Desenvolvimento de Software Gerencial para Empresa de Locação de Trajes de Noivos

Raphael Bochnia[1], Regiane Orlovski[2]

[1,2] Tecnologia em Análise e Desenvolvimento de Sistemas – Faculdade Guairacá
CEP 85010-000 – Guarapuava – PR – Brasil

[1]raphaelbochnia@hotmail.com, [2]regianeorlovski@hotmail.com

Abstract: *The aim of this paper is to show the development of a system directed to the manager of a marriage and related attires rental business, in order to achieve greater agility and organization in the business daily procedures. Created in PHP language along with its programing needs, such as HTML and JavaScript, the management software was developed following the prototyping life cycle and the object-oriented programming paradigm. The system was finalized with some tools help such as IDE (Integrated Development Environment) Netbeans 7.2, Astah Community, USBwServer 8.5 Management System and Database MySQL. Therefore, the agility and efficiency to the business were satisfactorily achieved with the system, according to the expected goals.*

Resumo: *O objetivo desse artigo é mostrar o desenvolvimento de um sistema direcionado ao empresário do ramo de aluguéis de trajes de noivos e afins, com o propósito de obter maior agilidade e organização em processos diários da empresa. Elaborado na linguagem PHP juntamente com suas necessidades de programação, como o HTML e JavaScipt, o software gerencial foi desenvolvido seguindo o ciclo de vida de prototipação e o paradigma de programação orientado a objetos. Chegou-se a finalização do sistema com o auxílio de ferramentas como o IDE (Integrated Development Environment) Netbeans 7.2, Astah Community, USBwServer 8.5 e Sistema de Gerenciamento de Banco de Dados MySQL. Dessa maneira alcançou-se a agilidade e eficiência com o uso do software de modo satisfatório para a empresa, e de acordo com os objetivos esperados.*

Introdução

No decorrer dos dias, cresce o número de pessoas que estão investindo em festas de casamento, isso faz os empresários verificarem a oportunidade e sentirem a necessidade de investir na área, principalmente em relação a tecnologia. Com o aumento de investimentos em festas de casamento e afins, fator que motiva os empresários do ramo, o aumento na demanda dos serviços é considerável, assim as empresas de locação de trajes de noivos devem estar preparadas com sistemas que agilizem suas tarefas, como buscas, cadastros e contratos de locações automáticos.

Devido ao crescimento do comércio de aluguéis de trajes de noivos e afins, empresários dessa área sentem a necessidade do auxílio da tecnologia no gerenciamento dos seus negócios para automatizar suas tarefas rotineiras de forma prática, eficaz, ágil e segura. Tarefas antes feitas manualmente, geravam um grande risco para a empresa,

pois o uso do papel era visto como o único meio para organizar os dados de clientes, funcionários, produtos e contratos de locação, e dessa forma, o uso desse meio poderia ocasionar a perda de dados. Por esse motivo a realização das tarefas por meio de um sistema computacional com o objetivo de manter a integridade dos seus dados é o mais adequado ao estabelecimento, tornando as informações da loja mais seguras e organizadas.

O sistema desenvolvido visou as necessidades reais da empresa, de maneira que fossem alcançados benefícios para o local. Realizado de acordo com as tarefas pertinentes a processos diários da loja de aluguel de trajes de noivos e afins, o sistema buscou melhor aproveitamento e agilidade das tarefas diárias da empresa. Para o desenvolvimento do sistema foi utilizado a linguagem PHP, juntamente com o *JavaScript*, frameworks como *Jquery* e *Bootstrap* utilizados para ações, estilo e designer das páginas, *software Astah* Community* 6.4.1 utilizado para construir os diagramas da UML, o ambiente *Netbeans* 7.2 para o desenvolvimento do código fonte e o *DBDesigner4* utilizado para a modelagem do banco de dados.

O objetivo do artigo é mostrar o desenvolvimento de um *software* que auxilie o empresário do ramo de aluguel de trajes de noivos e afins a manter a integridade dos dados de clientes, funcionários e produtos, e dessa forma organizar e agilizar todo o processo de locação e geração de contratos entre cliente e empresa.

Fundamentação Teórica

O mercado de matrimônios está em alta no Brasil e mostra novos caminhos para empreendedores, segundo WSCOM (2013) a expectativa de faturamento em 2013 é de R$ 16 bilhões com um aumento de 8% em relação ao ano anterior. Segundo IBGE (apud Gnidarchichi 2013) os casamentos no Brasil cresceram 25% entre 2003 e 2009. Os dados revelam que o aumento dos investimentos nessa área é real, e que a cada ano que passa os números tendem a subir. No ano de 2011 o país ultrapassou 1 milhão de cerimônias e em 2012 já estava com 1,028 milhões, é o que aponta uma pesquisa realizada pela Associação dos Profissionais, Serviços para casamento e eventos sociais (Abrafesta), em parceria com a Data popular.

Para Gnidarchichi (2013) esse aumento tem como fatores: a melhora da economia do país, a quantidade de casamentos organizados por instituições religiosas e civis que prestam o casamento comunitário, e o novo código civil que facilitou para os futuros casais a realização do matrimonio.

Segundo WSCOM (2013) com a demanda elevada no mercado de cerimônias, a abertura de empresas e negócios tende a aumentar, elas vem buscar novos caminhos para o empreendedorismo. Nesse aspecto pode-se considerar a tecnologia e o desenvolvimento de sistemas como uma opção de amparo a essas empresas como solução para cumprir metas esperadas e assim aumentar seus lucros. Ainda de acordo com WSCOM (2013), o empresário Hugo Kloppel afirma que as empresas procuram investir cada vez mais na tecnologia.

Para Barros (2013), com o investimento na tecnologia é possível agilizar e facilitar os processos administrativos dentro das organizações, de modo a reduzir custos a longo prazo. Com a informatização é possível oferecer um produto ou serviço de qualidade juntamente com agilidade, entre outros aspectos que tornam essas empresas

mais eficientes aos empresários e seus clientes, assim obtendo um passo à frente em relação aos seus concorrentes que não tem a tecnologia no seu negócio.

Conforme Bartié (2002) a dependência tecnológica que as organizações possuem aumenta de forma rápida e constante, levando as empresas a buscar a informatização nos seus negócios, de modo a reduzir custos e aumentar a eficiência. Dessa forma elas conseguem manter-se ativas em um mercado cada vez mais competitivo.

Segundo Oliveira (2004), as transformações resultantes de novas tecnologias de informatização, automação e robotização são constantes. Essas mudanças afetam as empresas de forma direta, pois com ela consegue-se mais produtividade e eficiência, os quais são princípios básicos e essenciais para uma empresa no mundo globalizado e com crescente concorrência. Ainda Oliveira (2004) afirma que não se pode negar que as tecnologias inovadoras são responsáveis pelo aumento da produtividade devido a melhorias na produção e na possibilidade de diminuição de mão de obra.

Segundo Viotti (2013) o uso frequente do documento em papel nas empresas não é uma solução racional, também não colabora com a preservação do meio ambiente, além de não ser uma maneira segura de arquivamento. Nesse contexto a tecnologia desempenha um papel importante na sociedade, que seria a apresentação de sistemas tecnológicos confiáveis, ergonômicos e de fácil acesso para a sociedade. A partir da percepção da necessidade de sistemas assim, vê-se também a necessidade da garantia da qualidade do *software* desenvolvido que, para Bartié (2002, pg.5) "[...] não é uma opção a ser estudada, mas parte de uma estratégia de sobrevivência em um mercado cada vez mais exigente e competitivo."

Segundo Pádua (2009, pg.7) "Para a maioria das pessoas, inclusive alguns profissionais da informática, a codificação parece ser a única tarefa de um desenvolvedor de *software*". Nesse aspecto para começar a desenvolver um sistema o programador deve primeiramente estudar sobre o assunto e elaborar um projeto de *software*. Segundo, Pfleeger e Lawrence (2004) é necessário entender o problema em questão para posteriormente efetuar o desenvolvimento da codificação.

Ainda Pfeeger e Lawrence (2004) afirmam que um sistema não pode começar do vazio. Deve-se analisar o pedido do cliente ou supervisor e assim questionar: esse sistema vai possuir um banco de dados pronto e vai trabalhar em conjunto com outro sistema? ou ele irá realizar cálculos e terá um banco de dados próprio? cada passo deve ser analisado cautelosamente. Assim a questão principal é onde esse *software* começa e onde ele termina. Levando em consideração a questão levantada por Pfleeger e Lawrence (2004), o *software* desenvolvido foi analisado cuidadosamente em conjunto com o cliente, para melhor qualidade final.

Ao iniciar o desenvolvimento do *software* o programador tem a necessidade de modelar o sistema utilizando a *Unified Modeling Language* (UML) a qual segundo Lima (2011, p.30) "[...] é uma linguagem para especificação, construção, visualização e documentação de artefatos de um sistema de *software* intensivo". Essa linguagem de modelagem proporciona ao desenvolvedor a possibilidade de escolha de qualquer linguagem de programação, métodos e processos, pois é independente, podendo ser realizada em qualquer *software*.

Para Koscianski e Soares (2007) a UML é uma linguagem muito bem aceita na indústria de *software*, e esta modelagem gráfica se define como semiformal e orientada

a objetos, além de ser um padrão mundial onde se permite desenvolver modelos em várias fases do desenvolvimento.

Utilizando a UML pode-se documentar um projeto visualmente com clareza e simplicidade, aumentando a qualidade e produtividade do mesmo. Para Lima (2011 pg.30) "Os modelos são documentados visualmente e possibilitam a produção de artefatos que podem ser publicados e mantidos com facilidade, qualidade e produtividade". Já para Koscianski e Soares (2007) a UML fornece uma visão das interações do *software* desenvolvido com o mundo exterior, tendo como objetivo principal a descrição dos requisitos funcionais e identificação dos erros, por meio de testes.

Primeiramente ao iniciar a UML precisa-se modelar os casos de uso que, segundo Lima (2011) serve como se fosse uma ponte entre o cliente e o desenvolvedor, mostrando de forma conceitual as funções que o *software* desempenhará, dessa forma a conversa com o cliente é indispensável, apresentando o modelo até o momento que ele der a ordem de execução.

De acordo com Koscianski e Soares (2007) existem alguns elementos básicos nos casos de uso, são eles: os atores, que são os agentes que integram com o sistema, os próprios casos de uso, que são um conjunto de sequências que determinam alguma ação, e os relacionamentos que fazem as ligações entre todos os casos de uso de um sistema.

Depois da visão dos casos de uso deve-se observar e desenvolver a visão lógica, que de acordo com Lima (2011, pg.32) "[...]permite estruturar e organizar o desenho do sistema de forma lógica". O desenvolvedor deve mostrar por meio desse, seu ponto de vista. A visão de implementação que é de exclusividade dos analistas e desenvolvedores do sistema tem o objetivo de definir e distribuir a implementação dos trabalhos e até mesmo decidir a quantidade de códigos que serão utilizados. Na última etapa da UML tem-se a visão de implantação na qual o analista deve observar o ambiente de implantação desse sistema, as máquinas e servidores de aplicações.

Ainda Lima (2011) afirma que seguindo a UML é possível obter um controle maior sobre o sistema simplificando e detalhando os passos até o seu resultado, garantindo assim maior manutenção e qualidade ao produto final. De acordo com Lima (2011), quanto mais detalhada ficar a modelagem do banco de dados, melhor será a visualização do projeto final já que a compreensão de sistemas é extremamente complexa sem esse auxílio. Desse modo os sistemas complementares tornam-se mais visíveis ao programador a partir do entendimento da modelagem.

O sistema desenvolvido é um *software* de aplicação que segundo Pressman (2011) é um sistema feito sob medida para atender específicas unidades de negócio, dessa forma precisa-se estabelecer um ciclo de vida para dar continuidade ao projeto. O modelo que mais se adaptou ao *software* foi o modelo evolucionário de prototipação. Para Pressman (2011, pg. 62) "Modelos evolucionários são interativos. Apresentam características que possibilitam desenvolver versões cada vez mais completas do software".

Ainda Pressman (2011) descreve que no projeto rápido a construção de um protótipo sempre antecipa a avaliação pelos envolvidos, os mesmos retornam um *feedback* para a melhora do *software* com os aprimoramentos necessários. Nesse tipo de

modelo o protótipo é como um mecanismo que leva a identificar todos os requisitos daquele *software*.

Depois do ciclo de vida definido faz-se necessário o uso da programação, o sistema para gerenciamento da loja de trajes foi desenvolvido para acesso via *Web*, dessa maneira é preciso abordar sobre a Linguagem para marcação de texto ou em inglês: *HyperText Markup Language* (HTML). Segundo Freeman e Freeman (2008) a tecnologia de marcação de texto é uma linguagem que mostra ao navegador como são estruturados textos de páginas utilizando *tags*, já Silva (2008) defende que o HTML pode ser definido como qualquer conteúdo que está dentro de um documento *web* e que tem a capacidade de interligar-se a outros documentos do mesmo tipo.

Segundo Freeman e Freeman (2008) com o HTML é possível a criação de páginas *Web* por meio de uma linguagem interpretada pelos navegadores de internet, dessa maneira é possível a visualização dos textos nas páginas pelo usuário do sistema. Mas somente com o HTML não é possível o desenvolvimento de um *designer* mais sofisticado, e para esse meio utiliza-se das Folhas de Estilo em Cascata / *Cascading Style Sheet* (CSS). Para a W3C (2013), criadora do CSS, o estilo em cascata é de fácil manuseio por se tratar de uma linguagem independente do HTML e com o uso dessa ferramenta é possível alterar cores, *layouts*, fontes e até mesmo o tamanho da tela. Por esse motivo pode-se dizer que com o CSS é possível mudar o visual das páginas de acordo com as necessidades do *software.*

Com o CSS definido no sistema, ainda complementou-se com a utilização do *Bootstrap*, ferramenta a qual tem estilos de CSS definidos para cada tipo de base, agilizando os processos do programador e poupando tempo na construção de *designers* para páginas *Web*. Para Magno (2012) o *Bootstrap* é uma poderosa ferramenta usada no desenvolvimento *Web* tanto para iniciantes quanto para programadores experientes devido a padronização e melhores práticas de programação.

Nesse aspecto é correto afirmar que com o domínio do HTML, CSS e *Bootstrap* é possível desenvolver páginas com *designers* ergonômicos e também as interligar umas às outras, mas não é possível desenvolver ações, como por exemplo um cadastro de clientes ou uma busca por produtos sem o auxílio da linguagem Pré-Processador de Hipertexto / *Hypertext Preprocessor* (PHP). Segundo Dall'Oglio (2009) o PHP foi desenvolvido por Rasmus Lerdorf com o intuito inicial de monitorar seu currículo na internet de forma dinâmica. Por meio de modificações o sistema passou a permitir a criação de códigos de forma simples, a partir de então recebeu a nomenclatura atual PHP e seu código foi disponibilizado por Rasmus em 1995 para ser compartilhado e melhorado por outras pessoas.

Para Melo (2007) o PHP é uma linguagem que tem como objetivo primário a criação de conteúdos dinâmicos para páginas *Web*, e que possui código aberto para a melhoria do mesmo. Melo (2007) ainda defende que o PHP possui várias outras vantagens e não apenas a criação de páginas dinâmicas, como por exemplo o custo, pois o PHP é um código livre e, desta forma, não possui um custo de licença podendo ser instalado em qualquer máquina e para qualquer número de usuários. O suporte da linguagem ocorre por meio de comunidades que veem crescendo, tanto no Brasil quanto no exterior. Ainda há outras vantagens como a manipulação de vários tipos de arquivos por meio do PHP, a geração de imagens dinâmicas, criptografia de dados, suporte para vários bancos de dados nativos, entre outras vantagens que a linguagem possui.

Nesse aspecto é possível afirmar que com a junção do PHP no desenvolvimento de sistemas consegue-se chegar a um resultado satisfatório para o usuário, como foi visto o HTML e o CSS tem a função de criar e estruturar os textos com *designers* apropriados para cada aplicação, e com o PHP pode-se realizar ações em elementos do HTML. Mas para que tudo funcione perfeitamente a programação deve ser o mais simplificada possível, para a geração de um código de fácil entendimento a qualquer programador que vier a utiliza-lo. Desse modo, é fácil afirmar que precisa-se recorrer a Orientação a Objetos (OO). Para Dall'Oglio (2009) com a orientação a objetos é possível programar utilizando meios mais próximos do mundo real, tornando o entendimento parecido com o que é pertinente do dia a dia das pessoas.

Para o desenvolvimento do *software* também foi necessário a utilização da linguagem *JavaScript* que segundo Silva (2010) foi criada com finalidade de adicionar maior interatividade as páginas desenvolvidas com o HTML, pois a linguagem de marcação de texto não possui a capacidade de adicionar interatividade avançada as páginas. Segundo Morrison (2010) os navegadores têm dentro do código uma parte especial do *JavaScript* que é responsável por executar os códigos da linguagem, por esse motivo a linguagem de interação também é descrita como interpretada.

Outra ferramenta utilizada foi o *Ajax* que para Riordan (2010) é mais interativo, rápido e fácil de usar e manipular do que outros meios para essa mesma finalidade, além de fazer solicitações assíncronas deixando o usuário do sistema continuar trabalhando em outras funções enquanto o *Ajax* carrega dados enviados em segundo plano.

Para armazenar todos os dados necessários para a empresa e o funcionamento do *software* faz-se necessário o uso de um Banco de Dados. Para Elmari e Navathe (2011) um Banco de Dados pode ser qualquer coleção de registro como, endereço, telefone, nome, sobrenome, entre outros, que se relacionam e tem um resultado implícito. Dessa maneira, tendo essa tecnologia consegue-se armazenar informações de clientes, produtos, funcionários e outras informações importantes para o funcionamento do sistema de gerenciamento de locações para trajes de noivos e afins.

Também é necessário um Sistema Gerenciador de Banco de Dados / *Database Management System* (SGBD) que, segundo Elmari e Navathe (2011), é uma coleção de programas que permite a criação, alteração, definição, compartilhamento e armazenamento de dados, facilitando qualquer operação que necessite manipular o conteúdo requerido para um sistema.

Para a manipulação dos dados, o SGBD escolhido foi o *MySql*. De acordo com Beighley e Morrison (2011) para a comunicação dos dados com o Sistema de Gerenciamento do Banco de Dados é necessário utilizar de uma linguagem que o mesmo entenda, para isso foi utilizada a linguagem *Structured Query Language* (SQL).

Silberschatz (2006) afirma que um Banco de Dados necessita da especificação de uma linguagem de definição de dados, e o SQL faz essas especificações de cada relação além de conjuntos e relações individuais de cada esquema. Para Oliveira (2011), SQL são conjuntos de comandos que criam e mantém estruturas de um banco de dados, podendo incluir, excluir, modificar e pesquisar informações em tabelas do *Database.*

O desenvolvimento do sistema foi organizado no padrão *Model, View, Controller* (MVC) que segundo Dall'Oglio (2009) tem os seguintes significados e funções: *Model* / Modelo, é um objeto que representa os dados do domínio de negócios

do *software*; *View* / Visualização, é onde ocorre a interação com o usuário, como por exemplo, uma tela de cadastro de clientes, onde a *View* é responsável por sua estrutura apenas com relação a visualização da página, que pode ser em HTML, não podendo exercer qualquer tipo de interação no fluxo de execução do *software*; *Controller* / Controle, é onde são recebidas, interpretadas e executadas as informações vindas da *View* e atualizadas na camada *Model*.

Ainda Dall'Oglio (2009) defende que as camadas devem interagir uma com as outras, a camada de Modelo não dependente das demais. A camada de Visualização e a de Controle fazem referências a objetos uma das outras e enviam para a camada de Modelo que recebe os dados e manda um retorno. Pode-se representar o modelo no desenho abaixo:

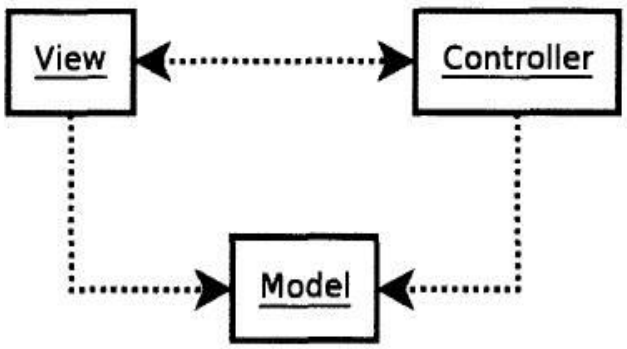


Figura1.Modelo MVC

Fonte Dall'Oglio (2009)

Dessa maneira, um auxílio no desenvolvimento é o *framework Calango*, que segundo Ribeiro (2013) é um projeto *OpenSource* (código aberto) e é um *framework* que trabalha no padrão MVC, ajudando no desenvolvimento de projetos que pretendem organizar-se dentro dessa estrutura. O Calango se adapta à vários tipos de *softwares*, bastando o desenvolvedor conhecer sua estrutura, e adaptar às necessidades de desenvolvimento.

Quando os ciclos de vida, diagramas e padrões de projetos estão estabelecidos há a necessidade do uso de determinadas ferramentas para o desenvolvimento de *softwares* e para esse sistema foram necessários IDE *Netbeans*, *Astah Community*, *DBdesigner* e USBw. O IDE, segundo o Netbeans.org (2013), é de código aberto e auxilia programadores com ferramentas para várias linguagens de programação, possuindo uma grande quantidade de usuários ao redor do mundo. O *software* para modelagem, segundo Astah.net (2013), é uma ferramenta dinâmica e poderosa que refina o processo de modelagem de diagramas da UML e possui sua versão livre. O *software* para modelagem de Banco de Dados, segundo FabForce.net (2013), é um sistema onde pode-se modelar e criar bancos de dados visualmente em um ambiente único, tendo seu código livre aos usuários. Ainda foi necessário o USBw que para UsbWebServer.net (2013) é uma ferramenta que combina vários *softwares* de servidor *web*, como por exemplo, *Apache*, *Mysql*, PHP e *PhpMyAdmin*, com a vantagem de ser executado a partir de um dispositivo móvel ou uma pasta do computador.

Etapas do Desenvolvimento do Projeto

Ao começar um projeto de um *software*, deve-se primeiramente realizar o levantamento de requisitos, no qual ficaram estabelecidas as necessidades do cliente. Na análise realizada com o cliente foi possível descobrir as necessidades da loja que são apresentados no apêndice X. A seguir, deve-se organizar um ciclo de vida para o desenvolvimento do projeto simultaneamente a criação de diagramas de UML. Só então é dado início a programação do *software*.

Tendo em vista a forma de organização dos dados de clientes, produtos e até mesmo contratos da loja de trajes festivos, que antes eram realizados por meio de documentos em papel, decidiu-se a nova forma de organização por intermédio da informatização afim de acompanhar a concorrência de mercado, facilitar e agilizar o trabalho do empresário.

Sabendo disso, o projeto de desenvolvimento do sistema teve início no levantamento de requisitos no qual percebeu-se quais deveriam ser as funções atribuídas ao *software* de gerenciamento de trajes festivos. Funções como cadastrar, pesquisar, excluir e editar foram estabelecidas para registrar dados de clientes. Da mesma maneira essas funções foram atribuídas para os dados de funcionários que recebem ainda, funcionalidades como senha e usuário de acesso. O cliente ainda registrou a necessidade de um controle de estoque de produtos onde o sistema deve realizar o cadastro, indicar datas de locação dos mesmos e pesquisar sua disponibilidade em estoque, podendo os produtos serem editados e excluídos.

Na análise de requisitos, ainda levantou-se a possibilidade de haver uma função destinada a gerar contratos de locações aos clientes da loja no qual o operador do sistema apenas indicaria o cliente, o produto e a data desejada e automaticamente o sistema deve gerar um documento em formato PDF para a assinatura do cliente que realizar a locação.

Posteriormente ao levantamento de requisitos, houve a necessidade da escolha do ciclo de vida o qual foi o de prototipação, assim como em outros modelos, tem-se como primeira etapa a análise de requisitos. Dessa forma, levanta-se todas as questões relacionadas ao *software* como suas funcionalidades, quais os cálculos a serem realizados e se devem existir relatórios e funções para impressão. Além disso é considerado como o sistema se adaptará melhor a empresa assim como a interface adequada às cores e marcas da empresa, que no caso do *software* que foi desenvolvido foram utilizados os próprios estilos existentes da empresa. Também deve ser levado em consideração o *hardware* que a empresa possui e quem irá utilizar esse sistema. Todos os dados serão levantados nessa etapa do ciclo, antes de dar continuidade ao projeto.

Depois de pronta à análise de requisitos, o desenvolvedor analisa cautelosamente os dados e projeta a melhor forma de ser construído esse *software*. Dessa forma percebeu-se a necessidade de um cadastro para funcionários o qual não havia sido estabelecido previamente na análise de requisitos. Sequencialmente deve-se voltar ao cliente com o projeto em mãos e apresentar ao mesmo, até ter certeza de que é isso mesmo que precisa para a sua empresa. Assim o ciclo continua, ou é projetado novamente até a autorização de execução.

Para a elaboração do projeto levado ao cliente, são utilizados os diagramas da UML, que por meio desses pode-se modelar o sistema antes mesmo de começar a

programar, como consta nos apêndices A, B e C, assim é possível identificar por exemplo, quem são os atores, quais as classes necessárias, e até mesmo as funções que o *software* realiza. Foi por intermédio dessa modelagem que chegou-se a concordância do cliente com a realização do sistema.

A construção do *software* teve como base o *framework Calango*, no qual trabalha com a estrutura MVC, dificuldades foram encontradas com a utilização do *framework*, pois o entendimento completo do mesmo só foi possível por meio da própria programação desenvolvida para adequar o *Calango* as necessidades do sistema para gerenciamento de trajes de noivos e afins. Como o MVC trabalha com a estrutura de *Model, View e Controller*, as classes de modelagens foram encaminhadas para o *Model*, as classes controladoras para o *Controller* e as páginas de apresentação para a *View*. Também foi utilizado os *Frameworks Jquery* e *Bootstrap*, para dar mais interatividade e ergonomia as páginas.

Ficou definido que o sistema precisaria de uma área de cadastro de clientes, com os seguintes dados: nome completo, documentos como RG e CPF, endereços completos, e contatos do cliente. Dessa maneira foi criado uma classe chamada "*Cliente_*", que é responsável por toda a modelagem dos dados de clientes com o banco de dados, como por exemplo, as funções salvar, editar, excluir, alterar, entre outras, elas passam por um controlador da classe chamado "*Model*" e então apresentam dados na tela do usuário por meio de páginas desenvolvidas em HTML que se encontram dentro da pasta "*View*", a qual encontram-se todas as demais páginas do sistema.

Do mesmo modo foi desenvolvido uma classe chamada "*Funcionário_*" com alguns atributos, por exemplo, um nome de usuário e senha para acesso ao sistema, que todo *software* deve possuir. Outra classe pertinente ao sistema é a "*Produto_*" onde é possível cadastrar o estoque da loja de trajes festivos, com as informações referentes ao produto, por exemplo, tamanho, cor, quantidade, descrição e o valor, assim como nas outras funções é permitido a pesquisa pelo fator desejado, diferenciando-se a pesquisa dos produto pelo motivo de apresentar também nessa as datas de locações e a disponibilidade no estoque, auxiliando o empresário e poupando o tempo que seria gasto para procurar pelo item sem a ajuda do sistema.

Para transformar o sistema em gerencial e não apenas um controle de estoque, foi criada uma classe chamada "*Locacao_*" na qual existem funções pertinentes a locação de trajes, salvando datas, gerenciando quantidades de produtos em estoque e ainda gerando arquivo no formato PDF para impressão e finalização do contrato de locação. Dessa maneira, agilizando o processo do empresário.

Foi necessário uma tela de devolução, para que ao retornar a loja um produto que antes estava em locação agora possa ser devolvido, retornando assim a quantidade daquele produto ao estoque, que posteriormente poderá ser locado outra vez. Ao realizar uma devolução o produto é automaticamente somado a quantidade do estoque e o contrato apagado do Banco de Dados.

O código em PHP foi utilizado seguindo a Orientação a Objetos por ser mais flexível esse tipo de programação. E também de fácil manutenção e melhor administração de problemas que venham à ocorrer, além disso a reutilização de código deixa o *software* mais leve, mais rápido e eficaz. Mas somente a linguagem PHP não seria suficiente para a construção desse programa, precisava-se de outras ferramentas

como HTML onde foi utilizado para construção dos *layouts*, assim como o *Jquery* e o *Bootstrap*, *frameworks* que foram necessários para melhor *designer* e ações das páginas.

O sistema ainda conta com várias outra classes e funções que são pertinentes ao funcionamento do mesmo, também fez-se necessário o uso de *plug-ins* como o mPDF para a geração de arquivos em formato PDF e o *framework Bootstrap* que tem como funcionalidade personalizar a aparência das páginas, deixando o *designer* das mesmas mais agradáveis aos olhos do usuário do *software*.

Durante a fase de programação o sistema sofreu várias reformas em seu banco de dados e códigos. Pois antes o banco de dados estava com informações repetidas a respeito de clientes e funcionários, então decidiu-se a organização dessa modelagem dos dados, por consequência os códigos também tiveram de ser recodificados para o novo esquema do banco.

Também no início da programação, não foi usado a orientação a objetos de forma correta, como por exemplo no código:

* Cadastrar *

Nesse caso o código estava dentro de um arquivo chamado *template.html* o qual faz a construção de todo o *template* do sistema, essa linha estava orientando o botão de menu Cadastrar para a página de visualização "salvarCliente.html", quando o correto seria enviar esse botão para uma classe de modelagem que só então chamaria a página HTML, de acordo com as necessidades de solução desse problema a linha de comando foi substituída por:

* Cadastrar .*

Dessa maneira o código era enviado para a classe de modelagem, obedecendo o padrão MVC e trabalhando de forma correta com a orientação a objetos.

O método de reutilização de código foi empregado na construção do *Create, Read, Update* e *Delete* (CRUD), nessa etapa foi gerado uma tabela com as informações necessárias e dependendo do ação que o sistema estiver trabalhando ele habilita certo botão, por exemplo se o usuário estiver na tela de alterar ele habilita o botão alterar mas se ele estiver na página pesquisar ele aparece apenas o botão de detalhes. Exemplo do código:

```
if (App::$action == 'pesquisaCliente') {
        $conteudoTabela_temp = str_replace('#BOTAO#', '', $conteudoTabela_temp);
    } else if (App::$action == 'alteraCliente') {
        $conteudoTabela_temp = str_replace('#BOTAO#', '<a href="#URL#Cliente_/formCliente/#ID#" role="button" class="btn btn-warning">Editar</a>', $conteudoTabela_temp);
    } else if (App::$action == 'buscarCliente') {
        $conteudoTabela_temp = str_replace('#BOTAO#', '<a onclick="detalhes()" role="button" class="btn btn-inverse">Adicionar</a>', $conteudoTabela_temp);
    } else {
```

$conteudoTabela_temp = str_replace('#BOTAO#', 'Excluir', $conteudoTabela_temp);

}

O código acima demonstra que se a ação do sistema for "*pesquisarCliente*", no momento que é montado a tabela o sistema adiciona um botão de pesquisa, se a ação for alterar, o botão de alterar aparece e caso a ação seja excluir, o sistema habilita o botão para excluir, cada um cumprindo com suas necessidades.

Durante a construção do sistema, percebeu-se a necessidade de uma avaliação do *software* até a etapa que se encontrava, desse modo como permite o ciclo de prototipação, foi levado ao cliente um modelo e questionado sobre a versão até o momento, o cliente mostrou-se satisfeito até então, mas acrescentou que ainda faltava no sistema a visualização das datas de locação dos produtos contratados.

O *software* voltou para a programação, resolvendo a questão levantada pelo cliente em relação as datas de locação. Dessa forma a continuidade do sistema foi até a finalização, passando pela faze de testes e entregue na loja de trajes festivos, onde também ocorreu a implantação e a introdução de funcionamento do *software*, esse treinamento foi realizado de maneira explicativa ao empresário e funcionários da loja, abordando-se questionamentos sobre a manipulação do *software*

Segundo o empresário e dono da loja de trajes festivos, o sistema concedeu melhoras significativas a sua empresa, aumentando a eficiência e agilidade dos seus negócios. Desse modo pode-se afirmar que o *software* alcançou a agilidade e eficiência esperada para a empresa.

Resultados

Por meio do *software* desenvolvido para gerenciamento da empresa de trajes de noivos e afins, percebeu-se a importância da análise detalhada de todos os requisitos, pois é a partir daí que se define um rumo de desenvolvimento do *software*. Quanto melhor for a análise de requisitos realizada, mais fácil e correto será o desenvolvimento e a finalização do sistema com a qualidade esperada.

Também percebeu-se a importância da modelagem do sistema por meio da UML, pois com ela se atinge uma visão do sistema, mesmo antes deste ser finalizado. Além disso, a modelagem ajuda o programador a identificar erros e melhorar suas funções antes do início da programação do *software*.

O estudo realizado no projeto para desenvolvimento do sistema possibilitou um conhecimento nas diversas áreas de programação. No entanto, para a realização do *software* foi necessário um aprofundamento deste estudo, o que proporcionou o fortalecimento de habilidades necessárias para a criação do sistema. A pesquisa bibliográfica teve também grande importância nesse processo de aprofundamento. Com ela pode-se distinguir maneiras e métodos para uma correta programação assim como as ferramentas ideais para cada tipo aplicação no sistema.

Com relação a empresa, a implantação do *software* trouxe inúmeros benefícios ao empresário. O mesmo indicou uma melhora significativa em relação a agilidade do

trabalho fornecida com o sistema. Com a função de cadastro de funcionários e clientes é proporcionada uma forma mais rápida de pesquisa a respeito de detalhes dos mesmos, como por exemplo, informações de endereço e contato. Havendo o cadastro dos clientes, é possível a geração automática de contratos.

Por intermédio da função estabelecida para geração de contrato automatizado, o empresário destacou a organização e a rapidez da documentação necessária para o aluguel. Também houve a diminuição de gastos já que não há mais a necessidade de blocos de contrato que agora são diretamente impressos por meio do sistema. Além disso, as funções de controle de estoque trouxeram a loja uma organização por meio de códigos empregados a cada produto. Ainda, o usuário do sistema pode verificar as datas de locação e entrega devido a tais códigos armazenados em bancos de dados, que antes eram arquivados em papéis, o que trazia a dificuldade de se procurar em todos os contratos para saber a data de locação do produto.

Considerações Finais

Considerando a necessidade das empresas em utilizar da informatização para o gerenciamento de seu negócio, objetivou-se com o presente estudo o desenvolvimento de um *software* que auxiliou o empresário do ramo de aluguéis de trajes de noivos e afins em suas tarefas cotidianas. O desenvolvimento do sistema *Web* baseou-se na linguagem PHP Orientado a Objetos juntamente com o Banco de Dados *MySQL*, com o intuito de garantir a segurança dos dados, agilizar o processo de locação e organização da empresa.

O sistema foi desenvolvido sob o ciclo de vida de prototipação, diante disso, se futuramente necessite de funções a mais do que aquelas inicialmente sugeridas, possa ser elaborado um estudo de casos e implementado as novas tecnologias, além da facilidade de criação de funções complementares por se tratar de um *software* feito no padrão MVC e Orientado a Objetos.

O projeto passou pelas fases de testes do programador e atualmente é utilizado pelo cliente. Dessa maneira pode-se afirmar que o *software* cumpriu as expectativas do usuário.

Referências

Pfleeger, S. L. (2004). “Engenharia de software: teoria e prática”. 3 edição. Editora: Prentice Hall, Brasil.

Oliveira, O. J, et AL (2004). “Gestão da Qualidade: Tópicos Avançados”. Editora: Thompson pioneira, Brasil.

Lima, A. d. S. (2011). “UML 2.3: do requisito a solução”. 1ª edição. Editora: Érica, Brasil.

Melo, A. A. d. (2007). “PHP profissional: aprendaa desenvolver sistemas profissionais orientados a objetos com padrões de projetos”. Editora: Novatec, Brasil.

Dall’Oglio, P. (2009). “PHP programando com orientação a objetos”. 2ª edição. Editora: Novatec, Brasil.

Koscianski, A. (2007). “Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software”. 2ª edição. Editora: Novatec, Brasil.

Bartié, A. (2002). “Garantia de Qualidade de Software: adquirindo maturidade organizacional”. 13ª Reimpressão. Editora: Elsevier, Brasil.

Silva, M. S. (2008). “Criando sites com HTML e CSS: sites de alta qualidade com HTML e CSS”. Editora: Novatec, Brasil.

Freeman, E.; Freeman E. (2008). “Use a cabeça HTML Com CSS e XHMTL”. 2ª edição. Editora: Alta Books, Brasil.

Oliveira, C. H. P. d. (2011). “SQL: Curso Prático”. 7ª edição. Editora: Novatec, Brasil.

Silberschatz, A.; Korth, H. F.; Sudarshan, S. (2006). “Sistema de banco de dados”. 9ª edição. Editora: Elsevier, Brasil.

Beighley, L.; Morrison, M. (2011). “Use a Cabeça: PHP & MySQL”. 1º reimpressão. Editora: Altabooks, Brasil.

Elmasri, R.; Navathe S. B. (2011). “Sistemas de banco de dados”. 6ª edição. Editora: Pearson Addison Wesley, Brasil.

Wscom. (2013) “Indústria do casamento deve crescer 8%”, http://www.wscom.com.br/noticia/economia/INDUSTRIA+DO+CASAMENTO+DEVE+CRESCER+8+-149544, Junho.

Gnidarchichi, T. (2013). “As estatísticas confirmam o casamento no Brasil está em alta”, http://www.casamentoclick.com.br/report/as-estatisticas-confirmam-o-casamento-no-brasil-em-alta.html, Outubro.

Viotti, G. d. M. (2011). “Tecnologia vs. Sustentabilidade – Meios eletrônicos na economia de papel”, http://www.tiespecialistas.com.br/2011/06/tecnologia-versus-sustentabilidade-meios-eletronicos-na-economia-de-papel/, Junho.

Magno, A. (2013). “Globo Bootstrap”. http://blog.alexandremagno.net/2012/08/globo-boostrap/, Outubro.

W3C (2013). “HTML & CSS”. http://www.w3.org/standards/webdesign/htmlcss, Outubro.

Barros, A. L. d. S. (2013). “Gastos em tecnologia – custo ou investimento”, http://www.administradores.com.br/artigos/tecnologia/gastos-em-tecnologia-custo-ou-investimento/73449/, Outubro.

Fabforce (2013). http://www.fabforce.net/dbdesigner4/, Outubro.

Netbeans (2013). “NetBeans IDE Features”, https://netbeans.org/features/index.html, Outubro.

Astah (2013). http://astah.net/editions/community, Outubro.

Usbwserver8 (2013). http://www.usbwebserver.net/en/, Outubro.

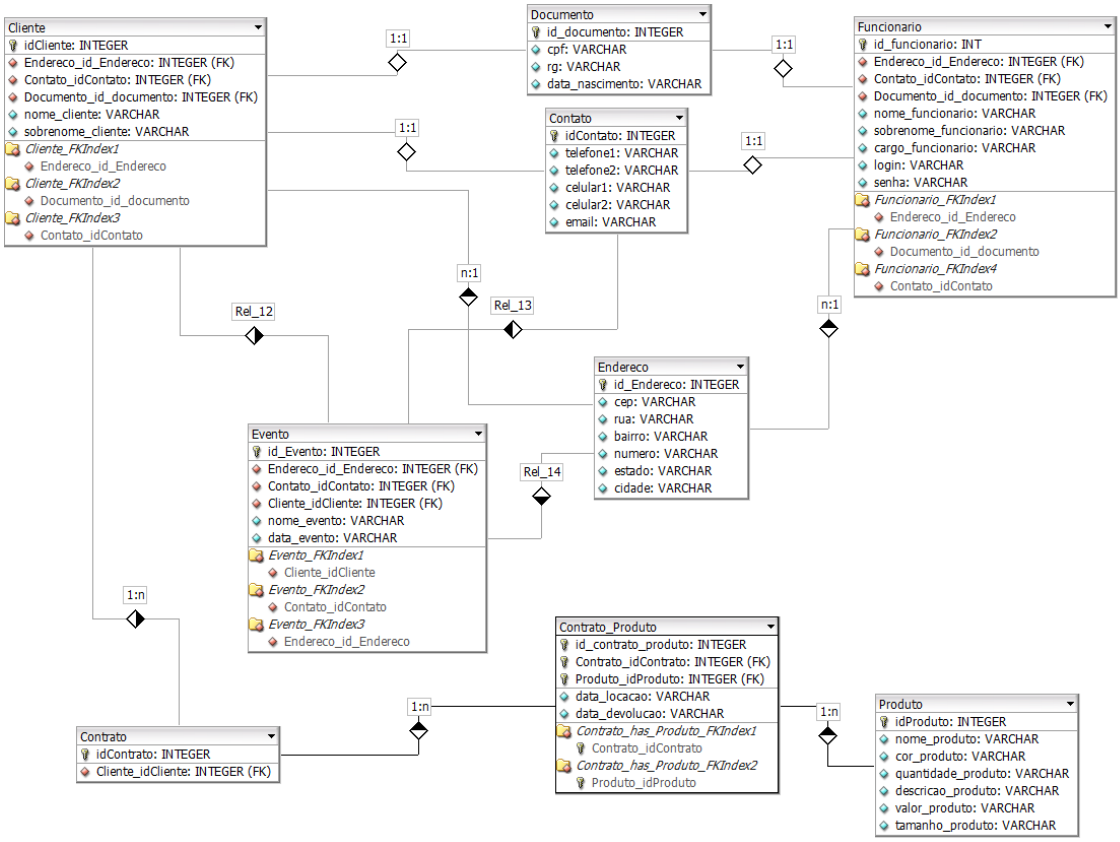
Ribeiro. S (2013) “Calango Framework”, http://calango.sergioribeiro.com.br/, Outubro.

Riordan, R. M. (2010). “Use a cabeça Ajax Profissional”. Editora: Alta Books, Brasil.

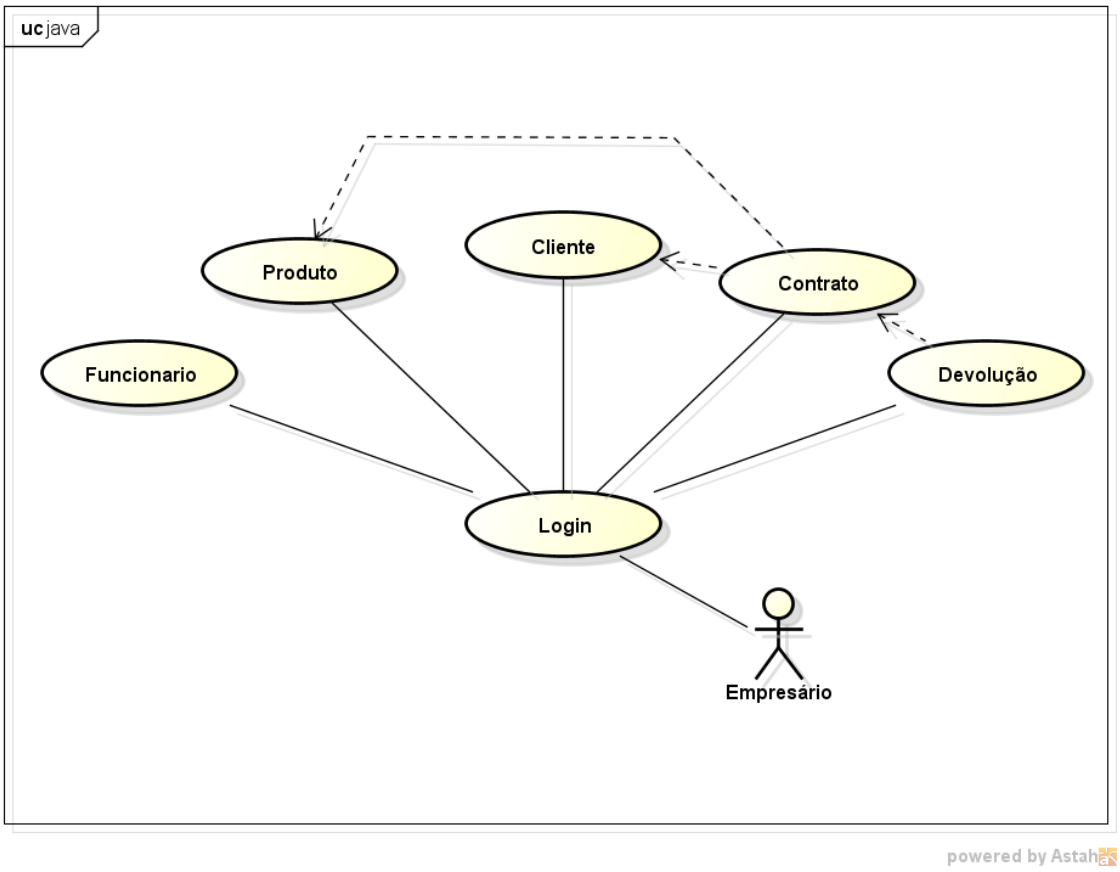
Silva. M. S. (2010). “JavaScript: Guia do Programador”. Editora: Novatec, Brasil.

Apêndices

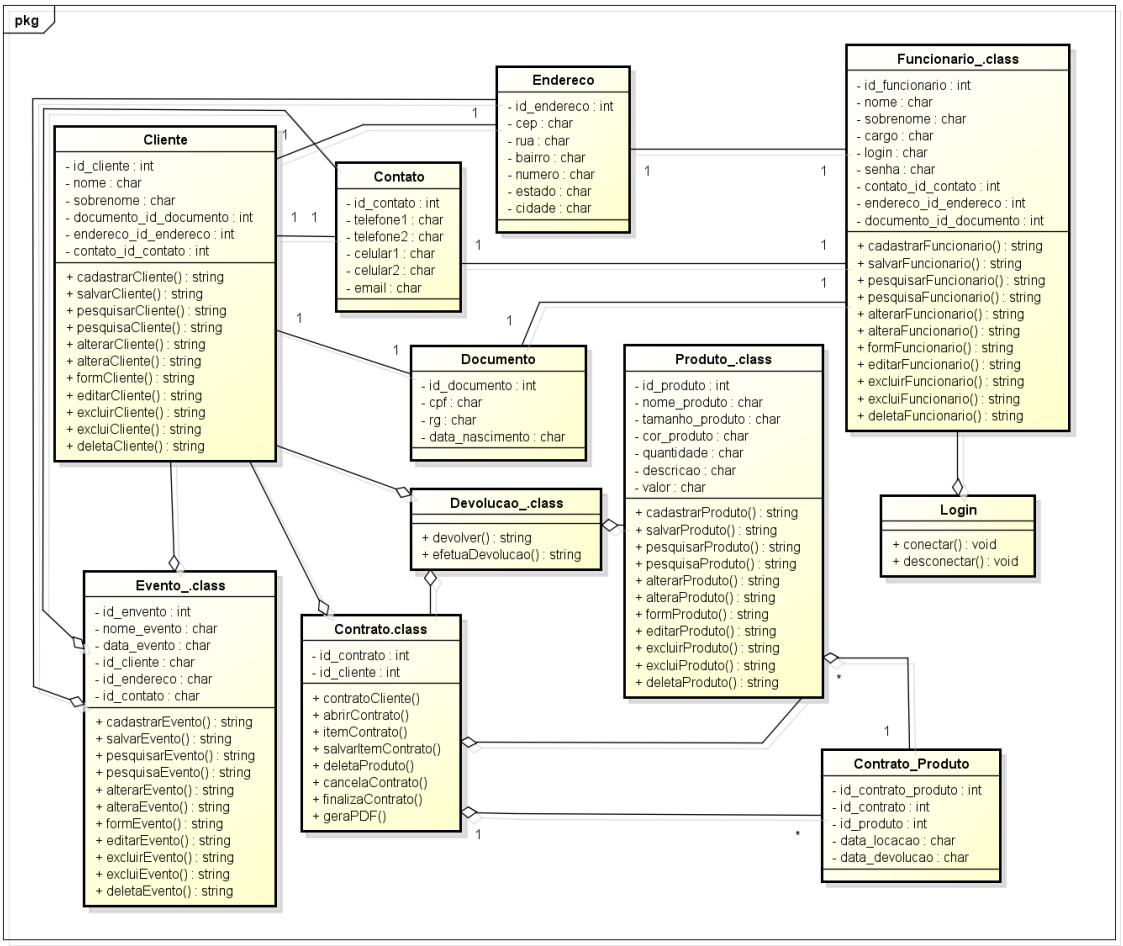
Apendice M: Modelagem do Banco de Dados



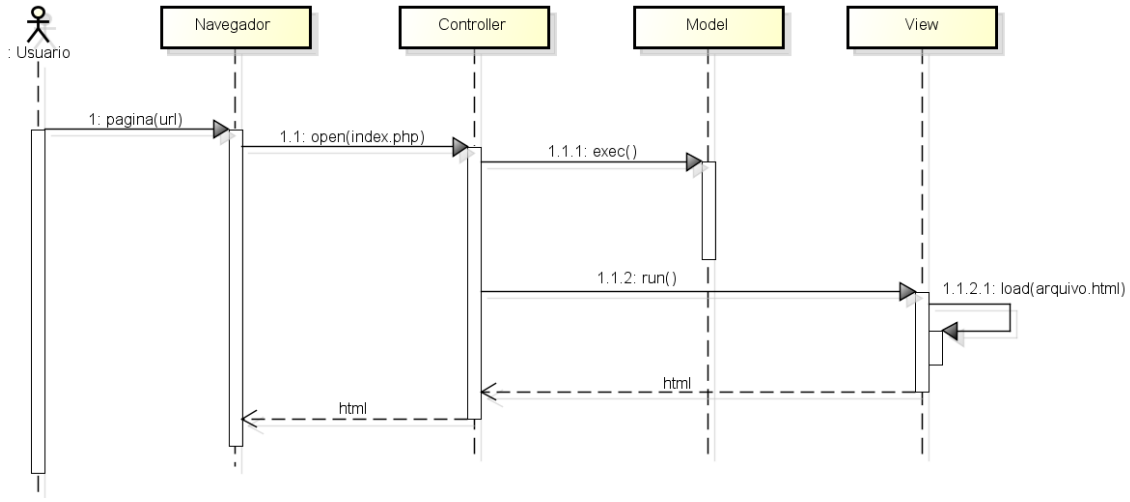
Apêndice A: Diagrama de casos de uso



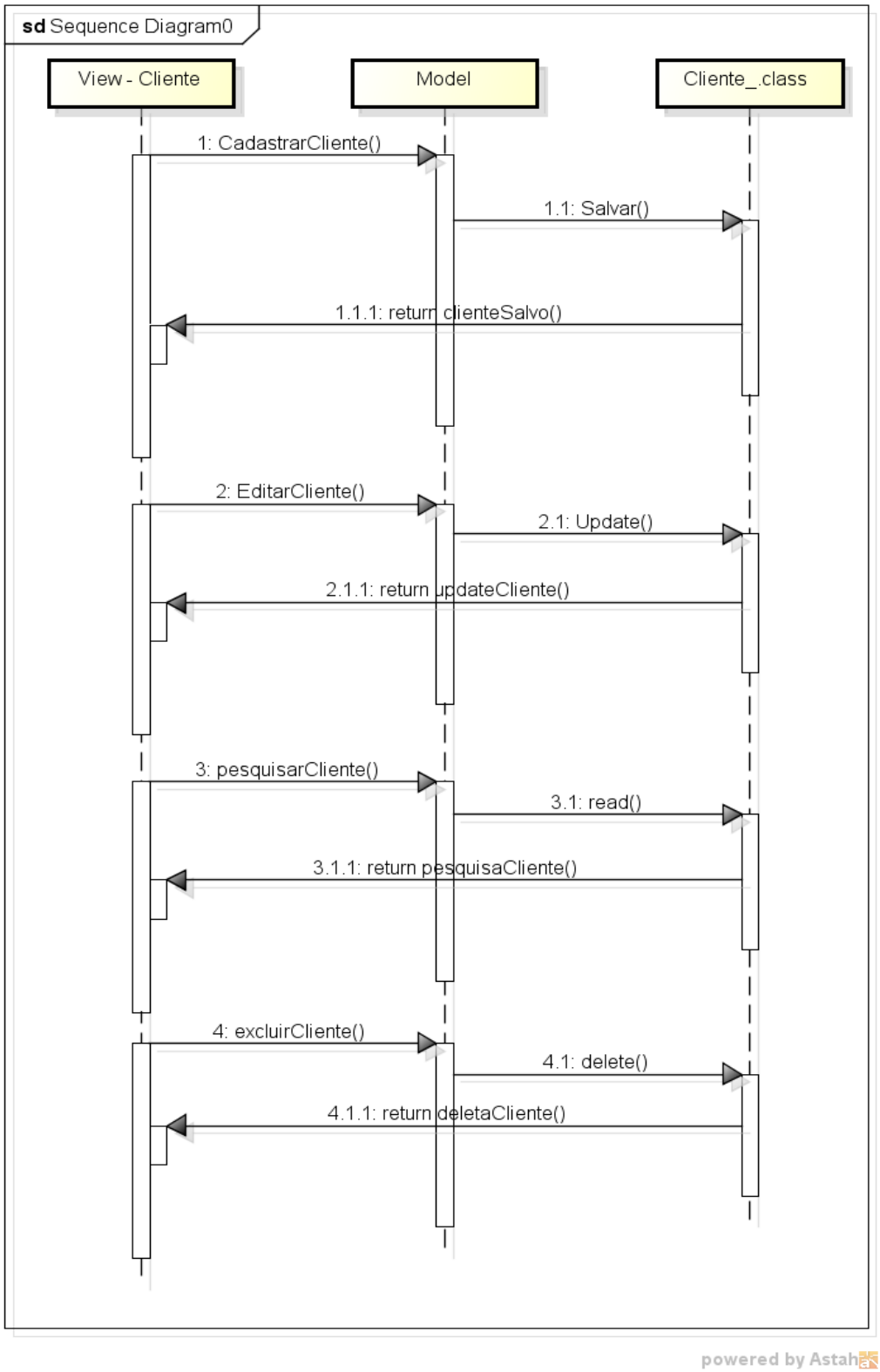
Apêndice B: Diagrama de classe



Apêndice Z: Diagrama de sequência do framework Calango padrão MVC



Apêndice C: Diagrama de sequência - Demonstração do CRUD de cliente



Apêndice X: Análise de requisitos

Análise de Requisitos

Principais funções do Sistema:

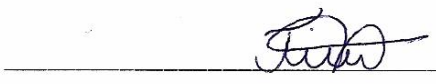
- Cadastrar produtos
- Pesquisar produtos
- Cadastrar clientes
- Pesquisar clientes
- Tela de acesso
- Gerar contratos
- Detalhes de produtos
- Pesquisar datas de locação

Funções detalhadas:

- Cadastrar produtos: deverá conter o código de cada produto gerado automaticamente, o nome do produto, o tamanho do produto, o valor do produto e a descrição do produto;
- Pesquisar produto: deverá pesquisar os produtos por meio do código ou do nome do produto;
- Cadastrar cliente: deverá cadastrar os clientes com os dados de nome, CPF, RG, endereço, telefone e e-mail;
- Pesquisar clientes: deverá realizar a pesquisa dos clientes por meio do CPF ou do nome do cliente e deverá mostrar detalhes do mesmo;
- Tela de acesso: deverá ser acessada por funcionários cadastrados antecipadamente no sistema;
- Gerar contratos: deverá gerar os contratos no formato PDF para impressão contendo as devidas cláusulas de contrato de locação. Deverá ser gerado a partir dos produtos selecionados e deverá conter informações do cliente e dos produtos, assim como valor final do contrato;
- Detalhes de produto: deverá mostrar detalhes de produtos ao se clicar em botão específico;
- Pesquisar datas de locação: deverá mostrar os produtos locados na data selecionada.

Designer do sistema:

O sistema deverá ser ergonômico de maneira a fazer com que o usuário se sinta bem ao utilizar o software com uma interface bonita e utilizando a logo da empresa.



Tiene Brandeleiro – Gerente da loja Véu de Noiva