Aplicação da Tecnologia da Informação em uma Vídeo Locadora por Meio de Desenvolvimento de Sistema Utilizando Software Livre

Rodrigo Florêncio de Lima[1], Regiane Orlovski[2], Willian Agner[3]

[1, 2, 3]Análise e desenvolvimento de Sistemas – Faculdade Guairacá
Guarapuava – PR – Brasil

[1]rodrigufl@hotmail.com, [2]regianeorlovski@hotmail.com,
[3]will.agner@hotmail.com

Abstract. *This article aims to show the development of a software alternative to video rental companies, so it will be discussed the importance of managing a company, and then present some types of information system to assist them, exposing the benefits of the tools CRM for customer loyalty and also present the steps for setting up the system, taking into account the requirements gathering, the life cycle, database, development of UML diagrams and development in Java, all with the exclusive use free software, that will result in the creation of a system that performs the specifications obtained in requirements elicitation. The implemented system has met all the requirements pre-determined by the client and it can realize the need to search for new technologies by small businesses.*

Resumo. *Este artigo tem como objetivo mostrar o desenvolvimento de uma alternativa de software para empresas de vídeo locação, para isso será abordada a importância do gerenciamento de uma empresa, e então apresentar alguns tipos de sistema de informação para auxiliá-las, expor os benefícios das ferramentas CRM para fidelização do cliente e, além disso, apresentar as etapas para a criação do sistema, levando em consideração a coleta de requisitos, o ciclo de vida, Banco de Dados, elaboração de diagramas UML e o desenvolvimento na linguagem Java, tudo isso com a utilização exclusiva de softwares livres, que resultarão na criação de um Sistema que realiza as especificações obtidas no levantamento de requisitos. O sistema implementado atendeu todos os requisitos pré-determinados pelo cliente e com isso pode-se perceber a necessidade da busca de novas tecnologias pelas pequenas empresas.*

Introdução

O desenvolvimento constante da tecnologia tem levado empresas de todos os tamanhos a buscarem aperfeiçoamento e conhecimento para se manter no mercado. As pessoas já sabem o valor da informação e tendem a buscá-la cada vez com mais facilidade devido aos avanços tecnológicos.

Esses avanços tem mudado a maneira das pessoas viverem, pois estão sempre à procura da maior comodidade para sua vida. Hoje é possível realizar compras sem sair de casa, ou assistir um filme sem a necessidade de ir a uma vídeo locadora ou em um

cinema. Pode-se perceber que o impacto da tecnologia, no entanto está afastando cada vez mais as pessoas e o contato pessoal entre cliente e empresa está ficando cada vez mais subjetivo.

Com as novas tecnologias, vários tipos de negócios estão surgindo e vários outros se extinguindo ou passando por muita dificuldade, como é o caso das vídeo locadoras que não são mais tão procuradas como antigamente, fato que se deve em grande parte devido ao aumento da velocidade da internet e da pirataria. Contudo, ainda existe um público para esse tipo de negócio e para alcançá-los é necessário que as locadoras invistam na relação com o cliente.

Portanto, este trabalho tem como objetivo auxiliar as vídeo locadoras por meio de um sistema capaz de gerencia-la e por meio de relatórios gerados pelo *software*, auxiliar os empreendedores na tomada de decisões. Para isso, foi apresentada a situação atual das vídeo locadoras, os principais Sistemas de Informação e concomitantemente a isso, apresentar maneiras de se fidelizar um cliente com o sistema. Também foram abordadas as fases para o desenvolvimento de um projeto, desde a coleta de requisitos, definição do ciclo de vida como todas as etapas deste ciclo, além de apontar alguns *softwares* livres que auxiliaram o desenvolvimento do projeto como por exemplo.

Fundamentação Teórica

Em razão do rápido crescimento do uso da internet e dos avanços contínuos da tecnologia, pessoas de qualquer lugar do mundo, independentemente de nível social, têm a possibilidade de ter acesso a diversos meios de alcançar informações. Conforto e Santarosa (2002) explanam que as utilizações constantes de novos tipos de tecnologia leva a sociedade a uma realidade quase utópica, onde todos tem acesso à informação.

O mundo está vivendo uma revolução tecnológica e os efeitos desta evolução afetam todas as esferas da atividade humana, moldando as relações sociais, a economia e o avanço da ciência e tecnologia (Souto *et al.*, 2010). Um desses efeitos se dá pelo aumento da velocidade da internet, o site do IBOPE (2013) apresenta uma pesquisa mostrando que 52,5% dos usuários ativos domiciliares, cerca de 23,8 milhões, estão utilizando a banda de conexão com velocidade superior a 2 *Mega bytes* por segundo.

Devido a essa evolução e a facilidade que a tecnologia proporciona, vários tipos de negócios estão surgindo, é o caso das lojas *online*, dos canais de vídeo que oferecem conteúdo gratuito e lucram por meio de patrocínios, entre outros. Em contrapartida, vários outros tipos de negócio estão tendo que se adaptar aos novos tempos, que é o caso de empresas que fundamentalmente sobreviviam por meio da proximidade e relacionamento com o cliente. Janissek (2000) afirma que nesse processo de mudança, as empresas têm a necessidade de diferenciar-se, compartilhar informações e atuarem rapidamente de forma a obter uma vantagem competitiva.

Um caso particular é o das empresas de vídeo locação, que de acordo com Caputo e Meirelles (2010) surgiram no Brasil por volta dos anos 80 com a chegada dos videocassetes, e juntamente das fitas VHS, essa nova sensação conquistou os lares brasileiros e que se manteve no mercado nacional ao longo de quase 20 anos. Na primeira metade da década os famosos videocassetes passaram a ser populares e em 1986, ocorreu a primeira explosão de consumo, onde os aparelhos nacionais venderam

200.000 unidades. Em 1991, 25% dos 30 milhões de televisores coloridos do país contavam com videocassetes, isto resultado da redução dos preços. Com este crescimento na utilização, a indústria do vídeo tornou-se um ótimo negócio.

Conforme Alves *et. al*. (2011), as fitas VHS também deram início as cópias não autorizadas de filmes. Apesar de que quanto mais cópias eram feitas a partir de um VHS, menos qualidade de reprodução se tinha da fita, que devido a isso, não ofereceu muito perigo para vídeo locadoras. A chegada da TV por assinatura no Brasil, em meados da década de 1990, não abalou o reinado do VHS. A verdadeira ameaça veio com o *Digital Video Disk* (DVD), novidade lançada no Japão em 1996 e que chegou ao Brasil em Dezembro de 1997. Com a popularização do CD e do DVD, as VHSs foram substituídas por essas mídias digitais. Com o uso dessas novas tecnologias era possível realizar cópias digitais, que garantiam as mesmas características do arquivo original, fato que popularizou a venda de filmes Piratas.

De acordo com Cesar (2013) percebe-se basicamente, que toda a utilização não autorizada de propriedade intelectual é denominada pela indústria como pirataria, entre outros exemplos cita-se a quebra de patentes para reprodução não autorizada de fármacos, e pirataria de recursos naturais para descobertas científicas.

A Associação Antipirataria de Cinema e Música – APCM (2013) definem Pirataria como a apropriação, reprodução e utilização de obras (escritas, musicais ou audiovisuais) protegidas por direitos autorais, sem devida autorização do autor. Ela pode ser caracterizada desde a compra de CDs e DVDs falsificados, até o *download* de arquivos pela internet, a ACPM (2013) afirma que pirataria é qualificada como crime independentemente do modo como é feita, e é punida como tal e devido ao crescimento exponencial, em grande parte por causa do aumento da velocidade da internet, é necessário saber identificar e combater a pirataria.

Além da pirataria, outro fator que ameaçou o reinado das vídeo locadoras depois tantos anos foi o surgimento de locadoras online, como a NetFlix.com, que em 2007 apresentou um portal de filmes onde uma pessoa poderia escolher o filme desejado e assistir online, esses tipos de sistema estão aumentando seu número de usuários a cada dia devido a facilidade que proporciona (Netflix.com, 2013).

Com isso, percebe-se que a internet em si não foi a única responsável pela diminuição da procura de vídeo locadoras, pois como cita Alves *et al*. (2011), somente na primeira década deste século houve uma expressiva quantidade de pessoas conectadas com a internet, e a utilização era com uma velocidade de conexão muito lenta, o que fazia com que um *download* de um filme demora-se vários dias. Nesse aspecto a venda de filmes piratas contribuiu muito nessa época.

Todas estas ações atingiam potencialmente pequenas empresas de vídeo locação, que teriam poucas ferramentas para permanecer no mercado. Sabendo que a principal competência que uma empresa necessita é a capacidade de estar sempre atendo as novas tecnologias (Janissek, 2000 apud Lesca 1998), então, é imprescindível que as vídeo locadoras busquem ferramentas para se diferenciar-se no mercado.

Diante disso, uma alternativa que pode auxiliar as vídeo locadoras é um Sistema de Informação (SI), O'Brien e Marakas (2012) definem SI como um conjunto de componentes relacionados, com limites bem definidos, trabalhando em conjunto para

alcançar um objetivo em comum. Palmisano e Rosini (2003) complementam elucidando que os sistemas podem ser classificados em sistemas abertos e fechados. O sistema aberto é aquele que interage com o meio com suas ações, os sistemas abertos tendem a se adaptar para garantir sua sobrevivência, já o sistema fechado por sua vez não sofre nenhum tipo de interação com o meio.

Com base nisso, Caiçara Junior (2008) comenta que as empresas e organizações são sistemas abertos devido às suas interações e influências com o meio, e estas influências são vitais para a organização, portanto um sistema computacional para gerenciar as informações de uma empresa também é um sistema aberto.

Além disso, os SI são divididos em três componentes básicos, para Batista (2006) estas etapas são divididas em:

- Entrada de dados, que trata-se do lançamento em um sistema de dados obtido pelas atividades corriqueiras da empresa.
- Processamento é a etapa de transformação desses dados obtidos em informações uteis para a tomada de decisões.
- Saída de dados, nada mais é do que a exteriorização das informações obtidas na etapa de processamento.

Cada vez mais os SI vem adotando um papel estratégico nas empresas, exigindo que as mesmas utilizem tecnologias para efetivar contratos, auxiliar a comunicação e o atendimento com seus clientes. Dessa maneira, cabe aos administradores utilizar os sistemas de informação para conseguir agilizar o trabalho, conservando sua empresa competitiva (Santos *et al*., 2012).

Complementando Rezende (2002) afirma que a competitividade entre as empresas torna os mercados acirrados e as empresas precisam ser mais ágeis e criativas para que se mantenham na ativa. Essa necessidade de que as organizações sejam atualizadas, diante das mudanças constantes da sociedade da informação, faz com que elas também alterem e criem novos planejamentos de suas informações auxiliadas pelos recursos da Tecnologia da Informação (TI).

O SI têm papel fundamental nas empresas e organizações, pois é por meio deles que um administrador consegue ter acesso facilitado as informações de todos os aspectos de sua empresa. Para Sperb e Neto (2006) a correta administração dessas informações é de suma importância para que todas as ações da empresa sejam realizadas com êxito, pois, com base nisso os executivos tem a possibilidade de conduzir seu negócio de uma maneira mais eficaz. Concomitantemente a isto Bazzotti e Garcia (2007) explanam a ideia de que as empresas precisam ser confiáveis, versáteis, eficientes e eficazes, por isso, torna-se de grande importância à utilização da TI para melhorar o desempenho das atividades das mesmas, pois os SI têm por objetivo gerar informações para a tomada de decisões.

Podem-se classificar os SI de acordo com a sua forma de utilização e o tipo de retorno que a ferramenta dá para a tomada de decisão de uma empresa. De acordo com O'Brien e Marakas (2012) os SI podem ser classificados em Sistemas de apoio as operações (SSO) e Sistemas de Suporte Gerencial (SSG), dentro desses existem subsistemas, cada um com um objetivo.

Pode-se perceber então que todos os sistemas se encaixam em alguma classificação e muitos se encaixam em vários itens. De acordo com O'Brien e Marakas (2012) os Sistemas de Suporte às Operações (SSO) tem o objetivo de processar as transações, controlar processos industriais e atualizar o banco de dados. Este sistema é utilizado no dia a dia da empresa, porém ele ainda necessita de um sistema gerencial, ou seja, mesmo com um SSO ainda é necessário a utilização em conjunto de um Sistema de Suporte Gerencial (SSG), que registram e processam os dados que resultam de transações de negócios.

De um modo geral, as empresas utilizam vários tipos de sistemas para a realização de suas atividades, e esses sistemas devem interagir entre si, fato que reforça a ideia de que os sistemas utilizados pelas empresas são sistemas abertos (Santos *et al*., 2012). Complementando, Laudon e Laudon (2004) trazem a ideia de que a interação entre estes sistemas é muito vantajosa, pois a informação pode fluir entre diversas partes da organização, porém esta integração é um processo lento e complexo, além de gerar custos, fato que faz exigir das empresas certa ponderação desses fatores.

Com o uso de SI, as empresas terão todos os dados e relatórios com facilidade, para que possam elaborar uma política capaz de fidelizar os seus clientes e concomitantemente a isso obter mais lucro. Segundo Bee e Frances (2000), para a fidelização do cliente é essencial que as informações do sistema sejam precisas, atualizadas e acessíveis.

Este fator proporciona ao empreendimento de se relacionar com o cliente da melhor maneira possível, pois tem acesso a diversas informações sobre o cliente e suas preferências. Esta atitude é chamada de *Custumer Relationship* Management/Gerenciamento do Relacionamento com o Cliente (*CRM),* e que segundo Caiçara Junior (2008), pode ser considerado uma arquitetura que faz uso combinado dos processos de negócios e tecnologias, e que tem por objetivo entender os clientes, bem como seus desejos e seus anseios.

Caiçara Junior (2008) ainda aponta que o CRM pode ser classificado em três tipos:

- Operacional que é o momento de interação com o cliente e a coleta de seus dados, este é a base para qualquer sistema CRM.
- Analítico que são sistemas de apoio para a tomada de decisão, utilizado após o CRM operacional.
- Colaborativo que é a última fase de implantação de um CRM, momento em que se compartilha todas as informações obtidas, é neste ponto que se pode obter o benéfico máximo da utilização do conceito de CRM.

Para Taurion (2004) o desenvolvimento de um sistema, ainda é necessário levar em consideração o custo do projeto e este pode ser reduzido com a utilização de *softwares* livres, que são aqueles que qualquer desenvolvedor pode ter acesso ao seu código fonte para usar, copiar alterar e redistribuir e de maneira geral são gratuitos. O conceito de *software* livre deve ser considerado uma forma de conhecimento cientifico, e, portanto, ser submetido aos mesmos critérios que regem a disseminação das pesquisas cientificas, isto se dá no *software* por meio da liberação de seu código fonte.

Complementado isso, Costa *et al.*(2011) elucida que esses tipos de *software* necessariamente precisam apresentar quatro características chamadas de liberdades: a liberdade para executar o programa para qualquer propósito, a liberdade de estudar como o programa funciona e adaptá-lo para as suas necessidades, de redistribuir cópias de modo que você possa ajudar ao seu próximo e a liberdade de modificar o programa e liberar estas modificações, de modo que toda a comunidade se beneficiem.

Com essas considerações sobre o custo do projeto definidas, o primeiro passo para o desenvolvimento do *software* é a coleta de requisitos, Pressman (2011) afirma que a coleta de requisitos fornece o entendimento de tudo o que o cliente deseja e isto e feito analisando as necessidades do cliente, avaliando a viabilidade e negociando a melhor solução e validando a especificação e gerenciando as necessidades à medida que são transformadas em um sistema. A coleta de requisitos continua durante todo o processo de modelagem do projeto.

Os requisitos funcionais são aqueles que definem como será o comportamento do sistema e são capturados por meio de diagramas de caso de uso, que documentão as entradas, os processos e as saídas do *software*. Os não funcionais são aqueles que não são necessariamente ligados ao comportamento do sistema, tal como usabilidade, confiabilidade, performance e suporte do *software* (Martins, 2007).

A coleta de requisitos pode ser realizada por meio de reuniões com envolvidos no projeto, Martins (2007) explica que nestas reuniões são obtidas todas as solicitações de como o sistema deve ser, em seguida, com base nestas informações pode-se formular um documento contendo a visão do sistema, logo após estes requisitos são traduzidos em requisitos detalhados com o uso de ferramentas *Unified Modeling Language*/Linguagem Unificada de Modelagem (UML), e dessa maneira é possível criar uma arquitetura para o sistema.

Com os requisitos em mãos, é possível definir todo o contexto de como será elaborado o sistema, bem como seu ciclo de vida, diagramas, seu Banco de Dados, as ferramentas utilizadas para o desenvolvimento e a linguagem de programação a ser utilizada, além disso pode-se definir também o tempo que o desenvolvimento do projeto irá levar para seu desenvolvimento.

Definir o ciclo de vida é uma etapa muito importante no desenvolvimento de um projeto, pois segundo Pfleeger (2004), descreve todas as fases do projeto, desde sua concepção até a implantação, entrega, utilização e manutenção. Para o desenvolvimento do sistema da vídeo locadora, tem de se levar em consideração as possíveis mudanças que podem ocorrer no decorrer do tempo, isto exige que o sistema seja dinâmico para que quando houver necessidade seja possível realizar alterações sem dificuldade.

Levando isso em consideração, optou-se pela utilização Modelo Evolucionário para o desenvolvimento do *software*, pois permite o desenvolvimento de um produto que evolui ao longo de seu desenvolvimento. Conforme afirma Pressman (2011), os modelos evolucionários são iterativos. Apresentam características que possibilitam desenvolver versões cada vez mais completas do *software*, os dois principais modelos evolucionários são o de prototipação e o modelo espiral.

Dentro dos modelos evolucionários, optou-se por utilizar a prototipação, pois com este modelo o cliente tem a possibilidade de visualizar o sistema mesmo sem ele

estar totalmente pronto, e isto faz com que ele possa dar sugestões de como melhorar e até mesmo perceber a necessidade de novos requisitos, isto torna o desenvolvimento do sistema dinâmico e proporciona a satisfação das expectativas do cliente.

O paradigma da prototipação começa com a comunicação, uma reunião entre os envolvidos. Com todos os requisitos colhidos é realizada uma modelagem rápida do projeto, como por exemplo, o *layout* ou formulários. Então, a partir do projeto rápido, gerar protótipo, que é levado até os envolvidos para fornecer um *feedback*, que auxilia na coleta de requisitos.

Presman (2011) afirma que existem dois tipos de protótipos os construídos como descartáveis e outros evolucionários, no sentido que evoluem lentamente até se transformar no sistema real, com base nisto, o sistema desenvolvido utilizou o paradigma da prototipação evolutiva. Com o ciclo de vida definido, o próximo passo é a modelagem dos diagramas na linguagem UML.

De acordo com Booch e Rumbaugh (2005), o UML é uma linguagem gráfica para a visualização, especificação, construção e documentação de projetos de sistemas de *softwares*. Por ser um modelo internacional, a UML permite a utilização de uma forma padrão para o desenvolvimento de um projeto, tanto nos aspectos conceituais, como processos de negócio e funções do sistema e também itens concretos como classes e linguagem de programação utilizada.

UML é uma linguagem de modelagem e é utilizada para desenvolver o projeto de um *software*, este poderá ser programado em qualquer linguagem, pois a modelagem definida será a mesma. Lima (2011) faz um comparativo entre a programação e a construção civil, onde o UML seria as plantas e maquetes utilizadas para representar os edifícios.

Para Melo (2011), o UML disponibiliza de cerca de 14 diagramas para auxiliar no desenvolvimento de um sistema, e estes são divididos em duas categorias, diagramas estruturais ou estáticos e diagramas comportamentais ou dinâmicos. Os diagramas estruturais mostram características do sistema que não mudam com o tempo e os diagramas comportamentais mostram como o sistema responde às requisições ou como este evolui durante o tempo. Porém, para o desenvolvimento de um projeto de *software* é interessante utilizar no mínimo três desses diagramas, o diagrama de caso de uso, diagrama de classe e diagrama de sequências.

O diagrama de caso de uso são utilizados para demonstrar a uma pessoa sem conhecimento de desenvolvimento de sistemas, o funcionamento do mesmo, além disso com sua ajuda uma pessoa leiga consegue entender as tarefas que o sistema irá realizar, o diagrama é composto por atores, ações e pelos relacionamentos dos atores com as ações, ou seja um caso de uso captura as interações que ocorrem entre produtores e consumidores de informação e o sistema em si (Pressman, 2011).

Conforme apresenta Lima (2011), o diagrama de classes é um dos diagramas mais utilizados da UML, pelo qual se pode representar todas as classes de Objetos de um sistema. É extremamente útil quando ao se pensar em uma modelagem Orientada a Objetos, pois oferece uma noção do funcionamento das classes.

O diagrama de sequência é utilizado para representar os eventos que irão acontecer em determinados casos de uso. Pressman (2011), explica que o diagrama de

sequência indica como os eventos provocam transições de objeto para objeto, ou seja, representa a comunicação exposta nos casos de uso.

Para o desenvolvimento dos diagramas UML é adequado a utilização de *softwares*, uma boa opção é um *software* chamado *Astah Community*, pois este permite a criação dos mais variados tipos de diagramas além de fornecer diversas ferramentas para a formatação permitindo criar diagramas de forma simplificada e profissional, além de ser de código aberto (Astah, 2010).

Posteriormente ao desenvolvimento dos diagramas, é a hora de se pensar no armazenamento as informações do sistema, existem diversos tipos de banco de dados no mercado, porém para o desenvolvimento do sistema optou-se pela utilização do MySQL pois ele tem o código aberto e atualmente é um dos sistemas de Banco de Dados mais utilizados no mundo. Além disso, com a utilização do MySQL, pode-se utilizar o *Workbench*, uma ferramenta utilizada para o desenvolvimento do esquema do banco de dados (MySQL, 2013).

O MySQL é uma linguagem de Banco de Dados que tem como base o *Structured English Query Language*/ Linguagem de Consulta Estruturada (SQL) que é um padrão no mundo dos ambientes de Banco de Dados. O SQL é uma linguagem que trabalha basicamente com tabelas e relações entre elas, este modelo é chamado de relacional. Todas as tabelas obrigatoriamente tem um campo identificador chamado de chave primaria, e esta chave garante que o registro seja único no Banco (MySQL, 2013).

Pensando no desenvolvimento do projeto optou-se utilizar a linguagem de programação Java visto que é executada hoje em cerca de 1,1 bilhão de *desktops* e são feitos em torno de 930 milhões de *downloads* por ano do *Java Runtime Environment*, utilizada para rodar aplicativos Java. Sabe-se também que o Java é uma linguagem livre e tem uma grande comunidade de desenvolvedores em todo o mundo (Java, 2013).

Como relata Arnold *et al.*(2007), a linguagem Java foi projetada para maximizar a portabilidade, uma vez que o código Java é executado em uma máquina virtual e a mesma pode ser disponibilizada em diversos tipos de dispositivos. Esta máquina virtual gerência toda a execução dos aplicativos bem como a segurança dos mesmos, o que a torna uma linguagem confiável e robusta.

Conforme Deitel (2010), a plataforma Java teve seu início em 1991 com um projeto de pesquisa financiado pela *Sun Microsystems*, que resultou em uma linguagem baseada em C++. Seu desenvolvedor, James Gosling inicialmente deu o nome à linguagem de Oak, mas esta, já se designava a outra linguagem de computação. O nome Java foi sugerido em 1995, sendo o ano de lançamento de sua primeira versão, enquanto a equipe de desenvolvimento da Sun visitava uma cafeteria local, pois Java é a cidade de origem de um tipo de café importado.

A linguagem Java é essencialmente Orientada a Objetos (OO), pois nela todos os Objetos criados ou instanciados estendem-se à classe *Object*. Segundo Lima (2011), este paradigma é um meio de prevenir erros, e também de facilitar a manutenção, surgiu por volta de 1960 com as linguagens Simula e *Smalltalk* e tem como objetivo trazer para as linguagens de programação uma abstração do mundo real. A OO não auxilia somente com a reutilização de código, mas também de requisitos, análise, projeto e

especificação, o que faz com que se tenha uma maior produtividade e desempenho nas aplicações.

Tendo como base que a linguagem Java é totalmente orientada a objetos, e o banco de dados MySQL é relacional, optou-se pela utilização de um *framework* para a persistência dos dados, o Hibernate. Para Kraemer e Vogt (2004) este *framework* permite a utilização de todos os paradigmas de OO em um Banco de Dados relacional, isto é, ele transforma os registros do banco de dados em objetos e vice versa, além de oferecer um maior desempenho e ter uma grande confiabilidade, além disso, por utilizar o Banco de Dados Relacional.

Kraemer e Vogt (2004) ainda explicam que o Hibernate usa arquivos XML para o mapeamento das classes em tabelas do Banco de Dados e com isso persiste *Plain Old Java Objects* (POJOS), também conhecidos como *beans*, que são classes que estão associadas a uma tabela do banco de dados e não possuem lógica em seu código, possuem um construtor sem parâmetros e os atributos mapeados possuam métodos *get*, utilizado para pegar o valor de um atributo e *set* que é utilizado para dar valor a um atributo.

Outro fator muito importante para o desenvolvimento de um sistema é a *Integrated Development Environment*/Ambiente de Desenvolvimento Integrado (IDE) utilizado, para o desenvolvimento do projeto optou-se pelo *Netbeans*, pois oferece suporte às mais novas tecnologias Java, possuindo também um editor de código rápido e inteligente fornecendo modelos de código, dicas de codificação e ferramentas de fatoração, provendo também de uma ferramenta para detecção de erros no código. Concomitantemente a isso, o desenvolvimento da interface gráfica fica simplificado, pois se pode arrastar e posicionar os componentes de uma paleta no editor para a tela projetada. O *Netbeans* possui uma grande comunidade de desenvolvedores que estão sempre criando *plug-ins* e *APIs* para facilitar o desenvolvimento (Netbeans, 2013).

Para o desenvolvimento dos relatórios, utilizou-se o *JasperReports* e para exibição dos relatórios o *iReport*. Conforme Jaspersoft (2013) são um dos mais populares mecanismos para geração de relatórios para o mundo *Open Source*. O *JasperReports* foi desenvolvido por Teodor Danciu utilizando a linguagem Java e é capaz de utilizar qualquer fonte de dados para a geração de relatórios, porém para utilizar o *JasperReports* é necessária outra ferramenta, o *iReport*, que é um editor visual dos relatórios, foi desenvolvido em 2002 por Giulio Toffoli, um usuário do *JasperReports* e membro da comunidade de código aberto.

Além destas ferramentas, também utilizou-se uma biblioteca para armazenar a foto do cliente no cadastro, o webcam-capture (2013) tem como objetivo fornecer uma biblioteca para acesso a câmeras diretamente do código Java de forma simplificada, ele fornece uma interface *Webcam* e vários construtores adicionais para utiliza-lo conforme a necessidade do sistema.

Devido a quantidade de campos de data presentes no sistema, e visando uma melhor usabilidade ao cliente utilizou-se a biblioteca *JCalendar*. Segundo Toedter (2013) está biblioteca fornece vários tipos de campos para se trabalhar com data graficamente, e isto simplifica e dá mais segurança na manipulação das datas.

Além disso, foi levado em consideração o aspecto visual do sistema, e com essa finalidade foi utilizada a biblioteca *Easy Look-and-Feel,* que de acordo com Code (2013) fornece grande facilidade ao se trabalhar com os temas das telas, e sua utilização depende de apenas uma linha de código.

Considerando também a organização e estruturação do código optou-se por utilizar o modelo *Model-View-Controller* (MVC) em qual, segundo Zanini (2006) divide-se o sistema em três camadas, de *Model* /Modelo que contém a as regras de negócio do sistema, o *View* /Visão que define a interface exibida ao usuário, ou seja, as telas do sistema, e o *Controller* /Controlador que é o mediador entre as regras de negócio e a interface. Sampaio (2007) complementa dizendo que o padrão MVC pode ser utilizado por qualquer tecnologia, no caso do desenvolvimento em Java com a utilização do pacote *Swing, este* modelo já é implementado por padrão.

Etapas do desenvolvimento

Para um *software* de uma vídeo locadora, deve-se pensar em uma ferramenta que possua características de um sistema de operações, quando houver a interação entre os funcionários e os clientes por meio de cadastro, empréstimos e devoluções, e além disso, deve possuir características de um sistema analítico, fornecendo relatórios para que o gestor do empreendimento possa tomar as melhores decisões para seu negócio.

Deve-se tomar por base todos os itens já citados, e levando em consideração a situação atual das vídeo locadoras, deve-se desenvolver um sistema que tenha um custo reduzido. Tendo isto como base, fez-se a utilização de *softwares* gratuitos para seu desenvolvimento.

Ao desenvolver um *software*, a primeira preocupação que existe é a da coleta de requisitos, então houve necessidade de se realizar uma reunião com o cliente para a definição dos requisitos gerais do sistema e do propósito do desenvolvimento. Como resultado desta reunião, obteve-se um documento contendo os principais requisitos para o começo do desenvolvimento do sistema que pode ser visualizado no Apêndice A. Além desta reunião inicial formal, também foram realizadas conversas informais para o levantamento de requisitos, concomitantemente a isto realizando observações em loco.

Com isto foi decidido que o sistema seria desenvolvido seguindo o ciclo de vida de prototipação, na linguagem Java, utilizando o paradigma de orientação a objetos para a programação e a IDE escolhida para o desenvolvimento foi o *Netbeans* com a utilização do *framework Hibernate* para persistência dos dados, além da necessidade de se obter a foto do cliente no momento do cadastro.

Visando um melhor entendimento e organização do sistema, optou-se por utilizar o MVC, que é um padrão de projeto que separa o sistema em três partes, o *Model* gerencia a troca de informações com o banco de dados, a *View* é a representação gráfica do sistema e o *controler* é responsável por realizar a comunicação entre visão e Modelo. No sistema todas as classes foram separadas em pastas, as *Views* estão na pasta *gui,* os *Models* na pasta *bean* e os *Controllers* na pasta *DAO.*

A primeira etapa da prototipação é a análise, para isso foram realizadas diversas conversas com o cliente, e pode-se observar a rotina da vídeo locadora, com todos os processos de funcionamento, com isto pode-se perceber a fragilidade do sistema com

que a empresa vinha trabalhando, sistema este que não oferecia nenhum tipo de relatório, além de ser um sistema muito antigo. Com base nas deficiências desse sistema foram definidas os principais objetivos do sistema a ser desenvolvido.

Para o melhor entendimento do sistema, foi desenvolvida uma versão inicial do *software*, que auxiliou tanto o cliente para expor suas opiniões e necessidades, quanto para o desenvolvedor, que pode refinar o sistema a cada interação. Além disso, a prototipação permitiu algumas alterações, pode-se perceber na Figura 1 que neste primeiro protótipo o sistema dispunha de quatro funções principais, são elas empréstimo, devolução, renovação e reserva, no entanto, em conjunto com o cliente, percebeu-se que a função de renovação não seria necessária, segundo o dono da empresa, devido à pouca ocorrência deste tipo de operação.

Neste primeiro protótipo pensou-se ainda na utilização de apenas um *Frame* para o desenvolvimento das telas, isto seria feito com o auxílio da *IDE Netbeans* que fornece ferramentas para o *layout* das telas, seria então utilizado o *layout* de cartão, que proporciona a troca dinâmica dos painéis adicionados a tela. Esta ideia inicial foi alterada e para uma melhor visualização das funcionalidades optou-se por utilizar um *Frame* principal e cada função em uma nova tela, para esta funcionalidade foi utilizado as classes *JFrame*, para a criação da tela principal, e do *JDialog* para as telas de interação do sistema.

Figura 1. Primeiro Protótipo do Sistema

A próxima etapa do desenvolvimento foi a criação dos diagramas de caso de uso, classe e sequência com base na coleta de requisitos que podem ser visualizados nos Apêndices B, C e D. A elaboração desses diagramas foi fundamental para o desenvolvimento do sistema, pois foi a base para a criação de todas as classes e funções do sistema. A visualização representada pelos diagramas oferece ao programador uma grande facilidade para absorver o que foi decidido na coleta de requisitos. Para elaboração desses diagramas foi utilizado o *software Astah Community 6.7.0* que permitiu a criação de diagramas com grande facilidade.

Com os diagramas prontos foi então projetado o esquema do Banco de Dados, ou seja, todas as tabelas e as relações entre elas, disponível no Apêndice E. Para este fim foi utilizado o sistema *MySQL Workbench 6.0 CE,* que permite a criação da modelagem do Banco. Uma das grandes vantagens deste *software* é que ele consegue se conectar com o banco de dados e sincronizar o esquema com o banco real, poupando muito tempo no desenvolvimento do banco de dados. O Sistema de Gerenciamento de Banco de Dados (SGBD) utilizado foi o *MySQL*, por ser compatível com todas as outras ferramentas utilizadas no desenvolvimento. A modelagem foi elaborada com nove tabelas, e elas podem ser divididas em quatro categorias,

- Cadastro de Cliente: tabela cliente e endereco.
- Cadastro de funcionário: tabela funcionario
- Cadastro de Material: tabela material, genero, tipoMaterial e tipoEmprestimo.
- Cadastro de operações: tabela emprestimo e reserva.

Depois de criado o banco de dados, iniciou-se a fase de programação, para a manipulação do banco de dados pelo sistema foi utilizado o *framework Hibernate*, que permite que um sistema em Java crie uma conexão com o SGBD e crie ou atualize suas tabelas por meio de *anotations/*anotações no código fonte das classes principais.

A conversão dos dados do SGBD para os objetos é realizado pelas classes *Data Access Object* (DAO), por elas são realizadas todas as consultas ao banco de dados com a biblioteca *Criteria*, presente no *Hibernate*, ferramenta que permite a consulta ao SGBD sem a utilização da linguagem SQL, e sim com a utilização de métodos das classes.

Além disso, outra ferramenta que o *Hibernate* disponibiliza é a de engenharia reversa, com ela é possível criar os POJOS, classes que tem uma tabela no banco de dados, por meio da sincronização com o banco de dados. Para este processo o primeiro passo foi a criação do banco de dados com o *MySQL Workbench 6.0 CE,* depois disso foi necessário criar uma conexão com o banco de dados e o sistema, isso foi feito com a criação do documento *hibernate.cfg.xml*, que possui todas as configurações para conexão com o SGBD.

Em seguida, com a conexão estabelecida, foi criado um documento chamado *hibernate.revang.xml*, este documento é responsável por listar todas as tabelas do SGBD automaticamente. A última etapa da reengenharia do *Hibernate* se dá pela execução do método "Arquivos de Mapeamento e POJOs de Banco de Dados", nesta etapa o *Hibernate* se conecta com o SGBD e, com o auxílio do *hibernate.revang.xml*, cria todas as principais classes no projeto Java, com todos os *geters* e *seters e anotations* para o relacionamento das tabelas.

Com a conexão do sistema efetuada, e a criação das principais classes do projeto, inicia-se a última etapa do modelo de prototipação, que é a refinação do sistema até o aceite do cliente. No início desta etapa foram criadas primeiramente o *Frame* principal do Sistema e as telas de *login*, de cadastro de cliente, de funcionário, de material, de gênero, de tipo de material e de tipo de empréstimo.

A tela principal foi alterada em relação a do primeiro protótipo, como mostra a figura 2, ela apresenta uma barra de menus, contendo as opções Sistema, Relatórios, Materiais e Cadastros, logo abaixo existem três botões, com as principais funcionalidades do sistema, empréstimo, devolução e reserva, e em seu rodapé o sistema apresenta o nome do usuário que acessou o *software*. Para o desenvolvimento desta tela utilizou-se um *JFrame* padrão, e por meio deste *Frame* todas as outras telas são acessadas.

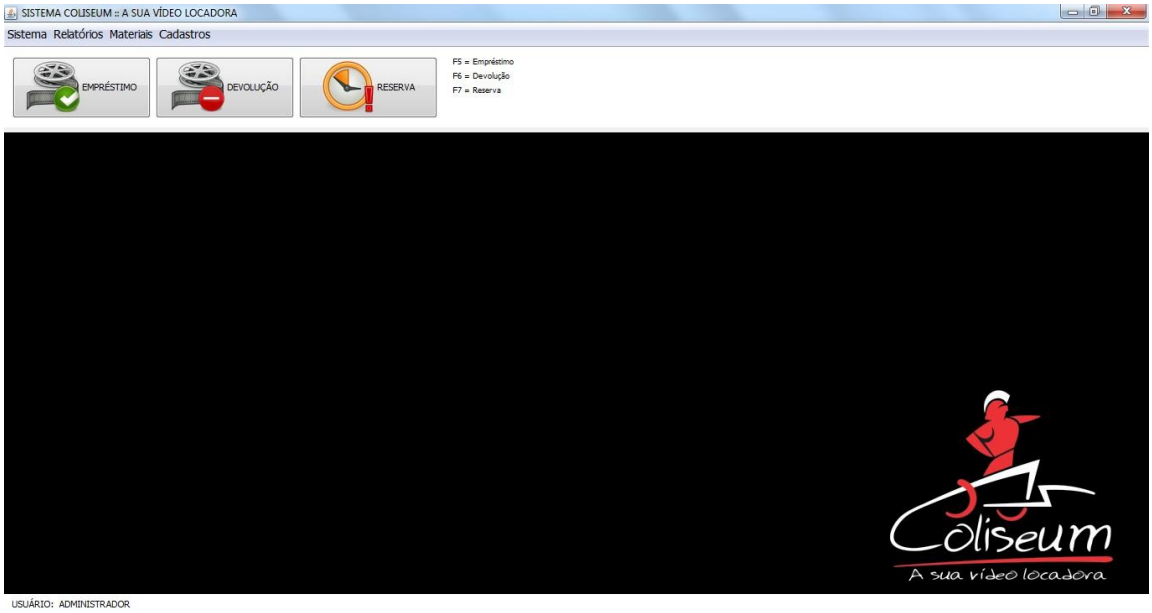


Figura 2. Tela Principal

Na criação das telas houve uma grande preocupação com a usabilidade, e para facilitar a operação foram adicionadas teclas de atalhos para estas funções principais, a instrução dos atalhos se encontra ao lado dos botões principais, os atalhos são;

- F5: abre a tela empréstimo
- F6: abre a tela devolução
- F7 abre a tela renovação

A primeira chamada do sistema é para a classe "TelaPrincipal" que contém o método *main(),* o método principal do todo aplicativo Java, no entanto, ela não é exibida de imediato e sim um *JDialog* para o *login* onde existe um método chamado *login()* que retorna uma variável de verdadeiro e falso, dependendo do resultado o *Frame* principal será exibido ou não, para isto foi elaborado a seguinte função no construtor do *JFrame* principal;

```
if (login.isLogin()) {
        funcionario = new Funcionario();
        funcionario = login.getFuncionario();
        initComponents();
}
```

Esta função verifica se a saída do método *login*, método que verifica se o usuário e a senha estão corretos, é verdadeira para o início dos componentes do *Frame* principal.

O próximo passo foi a criação das telas de cadastro, é importante citar que todas essas telas apresentam um método construtor diferenciado onde são passados, além dos valores comuns ao *JDialog*, um valor determinando o tipo da tela, ou seja se ela é de edição ou de um novo cadastro, e um valor com um objeto do tipo do cadastro caso seja uma tela de edição.

A primeira a ser desenvolvida foi a tela de cadastro de clientes, esta tela apresentou um dos maiores desafios impostos pelo levantamento de requisitos, a possibilidade de obter a foto do cliente no sistema. Para este método foi utilizada uma biblioteca para a linguagem Java chamada *webcam-capture,* que permite o acesso a *webcams* ligadas no USB pelo Java.

Foi então criado um *Frame* especial para a captura das fotos, onde foi adicionado um *WebcamPanel,* uma espécie de *JPanel* que permite a visualização da imagem da câmera*,* também é adicionado a classe *Webcam* que tem os métodos de controle. Ao tirar uma foto é executado um método que pega a imagem do *WebcamPanel,* transforma ela em *bits* e a envia para a tela de cadastro de clientes novamente pelo método *getFoto()* presente na tela da câmera. Ao salvar os dados do cliente, a foto é gravada em uma pasta do sistema, e o nome atribuído a foto é o número do código do cliente.

Depois disso, foram criadas as telas de cadastro de funcionário, onde foi necessário a utilização de um campo de senha *jPasswordField* e da seleção do tipo de usuário, ou seja, se o usuário do sistema será administrador ou funcionário. Dependendo do tipo de usuário, ao fazer o *login* no sistema, é exibido um conteúdo diferente nos menus, o administrador tem acesso a todas as funções do sistema, já o funcionário não tem acesso ao cadastro de funcionários e alguns relatórios. Este item também está presente na coleta de requisitos.

Feito isso, foram desenvolvidas as telas de cadastro de gênero, de tipo de empréstimo e de tipo de material, depois desses foi criado o de cadastro de materiais, onde é possível todos os outros cadastros referente ao material dinamicamente e esses novos cadastros estarão disponíveis automaticamente para a seleção. Para todos os tipos de cadastros foram criados *Frames* exibindo uma lista de todos os itens em uma tabela, e nessas listas é possível alterar os dados.

Com todos os cadastros terminados, a próxima etapa foi o desenvolvimento da tela de empréstimos, devolução e renovação, estas foram as telas que demandaram mais tempo de desenvolvimento, pois foi necessário criar um método de busca a partir de uma caixa de seleção, e de uma tabela que pudesse ser atualizada em tempo real.

O sistema de busca foi criado da seguinte maneira, primeiramente foi criada uma classe DAO genérica, e esta classe tem a função de buscar no banco de dados pelo nome. Foi criada então uma classe *JAutoCompletaComboBox* que é herdada de *JComboBox,* e nela foram desenvolvidos métodos para mostrar sugestões dependendo do que é escrito na busca. Com isso foi possível desenvolver um sistema de busca que pode ser usado em várias telas onde a busca por nome era necessária.

Para a criação das tabelas foi utilizado a classe padrão de tabelas do Java, a *DefaultTableModel.* No entanto, para a criação das tabelas não foi possível criar uma genérica como no campo de busca devido a diversidade de informações e objetos que cada tabela necessita. Pensando nisto, foi criado então um método para cada tabela do sistema, este método consiste na criação de um modelo de tabela que é adicionado a uma *JTable*.

Primeiramente é necessário definir o nome das colunas da tabela e depois é adicionar os valores de cada linha, para isso é necessário receber uma lista de objetos e percorre-la para adicionar todos os valores. Ou seja, enquanto a lista "objetos" apresentar um material, a *DefaultTableModel* adicionará a linha a tabela. Esse processo cria um modelo de uma tabela que é aplicado a uma tabela do sistema pelo método *setModel()* e a cada atualização da tabela será criado um novo modelo e excluído o antigo.

```
model.setColumnIdentifiers(new String[]{"Col 1", "col 2"});
for (Objeto objeto : objetos) {
    DefaultTableModel.add(objeto);
}
```

Na tela de empréstimo, primeiramente é necessário a busca de um cliente e a busca e seleção dos materiais, na seleção dos materiais o sistema captura o objeto buscado na pesquisa e adiciona na tabela. Ao realizar um empréstimo são verificados todos os dados, e atualizada a quantidade em estoque do material emprestado, além de adicionar o valor da dívida ao cliente.

Na tela devolução, após a escolha do cliente o usuário terá uma tabela com os itens emprestados, poderá então fazer a seleção dos materiais a serem devolvidos. Será também executado um método para o cálculo da multa, que tem o valor definido no tipo de empréstimo, a tela também apresenta o *JAutoCompletaComboBox* para a pesquisa de cliente, quando um cliente for selecionado a tabela com os materiais emprestados será preenchida.

A tela de reserva é parecida com a de empréstimo, porém, com a função principal diferente, a função só será efetuada se não houver mais nenhum exemplar disponível do material e depois de feito, este só poderá ser emprestado para outra pessoa se for retirado da lista de reservas.

Para uma melhor usabilidade e ganho no visual da interface foram utilizadas as bibliotecas *EasyLookAndFeel,* para que o tema das telas se adapte ao tema do sistema operacional, isto foi feito apenas utilizando o método *Start()* no método *main()* do sistema, e a biblioteca *JCalendar*, que fornece campos específicos para se trabalhar com datas, fator muito importante no sistema devido a quantidade de dados deste tipo.

Com todas as funções prontas, foram desenvolvidos então os relatórios, estes foram criados com a ajuda da biblioteca *JasperReports* e do editor de relatórios *iReports*. A utilização desta ferramenta necessitou de um novo tipo de conexão com o Banco de Dados independente do *Hibernate*, Devido a esse fato foi criado uma classe chamada *ConnectionFactory,* que faz a conexão com o banco de dados. Todas as consultas que geram os relatórios se utilizam da linguagem SQL por meio de *Querys.*

Resultados

Com o desenvolvimento deste projeto pode-se perceber o impacto sofrido pelas vídeo locadoras com o crescente desenvolvimento das tecnologias e do aumento da velocidade da internet, também foi possível verificar que a pirataria teve grade parte nesse impacto, tudo isso fez com que a procura pelas vídeo locadoras diminuíssem drasticamente.

Também se notou a necessidade de que as empresas tenham boas ferramentas para gestão do seu negócio, que possam auxiliar os empreendedores na hora da tomada de decisão com base em dados. Também foi visto alguns tipos de Sistema de Informação e algumas ferramentas para ajudar a fidelizar o cliente por meio de Sistemas.

Foi observado a necessidade da criação de um *software* para o gerenciamento de vídeo locadora que oferecesse ferramentas de auxílio a gerência, fornecendo relatórios para a tomada de decisões, e além disso que apresentasse um custo reduzido para implementação, por isso foram utilizados apenas *softwares* livres no projeto.

Para o desenvolvimento do sistema, observou-se a importância da coleta de requisitos para o desenvolvimento, pois é nesta hora que todos os anseios do cliente são expostos. A coleta de requisitos deve captar o máximo as necessidades do cliente sobre o projeto, para que este possa sair com qualidade e satisfazer o cliente.

A utilização do ciclo de vida de prototipação foi outro fator importante neste projeto, pois foi possível constatar a simplicidade deste método, que proporciona um desenvolvimento sem maiores problemas, pois permite um refinamento gradual do sistema e proporciona que o produto final saia exatamente de acordo com as expectativas do cliente.

Também observou-se a importância do desenvolvimento dos diagramas UML para o desenvolvimento do *software*, pode-se notar que eles são a base para todo o *software* pois auxiliam o desenvolvedor na hora da programação fazendo com que o foco do sistema não se perca. Neste projeto foram utilizados os diagramas de caso de uso, de classe e de sequência.

Percebeu-se também um aumento significativo no conhecimento no que diz respeito a linguagem de programação Java, Banco de Dados e todas as bibliotecas utilizadas para o desenvolvimento do sistema além do conhecimento adquirido sobre os tipos de sistema de informação e todas as ferramentas de auxílio a tomada de decisão.

Com a implementação do sistema pode-se perceber um ganho substancial para a tomada de decisões da empresa, pois por meio dos relatórios e dos gráficos o usuário tem a possibilidade de visualizar os dados de forma simplificada e com isso tomar a melhor decisão para empresa e, além disso, buscar sempre o que o cliente procura.

Considerações Finais

A evolução constante da tecnologia tem levado pequenas e grandes organizações a buscarem novos caminhos para a sobrevivência, em um mundo onde a tecnologia da informação tem feito parte da vida das pessoas.

Pode-se avaliar neste trabalho o impacto dessa evolução nas vídeo locadoras e propor uma possível solução desenvolvendo um sistema para o gerenciamento, e com

isto fornecer meios para o empreendedor sobreviver aos novos tempos, onde por meio de relatórios e gráficos é facilitada a tomada de decisão

Durante o desenvolvimento do sistema observou-se a importância de cada fase no processo de desenvolvimento de um sistema, desde sua coleta de requisitos até a implementação, também foi possível observar um crescimento significativo no aprendizado a respeito dos itens abordados.

Por fim, a implementação do sistema foi de acordo com o ciclo de vida estipulado, a prototipação, sendo que este ciclo permite uma nova coleta de requisitos do cliente e um nova interação para cada fase do projeto, isso faz com que o sistema seja moldável a qualquer nova necessidade do cliente.

Referências

Alves, S.; Dias, P. I. R. C.; Nogueira, A. R. R.; Figueiredo, K. F. (2011) "Webfilmes: Aluguel de Filmes em tempo de Pipoca Virtual", In: Tecnologias de Administração e Contabilidade, Curitiba-PR, V. 1, Nº 2, p.68-85, Jul./Dez.

Arnold, Ken; Gosling, James; Holmes, David. A linguagem de programação Java. Tradução: Maria Lúcia Blanck Lisboa. 4. ed. Porto Alegre: Bookman, 2007.

Associação Antipirataria de Cinema e Música (ACPM). (2013) "Pirataria na Internet", http://www.apcm.org.br/pirataria_internet.php, Setembro.

Astah (2013) http://astah.net/, Novembro.

Bazzotti C.; Garcia E. (2007) "A importância do sistema de informação gerencial para tomada de decisões". VI Seminário do Centro de Ciências Sociais Aplicadas, UNIOESTE, Cascavel.

Batista, E. O. (2006) "Sistemas de Informação: o uso consciente da tecnologia para o gerenciamento." Saraiva, São Paulo – SP.

Bee R.; Frances. (2000) "Fidelizar o cliente". Tradução Edite Sciulli, Nobel, São Paulo – SP.

Booch, G.; Rumbaugh, J; Jacobson, I. (2005) "UML: guia do usuário". Tradução: Fabio F. da Silva e Cristina de A. Machado. Elsevier, Rio de Janeiro – RJ.

Caiçara Junior, C. (2008) "Sistemas Integrados de Gestão - ERP: uma abordagem gerencial". 3. ed. rev. e atual., Ibpex, Curitiba-PR.

Caputo, E. F.; Meirelles, D. S. (2010) "Dinâmica tecnológica e evolução do modelo de negócios de vídeo locadoras". In: XIII SEMEAD (Seminário de administração), São Paulo – SP.

Cesar, D. J. T. (2013) "A Cultura da Cópia: estudo sobre o Compartilhamento de Arquivos e a Prática da Pirataria Virtual", Universidade de Brasília, Brasília,.

Code (2013) "Easy Look-and-Feel". https://code.google.com/p/easylookandfeel/, Outubro.

Conforto, D.; Santarosa, L. M. C. (2002) "Acessibilidade à Web: Internet para todos", In: Revista de Informática na Educação: Teoria, Prática – PGIE/UFRGS. V. 5 Nº 2, p. 87-102. Nov.

Costa M. M. A.; Cruz T. B.; Lopes M. M. (2011) “Os impactos da utilização de Softwares Livres na comunicação social das organizações, tanto no âmbito acadêmico quanto no mercado de trabalho” UFMG, Belo Horizonte.

Deitel, Harvey; Deitel, Paul. Java como Programar. Tradução: Edson Furmankiewicz; revisão técnica : Fabio Luis Picelli Lucchini. 8.ed. São Paulo: Pearson Prentice Hall, 2010.

IBOPE (2013) Disponível em: <http:// www.ibope.com.br>, Junho.

Janissek, R. (2000) “A Influência da Internet em Negócios Empresariais: Identificação e caracterização de elementos para análise de sites” UFRGS/URI, Porto Alegre – RS.

Jaspersoft (2013) Disponível em: http://community.jaspersoft.com/project/jasperreports-library, Junho.

Java (2013) www.java.com/pt_BR/ Outubro.

Kraemer, F.; Vogt,J. J. (2004) “Hibernate, um Robusto Framework de Persistência Objeto-Relacional.” Porto Alegre.

Laudon, K. C.; Laudon, J. P. (2004) “Sistemas de informação gerenciais”, 5. ed. Prentice hall, São Paulo-SP.

Lima, Adilson da Silva; (2011) “UML 2.3: do requisito à solução”, Érica, São Paulo-SP.

Martins, J. C. C. (2007) “Técnicas para gerenciamento de projetos de software”. Brasport, Rio de Janeiro – RJ.

Melo, A. C.. Desenvolvendo Aplicações com UML 2.2. 3. ed. Rio de Janeiro: Brasport, 2011.

MySQL (2013) www.mysql.com/, Novembro.

Netbeans (2013) netbeans.org/, Outubro.

Netflix (2013) Disponível em: <http://www.netflix.com >, Junho.

O’Brien J. A.; Marakas G. M; (2012) “Administração de sistemas de informação”. 15. ed. - Mcgraw Hill, Porto Alegre - RS.

Palmisano A.; Rosini A. M. (2003) “Administração de Sistemas de Informação e a Gestão do Conhecimento”. Pioneira Thomson Learning, São Paulo – SP.

Pfleeger, S. L. (2004) “Engenharia de Software: teoria e prática”. Tradução: Dino Franklin, 2. ed. Prentice Hall, São Paulo – SP.

Presman, R. (2011) “Engenharia de software: Uma abordagem profissional.” Tradução: Ariovaldo Griesi, Mario Moro Fecchio; revisão técnica Reginaldo Arakaki, Julio Arakaki, Renato Manzan de Andrade. 7. ed., AMGH, Porto Alegre - RS.

Rezende, D. A. (2002) “Tecnologia da Informação Integrada à Inteligência Empresarial.”, Atlas, São Paulo - SP.

Santos A. S.; Quatrin D. R.; Pinto L. M.; Stefanan A. A.; Costa V. M. F. (2012) “A Importância de Sistemas de Informação em Pequenas Empresas: um Estudo de Caso em uma Agência de Publicidade.” IX Simpósio de Excelência em Gestão e Tecnologia.

Sampaio, C. (2007) “Guia do Java Enterprise Edition 5: desenvolvendo aplicações corporativas”. Brasport, Rio de Janeiro-RJ.

Souto, A. A.; Cavalcanti, D. B.; Martins, R. P. (2010) “Um plano nacional para banda larga: O Brasil em alta velocidade” Ministério das comunicações.

Sperb, Chaiana Christine; NETO, Hercio Menegotto Ferraro. (2006) “A importância dos sistemas de informação na Gestão de empresas.” Minas Gerais.

Taurion, C. (2004) “Software Livre: potencialidades e modelos de negócio”. Brasport, Rio de Janeiro – RJ.

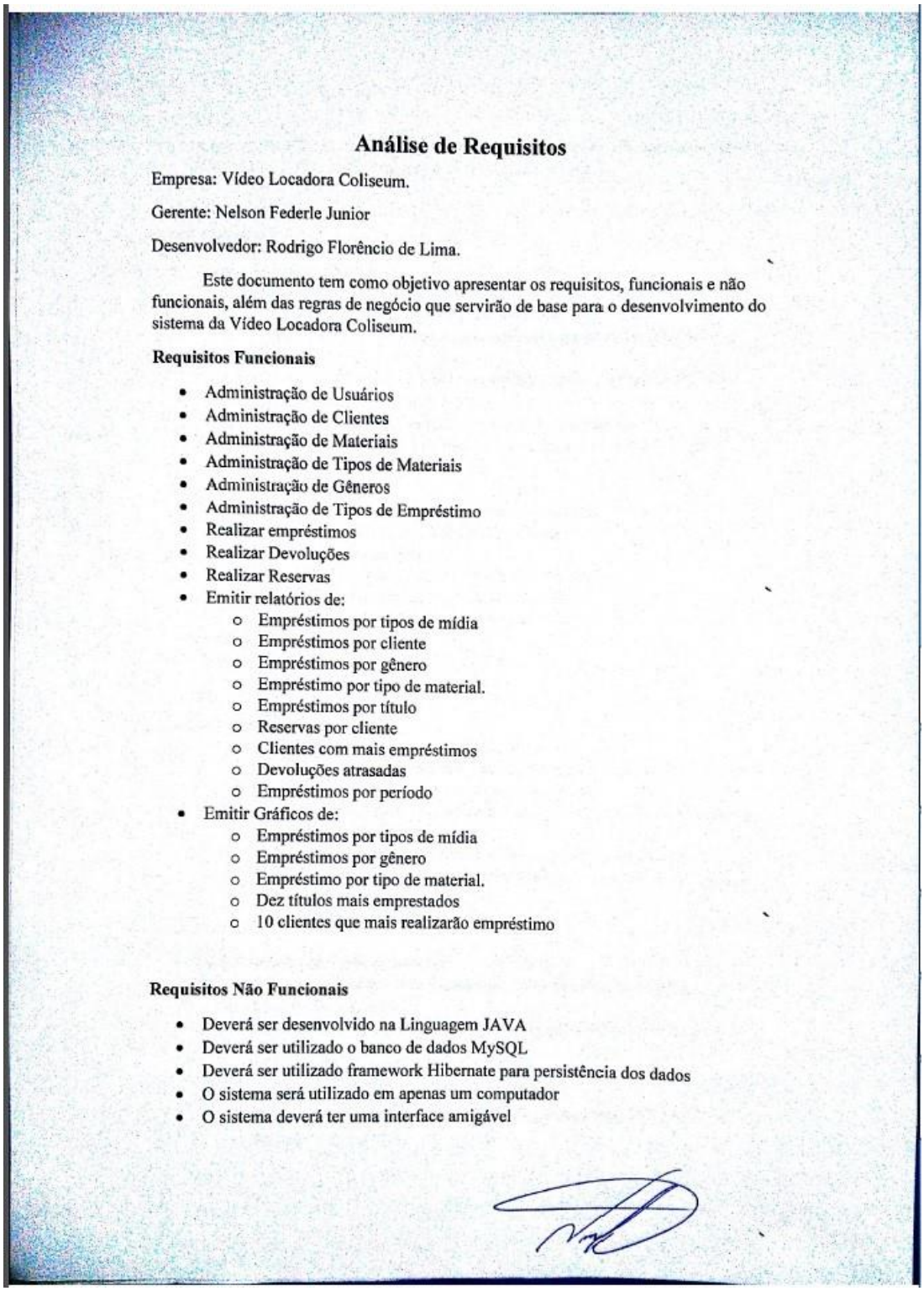
Toedter (2013) “JCalendar” Disponível em: <http://www.toedter.com/en/jcalendar>, Outubro.

Webcam-Capture (2013) “Webcam Capture: generic webcam Java utility” Disponível em: < http://webcam-capture.sarxos.pl/>, Outubro.

Zanini, M. (2006) “Framework de desenvolvimento de aplicações Java para Desktop”. Universidade Federal de Santa Catarina, Florianópolis – SC.

Apêndices

Apêndice A: Análise de requisitos



Análise de Requisitos

Empresa: Vídeo Locadora Coliseum.

Gerente: Nelson Federle Junior

Desenvolvedor: Rodrigo Florêncio de Lima.

Este documento tem como objetivo apresentar os requisitos, funcionais e não funcionais, além das regras de negócio que servirão de base para o desenvolvimento do sistema da Vídeo Locadora Coliseum.

Requisitos Funcionais

- Administração de Usuários
- Administração de Clientes
- Administração de Materiais
- Administração de Tipos de Materiais
- Administração de Gêneros
- Administração de Tipos de Empréstimo
- Realizar empréstimos
- Realizar Devoluções
- Realizar Reservas
- Emitir relatórios de:
  - Empréstimos por tipos de mídia
  - Empréstimos por cliente
  - Empréstimos por gênero
  - Empréstimo por tipo de material.
  - Empréstimos por título
  - Reservas por cliente
  - Clientes com mais empréstimos
  - Devoluções atrasadas
  - Empréstimos por período
- Emitir Gráficos de:
  - Empréstimos por tipos de mídia
  - Empréstimos por gênero
  - Empréstimo por tipo de material.
  - Dez títulos mais emprestados
  - 10 clientes que mais realizarão empréstimo

Requisitos Não Funcionais

- Deverá ser desenvolvido na Linguagem JAVA
- Deverá ser utilizado o banco de dados MySQL
- Deverá ser utilizado framework Hibernate para persistência dos dados
- O sistema será utilizado em apenas um computador
- O sistema deverá ter uma interface amigável

- A documentação do sistema deverá conter os Casos de uso, diagramas de classe, Modelagem do banco de dados e os diagramas de sequência.

Regras de Negócio

- Administração de Usuários:
  - Usuários do sistema serão definidos como Administrador e funcionário.
  - Administradores terão todos os acessos no sistema.
  - Funcionários não terão acesso aos relatórios, cadastro de usuários e cadastro de tipo de empréstimo.
  - Todos os usuários só poderão ser marcados como inativos, mais não excluídos do banco de dados.
  - Poderá ser realizada alteração dos dados do funcionário.
  - O sistema não aceitará a inclusão de usuários com o mesmo Login.

- Administração de Clientes:
  - Clientes são cadastrados pelos usuários do sistema.
  - Deverá ser armazenada a foto de cada cliente.
  - Não terão acesso ao sistema.
  - Não poderá ter dois usuários com o mesmo nome.
  - Não poderá ter dois usuários com o mesmo RG.
  - Não poderá ter dois usuários com o mesmo CPF.
  - Realizarão Empréstimos.
  - Realizarão Devoluções.
  - Realizarão Reservas

- Administração de Materiais
  - Materiais são cadastrados pelos usuários do sistema.
  - Serão emprestados, devolvidos e reservados pelos Clientes.
  - Deverá ser cadastrada uma quantidade em estoque.
  - Os exemplares disponíveis, ao cadastrar um novo material, será o mesmo da quantidade do estoque.
  - Ao se adicionar na quantidade de estoque, o número de exemplares disponíveis deve aumentar concomitantemente a quantidade de estoque.
  - Poderão ser alterados.
  - **Não poderão ser excluídos.**

- Administração de Tipos de Materiais
  - Tipos de Materiais serão cadastrados pelos usuários do sistema.
  - Poderão ser alterados.
  - Só poderão ser excluídos se não houver nenhum material cadastrado com o tipo selecionado.

- Administração de Gêneros
  - Gêneros serão cadastrados pelos usuários do sistema.
  - Poderão ser alterados.

- Só poderão ser excluídos se não houver nenhum material cadastrado com o Gênero selecionado.

- Administração de Tipos de Empréstimo
  - Tipos de empréstimo serão cadastrados pelos administradores do sistema.
  - Poderão ser alterados.
  - Só poderão ser excluídos se não houver nenhum material cadastrado com o Tipo de empréstimo selecionado.
  - Serão cadastrados o valor do empréstimo e o valor da multa por dia.

- Realizar empréstimos
  - Serão realizados pelos usuários do sistema
  - Será necessário a seleção de um cliente e de pelo menos um material
  - Para cada material será realizado um empréstimo
  - Ao realizar um empréstimo será subtraído 1 da quantidade de exemplares disponíveis do material.

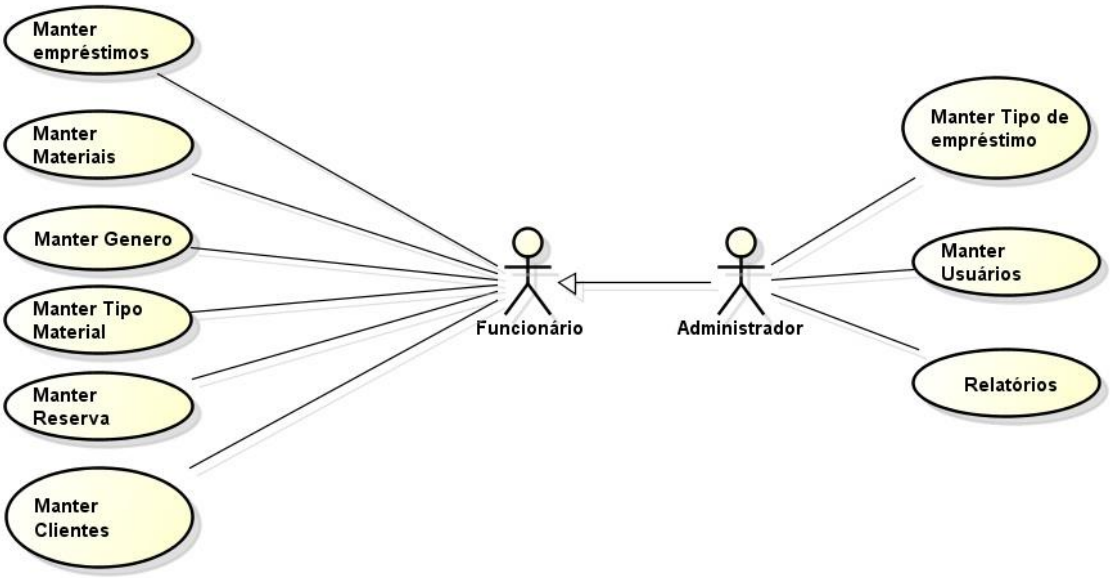
- Realizar Devoluções
  - Serão realizadas pelos usuários do sistema
  - Será necessário a seleção de um cliente e de pelo menos um material
  - Será incluído a data de devolução e alteração do status do empréstimo
  - Ao realizar uma devolução será adicionado 1 da quantidade de exemplares disponíveis do material.

- **Realizar Reservas**
  - **Serão realizadas pelos usuários do sistema**
  - Será necessário a seleção de um cliente e de um material
  - Um material reservado só poderá ser emprestado para o cliente que o reservou
  - Para emprestar para outro cliente deverá ser retirado todas as reservas do material.
  - O material só poderá ser emprestado se não houver nenhum exemplar disponível para locação.
  - Ao realizar uma devolução de um material reservado será dada uma mensagem ao usuário do sistema avisando que o material está reservado.
  - Quando o cliente que realizou a reserva emprestar o material, será alterado o status da reserva para "EMPRESTOU".

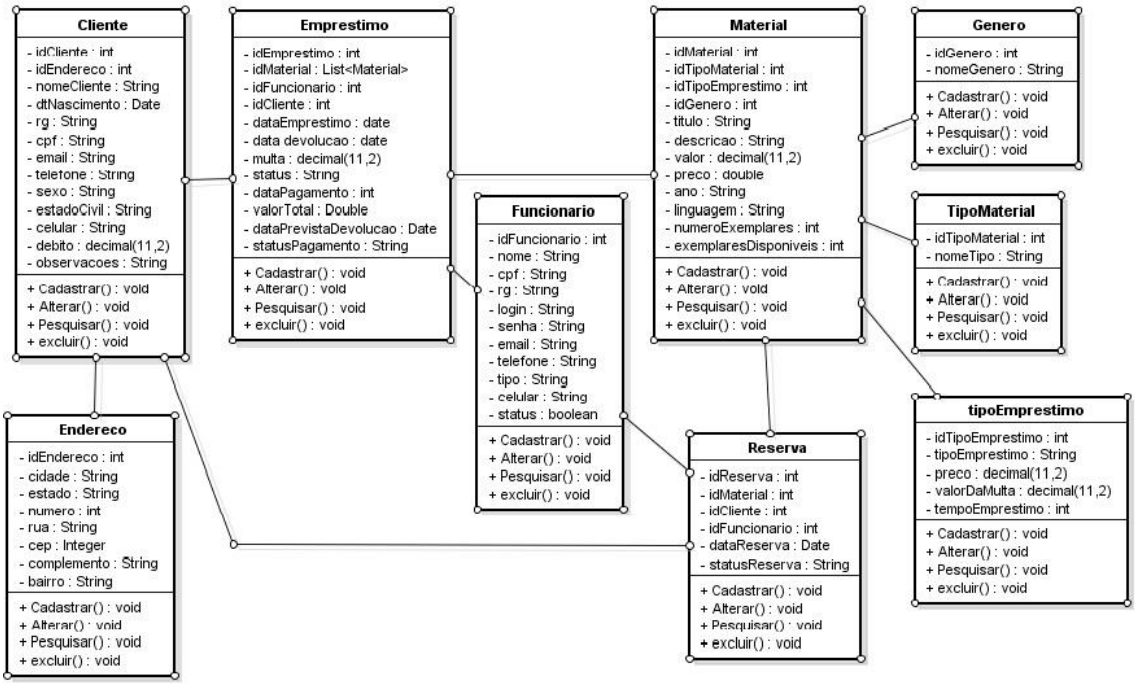
Rodrigo Florêncio Lima

Nelson Federle Junior

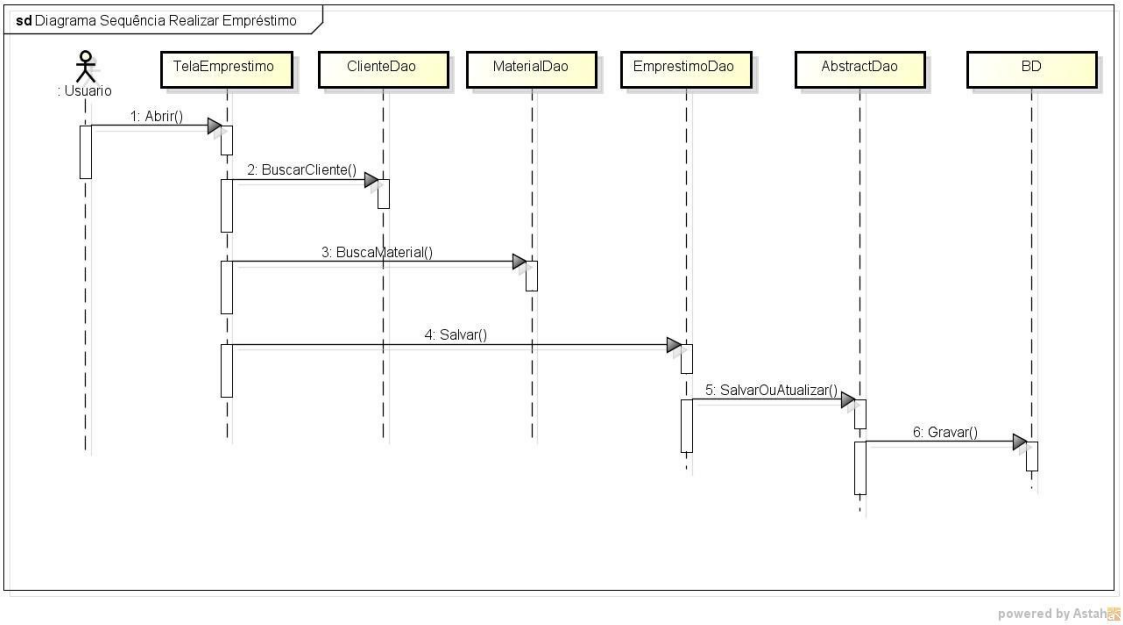
Apêndice B: Diagrama de Caso de Uso.



Apêndice C: Diagrama de Classes



Apêndice D: Diagrama de Sequência Realizar Empréstimo



Apêndice E: Modelagem Lógica do Banco de Dados