

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO
DO RIO GRANDE DO NORTE**

Luiz Antonio Ferreira Júnior

IMPLEMENTAÇÃO DE UM SISTEMA PARA LOCADORA VIA WEB

**NATAL-RN
2005**

LUIZ ANTONIO FERREIRA JÚNIOR

IMPLEMENTAÇÃO DE UM SISTEMA PARA LOCADORA VIA WEB

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte como requisito obrigatório para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador:

Fellipe Araújo Aleixo, M.Sc.

NATAL-RN
2005



M
004.4:658.91
F433 e
2005

Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Ferreira Júnior, Luiz Antonio

Educação Implementação de um Sistema para Locadora via Web / Luiz Antonio Ferreira Júnior. – Natal: FARN, 2005.
43 f.

Orientador: Fellipe Araújo Aleixo (M.Sc.)

Monografia (Bacharelado em Sistemas de Informação) – Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte.

1. Desenvolvimento de Software – Monografia. 2. Linguagem de Programação – Personal Home Page – Monografia. I. Título.

RN/FARN/BC

CDU 004.4:658.91

Luiz Antonio Ferreira Júnior

IMPLEMENTAÇÃO DE UM SISTEMA PARA LOCADORA VIA WEB

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte, aprovada em

17 de 12 de 2005 pela banca examinadora composta pelos seguintes

membros:


Felipe Araújo Aleixo, M.Sc.
Orientador


Bartira Paraguaçu Falcão Dantas Rocha, M.Sc.
Membro examinador


Nilda Maria Teixeira Cerqueira, M.Sc.
Membro examinador

Aos meus pais dedico este trabalho, como resultado de meu esforço e também do apoio que deles recebi em todos os projetos de minha vida, sempre me incentivando, confiando no meu potencial e fazendo o possível e impossível por mim.

AGRADECIMENTOS

Primeiramente a Deus por ter encontrado a profissão certa. Aos meus pais, pelo incentivo e por terem sempre acreditado no meu ideal. Aos mestres pela orientação constante e por auxiliar-me a trilhar este caminho árduo. Ao meu orientador Fellipe Aleixo que sempre se mostrou disponível nas horas que precisei de uma orientação. Ao meu amigo Gustavo Leitão que me auxiliou no desenvolvimento do sistema. Ao meu irmão e aos colegas de trabalho que por muito tempo ficaram sobrecarregados com as minhas ausências nas atividades do dia-a-dia.

RESUMO

O presente trabalho apresenta a análise e projeto do SLW (Sistema para Locadora via Web), um sistema que foi desenvolvido para a web. O SLW é um sistema que utiliza uma linguagem de programação dinâmica que é a PHP (*Personal Home Page*), é executado num servidor web, aonde todas as suas informações são armazenadas em um banco de dados (PostgreSQL) que também ficam sob a guarda do servidor, facilitando então futuras atualizações no sistema, manutenções e reparos. O sistema tem como objetivo principal facilitar o acesso do cliente, permitindo ao mesmo efetuar a locação de filmes sem se deslocar até a locadora.

Palavras-chave: Desenvolvimento de Software. Linguagem de Programação – *Personal Home Page*

ABSTRACT

The present work presents the analysis and project of the SLW (*System for video store through web*), a system that was developed for web. The SLW is a system that uses a dynamic programming language that is the PHP (*Personal Home Page*), is executed in a server web, where all its information are stored in a data base (PostgreSQL) that also is under the guard of the server, facilitating then future updates in the system, maintenances and repairs. The system has as objective main to facilitate the access of the customer exactly, allowing to effecting the location of films without if dislocating until the video store.

Word-key: Development the Software. Language the Programation - *Personal Home Page*

LISTA DE FIGURAS

Figura 1 – Código HTML básico	13
Figura 2 – Exemplo do comando condicional IF	16
Figura 3 – Exemplo de comando condicional SWITCH	16
Figura 4 – Exemplo do comando de repetição WHILE	17
Figura 5 – Exemplo do comando de repetição DO WHILE	18
Figura 6 – Exemplo do comando de repetição FOR	18
Figura 7 – Include para conexão com o bando de dados	19
Figura 8 – Módulo Cliente	24
Figura 9 – Módulo Administrador	24
Figura 10 – Diagrama de Atividades - AlugarFilme	26
Figura 11 – Tela Alugar Filme	27
Figura 12 – Código fonte Alugar Filme	28
Figura 13 – Diagrama de Atividades Reservar Filme	30
Figura 14 – Tela Reservar Filme	31
Figura 15 – Código fonte Reservar Filme	32
Figura 16 – Diagrama de Atividades Devolver Filme	34
Figura 17 – Diagrama de Atividades Cadastrar Cliente	35
Figura 18 – Diagrama de Atividades Consultar Filme	36
Figura 19 – Diagrama de Atividades Cadastrar Filme	38
Figura 20 – Diagrama de Atividades Imprimir Relatório dos filmes alugados	39
Figura 21 – Diagrama Entidade-Relacionamento (DER)	40

SUMÁRIO

1 Introdução	10
1.1 METODOLOGIA	10
2 Objetivos	11
2.1 OBJETIVOS GERAIS	11
2.2 OBJETIVOS EPECÍFICOS	11
3 Referencial Teórico	12
3.1 LOCAÇÃO	12
3.2 HTML (<i>Hiper Text Markup Language</i>)	12
3.2.1 Comandos HTML – Tags	13
3.2.2 Estrutura básica de uma página	13
3.3 PHP (<i>Personal Home Page</i>)	14
3.3.1 Estrutura de um programa PHP	15
3.3.2 Estruturas de controle em PHP	15
3.3.3 Comandos condicionais	15
3.3.4 Comandos de repetição	17
3.3.5 Funções de acesso ao banco de dados	19
3.4 UML (<i>Unified Modeling Language</i>)	20
3.4.1 Casos de uso	20
3.4.2 Diagramas de Atividade	20
3.5 POSTGRESQL	21
3.6 MYSQL	21
3.7 DREAMWEAVER MX	21
3.8 FIREWORKS MX	22
3.9 APACHE	22
4 Análise e projeto do Sistema	23
4.1 VISÃO GERAL DO SISTEMA	23
4.2 DIAGRAMAS DE CASOS DE USO	24
5 Implementação dos Casos de Uso	25
5.1 CASO DE USO ALUGAR FILME	25
5.1.1 Diagrama de Atividades – Alugar Filme	26
5.1.2 Tela e código fonte – Alugar Filme	27
5.2 CASO DE USO – RESERVAR FILME	29
5.2.1 Diagrama de Atividades – Reservar Filme	30

5.2.2 Tela e código fonte – Reservar Filme	31
5.3 CASO DE USO – DEVOLVER FILME	33
5.3.1 Diagrama de Atividades – Devolver Filme	34
5.4 CASO DE USO – CADASTRAR CLIENTE	34
5.4.1 Diagrama de Atividades – Cadastrar Cliente	35
5.5 CASO DE USO – CONSULTAR FILME	36
5.5.1 Diagrama de Atividades – Consultar Filme	36
5.6 CASO DE USO – CADASTRAR FILME	37
5.6.1 Diagrama de Atividades – Cadastrar Filme	37
5.7 CASO DE USO – IMPRIMIR RELATÓRIO DE FILMES ALUGADOS	38
5.7.1 Diagrama de Atividades – Imprimir relatório de filmes alugados	39
5.8 DIAGRAMA ENTIDADE-RELACIONAMENTO (DER)	40
6 Conclusão	41
Referências	42

1 INTRODUÇÃO

A Internet a cada dia que passa, está mais presente no dia-a-dia da maioria das pessoas, seja em casa ou no trabalho. Com o avanço que a tecnologia vem sofrendo, até mesmo através do celular consegue-se obter acesso a internet. Sem dúvida nenhuma, esses avanços só nos trazem benefícios tais como: agendar pagamentos, pesquisas para trabalhos, fazer compras, consultar a conta bancária, ficar por dentro do que ocorre no mundo, entre outros. Em virtude das inúmeras vantagens, soluções voltadas para Internet vêm aumentando a cada dia em diversas áreas.

O Sistema para Locadora via Web (SLW) implementa as etapas de uma locação via Web, partindo do cadastramento do cliente no site, até a devolução do filme.

O sistema flexibiliza o acesso do cliente, permitindo ao mesmo efetuar a locação de filmes sem se deslocar até a locadora. Com a ajuda do sistema, o locatário escolhe o filme via web e a locadora providencia a entrega na residência do cliente. O usuário terá que se cadastrar no site para poder usufruir os serviços oferecidos. Para uma maior garantia no cadastro, o sistema só irá cadastrar após verificar se o CPF digitado é verdadeiro. Quando alguma reserva ou locação é feita, a locadora é notificada através de um e-mail contendo a descrição da reserva ou da locação e também é enviado os dados do cliente para a locadora entrar em contato com o mesmo.

1.1 METODOLOGIA

Para a implementação desse sistema, foi feito uma pesquisa exploratória com o objetivo de encontrar vários conceitos e uma varredura em apostilas, tutoriais, livros, internet, etc. A pesquisa bibliográfica foi utilizada, pois ela é elaborada a partir de materiais já publicados, constituídos principalmente de livros, artigos de periódicos, internet, citações e referências. A pesquisa de laboratório também foi usada pelo fato de se tratar de uma implementação de sistema e porque durante o desenvolvimento e pesquisa, foi utilizado materiais do tipo hardware e software.

2 OBJETIVOS

2.1 OBJETIVOS GERAIS

Análise e implementação de um Sistema Web para automação de uma Locadora de Vídeo.

2.2 OBJETIVOS ESPECÍFICOS

Esse projeto de sistema está dividido nos seguintes objetivos específicos:

- Realizar um estudo sobre a linguagem PHP a fim de utilizar seus recursos;
- Realizar um estudo sobre os recursos do banco de dados PostgreSQL interagindo com PHP;
- Identificar os requisitos necessários para o sistema de uma locadora;
- Modelar o sistema proposto;
- Implementar o sistema proposto utilizando HTML, PHP e PostgreSQL.

3 REFERENCIAL TEÓRICO

Para o perfeito entendimento das sessões seguintes deste documento faz-se necessário a explanação de alguns conceitos que serão utilizados de forma recorrente na sequência do trabalho. Apresentaremos, portanto, os conceitos fundamentais e suficientes para o entendimento do trabalho em si, tais como: locação, HTML, PHP e análise estruturada.

3.1 LOCAÇÃO

Segundo Costa (1993), a locação, de acordo com a Tradição Romana, é o contrato em que uma das partes se obriga, mediante o pagamento de um preço, que a outra parte se obriga a pagar, a conceder-lhe, durante certo tempo, o uso ou gozo de uma coisa; ou a lhe prestar temporariamente os serviços; ou ainda a fazer por sua conta uma obra determinada.

A locação pode designar também dois contratos que se distinguem nitidamente e não pertencem à mesma categoria: A locação de coisas e a locação de serviços.

3.2 HTML (*HiperText Markup Language*)

É a linguagem de formatação que usamos para criar uma página Web, que por sua vez será composta de textos e comandos especiais que chamaremos de Tags.

No núcleo de toda página da Web, encontramos a HyperText Markup Language, uma linguagem de diagramação baseada em marcadores, ou tags como é mais comumente conhecido. A HTML (HyperText Markup Language) é uma linguagem simples baseada em texto que podemos visualizar em qualquer plataforma. (Chicayban, 2000, p. 1).

Segundo Cardozo (2003), a linguagem HTML é só o primeiro passo para quem começa a desenvolver sistemas para a web. Na hora de programar algo mais complexo, como serviços de comércio eletrônico, saber de uma linguagem que tenha acesso ao banco de dados é essencial.

3.2.1 Comandos HTML – TAGS

Os comandos em HTML são chamados de tags, e eles irão dizer ao browser como o texto, a informação e as imagens serão exibidas. Um *tag* também pode dizer que na verdade um determinado pedaço do texto é um endereço para outra página Web, que será acessado ao dar um click no endereço. (CHICAYBAN, 2000)

Os *tags* são identificados por serem envolvidos pelos sinais `< >` ou `</ >`. Entre os sinais `< >` são especificados os TAGS propriamente ditos. No caso de tags que precisem envolver um texto, a sua finalização deverá ser feita usando a barra de divisão `/`, indicando que o *tag* está finalizando a marcação de um texto, o que faz os tags normalmente andarem em pares. O formato genérico de um tag é:

`<Nome do TAG> Texto </Nome do TAG>`

3.2.2 Estrutura básica de uma página

Abaixo segue as tags básicas que determinam a estrutura básica de uma página HTML:

<code><HTML></code> <code></HTML></code>	Colocados no início e final do documento, delimitando a área onde serão colocados os TAGS HTML.
<code><HEAD></code> <code></HEAD></code>	Delimita uma área de cabeçalho com poucos TAGS.
<code><TITLE></code> <code></TITLE></code>	Serve apenas para colocar um título que aparecerá na barra de título do seu browser. É usado dentro do TAG <code><HEAD></code>
<code><BODY></code> <code></BODY></code>	Aqui serão colocados todos os comandos, texto e imagens da página. É o corpo da página.

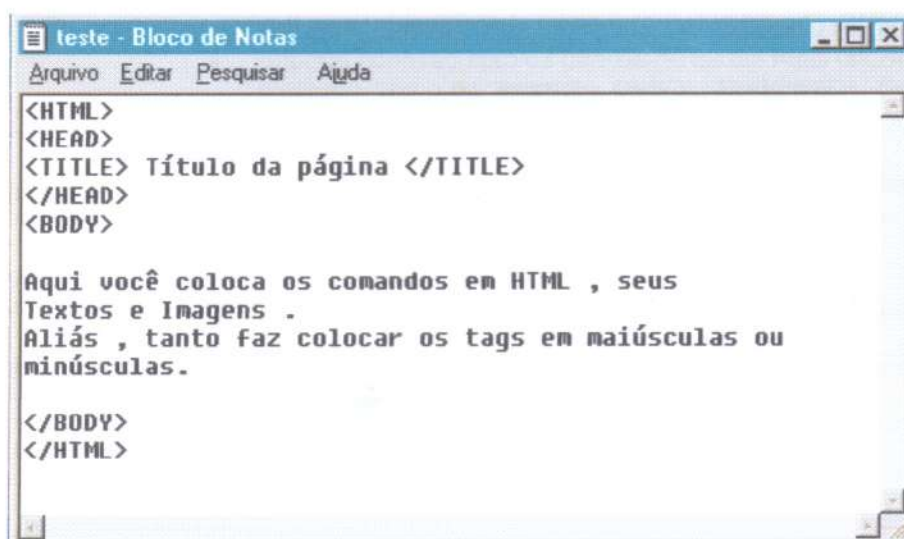


Figura 1 – Código HTML básico
Fonte: Elaborado pelo autor

3.3 PHP (Personal Home Page)

Arroyo; Santos (2003), afirmam que a linguagem PHP foi criada por volta de 1994 por Rasmus Lerdorf, que inicialmente o utilizava em sua home page pessoal (*Personal Home Page*).

Em meados de 1995 a linguagem passou a ser utilizada por outras pessoas e foi reescrito com novos recursos, sendo nomeado para *Personal Home Page Tools/FI* (Form Interpreter), e entre os novos recursos, passou a contar com suporte a MySQL. Dois anos mais tarde o PHP deixou de ser um projeto pessoal de Rasmus Lerdorf e passou a ser desenvolvido por uma equipe de colaboradores, e neste período, foi lançada a versão 3 da linguagem. A partir da versão 4 o PHP passou a utilizar a *engine de scripting (Máquina de script)* da Zend Technologies Ltda, para melhorar a performance e suportar uma variedade maior de bibliotecas externas e extensões. Até março de 2002, o PHP estava sendo utilizado em 9.000.000 de domínios.

Rocha (2003, p. 7) argumenta que:

PHP é uma linguagem que permite criar sites dinâmicos, possibilitando uma interação com o usuário através de formulários, parâmetros do URL e links. A diferença do PHP em relação às linguagens semelhantes a Javascript é que o código PHP é executado no servidor, sendo enviado para o cliente apenas HTML puro.

De acordo com Arroyo; Santos (2003):

O PHP tem inúmeras vantagens, como veremos a seguir:

- É uma linguagem de fácil aprendizado;
- Tem performance e estabilidade excelente;
- Seu código é aberto, não é preciso pagar por sua utilização, e é possível alterá-lo na medida da necessidade de cada usuário;
- Tem suporte nos principais servidores web do mercado, e suporte nativo no servidor web Apache (o mais utilizado no mundo);
- Suporta conexão com os bancos de dados mais utilizados no mercado, como por exemplo, MySQL, PostgreSQL, Oracle e DB2;
- É multiplataforma, tem suporte nos sistemas operacionais mais utilizados no mercado;
- Suporta uma variedade grande de padrões e protocolos, como o XML, DOM, IMAP, POP3, LDAP, HTTP, entre outros;
- Não precisa ser compilado.

Segundo Rocha (2003), o PHP encontra-se atualmente na versão 5, versão esta que oferece alguns recursos adicionais em relação às anteriores principalmente no que diz respeito à programação orientada a objetos, sem cópias redundantes de dados.

3.3.1 Estrutura de um programa PHP

Para o desenvolvimento de uma página em PHP se faz necessário possuir comandos HTML e códigos PHP.

De acordo com Niederauer (2004, p. 27):

Você pode ir concatenando *scripts* PHP com comandos HTML, podendo dessa forma, escrever vários *scripts* PHP em uma única página. Cada *script* PHP existente na página deve começar com a tag `<?php`, e terminar com `?>`. As linhas de programação que serão escritas entre as tags devem sempre terminar com `;` (ponto-e-vírgula), senão ocorrerão erros no momento da execução da página.

A mistura das linguagens PHP com HTML é muito útil, pois nós utilizamos o PHP para gerar os dados dinamicamente, enquanto o HTML é usado para formatar e exibir esses dados gerados nas páginas mostradas no browser do cliente.

3.3.2 Estruturas de controle em PHP

Segundo Rocha (2003), são comandos utilizados para estruturar os programas escritos em PHP. São comandos comuns à maioria das linguagens de programação, e o uso deles é fundamental para realizar decisões lógicas, testar, repetir um bloco de comandos, entre outras funções.

3.3.3 Comandos condicionais

Segundo Niederauer (2004), temos a oportunidade de avaliar uma expressão e, dependendo do resultado obtido, executar um trecho de código para uma tomada de decisão dentro de um programa.

IF – avalia uma expressão e, dependendo do resultado, é executado um conjunto diferente de instruções. Pode possuir como complemento o *elseif* e/ou *else*.

```

1  if ($casa=="true"){
2      $msg .= "ENTREGA: Em casa <BR>";
3  }else{
4      $msg .= "ENTREGA: Produto será retirado na locadora <BR>";
5  }
6  $car = $_SESSION["caminho"];
7  $msg .= "Filmes Alugados com Respectiva Quantidade: <BR>";

```

Figura 2 – Exemplo do comando condicional IF

Fonte: Elaborado pelo autor

O comando condicional IF acima checa se o cliente selecionou para receber o produto em casa ou não. A 1º linha checa o valor da variável \$casa, que contém o valor "true" caso o cliente selecionou para receber o filme em domicílio e contém o valor "false" caso deseje resgatar o produto na própria loja. A 2º linha somente será executada caso o valor da variável \$casa seja "true". Tal linha concatena à variável \$msg, no qual contém a mensagem a ser enviada ao e-mail do administrador do site, a STRING que informará como o produto será entregue. Semelhantemente a 4º linha só será executada se o valor da variável \$casa não seja igual a "true", ou seja, o produto deverá ser resgatado na loja. E para tal é concatenado a variável \$msg a STRING que informará ao administrador do site tal forma de resgate do produto.

SWITCH – muito parecido com o IF, porém, mais usado em ocasiões que tem-se a mesma variável a ser testada com valores diferentes. Trabalha basicamente com o operador de igualdade.

```

1  $busca = $_POST["busca"];
2  $tipo = $_POST["tipo"];
3  switch ($tipo){
4      case 0:
5          $sql = "select * from filmes WHERE genero like '%".$busca."%'";
6          break;
7      case 1:
8          $sql = "select * from filmes WHERE nome like '%".$busca."%'";
9          break;
10     case 2:
11         $sql = "select * from filmes WHERE atores like '%".$busca."%'";
12         break;
13     }
14

```

Figura 3 – Exemplo do comando condicional SWITCH

Fonte: Elaborado pelo autor

O comando condicional SWITCH tem como principal função selecionar os comandos que deverão ser executados, dependendo do valor da variável checada.

No exemplo acima o intuito é montar a STRING SQL de busca de acordo com o tipo selecionado pelo usuário do site. A variável \$tipo contém o tipo de busca selecionado: 0 para busca por gênero. 1 para busca por nome do filme e 2 para busca pelo elenco. A 3ª linha faz a checagem do tipo selecionado pelo usuário. Caso o valor da variável \$tipo seja 0 a 5ª linha será executada, montando adequadamente a STRING SQL para buscar por gênero. As linhas 9 e 12 funcionam de forma semelhante, sendo executadas de acordo com o valor da variável \$tipo: 1 e 2 respectivamente.

3.3.4 Comandos de repetição

Segundo Niederauer (2004), são comandos utilizados para que um conjunto de instruções seja executado repetidamente por um número determinado de vezes, ou até que determinada função seja atingida.

WHILE – composto por uma expressão e por um bloco de comandos, ele avalia a expressão, e enquanto essa expressão retornar o valor verdadeiro, a execução em questão será repetida.

1	<code><?php</code>
2	<code> \$cont = 1;</code>
3	<code> while (\$cont < 100)</code>
4	<code> {</code>
5	<code> echo "O valor atual do contador é \$cont
";</code>
6	<code> \$cont++;</code>
7	<code> }</code>
8	<code>?></code>

Figura 4 – Exemplo do comando de repetição WHILE
Fonte: Niederauer (2004, p.66)

A execução desse programa resultará em 99 linhas mostradas na tela, pois quando a variável \$cont atingir o valor 100, a expressão retornará o valor falso, pois 100 não é menor que o próprio 100, e isso fará com que o loop seja encerrado.

DO...WHILE – a expressão é avaliada no final do laço. Portanto, o laço será executado pelo menos uma vez. O laço só é encerrado quando a expressão avaliada é falsa.

```
1 <?php
2     $numero = 1;
3     do
4     {
5         echo "O valor atual do número é $numero <br>";
6         $numero++;
7     } while ( $numero < 4 );
8 ?>
```

Figura 5 – Exemplo do comando de repetição DO...WHILE

Fonte: Niederauer (2004, p.67)

A diferença entre o *while* e o *do...while* é que o *while* avalia a expressão no início do laço, e o *do...while* avalia a expressão no final do laço.

FOR – utilizado quando queremos executar um conjunto de instruções um número determinado de vezes.

```
1 for ($t=0;$t<sizeof($car);$t++){ //sizeof retorna o tamanho do array
2 $msg .= " COD FILME: ".$car[$t][0]. " QNT: ".$car[$t][1]. "<BR>";
3 $temp=$car[$t][0];
4 $resultado = pg_query($conexao, "SELECT qnt from filmes where
5 cod=$temp");
6 $qnts[$t] = pg_fetch_result($resultado,0,"qnt");
7 }
```

Figura 6 – Exemplo do comando de repetição FOR

Fonte: Elaborado pelo autor

O comando de repetição FOR é utilizado acima com duas funções básicas. As linhas 2 à 6 serão executadas de acordo com a quantidade de itens no carrinho de compras. Tais funções básicas serão as seguintes:

1º - Montar a mensagem a ser enviada ao administrador do site, informando-o quais filmes foram alugados, bem como a quantidade de cada filme que o cliente deseja fazer a locação. Esta mensagem é montada na 2ª linha onde é concatenada à variável \$msg o código e a quantidade do filme em questão.

2º Armazena no Array \$qnts a quantidade disponível em estoque de cada filme selecionado pelo cliente. Esta parte tem como intuito fazer a atualização da quantidade de produtos no estoque e a checagem se existe disponibilidade para quantidade selecionada pelo cliente.

3.3.5 Funções de acesso ao banco de dados

De acordo com Niederauer (2004, p. 97):

As *includes* são uma excelente opção oferecida pelo PHP. Elas nos permitem reaproveitar uma ou mais funções ou arquivos, utilizando-os em diversas páginas do site. As situações em que as *includes* são mais usadas são aquelas em que uma alteração deve ser refletida em todas as páginas do site automaticamente, e para isso precisamos alterar um arquivo, que será acessado por todas as páginas.

Sempre que o PHP necessita acessar o banco de dados, a *include* que foi criada é executada sempre no início do código. Observe na figura abaixo:

1	<?php
2	
3	<i>\$str_conexao = "dbname=locadora port=5432</i>
4	<i>user=postgres password=postgres";</i>
5	
6	<i>if(!(\$conexao=pg_connect(\$str_conexao)))</i>
7	<i>{</i>
8	<i> echo "Não foi possível estabelecer uma conexão</i>
9	<i>com o banco de dados.";</i>
10	<i> exit;</i>
11	<i>}</i>
12	
13	?>

Figura 7 – Include para conexão com o banco de dados
Fonte: Niederauer (2004, p.161)

Nas linhas 3 e 4 é feita a conectividade com o banco de dados, passando os dados do banco como: nome do banco, a porta que o banco está rodando, o nome do usuário no banco e a senha, já na linha 6 é feito o teste com a variável que recebe a conexão.

Abaixo segue algumas das funções que foram usadas para acessar o banco de dados:

- `pg_connect` Abre uma conexão PostgreSQL.
- `pg_query` Executa uma consulta.
- `pg_num_rows` Retorna o número de linhas de uma consulta.
- `pg_close` Fecha uma conexão PostgreSQL.
- `pg_fetch_result` Retorna valores a partir de um recurso do tipo *result*

3.4 UML (*Unified Modeling Language*)

De acordo com Ahmed; Umrysh (2002), a idéia central ao usar a UML é capturar os detalhes significantes sobre um sistema de modo que o problema seja claramente compreendido, a arquitetura da solução seja desenvolvida e a implementação escolhida seja claramente identificada e construída.

A UML é uma linguagem gráfica para a modelagem e o desenvolvimento de sistemas de software. Fornece um suporte de modelagem e visualização para todas as fases de desenvolvimento do software, desde a análise das exigências até a especificação, construção e distribuição. Durante essa análise com UML, há uma ênfase na descoberta e na descrição dos objetos, ou conceitos, do domínio do problema, já durante o projeto em UML, a ênfase passa a definição dos elementos lógicos de software os quais serão implementados em uma linguagem de programação. (Larman, 2000).

A UML define artefatos como: casos de uso, diagramas, atores entre outros para padronizar a linguagem de entendimento entre o analista de sistema e o programador. (Larman, 2000).

3.4.1 Casos de uso

Segundo Ahmed; Umrysh (2002), um diagrama do caso de uso mostra os casos de uso, os atores e suas relações. Os diagramas do caso de uso capturam as exigências precisas para o sistema da perspectiva de um usuário.

3.4.2 Diagramas de Atividade

De acordo com Ahmed; Umrysh (2002), diagrama de atividade é uma extensão de um diagrama do gráfico do estado e é parecido em conceito com um fluxograma. Ele permite modelar o comportamento do sistema em termos de interação ou fluxo do controle entre as atividades distintas ou objetos.

3.5 PostgreSQL

Segundo Stefan (1998), o PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional (SGBDOR), desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley. O PostgreSQL foi pioneiro em vários conceitos que somente se tornaram disponíveis muito mais tarde em alguns sistemas de banco de dados comerciais.

3.6 MySQL

Segundo Tridapali; Santanna (2001), O MySQL é um servidor de banco de dados relacional multi-usuário. SQL é a linguagem de banco de dados mais popular no mundo. MySQL é uma implementação cliente-servidor que consiste de um servidor e diferentes programas clientes e bibliotecas.

Tridapali; Santanna (2001) afirmam que:

As principais características do MySQL são:

- Manipula um número ilimitado de usuários simultâneos;
- Alta velocidade de execução;
- Possui APIs C, C++, Eiffel, Java, Perl, PHP, Python e TCL;
- Trabalha com diferentes plataformas: Unix, Windows etc.;
- Disponibiliza registros de tamanho fixos e variados;
- O servidor pode emitir mensagens de erros em diversas linguagens;

3.7 Dreamweaver MX

Segundo Silva (2004), o Macromedia Dreamweaver é uma poderosa ferramenta para criação de páginas de Internet, hoje na versão 7 (MX2004). Suporte a vários tipos de linguagens de programação (ASP, PHP, XML, JSP, etc.).

Com o Dreamweaver podemos criar sites para a Web, graças à sua interface gráfica de fácil utilização que mostra as modificações à medida que são efetuadas. Além disso, teremos a certeza de que o código HTML gerado nos é consistentemente limpo, conciso e perfeitamente editável, ao contrário do que acontece com a maioria dos programas de edição visual para criação da Web, o Dreamweaver não altera o código HTML quando importa documentos de um outro programa HTML.

3.8 Fireworks MX

O Fireworks possibilita a combinação de vetores e bitmaps, a aplicação de efeitos do tipo *rollover*, aquele que muda a imagem quando o mouse passa por cima, definir blocos de texto como GIFs, colocar fotos como JPEGs totalmente coloridas e gerar (exportar) automaticamente uma tabela HTML para reorganizar suas imagens divididas.

3.9 Apache

Segundo Alecrim (2003), o Apache existe para oferecer uma implementação robusta do protocolo HTTP. Para isso, é necessário que ele rode em plataformas de código-fonte aberto, onde pessoas e empresas possam utilizar o software de acordo com suas necessidades e pesquisas. o Apache é desenvolvido por uma entidade colaborativa, onde quem quer (e pode) ajudar, participa, e usufrui os benefícios de um software feito por muitas pessoas e ajuda outras, com suas ações no projeto.

4 ANÁLISE E PROJETO DO SISTEMA PARA LOCADORA VIA WEB

4.1 Visão geral do sistema

O sistema possui várias funcionalidades, entre elas, efetuar locação via web é a principal, confirmar devolução do filme via web, fazer reservas de filmes, permite que o cliente se cadastre via web, consultar filmes para locar ou reservar, área de administrador para cadastrar os filmes e controlar os níveis de acesso dos clientes e imprimir relatórios dos filmes locados.

Id.	Caso de Uso	Descrição	Entrada	Classificação
01	AlugarFilme	Cliente aluga filme	DadosAluguel	Primário
02	ReservarFilme	Cliente solicita reserva	DadosReserva	Primário
03	DevolverFilme	Cliente devolve filme	DadosDevolucao	Primário
04	CadastrarCliente	Cliente solicita cadastro	DadosCliente	Secundário
05	ConsultarFilme	Cliente realiza consulta	DadosConsulta	Secundário
06	CadastrarFilme	Gerente cadastra filme	DadosFilme	Secundário
07	ImprimirRel FilmesAlugados	Gerente solicita relatório de filmes alugados	SolicitacaoRel FilmesAlugados	Secundário

Classificação:

- Primário: São as ações mais importantes para que se concretize uma locação via web, são elas (Cliente aluga filme, devolve filme e reserva filme). Todas elas ilustrando a interação do cliente com o sistema.

- Secundário: Não são de fundamental importância, pois a ênfase maior do sistema é efetuar locação via web, em segundo plano temos operações como: consultar por filmes, impressão relatórios, etc.

4.2 Diagramas de Casos de Uso

O sistema foi dividido em módulos, para deixar bem claro a diferença entre a interação do cliente com o sistema e do administrador com o sistema.

O módulo cliente é subordinado ao módulo administrador, ou seja, quem estabelece as regras de permissão é o administrador do sistema. Portanto, o cliente interage com o sistema apenas para fazer operações não complexas como, alugar um filme, se cadastrar no sistema, consultar filme, etc.

O módulo administrador como o próprio nome já diz, ele tem a função de administrar o sistema, ou seja, cadastrar filmes no banco de dados, imprimir relatórios, controle de todos os clientes cadastrados no sistema, etc.

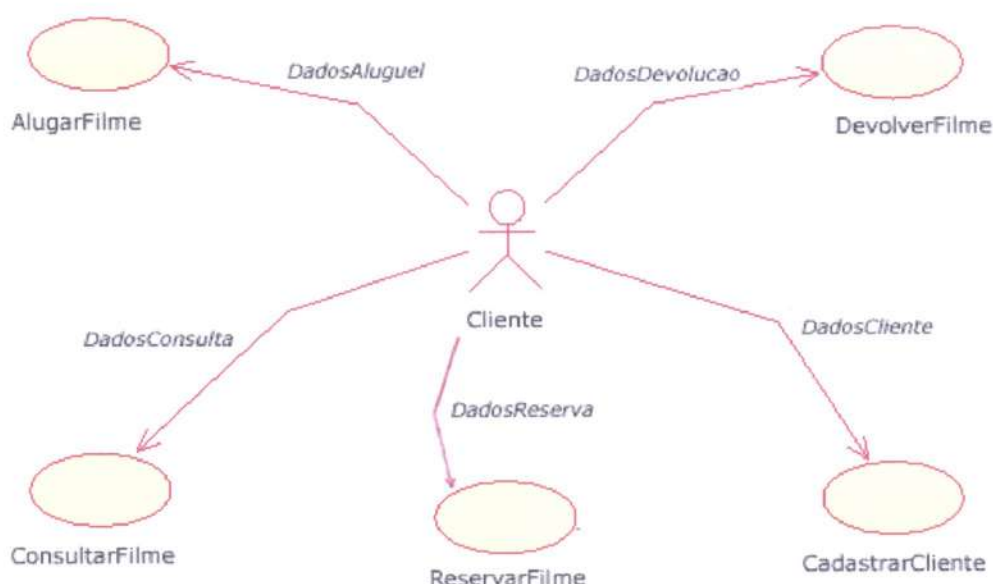


Figura 8 – Módulo Cliente
Fonte: Elaborado pelo autor



Figura 9 – Módulo Administrador
Fonte: Elaborado pelo autor

5 IMPLEMENTAÇÃO DOS CASOS DE USO

Para cada caso de uso, que representa a interação do ator com o sistema, tem-se representado os seus relacionamentos com os atores e uma descrição do seu comportamento.

5.1 Caso de uso – Alugar Filme

Identificação: 01

Caso de Uso: AlugarFilme

Descrição: Este caso de uso trata do aluguel de um filme.

Ator: Cliente

Seqüência Normal

1. Cliente solicita o aluguel de um filme.
2. Sistema solicita que o cliente efetue o login.
3. Cliente efetua login.
4. Sistema verifica no banco de dados se o cliente está cadastrado.
5. Sistema encontra o cliente no banco de dados e autoriza.
6. O cliente pesquisa os filmes disponíveis para locação.
7. Sistema guarda as informações do aluguel associadas ao cliente.
8. Sistema libera o aluguel.

Seqüências Alternativas

5. Sistema não encontra o cliente no banco de dados.
 - 5.1 Sistema cancela o aluguel e encerra o caso de uso.

5.1.1 Diagrama de Atividades – Alugar Filme

Para que o processo de locação seja concretizado, o cliente terá que se cadastrar no sistema antes de tudo. O cadastro pode ser feito pelo próprio cliente através do site da locadora. Logo após o cadastro, o novo cliente já poderá efetuar *login* e assim então, iniciar o processo de locação.

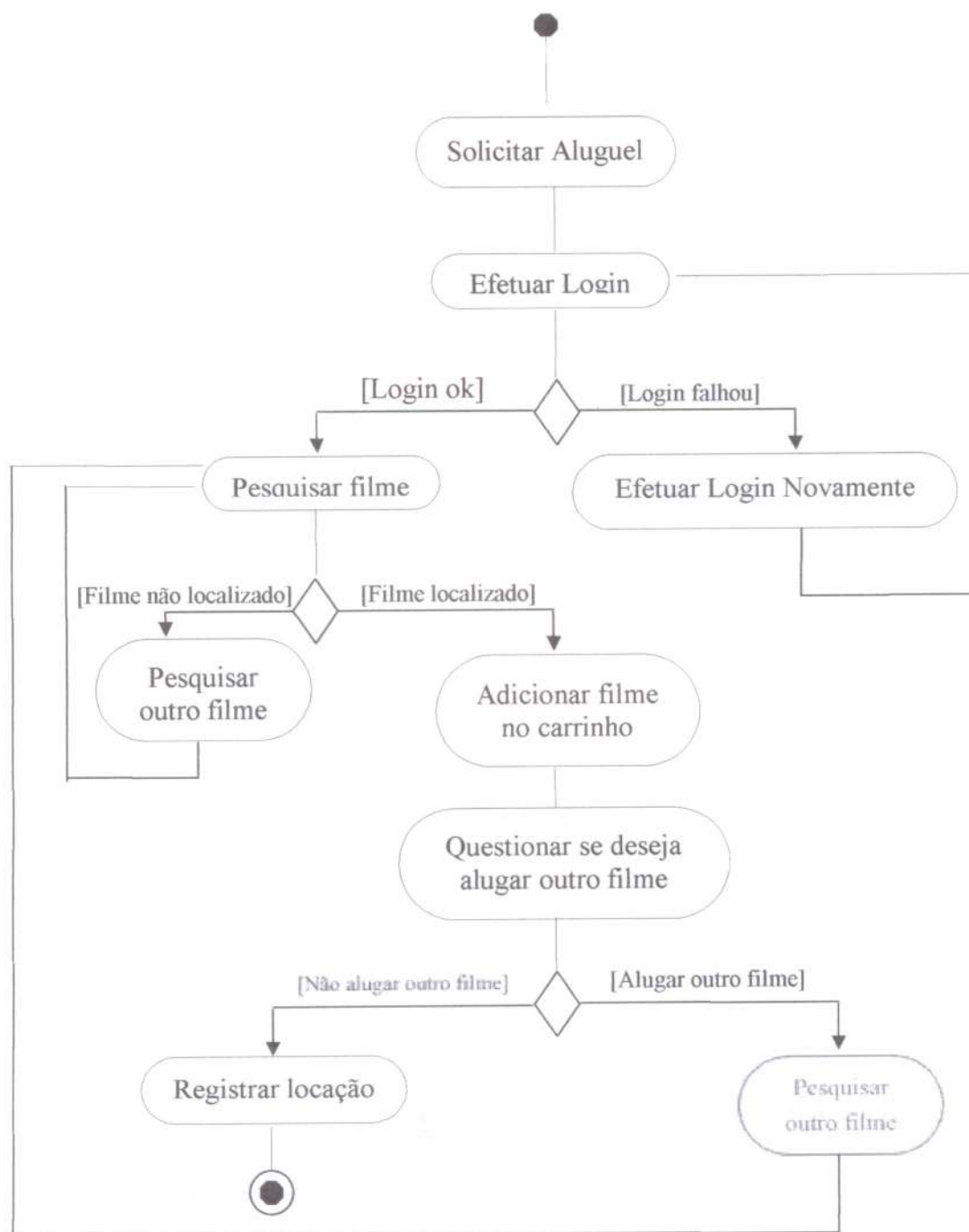


Figura 10 – Diagrama de Atividades - AlugaFilme
Fonte: Elaborado pelo autor

5.1.2 Tela e código fonte – Alugar Filme

Para chegar a essa tela da Figura 11, é obrigatório passar pelo cadastramento no sistema, como foi explicado na Figura 10. Depois de cadastrado, o cliente irá realizar uma consulta e selecionar o(s) filme(s) que deseja para então chegar nessa tela de confirmação de aluguel.

NETVIDEO.COM

HOME PREÇOS PERGUNTAS FREQUENTES CONTATOS MEU CARRINHO Bem vindo: luiz Encerrar Sessão

LOGIN

Login

Senha

ok

Cadastre-se!

MENU

- EM BREVE
- FILMES DO MÊS
- TOP 10
- CATÁLOGO
- PROMOÇÕES

LOCALIZAR

Dados Cadastrais:

Nome:
Lutz Antonio Ferreira Júnior

Endereço:
Av Jaguarari, 1855

Bairro:
Jiqui

CEP:
55874489

Qnt.	Nome:	Preço (R\$) :
2	mergulho radical	4,5
1	cidade baixa	4,5
TAXA DE ENTREGA:		3,00
TOTAL A PAGAR		16,5

* TAXA DE ENTREGA ÚNICA PARA TODA GRANDE NATAL

* SÓ ENTREGAMOS PARA A GRANDE NATAL E PARA O ENDEREÇO DE CLIENTES CADASTRADOS CONOSCO

[CONFIRMAR ALUGUEL](#)

Figura 11 – Tela Alugar Filme
Fonte: Elaborado pelo autor

```

1  <?
2  session_start();
3
4  include "conecta_pg.inc"; //conecta com o banco
5  $usuario = $_SESSION["sess_usuario"];
6  $sql = "SELECT * FROM clientes WHERE usuario='$usuario'";
7  $resultado = pg_query($conexao, $sql); //pega infos do usuario logado
8
9  $casa = $_GET["casa"]; //pega qual o modo de entrega
10
11 $msg = "NOME: ".pg_fetch_result($resultado,0,"nome") . "<BR>";
12 $msg .= "ENDEREÇO: ".pg_fetch_result($resultado,0,"endereco").",
13 ".pg_fetch_result($resultado,0,"numero"). "<BR>";
14 $msg .= "BAIRRO: ".pg_fetch_result($resultado,0,"bairro"). "<BR>";
15
16 if ($casa=="true"){
17 $msg .= "ENTREGA: Em casa <BR>";
18 }else{
19 $msg .= "ENTREGA: Produto será retirado na locadora <BR>";
20 }
21 $car = $_SESSION["carrinho"];
22 $msg .= "Filmes Alugados com Respectiva Quantidade: <BR>";
23
24 $qnts = array(); //primeiro for: gera msg com cod e quantida de cada filme
25 for ($t=0;$t<sizeof($car);$t++){ //sizeof retorna o tamanho do array
26 $msg .= " COD FILME: ".$car[$t][0]. " QNT: ".$car[$t][1]. "<BR>";
27 $temp=$car[$t][0];
28 $resultado = pg_query($conexao, "SELECT qnt from filmes where
29 cod=$temp");
30 $qnts[$t] = pg_fetch_result($resultado,0,"qnt");
31 }
32
33 for ($t=0;$t<sizeof($qnts);$t++){ //atualiza a quantidade de filmes no banco
34 $qnts[$t] -= $car[$t][1];
35 $temp0 = $car[$t][0];
36 $temp1 = $qnts[$t];
37 pg_query($conexao, "UPDATE filmes SET qnt=$temp1 where cod=$temp0");
38 }
39
40 $email_dest = "locadora@nvideos.com";
41 $assunto = "Aluguel";
42 $email_rem = "aluguel@nvideos.com";
43
44 mail($email_dest,$assunto,$msg,"From:$email_rem Content-Type: text/html;
45 charset=us-ascii");
46
47 $_SESSION["carrinho"] = array(); //zerando o carrinho de compras
48
49 $msg = "Aluguel Realizado com Sucesso!!";
50 header ("Location: sucesso.php?msg=$msg");
51
52 ?>

```

Figura 12 – Código fonte Alugar Filme
 Fonte: Elaborado pelo autor

De acordo com a Figura 11 e 12, após visualizar e confirmar os dados e valores, o usuário deverá clicar em “CONFIRMAR ALUGUEL” para que o processo seja então finalizado. Após clicar nesta opção será enviado um e-mail para o administrador da locadora com especificações do pedido realizado, bem como o produto é automaticamente retirado de estoque, para evitar aluguéis acima da oferta.

Da linha 10 à 14 é montada a mensagem a ser enviada para o administrador da locadora referente aos dados do cliente que está finalizando o processo de locação. Entre as linhas 17 e 19 é concatenada a mensagem se o usuário vai retirar o produto na locadora ou se vai ser entregue em casa. A partir das linhas 24 até a linha 31 é concatenada a mensagem dos dados referentes aos filmes escolhidos para locação pelo cliente. Simultaneamente é checada no banco a quantidade de cópias disponíveis e guardadas em um array para se poder fazer a atualização do estoque em seguida. Na linha 37 é feita uma atualização (update) na quantidade de mídias disponíveis de um filme, ou seja, o estoque do filme é atualizado subtraindo a quantidade que já se tinha em estoque da quantidade desejada à locação pelo cliente. Após tal procedimento, da linha 40 em diante é enviado o e-mail para o administrador da locadora para o mesmo providenciar a entrega ou reserva dos produtos desejados pelo cliente. Logo após o carrinho de compras é reinicializado (linha 47) e é enviado uma mensagem de sucesso para o usuário avisando que o processo de aluguel foi finalizado com sucesso.

5.2 Caso de uso – Reservar Filme

Identificação: 02

Caso de Uso: ReservarFilme

Descrição: Este caso de uso trata da reserva de um filme.

Ator: Cliente

Seqüência Normal

1. Cliente digita nome do filme no formulário de busca.
2. Sistema realiza a consulta pelo filme.
3. Sistema exibe o resultado da consulta na tela.

4. Cliente seleciona o(s) filme(s).
5. Sistema exibe os dados do filme na tela.
6. Cliente confirma a reserva.

Seqüências Alternativas

4. Cliente não seleciona nenhum filme.
 - 4.1 Sistema não localizou nenhum filme na consulta.
 - 4.2 Sistema volta para a tela de nova consulta de filmes.

5.2.1 Diagrama de Atividades – Reservar Filme

Para que uma reserva seja realizada o cliente terá que fazer uma consulta pelo(s) filme(s) que deseja. O sistema irá pesquisar no banco de dados se o filme existe, caso exista, será exibido na tela os dados sobre o filme, caso contrário terá que ser feita uma nova consulta.

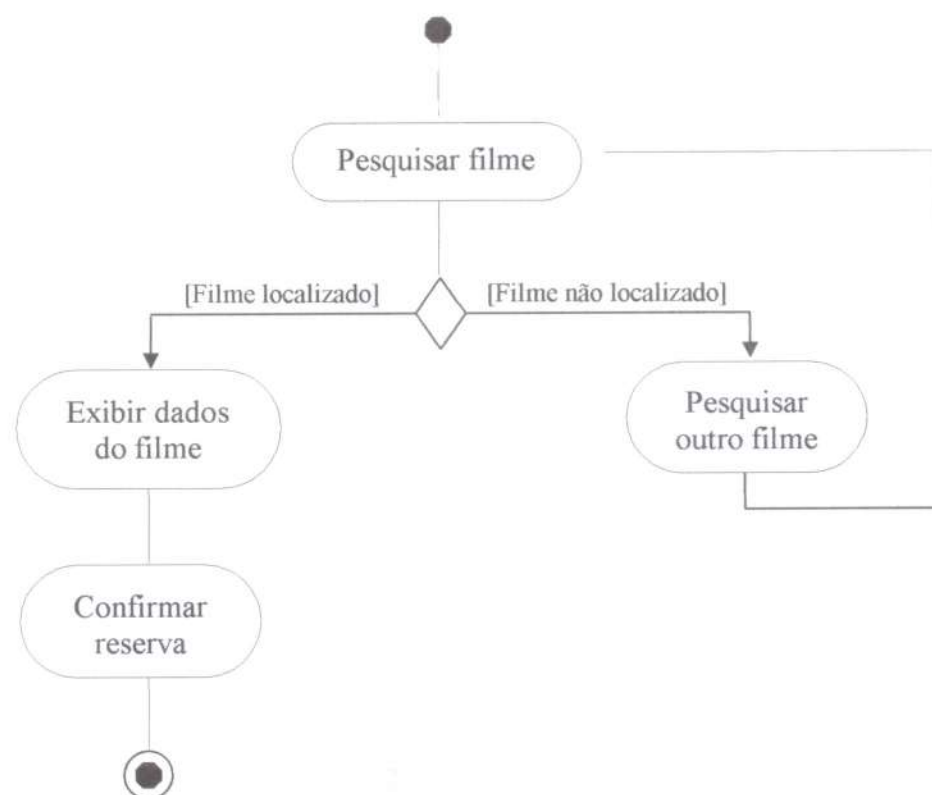


Figura 13 – Diagrama de atividades Reservar Filme
Fonte: Elaborado pelo autor

5.2.2 Tela e código fonte – Reservar Filme

Ao chegar nessa tela (Figura 14), significa que o usuário já logou no sistema, selecionou o(s) filme(s) desejado e falta decidir se o filme vai ser retirado na loja ou entregue em casa.

Carrinho de Compras			
Qnt.	Nome:	Preço (R\$) :	Ação:
3	mergulho radical	4.5	Excluir
2	cidade baixa	4.5	Excluir
TOTAL:			22.5
☐ Desejo Apenas Reservar e Pegar na Loja			
☐ Desejo Receber Meu Filme em Casa			
PROSSEGUIR			

Figura 14 – Tela Reservar Filme.
Fonte: Elaborado pelo autor

```

1  <?
2  session_start();
3  $msg="";
4  if ($_SESSION["sess_usuario"] == ""){ //testa se o usuário está logado
5      $msg="É necessário efetuar login";
6
7      if (isset($_GET["action"]) || (isset($_GET["cod"]))){//verifica se o filme foi selec..
8          if (!isset($_GET["cod"])){
9              $_SESSION["back"] = "aluga.php?action=&cod=".$_GET["cod"];
10             }elseif (isset($_GET["action"])){
11                 $_SESSION["back"] = "aluga.php?action=".$_GET["cod"]."&cod=";
12             }else{
13                 $_SESSION["back"] = "aluga.php?action=".$_GET["cod"]."&cod=".$_GET["cod"];
14             }
15         }
16     }
17
18     $acao = $_GET["action"];
19     $id = $_GET["cod"];
20
21     $car = $_SESSION["carrinho"];
22
23     if ($acao == "alugar"){
24
25         if (!empty($id)){
26
27             $indice = sizeof($car);
28             $teste = false;
29
30             for ($t=0;$t<sizeof($car);$t++){ //procura se o filme já for a alugado
31                 if ($car[$t][0] == $id){
32                     $indice = $t; //guarda o índice do array que tem o filme que já fora alugado
33                     $teste = true; //indica que existe no array um filme do mesmo cod do que se deseja.
34                 }
35             }
36
37             $car[$indice][0] = $id;
38             if (!$teste){
39                 $car[$indice][1] = 1;
40             }else{
41                 $car[$indice][1] += 1;
42             }
43         }
44
45         }elseif ($acao == "excluir") {
46             for ($t=0;$t<sizeof($car);$t++){
47                 if ($car[$t][0] == $id){
48                     $car[$t] = $car[sizeof($car)-1];
49                     unset($car[sizeof($car)-1]);
50                 }
51             }
52         }
53
54     $_SESSION["carrinho"] = $car;
55
56     ?>

```

Figura 15 – Código fonte Reservar Filme
Fonte: Elaborado pelo autor

O carrinho de compras nada mais é do que um *Array* de duas dimensões. A primeira dimensão fica responsável por guardar qual o código do filme e a segunda a quantidade de cópias a serem alugadas. Tal *Array* é guardado em uma variável de

sessão para que o usuário possa acessar e retornar para outras páginas sem perder tais informações.

A rotina de inclusão do filme escolhido no carrinho encontra-se implementada da linha 25 à 35 como mostra a Figura 15. Tais linhas consistem em checar se o filme escolhido já pertence ao carrinho, caso positivo é incrementado apenas a dimensão responsável pela quantidade de cópias, caso não pertença ao carrinho ainda, será incluído uma nova linha ao array com o código do filme escolhido e sua correspondente quantidade.

O sistema também possibilita a exclusão de itens do carrinho de compras. Tal rotina está implementada da linha 38 à 42 e consiste em procurar no array pelo item a ser excluído. Após achar tal elemento o último elemento do array ocupa o lugar de tal item, evitando assim que o array contenha “buracos” em sua estrutura, facilitando sua manutenção e busca.

Após a conclusão da compra pelo site o usuário deverá escolher a forma como resgatará o filme, tendo para isso duas opções: Ir buscar na locadora ou receber o filme em casa. Logo em seguida deverá clicar no botão prosseguir, que transferirá a execução da página para o arquivo “*confirma.php*” que fará a confirmação do aluguel em questão.

5.3 Caso de uso – Devolver Filme

Identificação: 03

Caso de Uso: DevolverFilme

Descrição: Este caso de uso trata da devolução de um filme.

Ator: Cliente

Seqüência Normal

1. Cliente solicita devolução de um filme.
2. Cliente escolhe o(s) filme(s) à ser(em) devolvido(s)
3. Cliente confirma dados de cadastro e valor do pagamento
4. Cliente confirma a devolução.
5. Sistema solicita que seja feito o resgate do filme e do pagamento.

Seqüências Alternativas

3. Sistema não encontra o filme associado ao código digitado.

3.1 Sistema cancela o processo de devolução e encerra o caso de uso .

5.3.1 Diagrama de Atividades – Devolver Filme

Após a solicitação para devolução do filme, o sistema irá pedir que o cliente digite o código do filme para que seja verificado se o pagamento foi feito e imprimir o recibo de devolução.

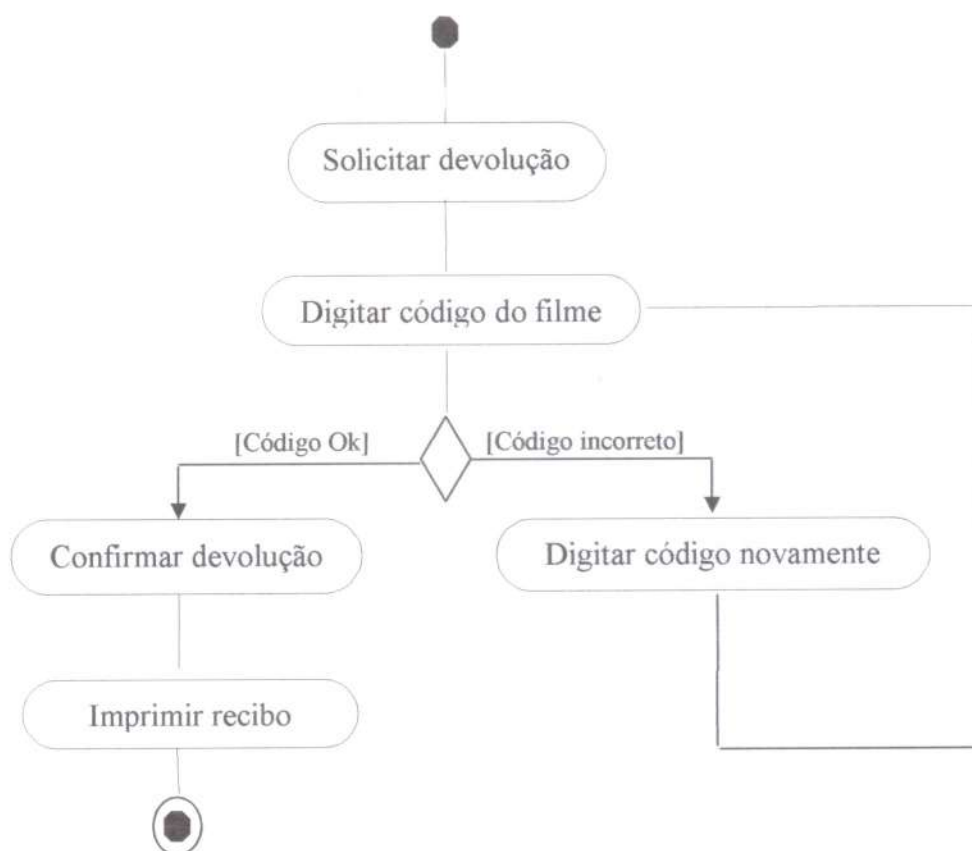


Figura 16 – Diagrama de atividades Devolver Filme
Fonte: Elaborado pelo autor

5.4 Caso de uso – Cadastrar Cliente

Identificação: 04

Caso de Uso: CadastrarCliente

Descrição: Este caso de uso trata do cadastro do cliente.

Ator: Cliente

Seqüência Normal

1. Cliente deseja fazer seu cadastro.
2. Sistema solicita que o cliente preencha um formulário com dados pessoais.
3. Cliente preenche seus dados e confirma seu cadastro.
4. Sistema verifica se já existe cliente cadastrado com o mesmo login.
5. Sistema cadastra cliente.

Seqüências Alternativas

4. Sistema verifica que existe cliente cadastrado com o mesmo login.
 - 4.1 Sistema solicita que o cliente digite outro login e tente o cadastro novamente.

5.4.1 Diagrama de Atividades – Cadastrar Cliente

O cadastro para novos clientes poderá ser feito pelo próprio cliente através do site, bastando preencher o formulário de cadastros. O sistema irá verificar se já existe algum login idêntico ao que foi digitado para novo cadastro, caso exista, o cliente volta para a página anterior para escolher outro login, caso o cadastro seja realizado com sucesso o cliente já poderá efetuar login no sistema.

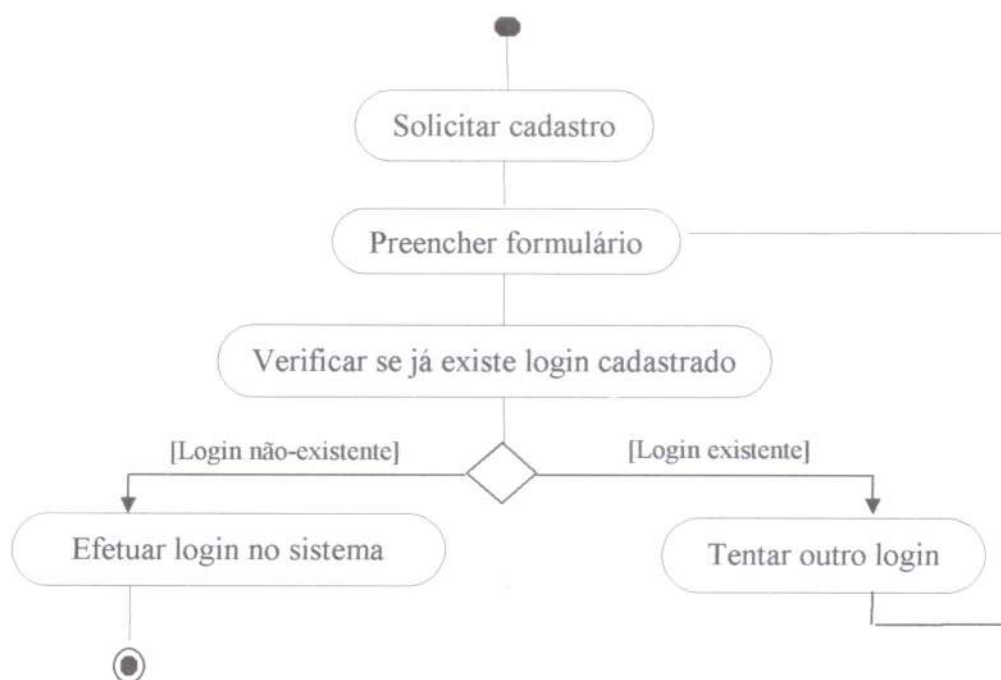


Figura 17 – Diagrama de Atividades Cadastrar Cliente
Fonte: Elaborado pelo autor

5.5 Caso de uso – Consultar Filme

Identificação: 05

Caso de Uso: ConsultarFilme

Descrição: Este caso de uso trata da consulta de filmes.

Ator: Cliente

Seqüência Normal

1. Cliente digita o nome do filme que deseja consultar.
2. Sistema procura no banco de dados pelo filme digitado.
3. Sistema encontra o filme.
4. Sistema exibe na tela as informações do filme.
5. Sistema encerra caso de uso.

Seqüências Alternativas

3. Sistema não encontra o filme.
 - 3.1 Sistema solicita que seja realizada uma nova consulta.

5.5.1 Diagrama de Atividades – Consultar Filme

Para consultar filmes o cliente terá que utilizar o formulário de busca. O sistema exibe o resultado da consulta na tela para que o cliente selecione o(s) filme(s) que deseja. Caso a consulta não encontre nenhum filme o sistema solicita que seja realizada uma nova consulta.

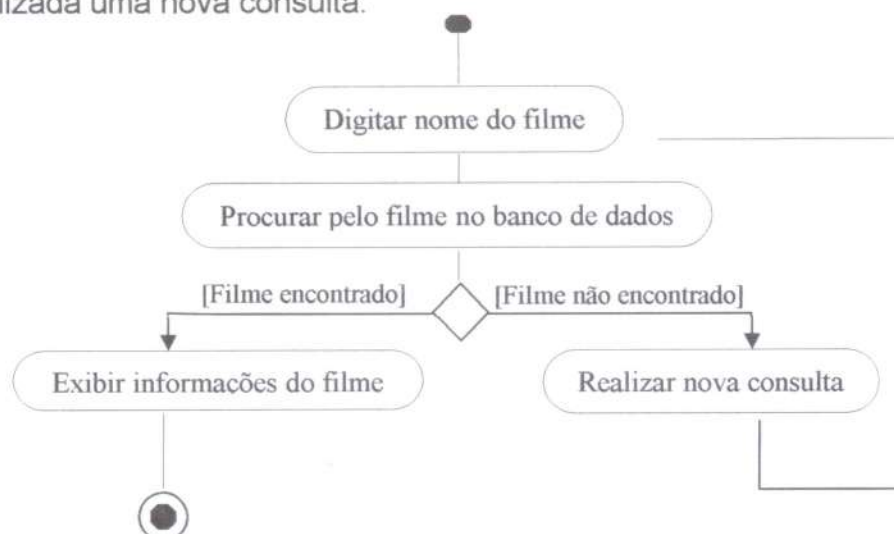


Figura 18 – Diagrama de Atividades Consultar Filme
Fonte: Elaborado pelo autor

5.6 Caso de uso – Cadastrar Filme

Identificação: 06

Caso de Uso: CadastrarFilme

Descrição: Este caso de uso trata do cadastramento de filmes.

Ator: Administrador

Seqüência Normal

1. Administrador se autentica no sistema.
2. Sistema encontra e autoriza o gerente.
3. Administrador solicita cadastro de filmes.
4. Sistema exibe na tela o formulário para cadastro de filmes.
5. Administrador preenche o formulário e confirma cadastro.
6. Sistema cadastra e volta para a tela de cadastrar novos filmes.

Seqüências Alternativas

- 6 Sistema não cadastra o filme.
 - 6.1 Sistema informa que o filme já existe.
 - 6.2 Sistema volta para a tela de cadastrar novos filmes.

5.6.1 Diagrama de Atividades – Cadastrar Filme

O processo para cadastrar filmes no banco de dados só pode ser feito pelo administrador do sistema que terá que se autenticar antes para poder ter a permissão geral do sistema. Após autenticar-se, será solicitado o formulário para cadastro de filmes. Se já existir um filme idêntico ao que está sendo cadastrado, o sistema solicita que seja preenchido novamente o formulário. Caso contrário, o filme é cadastrado com sucesso.

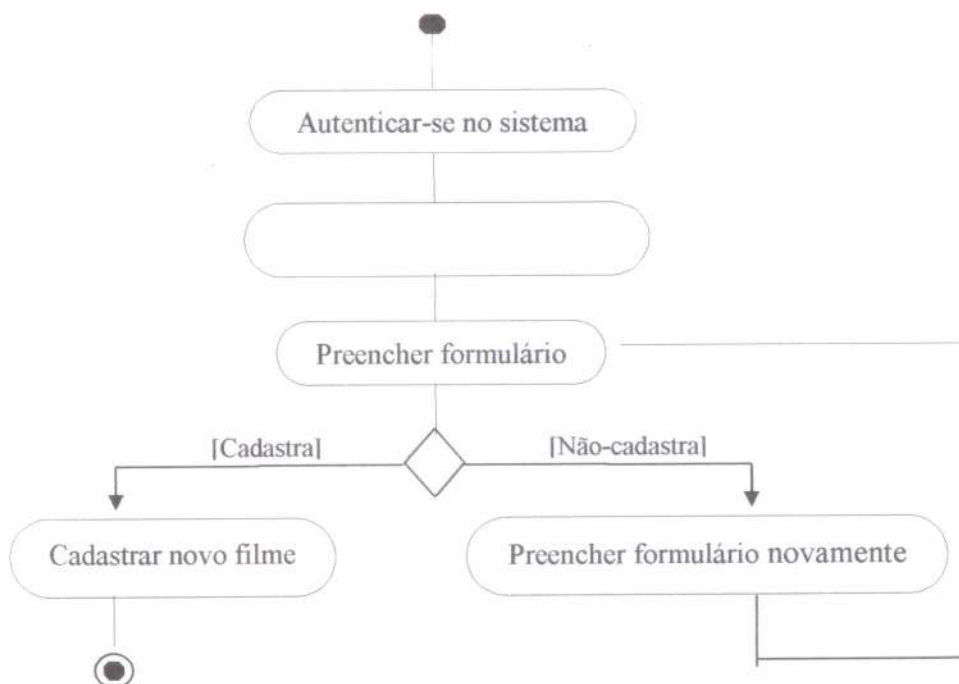


Figura 19 – Diagrama de Atividades Cadastrar Filme
Fonte: Elaborado pelo autor

5.7 Caso de uso – Imprimir relatório de filmes alugados

Identificação: 07

Caso de Uso: ImprimirRelFilmesAlugados

Descrição: Este caso de uso trata da impressão do relatório dos filmes alugados.

Ator: Administrador

Seqüência Normal

1. Administrador se autentica no sistema.
2. Administrador solicita relatório dos filmes locados.
3. Sistema gera relatório e exibe na tela para o gerente.
4. Administrador solicita impressão do relatório.
5. Sistema imprime relatório e volta para a tela de solicitação de relatórios.

Seqüências Alternativas

2. Sistema não gera relatório.

3.1 Sistema informa que o relatório não foi impresso porque nenhum filme foi locado até o momento.

2.2 Sistema volta para a tela de solicitação de relatórios.

5.7.1 Diagrama de Atividades – Imprimir relatório de filmes alugados

Após autenticação do administrador no sistema, o mesmo irá solicitar ao sistema o relatório de filmes locado por mês. O relatório será exibido na tela e impresso após solicitação do administrador. Caso não haja nenhum filme locado no mês que foi solicitado, o sistema emite uma mensagem informando o motivo.

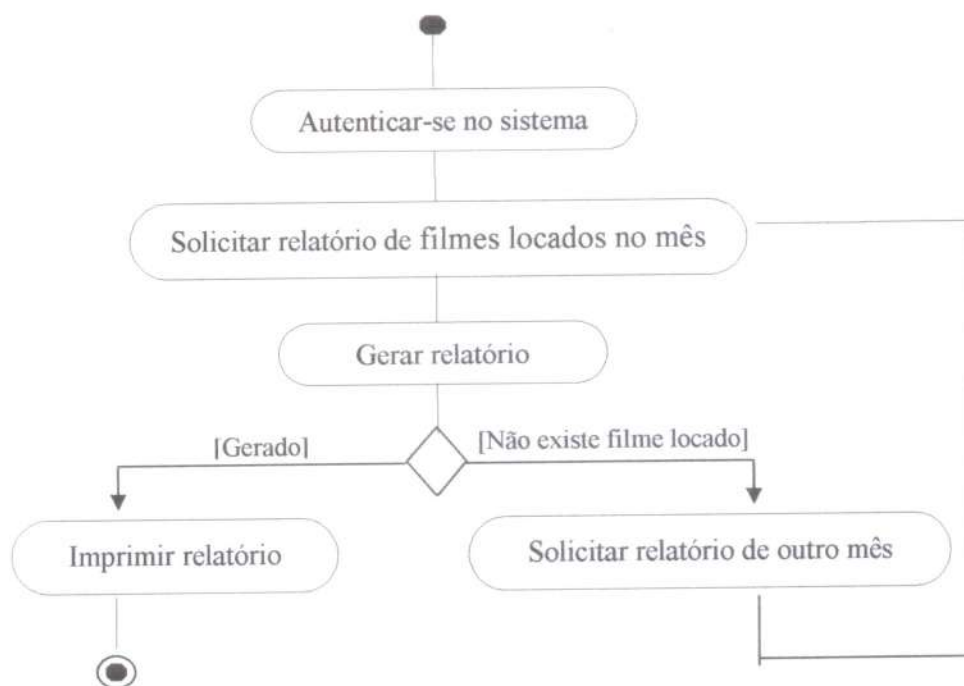


Figura 20 – Diagrama de Atividades Imprimir Relatório de Filmes Alugados
Fonte: Elaborado pelo autor

5.8 Diagrama Entidade-Relacionamento (DER)

O Diagrama de Entidade-Relacionamento tem por base a percepção do mundo real, consistindo em um conjunto de objetos básicos chamados de entidades e relacionamentos.

Entidade é uma coisa ou objeto no mundo real que pode ser identificada de forma unívoca em relação a todos os outros objetos, ou seja, é um objeto que existe e é distinguível de outros objetos. (Silberschatz, Korth e Sudarshan, 1999).

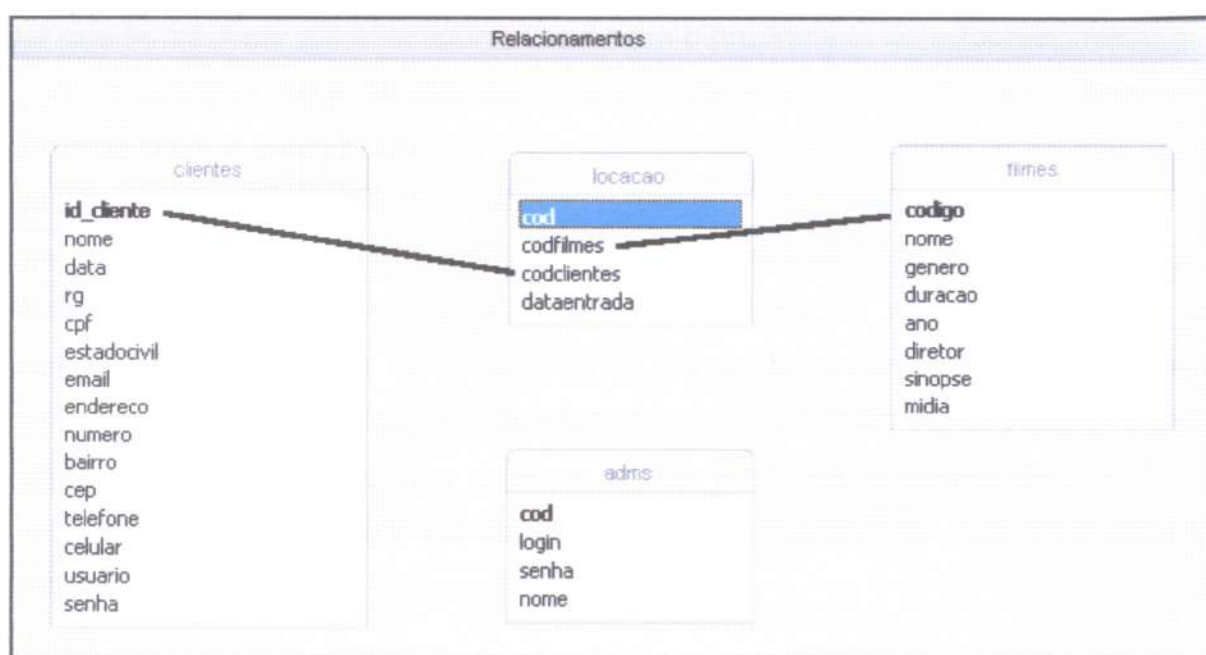


Figura 21 – Diagrama Entidade-Relacionamento (DER)

Fonte: Elaborado pelo autor

O sistema é composto pelas 4 (quatro) tabelas mostradas na Figura 21. A tabela *locacao* é responsável por armazenar as informações referente aos alugueis. Para que isto ocorra é necessário haver uma ligação entre as tabelas *clientes*, *locacao* e *filmes*. Essa ligação é feita da seguinte forma: A tabela *clientes* possui uma chave primária chamada *id_cliente*, esta que por sua vez tem ligação com *codclientes* que pertence a tabela *locacao*. A mesma coisa ocorre na tabela *filmes* que também possui uma chave primária chama *codigo*, que por sua vez tem ligação com *codfilmes* que pertence a tabela *locacao*. Já na tabela *adms* não existe relacionamento com nenhuma outra, pois não há necessidade já que nela será armazenado apenas os dados dos administradores.

6 CONCLUSÃO

De acordo com as pesquisas feitas na cidade de Natal/RN, foi observado a carência de locadoras de vídeo que disponibilizam seus serviços via web. É um fato e não se pode negar que na cidade existem muitas locadoras de vídeo, porém, poucas perceberam a importância de conquistar esse diferencial entre as demais, oferecendo seus serviços via web.

O sistema traz comodidade tanto para o cliente que pode alugar um filme sem ter que se deslocar até a locadora, como para o proprietário do estabelecimento que pode a qualquer hora do dia, em qualquer dia e em qualquer lugar, acessar o sistema e ter o controle dos clientes, aluguéis e reservas. Como já foi dito, é um sistema que traz comodidade, praticidade, pois com o sistema sendo executado num servidor web, não há a necessidade de instalação em várias máquinas caso surjam filiais da locadora. Caso haja alguma atualização a ser feita, reparos ou manutenções no sistema, as novas versões tornam-se automaticamente disponíveis para todos quando instaladas no servidor. Segurança é outro ponto positivo e muito importante do sistema, já que todos os dados ficam sob a guarda do servidor. O computador servidor ficará restrito apenas ao administrador do sistema e dono do estabelecimento.

A princípio, foi definido que o banco de dados a ser trabalhado seria o MySQL, porém, após algumas reuniões com o orientador, se chegou em um consenso que seria melhor trabalhar com outro banco de dados, o PostgreSQL. Ambos interagem muito bem com a linguagem de programação usada no desenvolvimento, que são elas PHP (*Personal Home Page*) e HTML (*Hiper Text Markup Language*).

REFERÊNCIAS

AHMED, Khawar Zaman; UMRYSH, Cary E. **Desenvolvendo Aplicações Comerciais em Java com J2EE e UML**. Rio de Janeiro: Ciência Moderna, 2002.

ALECRIM, Emerson. **Servidor Apache – o projeto**. Disponível em: <<http://www.infowester.com/servapach.php>>. Acesso em: 10 dez. 2005.

AROCA, Rafael V. **Tutorial MySQL**. Disponível em: <<http://www.linuxqos.cjb.net>>. Acesso em: 15 jun. 2005.

ARROYO, Alexandre; SANTOS, Fábio. **Programação para a Web utilizando PHP**. Disponível em: <<http://php.tutorialonline.pro.br>>. Acesso em: 29 abr. 2005.

CARDOZO, André. Guia do Webmaster. **Coleção Info Exame**. São Paulo, n.5, p.24. 2003.

CHICAYBAN, Daniel. **A HyperText Markup Language e a Web**. Disponível em: <<http://www.tol.pro.br>>. Acesso em: 28 abr. 2005.

COSTA, Dilvanir José da. **A Locação no Direito Brasileiro**. Belo Horizonte: Del Rey, 1993.

LARMAN, Craig. **Utilizando UML e Padrões: Uma Introdução a Análise e ao Projeto Orientado a Objeto**. Porto Alegre: Bookman, 2000.

LOZANO, Fernando. **Apache + PHP + MySQL**. Disponível em: <<http://www.evolucomp.com.br>>. Acesso em: 26 mai. 2005.

MACROMEDIA **Fireworks: guia do currículo**. Disponível em: <<http://www.wmonline.com.br/?area=arquivos&categoria=fireworks&subcategoria=tutoriais>>. Acesso em: 27 nov. 2005.

NIEDERAUER, Juliano. **Desenvolvendo Websites com PHP: Aprenda a criar Websites dinâmicos e interativos com PHP e banco de dados**. São Paulo: Novatec, 2004.

ROCHA, Cerli Antônio da. **Desenvolvendo Web Sites Dinâmicos: PHP, ASP e JSP**. Rio de Janeiro: Campus, 2003.

SILBERSCHATZ, A., KORTH, Henry F., SUDARSHAN, S. **Sistemas de banco de dados**. São Paulo: MAKRON Books, 1999.

SILVA, Paulo Barbosa da. **Essencial do Dreamweaver MX e MX 2004**. Disponível em: <<http://www.apostilando.com/sessao.php?cod=21>>. Acesso em: 27 nov. 2005.

STEFAN, Simkovics. **Documentação do PostgreSQL 8.0.0**. Disponível em: <<http://pgdocptbr.sourceforge.net/pg80/appendixes.html>>. Acesso em: 07 dez. 2005.

TIZZO, Neil Paiva. **Desenvolvimento de Web Sites I**. Disponível em: <<http://www.tol.pro.br>>. Acesso em: 29 abr. 2005.

TRIDAPALI, Graciela W.; SANTANNA, Juliana S. **Construindo uma aplicação de comércio eletrônico**. Disponível em: <http://www.javenessi.com.br/contri_apo.htm> Acesso em: 26 maio. 2005.

VÍVAS, Maurício. **Curso de Aplicações WEB em PHP**. Disponível em: <<http://www.tol.pro.br>>. Acesso em: 29 abr. 2005.