DAS ESTRELAS
O FETICO DAS TREVAS A.P. CABRAL
COMO APROVEITAR AO MÁXIMO
MARKUS ZUSAK
WALTER KIRN

Home	Minha Estante	Lista de Desejos	Solicitações	Como Funciona
Meus livros disponíveis para troca			Adicionar livro para troca	
	Título: Cem anos de solidão Autor: Gabriel Garcia Marquez EDITAR		Título: O amor em tempos de cólera Autor: Gabriel Garcia Marquez EDITAR	<div style="border: 1px solid #0070c0; padding: 10px;"> <p>No image Available</p> <p>Escolher arquivo Nenhum arquivo selecionado</p> <p>Título: <input type="text"/></p> <p>Autor: <input type="text"/></p> <p>Editora: <input type="text"/> Ano de Publicação: <input type="text"/></p> <p>Descrição: <input type="text"/></p> <p>Adicionar Cancelar</p> </div>
	Título: José Saramago Autor: O evangelho segundo Jesus Cristo EDITAR		Título: Admirável mundo novo Autor: Aldous Huxley EDITAR	
	Título: O feitiço das trevas Autor: A. P. Cabral EDITAR		Título: A culpa é das estrelas Autor: John Green EDITAR	

Home	Minha Estante	Lista de Desejos	Solicitações	Como Funciona	
		Olá, você está logado como Vinicius! sair Você possui 3 créditos		<input type="text" value="Busque por Título, Autor ou Editora"/>	
Solicitações Recebidas					
	Título: Cem anos de solidão Autor: Gabriel Garcia Marquez Status: Pendente RESPONDER				
Solicitações Enviadas					
	Título: A menina que roubava livros Autor: Markus Zusak Status: Pendente GERENCIAR		Título: Como aproveitar ao máximo o seu tempo e potencial Autor: Edward R. Dayton Status: Pendente GERENCIAR		Título: Quando a máscara cai Autor: Walter Kirin Status: Pendente GERENCIAR