# 

# SENAC LAPA TITO

**CURSO TÉCNICO EM INFORMÁTICA**

**Módulo III – Desenvolvimento de Sistemas Orientado a Objetos**

**T OP MOVI ES**

Filipe Amaral

Mayara Pedroso

Projeto de Pesquisa apresentado no SENAC Lapa Tito como um dos pré-requisitos para conclusão do Módulo III do Curso Técnico em Informática, sob orientação

do Docente: Valter Vigati

#### **São Paulo**

**Setembro/ 2009**

**ÍNDICE**

##### Introdução 04

##### Título 05

##### Tema: 06

##### Objetivo Geral 07

##### Objetivos Específicos 08

##### Ata de Reunião com o Cliente: 09-10

##### Cronograma do Projeto 11

##### Contrato com o Cliente 12-18

##### Nota Fiscal 19

##### Configuração de Hardware e Software 20

##### a) Mínima 20

##### b) Recomendada 20

##### Diagrama de Atividades 21

##### Diagrama de Classes 22

##### Modelagem de Dados 23

##### Diagrama de Fluxo de Dados. 24

##### Estrutura do Banco de Dados 25

##### a) Relacionamentos 25-26

##### b) Nomes dos Campos 27-29

##### Plano de Teste do Sistema 30-31

##### Manual do Usuário 32-41

##### Manual de Comandos 42-118

##### Dicionário de Dados 119-121

##### Conclusão 122

##### Referências Bibliográficas 123

**INTRODUÇÃO**

Com crescimento acelerado da tecnologia é preciso estar conectado a tudo que for preciso para o grande salto profissional. É com esse propósito que os profissionais da área de T.I devem trabalhar e cada vez mais se dedicar.

Nesse projeto iremos mostrar um pouco dessa dedicação.

Esse projeto será avaliado pelos docentes do SENAC que estarão analisando passo a passo tudo o que nos foi ensinado durante o período de aula. O objetivo desse nosso projeto não é só apresentar um software desenvolvido para uma vídeo locadora, mais sim provar e mostrar todo nosso conhecimento e aprendizado desse terceiro módulo.

Este projeto foi muito trabalhoso, porém, obtivemos ótimos resultados. Neste foi criado um software de cadstro de funcionários, clientes, fornecedores, mídias e filmes para uma locadora de pequeno porte, com toda a pesquisa necessária para saber da necessidade do cliente.

Contudo, será visto que, além de um projeto de finalização de módulo, se fez também um projeto de qualidade que possa ser usado mais adiante, tornando-se algo verdadeiro e utilizável.

**TOP MOVIES**

Título dado para representar a empresa para a qual estamos prestando serviços.

A Top Movies é uma pequena empresa que trabalha com locação de filmes de quase todos os gêneros. Nesta, implementaremos um software de cadastro de funcionários, clientes, fornecedores, mídias e filmes, ou seja, um software simples utilizado para armazenamento de dados necessários para qualquer tipo de informação, conforme a necessidade inicial da empresa.

**TEMA**

O tema deste projeto é a implementação de um software básico à empresa Top Movies. Este projeto abrange as necessidades dos usuários, contudo, a sua necessidade de armazenamento de informações importantes e necessárias para seu trabalho.

Com o nosso software, elaborado especialmente para a Top Movies, a empresa terá a melhor agilidade e confiança nos funcionários, que trabalharão com um software adaptado para seu tipo de função.

Instalaremos no computador o software que fará a manipulação das informações de todos aqueles que tenham necessidade de serem armazenados.

**OBJETIVO GERAL**

A Imhotep Technology estará implantando na Empresa Top Movies um software de cadastro de clientes.

Será feito todo o levantamento de dados sobre a necessidade de armazenamento de dados dos funcionários, clientes, fornecedores, mídias e filmes.

Nosso maior objetivo é implementar um software simples de ser utilizado e entendido pelos seus usuários.

**OBJETIVO ESPECÍFICO**

Com o intuito de instalar um software básico, ágil e simples na empresa Top Movies, para que tudo ocorra corretamente conforme o que nos foi passado sobre a necessidade da empresa, será feito um levantamento de dados sobre os dados que serão necessários armazenar e também a importância de cada um deles.

Após o levantamento de dados, será feita a pesquisa de tipos de formulários adaptáveis para uma locadora. Depois, serão listados os campos para cadastro dos funcionários, dos clientes, dos fornecedores, das mídias e dos filmes. E assim, será feita toda a manipulação e desenvolvimento dos dados do formulário.

Com todos os formulários de cadastro funcionando corretamente, serão feitas as instalações nos computadores dos usuários e o treinamento adequado dos mesmos.

**ATA DE REUNIÃO COM O CLIENTE**

Foi solicitado pelo cliente um software que fizesse o cadastro e guardasse informações sobre seus:

* Cliente Titular e Adicional;
* Funcionários;
* Fornecedores;
* Mídias;
* Filmes;
* Locações e Reservas.

O cliente necessita de um software que faça o cadastro do cliente titular, armazenando seu nome, RG, CPF, endereço completo contendo seu CEP, bairro, cidade, seus números de telefone, celular e seu e-mail.

Devem ter campos de cadastro de cliente titular e seus adicionais, contendo os mesmos dados que o primeiro.

Para os fornecedores, será necessário o armazenamento da Razão Social, Nome Fantasia, CNPJ, algum nome de contato da empresa, seus telefones, e-mail, endereço, CEP, bairro, cidade e estado.

É necessário também cadastro dos funcionários, com nome, horário de trabalho, data de admissão, salário, cargo, seus dados pessoais, que será padrão no software e endereço completo, também padrão.

Por ser uma locadora de pequeno porte, o cliente não quis aderir a grandes tecnologias ou recursos que poderíamos no software oferecer, porém, nos comprometemos à, assim que o cliente solicitar, dar um Up-date no seu programa.

É necessário que tenha uma maneira rápida de busca através de código ou nome. Isso de adere para todos os formulários.

Os formulários de cadastro devem ser simples e de fácil entendimento para o usuário.

A locadora é simples e não necessita de muitos dados para seu funcionamento. Ela é uma locadora de bairro e, por isso, trabalha apenas com a mídia de DVD, que é a mais popular entre os clientes.

Será feito o cadastro das locações de filmes no nome do cliente e a reserva de filmes que não estiverem disponíveis na locadora para o cliente.

Os gêneros de filmes serão os mais simples, por isso não é necessário o bloqueio de alguns para menores de 18 anos.

Assim sendo, finaliza-se a reunião com todos os dados ditos como necessário para um bom funcionamento do sistema e para melhor agilidade e facilidade dos usuários em seu uso.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CRONOGRAMA DO PROJETO** | | | | |
| **Ident.** | **Nome da Tarefa** | **Início** | **Término** | **Duração** |
| 1 | Reunião com o Cliente | 22/06/2009 | 22/06/2009 | 1 Dia |
| 2 | Levantamento de Dados | 23/6/2009 | 29/6/2009 | 05 Dias |
| 3 | Elaboração e Entrega da Proposta | 30/6/2009 | 10/7/2009 | 09 Dias |
| 5 | Aprovação da Proposta | 13/7/2009 | 17/7/2009 | 05 Dias |
| 6 | Desenvolvimento do Sistema | 20/7/2009 | 23/09/2009 | 51 Dias |
| 7 | Entrega da versão Beta do Sistema | 23/09/2009 | 23/09/2009 | 1 Dia |
| 8 | Teste dos usuários na versão beta | 23/09/2009 | 29/09/2009 | 5 Dias |
| 9 | Entrega e Apresentação do Sistema | 30/9/209 | 30/9/2009 | 1 Dia |
| 10 | Treinamento dos Usuários | 1/10/2009 | 5/10/2009 | 5 Dias |
| 11 | Finalização do Projeto | 5/10/2009 | 5/10/2009 | 1 Dia |

**CONTRATO DE PRESTAÇÃO DE SERVIÇOS DE INFORMÁTICA**

**CONTRATANTE:** VIDEO LOCADORATOP MOVIES ME., inscrita no CNPJ n.º 01.305.089.0001/23, com sua localização na Avenida dos Remédios n.º 190, – CEP. 01310-200 – Vila dos Remédios - São Paulo, no ramo de locações de filmes, neste ato representada por seu proprietário ROBERTO DA SILVA SANTOS, brasileiro, solteiro, administrador, portador da cédula de identidade RG n.º 24.707.701 -3, inscrito no CPF/MF n.º 123.321.456-0.

**CONTRATADA:** COMPANHIA DE ASSESSORIA TÉCNICA EM INFORMÁTICA IMHOTEP TECNOLOGY LTDA., inscrita no CNPJ n.º 01.302.509.0001/29, com sua localização na Avenida Paulista n.º 1912, conjunto 31, 3º andar – CEP. 01310-200 – Jardim Bela Vista – São Paulo, representada por MAYARA PEDROSO, solteira, empresária técnica, portadora da cédula de identidade RG n.º 23.121.988-1 SSP/SP, inscrita no CPF/MF n.º 343.367.256-02.

**A CONTRATANTE**, empresa cuja atividade é a área de locações de filmes, a fim de atender ao que determinam os artigos 27 e 28 da Lei nº 2.800/56, firma o presente contrato com a CONTRATADA, a qual obriga-se a prestar à CONTRATANTE serviços profissionais atinentes a sua formação tecnológica e habilitação profissional na área de informática e a assumir a responsabilidade técnica perante o referido órgão de fiscalização profissional que lhe exijam.

Parágrafo único – A CONTRATADA prestará a CONTRATANTE as seguintes atividades: desenvolver um software de cadastro de filmes, clientes, funcionários e fornecedores, a pós o programa ser instalado, assessoria, orientação técnica, enfim, tudo o que for pertinente à sua capacitação técnica em informática para manter a regularidade da atividade explorada pela CONTRATANTE.

Pelo presente instrumento particular de Contrato de Prestação de Serviços e de Assunção de Responsabilidade Técnica, as partes acima qualificadas têm entre si justo e avençado o seguinte:

**CLÁUSULA PRIMEIRA: OBJETO**

1.1 A CONTRATADA obriga-se a prestar a CONTRATANTE, serviços de desenvolvimento, intalação, manutenção e atualização do software, pelo período estipulado por ela mesma “A CONTRATADA” para a realização dos serviços pelos quais ela se comprometeu a realizar nas instalações da CONTRATANTE identificadas no preâmbulo, serviços estes que serão mais bem descritos e caracterizados na cláusula Segunda, item 2.7.

1.2 A CONTRATADA se compromete a disponibilizar 01 (um) Analista na CONTRATANTE em um dia fixo da semana, pelo período do plano escolhido para atendimento semanal ou quinzenal, conforme plano escolhido. Caso haja necessidade, será disponibilizado um Analista para atender um Chamado extra, na CONTRATANTE, que será faturado de acordo com o disposto no Anexo 1, item 2, (trabalho extra).

**CLÁUSULA SEGUNDA: EXECUÇÃO DOS SERVIÇOS**

2.1 A CONTRATADA executará os serviços de desenvolvimento, intalação, manutenção e atualização do software, colocando à disposição da CONTRATANTE, de acordo com o escopo descrito na cláusula Segunda item 2.7, analista em informática pelo período contratado, a título de atendimento semanal ou quinzenal.

2.2 A CONTRATADA se obriga a atender aos Chamados Extras, no prazo máximo de 48 (quarenta e oito horas) depois de caracterizado o chamado, para manutenção, sendo cobrado o valor conforme descrito no Anexo 1, mais os acréscimos decorrentes de atividades extras, que ocasionalmente possam ocorrer em horário extraordinário ou excedente à quantidade de horas, cobertas por este contrato.

2.3 A CONTRATADA se obriga a manter os profissionais à disposição da CONTRATANTE durante o tempo contratado no horário comercial, ficando, desde já, estabelecido como horário comercial o citado no Anexo I.

2.4 A CONTRATANTE é a única responsável pelos direitos sobre o programa, informações, relatórios, projetos e documentos por ela elaborados ou adquiridos.

2.5 Os serviços contratados, serão cobrados mensalmente, por meio de cobrança bancária. Os serviços prestados, excedentes aos contratados, serão cobrados mensalmente, por meio de cobrança bancária, baseada na ficha de Ficha de Atendimento, preenchida pela CONTRATADA e aprovada pela CONTRATANTE.

2.6 Caso haja necessidade de prestação de serviços em laboratório, será fornecido orçamento prévio e haverá cobrança extra. Todos os reparos e configurações, são de inteira responsabilidade da CONTRATANTE, serão adquiridos através da CONTRATADA, após orçamentos previamente aprovados pela CONTRATANTE. Caso a CONTRATANTE adquira atualização ou manutençaõ do software de outra que não seja a CONTRATADA esse deverá ser inspecionada e aprovada pela CONTRATADA por escrito.

2.7 A CONTRATADA se obriga a prestar os serviços até o praso combinado com a CONTRATANTE.

**CLÁUSULA TERCEIRA: PRAZO**

3.1 O presente contrato vigorará por um prazo de (30) trinta dias, a partir da data do aceite e primeiro pagamento, podendo ser rescindido a qualquer tempo, sem prejuízo no disposto na cláusula décima quarta, por qualquer uma das partes, desde que dada a ciência expressa à outra parte com 30 (trinta) dias de antecedência, independentemente de qualquer multa ou indenização.

**CLÁUSULA QUARTA: PREÇO E PAGAMENTO**

4.1 Pelos serviços ora contratados, a CONTRATANTE pagará o valor R$ 60,00 a hora de trabalho por 53 dias trabalhando por 06 horas diárias, totalizando o valor de de R$ 19.000,00, referente aos serviços de desenvolvimento, instalação, manutenção e atualização do software, e treinamento dos usuarios.

**CLÁUSULA QUINTA: OBRIGAÇÕES DA CONTRATANTE**

5.1 A CONTRATANTE obriga-se a:

- Definir com precisão as prioridades para o trabalho da CONTRATADA.

- Fornecer todas as informações que entender necessárias para colaborar no desenvolvimento dos trabalhos da CONTRATADA.

- Disponibilizar acomodações para o profissional no período em que este estiver residente.

**CLÁUSULA SEXTA: ENCARGOS SOCIAIS**

6.1 O presente contrato não importa em vínculo de ordem trabalhista entre a CONTRANTE e os profissionais/cooperados da CONTRATADA, já que a mesma, como prestadora de serviço não é subordinada à CONTRATANTE, eis que é civil a relação decorrente do presente contrato, decisão esta ratificada pela lei nº 8.949/94 que acrescentou parágrafo único artigo 442 da CLT

**CLÁUSULA SÉTIMA: DESPESAS CONTRATUAIS**

7.1 Todas as despesas decorrentes do presente contrato, tais como deslocamento e alojamento referentes a atendimentos que não seja no local especificado, serão de responsabilidade exclusiva da CONTRATANTE.

**CLÁUSULA OITAVA: ENCARGOS SOCIAIS**

8.1 Este contrato não criará qualquer vínculo empregatício entre a CONTRATANTE e a CONTRATADA, seus funcionários e/ou prepostos, correndo por conta desta, todos os encargos e exigências fiscais, trabalhistas, previdenciárias e acidentários de seus funcionários e/ou prepostos que executarem os serviços.

8.2 A CONTRATANTE se obriga a não contratar qualquer profissional da CONTRATADA, durante 5 (Cinco) anos a partir da expiração do prazo contratual, sob pena de uma multa de 50 (Cinqüenta) vezes o valor médio mensal pago pela CONTRATANTE para a CONTRATADA, conforme apurado na data do respectivo evento.

**CLÁUSULA NONA: CESSÃO DO CONTRATO**

9.1 É vedado a qualquer das partes ceder ou transferir, total ou parcialmente, a título oneroso ou gratuito, os direitos e obrigações decorrentes deste contrato.

**CLÁUSULA DÉCIMA: SIGILO**

10.1 A CONTRATADA obriga-se a manter o mais absoluto sigilo e confidenciabilidade de todas e quaisquer informações obtidas e consideradas confidenciais pela CONTRATANTE, direta e indiretamente, em decorrência do presente contrato.

10.2 A CONTRATADA fica expressamente proibida de utilizar material ou informações da CONTRATANTE para executar trabalhos similares a terceiros.

**CLÁUSULA DÉCIMA PRIMEIRA: DOCUMENTAÇÃO**

11.1 Toda documentação contratual, de produtos e softwares, necessários à fiscalização deverá permanecer no local da prestação de serviços, sendo de inteira responsabilidade da CONTRATANTE.

**CLÁUSULA DÉCIMA SEGUNDA: NORMAS**

12.1 Obriga-se a CONTRATADA a respeitar todas as normas de segurança e de ordem disciplinar inerentes ao seu ramo de atividade.

**CLÁUSULA DÉCIMA TERCEIRA: OBRIGAÇÕES DOS SUCESSORES**

13.1 O presente contrato obriga as partes contratantes e seus sucessores.

**CLÁUSULA DÉCIMA QUARTA: RESCISÃO E MULTA**

14.1 O presente poderá ser considerado rescindido, independentemente de notificação judicial ou extrajudicial, e de qualquer ônus para a parte inocente, nas seguintes hipóteses, além das previstas em lei:

- Descumprimento das obrigações assumidas neste contrato por um período superior a 15 (Quinze) dias;

- Dissolução judicial ou extrajudicial da CONTRATANTE;

14.2 A parte que inadimplir qualquer das cláusulas e condições deste contrato, incorrerá na multa de 20% (vinte por cento) sobre o valor total do contrato, sem prejuízo da apuração das perdas e danos decorrentes do inadimplemento.

14.3 Na hipótese de rescisão de contrato, os equipamentos com trabalhos em andamento serão entregues à CONTRATANTE, no estágio que estiverem.

**CLÁUSULA DÉCIMA QUINTA: CONDIÇÕES COMERCIAIS**

15.1 Horário Comercial:

Fica estabelecido como horário comercial, o horário entre 09h00min às 16h00min.

15.2 Horário e Atendimento Extraordinário:

Fica estabelecido como horário extraordinário, o horário não correspondente ao horário comercial.

15.3 Freqüência do Atendimento:

Fica estabelecido o atendimento em dias fixos da semana, apenas um atendimento contratado poderá ocorrer na mesma semana. Onde é considerado para este contrato as quatro semanas do mês.

15.4 Os serviços serão faturados por cobrança bancária, cujos boletos serão enviados por e-mail à CONTRATANTE, onde os vencimentos serão ao dia 01 (primeiro) de cada mês.

Caso haja a necessidade de trabalhos extras, (atividades ou horários) os mesmos serão faturados para o próximo mês.

**E assim, por estarem justos e combinados, assinam em (2) duas vias de igual teor diante as testemunhas abaixo**:

**São Paulo, 30 de Setembro de 2009.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**VÍDEO LOCADORA TOP MOVIES ME.**

**ROBERTO DA SILVA SANTOS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**COMPANHIA DE ASSESSORIA TÉCNICA EM INFORMÁTICA IMHOTEP TECNOLOGY LTDA.**

**MAYARA PEDROSO**

Testemunhas Testemunhas

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **IMHOTEP TECNOLOGY**  COMPANHIA DE ASSESSORIA TÉCNICA EM INFORMÁTICA IMHOTEP TECNOLOGY LTDA  E-mail: imhotep@tecnology.com.br  Fone: 3283-3303  Av. Paulista n.º 1912, conjunto 31, 3º andar – Jardim Bela Vista – São Paulo | | | | | CNPJ n.º 01.302.509.0001/29 I.E. 116.617.616.114  N. FISCAL DE VENDA A CONSUMIDOR Nº 02313  1ª Via Cliente – 2ª Via Controle – 3ª Fixa    DATA: 1/ 09 / 2009  CLIENTE: VÍDEO LOCADORA TOP MOVIES ME  ENDEREÇO: Avenida dos Remédios n.º 190 | | | | | **Qtde.** | **Discriminação / Produto** | **Unitário** | **Valor Hora** | | 1 | Software de controle de locadora | 1 | R$ 60,00 | | **Obs.:** |  |  |  | | **TOTAL** | **R$ 19.000,00** | | |

**CONFIGURAÇÃO DE HARDWARE E SOFTWARE**

1. Mínima
   1. Processador Intel Pentium 4 - 1.8 GHz
   2. Memória RAM de 512 MB
   3. Espaço livre no disco rígido de 10 GB
   4. Display- 800x600, 16 - bit color
   5. Windows XP/Vista
   6. Servidor de Banco de Dados
2. Recomendada
   1. Processador Intel Pentium Dual Core – 2.1 GHz
   2. Memória RAM de 1 GB
   3. Espaço livre no disco rígido de 20 GB
   4. Display- 800x600, 16 - bit color
   5. Windows XP/Vista
   6. Servidor de Banco de Dados

**DIAGRAMA DE ATIVIDADES**



**DIAGRAMA DE CLASSES**



**MODELAGEM DE DADOS**

**DIAGRAMA DE FLUXO DE DADOS**



**ESTRUTURA DO BANCO DE DADOS**

1. Relacionamentos

Tabela Cliente Titular – Campo ID Adcional com

Tabela Cliente Adicional – Campo ID Adicional

Tabela Cliente Titular – Campo CEP com

Tabela Endereço – Campo CEP

Tabela Cliente Titular – Campo ID Dados Pessoais com

Tabela Dados Pessoais – Campo ID Dados Pessoais

Tabela Cliente Adcional – Campo CEP com

Tabela Endereços – Campo CEP

Tabela Cliente Adicional – Campo ID Dados Pessoais com

Tabela Dados Pessoais – Campo ID Dados Pessoais

Tabela Funcionários – Campo CEP com

Tabela Endereços – Campo CEP

Tabela Funcionários – Campo ID Dados Pessoais com

Tabela Dados Pessoais – Campo ID Dados Pessoais

Tabela Fornecedor – Campo CEP com

Tabela Endereços – Campo CEP

Tabela Mídia – Campo ID Filme com

Tabela Filmes – Campo ID Filme

Tabela Locação – Campo ID Titular com

Tabela Cliente Titular – Campo ID Titular

Tabela Locação – Campo ID Adicional com

Tabela Cliente Adicional – Campo ID Adicional

Tabela Locação – Campo ID Filme com

Tabela Filmes – Campo ID Filme

Tabela Reserva – Campo ID Titular com

Tabela Cliente Titular – Campo ID Titular

Tabela Reserva – Campo ID Adicional com

Tabela Cliente Adicional – Campo ID Adicional

Tabela Reserva – Campo ID Filme com

Tabela Filmes – Campo ID Filme

1. Nomes dos Campos

|  |  |
| --- | --- |
| **CLIENTE\_ADICIONAL** | |
| **Campos** | **Tipo de Campo** |
| ID\_Adicional | int - NotNull |
| ID\_Titular | int - Null |
| Nome | varchar (80) - Null |
| Cep | char (9) - Null |
| ID\_Dados\_Pessoais | int - Null |
| Grau\_Parentesco | varchar (10) - Null |

|  |  |
| --- | --- |
| **CLIENTE\_TITULAR** | |
| **Campos da Tabela** | **Tipo de Campo** |
| ID\_Titular | Int – NotNull |
| Nome | varchar (80) - Null |
| Cep | char (9) - Null |
| ID\_Dados\_Pessoais | Int - Null |

|  |  |
| --- | --- |
| **DADOS\_PESSOAIS** | |
| **Campos** | **Tipo de Campo** |
| ID\_Dados\_Pessoais | int - NotNull |
| RG | char (12) - Null |
| CPF | char (14) - Null |
| Sexo | char (10) - Null |
| Data\_Nascimento | Datetime - Null |
| Telefone | varchar (15) - Null |
| Celular | varchar (15) - Null |
| E\_mail | varchar (30) - Null |

|  |  |
| --- | --- |
| **ENDEREÇOS** | |
| **Campos** | **Tipo de Campo** |
| Cep | char (9) - NotNull |
| Logradouro | varchar (50) - Null |
| Número | char (10) - Null |
| Complemento | char (10) - Null |
| Bairro | varchar (30) – Null |
| Cidade | varchar (30) – Null |
| Estado | char (10) – Null |

|  |  |
| --- | --- |
| **FILMES** | |
| **Campos** | **Tipo de Campo** |
| ID\_Filme | int - NotNull |
| ID\_Midia | int - Null |
| Filme | varchar (50) - Null |
| Ano\_lançamento | char (5) - Null |
| Gênero | varchar (30) - Null |
| Nacionalidade | varchar (30) - Null |
| Duração | int - Null |
| Direção | varchar (50) - Null |
| Sinopse | text - Null |
| Preço | smallmoney - Null |

|  |  |
| --- | --- |
| **FORNECEDORES** | |
| **Campos** | **Tipo de Campo** |
| ID\_Fornecedor | int - NotNull |
| Razão\_Social | varchar (80) - Null |
| Nome\_Fantasia | varchar (80) - Null |
| CNPJ | char (18) - Null |
| Contato | varchar (30) - Null |
| Telefone | varchar (20) - Null |
| Fax | varchar (20) - Null |
| E\_mail | varchar (50) - Null |
| Cep | char (9) - Null |

|  |  |
| --- | --- |
| **FUNCIONÁRIOS** | |
| **Campos** | **Tipo de Campo** |
| ID\_Funcionário | int – NotNull |
| Nome | varchar (50) – Null |
| Horário\_Trabalho | varchar (10) – Null |
| Data\_Admissão | smalldatetime – Null |
| Cargo | varchar (20) – Null |
| Cep | char (9) |
| ID\_Dados\_Pessoais | int – Null |
| Salário | smallmoney – Null |
| Estado\_Civil | varchar (15) |

|  |  |
| --- | --- |
| **LOCAÇÃO** | |
| **Campos** | **Tipo de Campo** |
| ID\_Locacao | int - NotNull |
| Cod\_Cliente | int - Null |
| Nome | varchar (50) - Null |
| Tipo\_Cliente | char (3) - Null |
| Cod\_Filme | int - Null |
| Filme | varchar (50) - Null |
| Data\_Retirada | smalldatetime - Null |
| Data\_Devolucao | smalldatetime - Null |
| Preco | smallmoney - Null |
| Valor\_Locacao | smallmoney - Null |
| Locacao\_Paga\_Ato | char (3) - Null |

|  |  |
| --- | --- |
| **Mídias** | |
| **Campos** | **Tipo de Campo** |
| ID\_Midia | int - NotNull |
| Midia | varchar (15) - Null |

|  |  |
| --- | --- |
| **RESERVA** | |
| **Campos** | **Tipo de Campo** |
| ID\_Reserva | int - NotNull |
| Cod\_Cliente | int - Null |
| Nome | varchar (50) - Null |
| Tipo\_Cliente | char (3) - Null |
| Telefone | varchar (15) - Null |
| Cod\_Filme | int - Null |
| Filme | varchar (50) - Null |
| Data\_Reserva | smalldatetime - Null |
| Preco | smallmoney - Null |

**PLANO DE TESTE NO SISTEMA**

Abrimos o formulário de entrada de dados dos clientes adicional e titular e fizemos cadastros com o botão de Incluir. Após o teste de inclusão de dados, foram feitos os testes de gravação dos mesmos no Banco de Dados. Fizemos alterações com o botão Alterar e fizemos exclusões com o botão Excluir. Todos os botões testados, quando indicaram erros, foram reconfigurados e testados novamente.

Abrimos o formulário de entrada de dados dos fornecedores e fizemos cadastros com o botão de Incluir. Após o teste de inclusão de dados, foram feitos os testes de gravação dos mesmos no Banco de Dados. Fizemos alterações com o botão Alterar e fizemos exclusões com o botão Excluir. Todos os botões testados, quando indicaram erros, foram reconfigurados e testados novamente.

Abrimos o formulário de entrada de dados dos funcionários e fizemos cadastros com o botão de Incluir. Após o teste de inclusão de dados, foram feitos os testes de gravação dos mesmos no Banco de Dados. Fizemos alterações com o botão Alterar e fizemos exclusões com o botão Excluir. Todos os botões testados, quando indicaram erros, foram reconfigurados e testados novamente.

Abrimos o formulário de entrada de dados dos filmes e mídias e fizemos cadastros com o botão de Incluir. Após o teste de inclusão de dados, foram feitos os testes de gravação dos mesmos no Banco de Dados. Fizemos alterações com o botão Alterar e fizemos exclusões com o botão Excluir. Todos os botões testados, quando indicaram erros, foram reconfigurados e testados novamente.

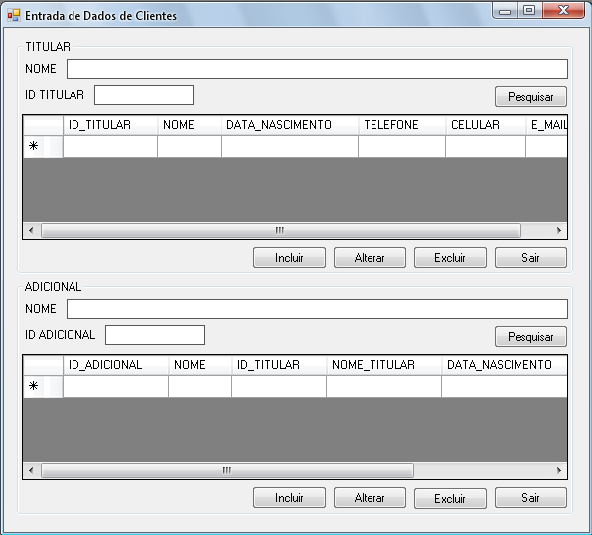
Após o cadastro dos clientes, das mídias e dos filmes, fizemos algumas locações e reserva de filmes no nome dos clientes. Assim, pudemos testar o botão de Inserir mais de um filme para um só cliente, gravar a locação e a reserva feita pelo cliente e também excluir uma locação ou reserva de algum filme já salvo.

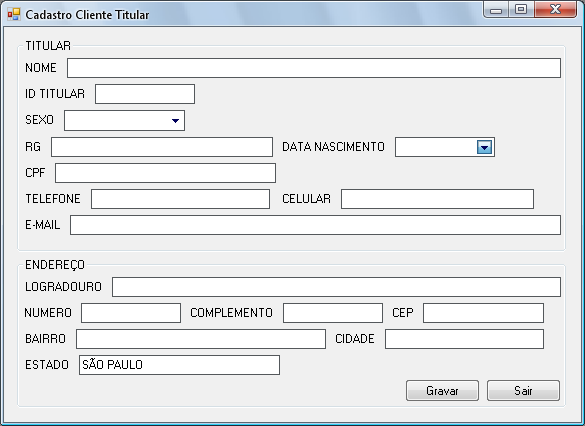
Assim então, pudemos depois confirmar nas telas de consulta se as locações e reservas foram salvas no Banco de Dados, conforme o que se foi programado.

Todas as telas e botões que apresentaram erro no teste, foram analisadas, reprogramadas e novamente testadas, para a apresentação do software para a Top Movies estar conforme o solicitado pela empresa.

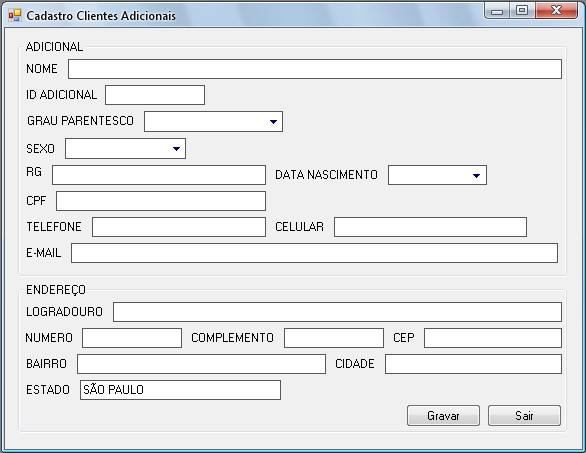
.

**MANUAL DO USUÁRIO**

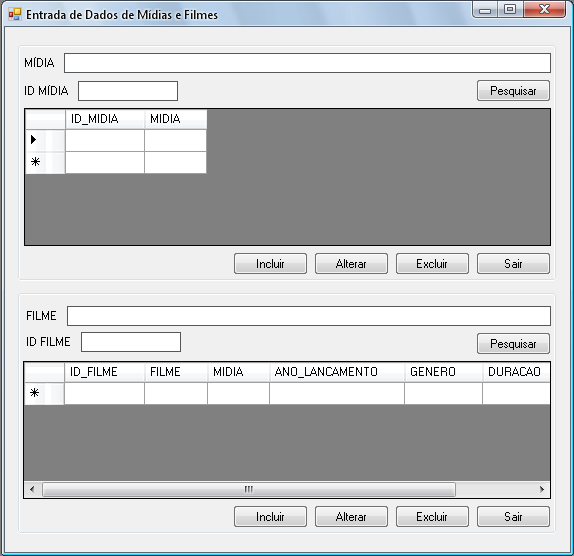
Esta é a tela de entrada de dados dos clientes. Nela é possível fazer a busca do nome do cliente adicional ou titular ou também fazer a busca pelo ID, ou seja, o código do cliente.Tem a opção de incluir, alterar ou excluir algum cliente. Os botões incluir e alterar próximo à cliente titular chama a tela de cadastro do cliente titular, e os botões incluir e alterar próximo à cliente adicional chama a tela de cadastro do cliente adicional.



Esta é a tela de cadastro de cliente titular citada logo acima.

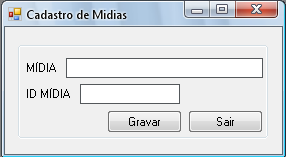


Esta é a tela de cadastro de cliente adicional citada logo acima.

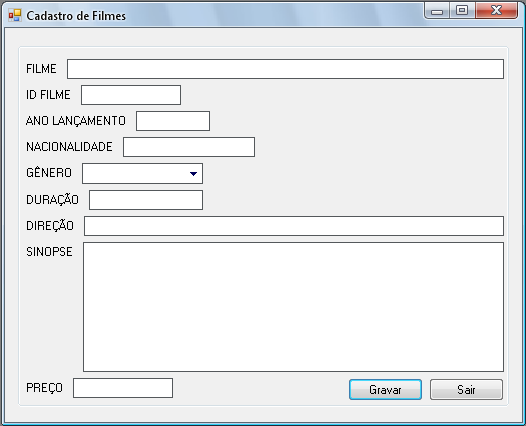


Esta é a tela de entrada de dados dos filmes e das mídias. Nela é possível fazer a busca da mídia ou nome do filme ou também fazer a busca pelo ID, ou seja, o código da mídia ou filme.

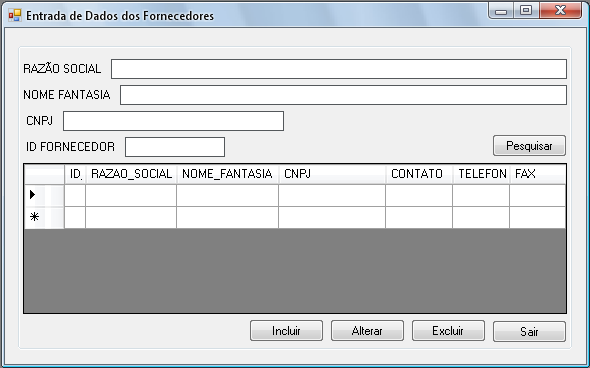
Tem a opção de incluir, alterar ou excluir alguma mídia ou filme. Os botões incluir e alterar próximo à mídia chama a tela de cadastro da mídia, e os botões incluir e alterar próximo ao filme chama a tela de cadastro dos filmes.



Esta é a tela de cadastro das mídias citada logo acima.



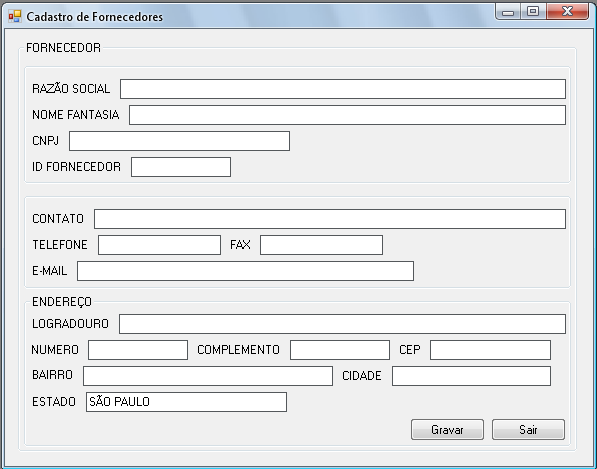
Esta é a tela de cadastro dos filmes citada logo acima.



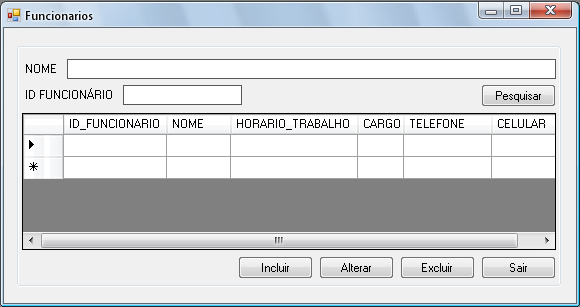
Esta é a tela de entrada de dados dos fornecedores. Nela é possível fazer a busca da razão social, nome fantasia, CNPJ ou ID, ou seja, o código do fornecedor.

Tem a opção de incluir, alterar ou excluir algum fornecedor.

Os botões incluir e alterar chama a tela de cadastro do fornecedor.

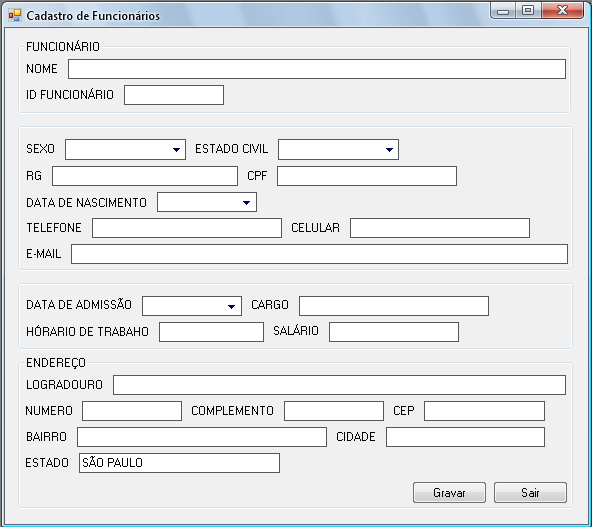


Esta é a tela de cadastro dos fornecedores citada logo acima.

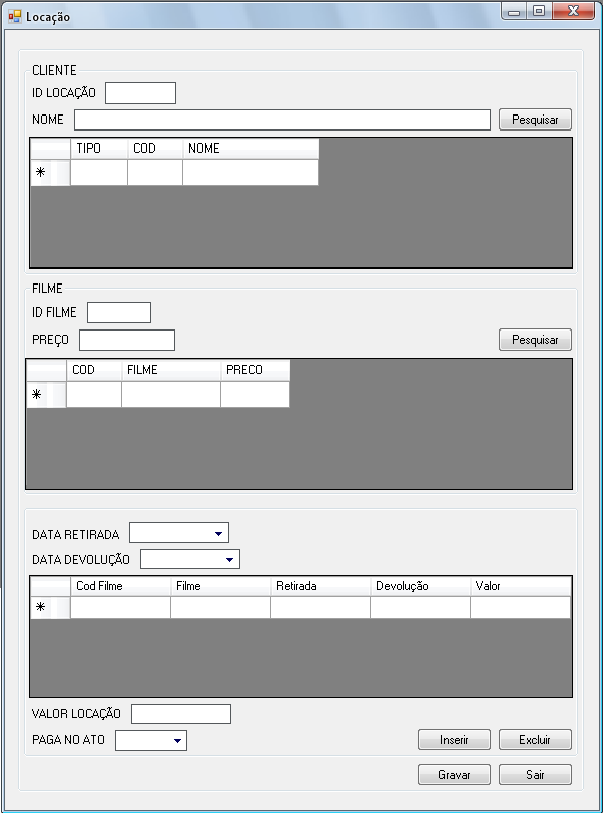


Esta é a tela de entrada de dados dos funcionários. Nela é possível fazer a busca do funcionário pelo nome ou pelo ID, ou seja, o código do funcionário.

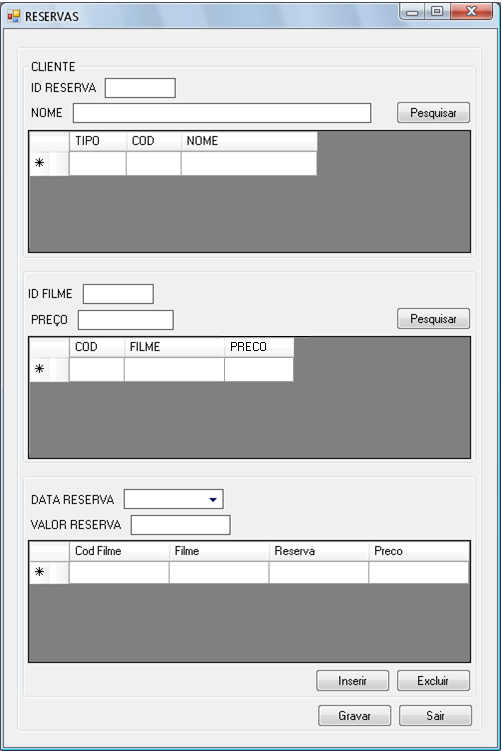
Tem a opção de incluir, alterar ou excluir algum funcionário. Os botões incluir e alterar chama a tela de cadastro do funcionário.



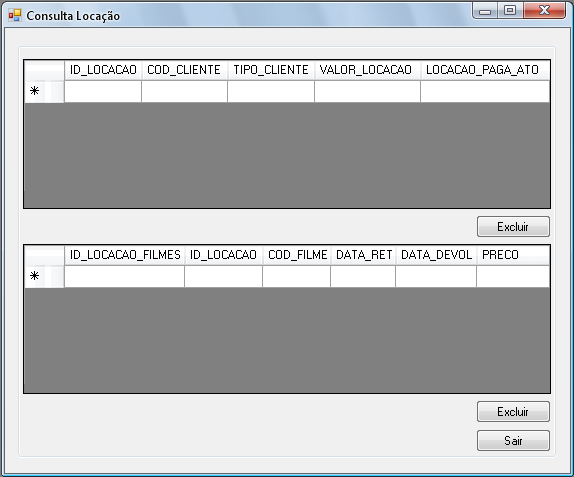
Esta é a tela de cadastro dos funcionários citada logo acima.



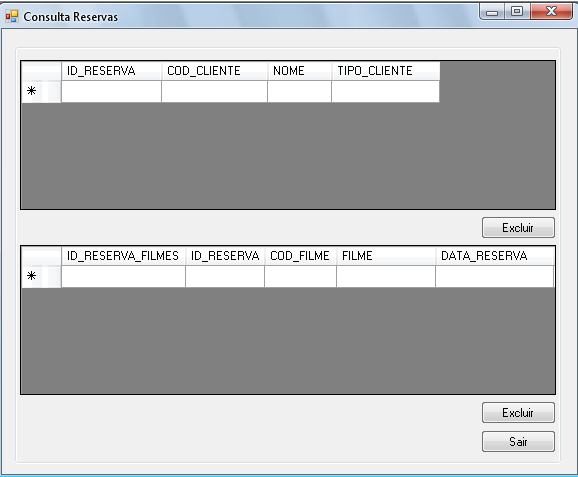
Esta é tela que o usuário fará a locação de um ou de mais filmes. Ela fará a gravação dos filmes no código de locação do cliente.



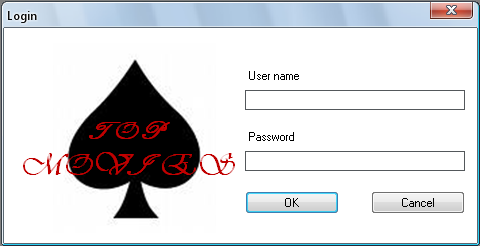
Esta é tela que o usuário fará a reserva de um ou de mais filmes. Ela fará a gravação de reserva dos filmes no código de locação do cliente.



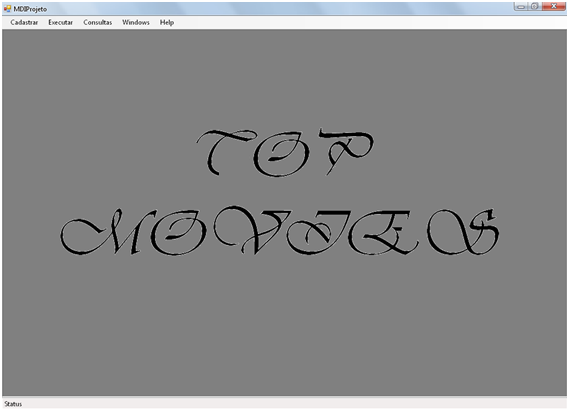
Essa é a tela em que o usuário fará consulta das locações que foram feitas.

****

Essa é a tela em que o usuário fará consulta das reservas de filmes que foram feitas.

****

Essa é a tela em que o usuário fará o Logon no sistema.



Essa é a tela em que todos os formulários são abertos para cadastro, consultas e para executar locações e reservas.

**MANUAL DE COMANDOS**

Public Class Cliente\_Adicional

Inherits Dados\_Pessoais

Private id\_adicional As Integer

Private id\_titular As Integer

Private nome As String

Private cep As String

Private id\_dados\_pessoais As Integer

Private grau\_parentesco As String

Public Property p\_id\_adicional() As Integer

Get

Return id\_adicional

End Get

Set(ByVal value As Integer)

id\_adicional = value

End Set

End Property

Public Property p\_id\_titular() As Integer

Get

Return id\_titular

End Get

Set(ByVal value As Integer)

id\_titular = value

End Set

End Property

Public Property p\_nome() As String

Get

Return nome

End Get

Set(ByVal value As String)

nome = value

End Set

End Property

Public Property p\_dados\_pessoais() As Integer

Get

Return id\_dados\_pessoais

End Get

Set(ByVal value As Integer)

id\_dados\_pessoais = value

End Set

End Property

Public Property p\_grau\_parentesco() As String

Get

Return grau\_parentesco

End Get

Set(ByVal value As String)

grau\_parentesco = value

End Set

End Property

'lista dos itens da grid

Public Function ListagemDataTableCliente\_Adicional(ByVal C\_Adicional As Cliente\_Adicional) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM CLIENTE\_ADICIONAL"

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function LstGridCliente\_adicional(ByVal C\_Adicional As Cliente\_Adicional) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere As String = " WHERE "

Dim sSql As String = ""

sSql = " SELECT "

sSql &= "ca.ID\_ADICIONAL, ca.NOME,"

sSql &= "ct.ID\_TITULAR as ID\_TITULAR,"

sSql &= "ct.NOME as NOME\_TITULAR,"

sSql &= "dp.DATA\_NASCIMENTO as DATA\_NASCIMENTO,"

sSql &= "dp.TELEFONE as TELEFONE,"

sSql &= "dp.CELULAR as CELULAR "

sSql &= " FROM CLIENTE\_ADICIONAL ca "

sSql &= " LEFT OUTER JOIN CLIENTE\_TITULAR ct ON ca.ID\_TITULAR = ct.ID\_TITULAR "

sSql &= " LEFT OUTER JOIN DADOS\_PESSOAIS dp ON ca.ID\_DADOS\_PESSOAIS = dp.ID\_DADOS\_PESSOAIS "

If C\_Adicional.p\_nome <> "" Then

sSql &= msWhere & " ca.NOME like '%" & C\_Adicional.p\_nome & "%'"

msWhere = " AND "

End If

If C\_Adicional.p\_id\_adicional > 0 Then

sSql &= msWhere & " ID\_ADICIONAL = " & C\_Adicional.p\_id\_adicional

msWhere = " AND "

End If

If C\_Adicional.p\_id\_titular > 0 Then

sSql &= msWhere & "ct.ID\_TITULAR = " & C\_Adicional.p\_id\_titular

msWhere = " AND "

End If

If C\_Adicional.p\_id\_dados\_pessoais > 0 Then

sSql &= msWhere & "dp.ID\_DADOS\_PESSOAIS = " & C\_Adicional.p\_id\_dados\_pessoais

msWhere = " AND "

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'retornar os dados do registro para alteração

Public Function RetornarDadosCliente\_Adicional(ByVal C\_Adicional As Cliente\_Adicional) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere As String = " WHERE "

Dim sSql As String = ""

If C\_Adicional.p\_id\_adicional <> 0 Then

sSql = "SELECT "

sSql &= "ca.ID\_ADICIONAL, ca.GRAU\_PARENTESCO,"

sSql &= "ca.ID\_DADOS\_PESSOAIS as ID\_DADOS\_PESSOAIS, "

sSql &= "ca.NOME as NOME, "

sSql &= "ca.ID\_TITULAR as ID\_TITULAR,"

sSql &= "dp.DATA\_NASCIMENTO as DATA\_NASCIMENTO,"

sSql &= "dp.SEXO as SEXO,"

sSql &= "dp.RG as RG,"

sSql &= "dp.CPF as CPF,"

sSql &= "dp.TELEFONE as TELEFONE,"

sSql &= "dp.CELULAR as CELULAR,"

sSql &= "dp.E\_MAIL as E\_MAIL,"

sSql &= "ende.LOGRADOURO as LOGRADOURO,"

sSql &= "ende.NUMERO as NUMERO,"

sSql &= "ende.COMPLEMENTO as COMPLEMENTO,"

sSql &= "ende.BAIRRO as BAIRRO,"

sSql &= "ende.CIDADE as CIDADE,"

sSql &= "ende.ESTADO as ESTADO,"

sSql &= "ende.CEP as CEP "

sSql &= " FROM CLIENTE\_ADICIONAL ca "

sSql &= " LEFT OUTER JOIN CLIENTE\_TITULAR ct ON ca.ID\_TITULAR = ct.ID\_TITULAR sSql &= " LEFT OUTER JOIN DADOS\_PESSOAIS dp ON ca.ID\_DADOS\_PESSOAIS = dp.ID\_DADOS\_PESSOAIS "

sSql &= " LEFT OUTER JOIN ENDERECOS ende ON ca.CEP = ende.CEP "

If C\_Adicional.p\_id\_titular > 0 Then

sSql &= "WHERE ca.ID\_TITULAR = " & C\_Adicional.p\_id\_titular

End If

If C\_Adicional.p\_id\_dados\_pessoais > 0 Then

sSql &= " WHERE ca.ID\_DADOS\_PESSOAIS = " & C\_Adicional.p\_id\_dados\_pessoais

End If

If C\_Adicional.p\_cep > 0 Then

sSql &= " WHERE ca.CEP = " & C\_Adicional.p\_cep

End If

If C\_Adicional.p\_id\_adicional > 0 Then

sSql &= " WHERE ID\_ADICIONAL = " & C\_Adicional.p\_id\_adicional

End If

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function InserirCA(ByVal C\_Adicional As Cliente\_Adicional) As DataTable

Return ExecutaConsultaCA(C\_Adicional, "I")

End Function

'alterando dados

Public Function AlterarCA(ByVal C\_Adicional As Cliente\_Adicional) As DataTable

Return ExecutaConsultaCA(C\_Adicional, "U")

End Function

'excluindo itens

Public Function ExcluirCA(ByVal C\_Adicional As Cliente\_Adicional) As DataTable

Return ExecutaConsultaCA(C\_Adicional, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultaCA(ByVal C\_Adicional As Cliente\_Adicional, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String

sSql = "sp\_CLIENTE\_ADICIONAL "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_ID\_ADICIONAL = " & C\_Adicional.p\_id\_adicional & ", "

sSql &= "@p\_ID\_TITULAR = " & C\_Adicional.p\_id\_titular & ", "

sSql &= "@p\_NOME = '" & C\_Adicional.p\_nome & "', "

sSql &= "@p\_CEP = '" & C\_Adicional.p\_cep & "', "

sSql &= "@p\_ID\_DADOS\_PESSOAIS = " & C\_Adicional.p\_id\_dados\_pessoais & ", "

sSql &= "@p\_GRAU\_PARENTESCO = '" & C\_Adicional.p\_grau\_parentesco & "' "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Cliente\_Titular

Inherits Dados\_Pessoais

Private id\_titular As Integer

Private nome As String

Private cep As String

Private id\_dados\_pessoais As Integer

Public Property p\_id\_titular() As Integer

Get

Return id\_titular

End Get

Set(ByVal value As Integer)

id\_titular = value

End Set

End Property

Public Property p\_nome() As String

Get

Return nome

End Get

Set(ByVal value As String)

nome = value

End Set

End Property

Public Function ListagemDataTableCliente\_Titular(ByVal C\_Titular As Cliente\_Titular) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM CLIENTE\_TITULAR "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function LstGridCliente\_Titular(ByVal C\_Titular As Cliente\_Titular) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere = "WHERE"

Dim sSql As String = ""

sSql = " SELECT "

sSql &= "ct.ID\_TITULAR, ct.NOME,"

sSql &= "dp.DATA\_NASCIMENTO as DATA\_NASCIMENTO,"

sSql &= "dp.TELEFONE as TELEFONE,"

sSql &= "dp.CELULAR as CELULAR,"

sSql &= "dp.E\_MAIL as E\_MAIL"

sSql &= " FROM CLIENTE\_TITULAR ct "

sSql &= " LEFT OUTER JOIN DADOS\_PESSOAIS dp ON ct.ID\_DADOS\_PESSOAIS = dp.ID\_DADOS\_PESSOAIS "

If C\_Titular.p\_nome <> "" Then

sSql &= msWhere & " NOME like '%" & C\_Titular.p\_nome & "%'"

msWhere = " AND "

End If

If C\_Titular.p\_id\_titular > 0 Then

sSql &= msWhere & " ID\_TITULAR = " & C\_Titular.p\_id\_titular

msWhere = " AND "

End If

If C\_Titular.p\_id\_dados\_pessoais > 0 Then

sSql &= msWhere & "ct.ID\_DADOS\_PESSOAIS = " & C\_Titular.p\_id\_dados\_pessoais

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'retornar os dados do registro para alteração

Public Function RetornarDadosCiente\_Titular(ByVal C\_Titular As Cliente\_Titular) As DataTable

ProgramaSQL = New ProgramaSQL

Dim v\_Dados\_Pessoais As New Dados\_Pessoais

Try

Dim sSql As String = ""

If C\_Titular.p\_id\_titular <> 0 Then

sSql = "SELECT "

sSql &= "ct.ID\_TITULAR, ct.NOME, "

sSql &= "ct.ID\_DADOS\_PESSOAIS as ID\_DADOS\_PESSOAIS, "

sSql &= "dp.DATA\_NASCIMENTO as DATA\_NASCIMENTO,"

sSql &= "dp.SEXO as SEXO,"

sSql &= "dp.RG as RG,"

sSql &= "dp.CPF as CPF,"

sSql &= "dp.TELEFONE as TELEFONE,"

sSql &= "dp.CELULAR as CELULAR,"

sSql &= "dp.E\_MAIL as E\_MAIL,"

sSql &= "ende.LOGRADOURO as LOGRADOURO,"

sSql &= "ende.NUMERO as NUMERO,"

sSql &= "ende.COMPLEMENTO as COMPLEMENTO,"

sSql &= "ende.BAIRRO as BAIRRO,"

sSql &= "ende.CIDADE as CIDADE,"

sSql &= "ende.ESTADO as ESTADO,"

sSql &= "ende.CEP as CEP "

sSql &= " FROM CLIENTE\_TITULAR ct "

sSql &= " LEFT OUTER JOIN DADOS\_PESSOAIS dp ON ct.ID\_DADOS\_PESSOAIS = dp.ID\_DADOS\_PESSOAIS "

sSql &= " LEFT OUTER JOIN ENDERECOS ende ON ct.CEP = ende.CEP"

If C\_Titular.p\_id\_dados\_pessoais > 0 Then

sSql &= " WHERE ct.ID\_DADOS\_PESSOAIS = " & C\_Titular.p\_id\_dados\_pessoais

End If

If C\_Titular.p\_cep > 0 Then

sSql &= " WHERE ct.CEP = " & C\_Titular.p\_cep

End If

If C\_Titular.p\_id\_titular > 0 Then

sSql &= " WHERE ID\_TITULAR = " & C\_Titular.p\_id\_titular

End If

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function InserirCT(ByVal C\_Titular As Cliente\_Titular) As DataTable

Return ExecutaConsultaCT(C\_Titular, "I")

End Function

'alterando dados

Public Function AlterarCT(ByVal C\_Titular As Cliente\_Titular) As DataTable

Return ExecutaConsultaCT(C\_Titular, "U")

End Function

'excluindo itens

Public Function ExcluirCT(ByVal C\_Titular As Cliente\_Titular) As DataTable

Return ExecutaConsultaCT(C\_Titular, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultaCT(ByVal C\_Titular As Cliente\_Titular, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String

sSql = "sp\_CLIENTE\_TITULAR "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_ID\_TITULAR = " & C\_Titular.p\_id\_titular & ", "

sSql &= "@p\_NOME = '" & C\_Titular.p\_nome & "', "

sSql &= "@p\_CEP = '" & C\_Titular.p\_cep & "', "

sSql &= "@p\_ID\_DADOS\_PESSOAIS = " & C\_Titular.p\_id\_dados\_pessoais & " "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Consulta\_locacao

Private id\_locacao As Integer

Private cod\_cliente As Integer

Private nome As String

Private tipo\_cliente As String

Private locacao\_paga\_ato As String

Private valor\_locacao As Double

Private cod\_filme As Integer

Private filme As String

Private data\_retirada As Date

Private data\_devolucao As Date

Private preco As Double

Public Property p\_id\_locacao() As Integer

Get

Return id\_locacao

End Get

Set(ByVal value As Integer)

id\_locacao = value

End Set

End Property

Public Property p\_cod\_cliente() As Integer

Get

Return cod\_cliente

End Get

Set(ByVal value As Integer)

cod\_cliente = value

End Set

End Property

Public Property p\_nome() As String

Get

Return nome

End Get

Set(ByVal value As String)

nome = value

End Set

End Property

Public Property p\_tipo\_cliente() As String

Get

Return tipo\_cliente

End Get

Set(ByVal value As String)

tipo\_cliente = value

End Set

End Property

Public Property p\_locacao\_paga\_ato() As String

Get

Return locacao\_paga\_ato

End Get

Set(ByVal value As String)

locacao\_paga\_ato = value

End Set

End Property

Public Property p\_valor\_locacao() As String

Get

Return valor\_locacao

End Get

Set(ByVal value As String)

valor\_locacao = value

End Set

End Property

Public Property p\_cod\_filme() As Integer

Get

Return cod\_filme

End Get

Set(ByVal value As Integer)

cod\_filme = value

End Set

End Property

Public Property p\_filme() As String

Get

Return filme

End Get

Set(ByVal value As String)

filme = value

End Set

End Property

Public Property p\_data\_retirada() As date

Get

Return data\_retirada

End Get

Set(ByVal value As Date)

data\_retirada = value

End Set

End Property

Public Property p\_data\_devolucao() As Date

Get

Return data\_devolucao

End Get

Set(ByVal value As Date)

data\_devolucao = value

End Set

End Property

Public Property p\_preco() As String

Get

Return preco

End Get

Set(ByVal value As String)

preco = value

End Set

End Property

Protected ProgramaSQL As ProgramaSQL

Public Function ListagemDataTableConsulta\_Locacao(ByVal C\_Locacao As Consulta\_locacao) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM LOCACAO\_FILMES "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function ListagemDataTable2Consulta\_Locacao(ByVal C\_Locacao As Consulta\_locacao) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM LOCACAO "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Consulta\_reserva

Private id\_reserva As Integer

Private cod\_cliente As Integer

Private nome As String

Private tipo\_cliente As String

Private cod\_filme As Integer

Private filme As String

Private data\_reserva As Date

Private preco As Double

Public Property p\_id\_reserva() As Integer

Get

Return id\_reserva

End Get

Set(ByVal value As Integer)

id\_reserva = value

End Set

End Property

Public Property p\_cod\_cliente() As Integer

Get

Return cod\_cliente

End Get

Set(ByVal value As Integer)

cod\_cliente = value

End Set

End Property

Public Property p\_nome() As String

Get

Return nome

End Get

Set(ByVal value As String)

nome = value

End Set

End Property

Public Property p\_tipo\_cliente() As String

Get

Return tipo\_cliente

End Get

Set(ByVal value As String)

tipo\_cliente = value

End Set

End Property

Public Property p\_cod\_filme() As Integer

Get

Return cod\_filme

End Get

Set(ByVal value As Integer)

cod\_filme = value

End Set

End Property

Public Property p\_filme() As String

Get

Return filme

End Get

Set(ByVal value As String)

filme = value

End Set

End Property

Public Property p\_data\_reserva() As String

Get

Return data\_reserva

End Get

Set(ByVal value As String)

data\_reserva = value

End Set

End Property

Public Property p\_preco() As String

Get

Return preco

End Get

Set(ByVal value As String)

preco = value

End Set

End Property

Protected ProgramaSQL As ProgramaSQL

Public Function ListagemDataTableConsulta\_Reserva(ByVal C\_Reserva As Consulta\_reserva) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM RESERVA\_FILMES "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function ListagemDataTable2Consulta\_Reserva(ByVal C\_Reserva As Consulta\_reserva) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM RESERVA "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Dados\_Pessoais

Inherits Enderecos

Private id\_dados\_pessoais As Integer

Private rg As String

Private cpf As String

Private sexo As String

Private data\_nascimento As Date

Private telefone As String

Private celular As String

Private e\_mail As String

Public Property p\_id\_dados\_pessoais() As Integer

Get

Return id\_dados\_pessoais

End Get

Set(ByVal value As Integer)

id\_dados\_pessoais = value

End Set

End Property

Public Property p\_rg() As String

Get

Return rg

End Get

Set(ByVal value As String)

rg = value

End Set

End Property

Public Property p\_cpf() As String

Get

Return cpf

End Get

Set(ByVal value As String)

cpf = value

End Set

End Property

Public Property p\_sexo() As String

Get

Return sexo

End Get

Set(ByVal value As String)

sexo = value

End Set

End Property

Public Property p\_data\_nascimento() As Date

Get

Return data\_nascimento

End Get

Set(ByVal value As Date)

data\_nascimento = value

End Set

End Property

Public Property p\_telefone() As String

Get

Return telefone

End Get

Set(ByVal value As String)

telefone = value

End Set

End Property

Public Property p\_celular() As String

Get

Return celular

End Get

Set(ByVal value As String)

celular = value

End Set

End Property

Public Property p\_e\_mail() As String

Get

Return e\_mail

End Get

Set(ByVal value As String)

e\_mail = value

End Set

End Property

'retornar os dados do registro para alteração

Public Function RetornarDadosDP(ByVal v\_Dados\_Pessoais As Dados\_Pessoais) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

If v\_Dados\_Pessoais.id\_dados\_pessoais <> "" Then

sSql = "SELECT \* FROM DADOS\_PESSOAIS "

sSql &= " WHERE ID\_DADOS\_PESSOAIS = " & v\_Dados\_Pessoais.p\_id\_dados\_pessoais & " " End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function InserirDP(ByVal v\_Dados\_Pessoais As Dados\_Pessoais) As DataTable

Return ExecutaConsultaDP(v\_Dados\_Pessoais, "I")

End Function

'alterando dados

Public Function AlterarDP(ByVal v\_dados\_pessoais As Dados\_Pessoais) As DataTable

Return ExecutaConsultaDP(v\_dados\_pessoais, "U")

End Function

'excluindo itens

Public Function ExcluirDP(ByVal v\_dados\_pessoais As Dados\_Pessoais) As DataTable

Return ExecutaConsultaDP(v\_dados\_pessoais, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultaDP(ByVal v\_dados\_pessoais As Dados\_Pessoais, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String

sSql = "sp\_DADOS\_PESSOAIS "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_ID\_DADOS\_PESSOAIS = " & v\_dados\_pessoais.p\_id\_dados\_pessoais & ", "

sSql &= "@p\_RG = '" & v\_dados\_pessoais.p\_rg & "',"

sSql &= "@p\_CPF = '" & v\_dados\_pessoais.p\_cpf & "',"

sSql &= "@p\_SEXO = '" & v\_dados\_pessoais.p\_sexo & "',"

sSql &= "@p\_DATA\_NASCIMENTO = '" & v\_dados\_pessoais.p\_data\_nascimento & "',"

sSql &= "@p\_TELEFONE = '" & v\_dados\_pessoais.p\_telefone & "',"

sSql &= "@p\_CELULAR = '" & v\_dados\_pessoais.p\_celular & "',"

sSql &= "@p\_E\_MAIL = '" & v\_dados\_pessoais.p\_e\_mail & "' "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Enderecos

Private cep As String

Private logradouro As String

Private numero As Integer

Private complemento As String

Private bairro As String

Private cidade As String

Private estado As String

Public Property p\_cep() As String

Get

Return cep

End Get

Set(ByVal value As String)

cep = value

End Set

End Property

Public Property p\_logradouro() As String

Get

Return logradouro

End Get

Set(ByVal value As String)

logradouro = value

End Set

End Property

Public Property p\_numero() As Integer

Get

Return numero

End Get

Set(ByVal value As Integer)

numero = value

End Set

End Property

Public Property p\_complemento() As String

Get

Return complemento

End Get

Set(ByVal value As String)

complemento = value

End Set

End Property

Public Property p\_bairro() As String

Get

Return bairro

End Get

Set(ByVal value As String)

bairro = value

End Set

End Property

Public Property p\_cidade() As String

Get

Return cidade

End Get

Set(ByVal value As String)

cidade = value

End Set

End Property

Public Property p\_estado() As String

Get

Return estado

End Get

Set(ByVal value As String)

estado = value

End Set

End Property

Protected ProgramaSQL As ProgramaSQL

'retornar os dados do registro para alteração

Public Function RetornarDadosEnderecos(ByVal v\_Enderecos As Enderecos) As DataTable

ProgramaSQL = New ProgramaSQL

Dim sSql As String = ""

Try

If v\_Enderecos.p\_cep <> "" Then

sSql &= "SELECT \* FROM ENDERECOS "

sSql &= " WHERE CEP = '" & v\_Enderecos.p\_cep & "' "

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function InserirEnderecos(ByVal v\_Enderecos As Enderecos) As DataTable

Return ExecutaConsultaEnderecos(v\_Enderecos, "I")

End Function

'alterando dados

Public Function AlterarEnderecos(ByVal v\_Enderecos As Enderecos) As DataTable

Return ExecutaConsultaEnderecos(v\_Enderecos, "U")

End Function

'excluindo itens

Public Function ExcluirEnderecos(ByVal v\_Enderecos As Enderecos) As DataTable

Return ExecutaConsultaEnderecos(v\_Enderecos, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultaEnderecos(ByVal v\_Enderecos As Enderecos, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String

sSql = "sp\_ENDERECOS "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_CEP = '" & v\_Enderecos.p\_cep & "', "

sSql &= "@p\_LOGRADOURO = '" & v\_Enderecos.p\_logradouro & "', "

sSql &= "@p\_NUMERO = " & v\_Enderecos.p\_numero & ", "

sSql &= "@p\_COMPLEMENTO = '" & v\_Enderecos.p\_complemento & " ',"

sSql &= "@p\_BAIRRO = '" & v\_Enderecos.p\_bairro & "', "

sSql &= "@p\_CIDADE = '" & v\_Enderecos.p\_cidade & "', "

sSql &= "@p\_ESTADO = '" & v\_Enderecos.p\_estado & "' "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Filmes

Private id\_filme As Integer

Private id\_midia As Integer

Private filme As String

Private ano\_lancamento As String

Private genero As String

Private nacionalidade As String

Private duracao As Integer

Private direcao As String

Private sinopse As String

Private preco As Double

Public Property p\_id\_filme() As Integer

Get

Return id\_filme

End Get

Set(ByVal value As Integer)

id\_filme = value

End Set

End Property

Public Property p\_id\_midia() As Integer

Get

Return id\_midia

End Get

Set(ByVal value As Integer)

id\_midia = value

End Set

End Property

Public Property p\_filme() As String

Get

Return filme

End Get

Set(ByVal value As String)

filme = value

End Set

End Property

Public Property p\_ano\_lancamento() As String

Get

Return ano\_lancamento

End Get

Set(ByVal value As String)

ano\_lancamento = value

End Set

End Property

Public Property p\_genero() As String

Get

Return genero

End Get

Set(ByVal value As String)

genero = value

End Set

End Property

Public Property p\_nacionalidade() As String

Get

Return nacionalidade

End Get

Set(ByVal value As String)

nacionalidade = value

End Set

End Property

Public Property p\_duracao() As Integer

Get

Return duracao

End Get

Set(ByVal value As Integer)

duracao = value

End Set

End Property

Public Property p\_direcao() As String

Get

Return direcao

End Get

Set(ByVal value As String)

direcao = value

End Set

End Property

Public Property p\_sinopse() As String

Get

Return sinopse

End Get

Set(ByVal value As String)

sinopse = value

End Set

End Property

Public Property p\_preco() As Double

Get

Return preco

End Get

Set(ByVal value As Double)

preco = value

End Set

End Property

Protected ProgramaSQL As ProgramaSQL

Public Function ListagemDataTableFilmes(ByVal F\_Filmes As Filmes) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM FILMES "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function LstGridFilmes(ByVal F\_Filmes As Filmes) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere As String = " WHERE "

Dim sSql As String = ""

sSql = "SELECT "

sSql &= "f.ID\_FILME, f.FILME, "

sSql &= "m.MIDIA as MIDIA,"

sSql &= "f.ANO\_LANCAMENTO, f.GENERO,"

sSql &= "f.DURACAO, f.PRECO"

sSql &= " FROM FILMES f "

sSql &= " LEFT OUTER JOIN MIDIAS m ON f.ID\_MIDIA = m.ID\_MIDIA "

If F\_Filmes.p\_filme <> "" Then

sSql &= msWhere & " FILME like '%" & F\_Filmes.p\_filme & "%'"

msWhere = " AND "

End If

If F\_Filmes.p\_id\_filme > 0 Then

sSql &= msWhere & " ID\_FILME = " & F\_Filmes.p\_id\_filme & " "

msWhere = " AND "

End If

If F\_Filmes.p\_id\_midia > 0 Then

sSql &= msWhere & "m.ID\_MIDIA = " & F\_Filmes.p\_id\_midia & " "

msWhere = " AND "

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'retornar os dados do registro para alteração

Public Function RetornarDadosFilmes(ByVal F\_Filmes As Filmes) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere As String = " WHERE "

Dim sSql As String = ""

If F\_Filmes.id\_filme <> 0 Then

sSql = "SELECT "

sSql &= "f.ID\_FILME,"

sSql &= "m.ID\_MIDIA as ID\_MIDIA, "

sSql &= "m.MIDIA as MIDIA,"

sSql &= "f.FILME, f.ANO\_LANCAMENTO, f.GENERO,"

sSql &= "f.NACIONALIDADE, f.DURACAO, f.DIRECAO,"

sSql &= "f.SINOPSE, f.PRECO"

sSql &= " FROM FILMES f "

sSql &= " LEFT OUTER JOIN MIDIAS m ON f.ID\_MIDIA = m.ID\_MIDIA "

If F\_Filmes.p\_id\_midia > 0 Then

sSql &= msWhere & " m.ID\_MIDIA = " & F\_Filmes.p\_id\_midia & " "

End If

If F\_Filmes.p\_filme <> "" Then

sSql &= msWhere & " FILME like '%" & F\_Filmes.p\_filme & "%'"

msWhere = " AND "

End If

If F\_Filmes.p\_id\_filme > 0 Then

sSql &= msWhere & " ID\_FILME = " & F\_Filmes.p\_id\_filme & " "

msWhere = " AND "

End If

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function InserirFilmes(ByVal F\_Filmes As Filmes) As DataTable

Return ExecutaConsultaFilmes(F\_Filmes, "I")

End Function

'alterando dados

Public Function AlterarFilmes(ByVal F\_Filmes As Filmes) As DataTable

Return ExecutaConsultaFilmes(F\_Filmes, "U")

End Function

'excluindo itens

Public Function ExcluirFilmes(ByVal F\_Filmes As Filmes) As DataTable

Return ExecutaConsultaFilmes(F\_Filmes, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultaFilmes(ByVal F\_Filmes As Filmes, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String

sSql = "sp\_FILMES "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_ID\_FILME = " & F\_Filmes.p\_id\_filme & ", "

sSql &= "@p\_ID\_MIDIA = " & F\_Filmes.p\_id\_midia & ", "

sSql &= "@p\_FILME = '" & F\_Filmes.p\_filme & "', "

sSql &= "@p\_ANO\_LANCAMENTO = '" & F\_Filmes.p\_ano\_lancamento & "', "

sSql &= "@p\_GENERO = '" & F\_Filmes.p\_genero & "', "

sSql &= "@p\_NACIONALIDADE = '" & F\_Filmes.p\_nacionalidade & "', "

sSql &= "@p\_DURACAO = " & F\_Filmes.p\_duracao & ", "

sSql &= "@p\_DIRECAO = '" & F\_Filmes.p\_direcao & "', "

sSql &= "@p\_SINOPSE = '" & F\_Filmes.p\_sinopse & "', "

sSql &= "@p\_PRECO = '" & F\_Filmes.p\_preco.ToString.Replace(",", ".") & "' "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class f\_Fornecedores

Inherits Enderecos

Private id\_fornecedor As Integer

Private razao\_social As String

Private nome\_fantasia As String

Private cnpj As String

Private contato As String

Private telefone As String

Private fax As String

Private e\_mail As String

Private cep As String

Public Property p\_id\_fornecedor() As Integer

Get

Return id\_fornecedor

End Get

Set(ByVal value As Integer)

id\_fornecedor = value

End Set

End Property

Public Property p\_razao\_social() As String

Get

Return razao\_social

End Get

Set(ByVal value As String)

razao\_social = value

End Set

End Property

Public Property p\_nome\_fantasia() As String

Get

Return nome\_fantasia

End Get

Set(ByVal value As String)

nome\_fantasia = value

End Set

End Property

Public Property p\_cnpj() As String

Get

Return cnpj

End Get

Set(ByVal value As String)

cnpj = value

End Set

End Property

Public Property p\_contato() As String

Get

Return contato

End Get

Set(ByVal value As String)

contato = value

End Set

End Property

Public Property p\_telefone() As String

Get

Return telefone

End Get

Set(ByVal value As String)

telefone = value

End Set

End Property

Public Property p\_fax() As String

Get

Return fax

End Get

Set(ByVal value As String)

fax = value

End Set

End Property

Public Property p\_e\_mail() As String

Get

Return e\_mail

End Get

Set(ByVal value As String)

e\_mail = value

End Set

End Property

'lista dos itens da grid

Public Function ListagemDataTableF\_Fornecedores(ByVal Fornecedores\_F As f\_Fornecedores) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM FORNECEDORES "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function LstGridf\_Fornecedores(ByVal Fornecedores\_F As f\_Fornecedores) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere = "WHERE"

Dim sSql As String = ""

sSql = " SELECT "

sSql &= "forn.ID\_FORNECEDOR, forn.RAZAO\_SOCIAL, forn.NOME\_FANTASIA,"

sSql &= "forn.CNPJ, forn.CONTATO, forn.TELEFONE, forn.FAX"

sSql &= " FROM FORNECEDORES forn "

If Fornecedores\_F.p\_razao\_social <> "" Then

sSql &= msWhere & " RAZAO\_SOCIAL like '%" & Fornecedores\_F.razao\_social & "%'"

msWhere = " AND "

End If

If Fornecedores\_F.p\_id\_fornecedor > 0 Then

sSql &= msWhere & " ID\_FORNECEDOR = " & Fornecedores\_F.p\_id\_fornecedor

msWhere = " AND "

End If

If Fornecedores\_F.p\_nome\_fantasia <> "" Then

sSql &= msWhere & " NOME\_FANTASIA like '%" & Fornecedores\_F.p\_nome\_fantasia & "%'"

msWhere = " AND "

End If

If Fornecedores\_F.p\_cnpj <> "" Then

sSql &= msWhere & " CNPJ like '%" & Fornecedores\_F.p\_cnpj & "%'"

msWhere = " AND "

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'retornar os dados do registro para alteração

Public Function RetornarDadosFornecedores(ByVal fornecedores\_f As f\_Fornecedores) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

If fornecedores\_f.id\_fornecedor <> 0 Then

sSql = "SELECT "

sSql &= "forn.ID\_FORNECEDOR, forn.RAZAO\_SOCIAL, "

sSql &= "forn.NOME\_FANTASIA, forn.CNPJ, forn.CONTATO,"

sSql &= "forn.TELEFONE, forn.FAX, forn.E\_MAIL,"

sSql &= "ende.LOGRADOURO as LOGRADOURO,"

sSql &= "ende.NUMERO as NUMERO,"

sSql &= "ende.COMPLEMENTO as COMPLEMENTO,"

sSql &= "ende.BAIRRO as BAIRRO,"

sSql &= "ende.CIDADE as CIDADE,"

sSql &= "ende.ESTADO as ESTADO,"

sSql &= "ende.CEP as CEP "

sSql &= " FROM FORNECEDORES forn "

sSql &= " LEFT OUTER JOIN ENDERECOS ende ON forn.CEP = ende.CEP "

If fornecedores\_f.p\_cep > 0 Then

sSql &= " WHERE forn.CEP = " & fornecedores\_f.p\_cep

End If

If fornecedores\_f.p\_id\_fornecedor > 0 Then

sSql &= " WHERE forn.ID\_FORNECEDOR = " & fornecedores\_f.p\_id\_fornecedor

End If

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function InserirFornecedores(ByVal Fornecedores\_F As f\_Fornecedores) As DataTable

Return ExecutaConsultaFornecedores(Fornecedores\_F, "I")

End Function

'alterando dados

Public Function AlterarFornecedores(ByVal Fornecedores\_F As f\_Fornecedores) As DataTable

Return ExecutaConsultaFornecedores(Fornecedores\_F, "U")

End Function

'excluindo itens

Public Function ExcluirFornecedores(ByVal Fornecedores\_F As f\_Fornecedores) As DataTable

Return ExecutaConsultaFornecedores(Fornecedores\_F, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultaFornecedores(ByVal Fornecedores\_F As f\_Fornecedores, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String

sSql = "sp\_FORNECEDORES "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_ID\_FORNECEDOR = " & Fornecedores\_F.p\_id\_fornecedor & ", "

sSql &= "@p\_RAZAO\_SOCIAL= '" & Fornecedores\_F.p\_razao\_social & "', "

sSql &= "@p\_NOME\_FANTASIA = '" & Fornecedores\_F.p\_nome\_fantasia & "', "

sSql &= "@p\_CNPJ = '" & Fornecedores\_F.p\_cnpj & "', "

sSql &= "@p\_CONTATO = '" & Fornecedores\_F.p\_contato & "', "

sSql &= "@p\_TELEFONE= '" & Fornecedores\_F.p\_telefone & "', "

sSql &= "@p\_FAX = '" & Fornecedores\_F.p\_fax & "', "

sSql &= "@p\_E\_MAIL = '" & Fornecedores\_F.p\_e\_mail & "', "

sSql &= "@p\_CEP = '" & Fornecedores\_F.p\_cep & "' "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Funcionarios

Inherits Dados\_Pessoais

Private id\_funcionario As Integer

Private nome As String

Private horario\_trabalho As String

Private data\_admissao As Date

Private cargo As String

Private salario As Double

Private estado\_civil As String

Private id\_dados\_pessoais As Integer

Private cep As String

Public Property p\_id\_funcionario() As Integer

Get

Return id\_funcionario

End Get

Set(ByVal value As Integer)

id\_funcionario = value

End Set

End Property

Public Property p\_nome() As String

Get

Return nome

End Get

Set(ByVal value As String)

nome = value

End Set

End Property

Public Property p\_horario\_trabalho() As String

Get

Return horario\_trabalho

End Get

Set(ByVal value As String)

horario\_trabalho = value

End Set

End Property

Public Property p\_data\_admissao() As Date

Get

Return data\_admissao

End Get

Set(ByVal value As Date)

data\_admissao = value

End Set

End Property

Public Property p\_cargo() As String

Get

Return cargo

End Get

Set(ByVal value As String)

cargo = value

End Set

End Property

Public Property p\_salario() As Double

Get

Return salario

End Get

Set(ByVal value As Double)

salario = value

End Set

End Property

Public Property p\_estado\_civil() As String

Get

Return estado\_civil

End Get

Set(ByVal value As String)

estado\_civil = value

End Set

End Property

'lista dos itens da grid

Public Function ListagemDataTablef\_Funcionarios(ByVal funcionarios As f\_funcionarios) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM FUNCIONARIOS "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function LstGridFuncionarios(ByVal f\_funcionarios As Funcionarios) As DataTable

ProgramaSQL = New ProgramaSQL

Try

dim msWhere = "WHERE"

Dim sSql As String = ""

sSql = " SELECT "

sSql &= "fun.ID\_FUNCIONARIO, fun.NOME,"

sSql &= "fun.HORARIO\_TRABALHO,"

sSql &= "fun.CARGO,"

sSql &= "dp.TELEFONE as TELEFONE,"

sSql &= "dp.CELULAR as CELULAR "

sSql &= " FROM FUNCIONARIOS fun "

sSql &= " LEFT OUTER JOIN DADOS\_PESSOAIS dp ON fun.ID\_DADOS\_PESSOAIS = dp.ID\_DADOS\_PESSOAIS "

If f\_funcionarios.id\_dados\_pessoais > 0 Then

sSql &= msWhere & " dp.ID\_DADOS\_PESSOAIS = " & f\_funcionarios.id\_dados\_pessoais

msWhere = " AND "

End If

If f\_funcionarios.id\_funcionario > 0 Then

sSql &= msWhere & " ID\_FUNCIONARIO = " & f\_funcionarios.id\_funcionario

msWhere = " AND "

End If

If f\_funcionarios.p\_nome <> "" Then

sSql &= msWhere & " NOME like '%" & f\_funcionarios.p\_nome & "%'"

msWhere = " AND "

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'retornar os dados do registro para alteração

Public Function RetornarDadosFuncionarios(ByVal f\_funcionarios As Funcionarios) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere = "WHERE"

Dim sSql As String = ""

If f\_funcionarios.id\_funcionario <> 0 Then

sSql = "SELECT "

sSql &= "fun.ID\_FUNCIONARIO, fun.NOME, "

sSql &= "fun.ID\_DADOS\_PESSOAIS as ID\_DADOS\_PESSOAIS, "

sSql &= "fun.HORARIO\_TRABALHO, fun.DATA\_ADMISSAO,"

sSql &= "fun.CARGO, fun.SALARIO, fun.ESTADO\_CIVIL,"

sSql &= "dp.DATA\_NASCIMENTO as DATA\_NASCIMENTO,"

sSql &= "dp.SEXO as SEXO,"

sSql &= "dp.RG as RG,"

sSql &= "dp.CPF as CPF,"

sSql &= "dp.TELEFONE as TELEFONE,"

sSql &= "dp.CELULAR as CELULAR,"

sSql &= "dp.E\_MAIL as E\_MAIL,"

sSql &= "ende.LOGRADOURO as LOGRADOURO,"

sSql &= "ende.NUMERO as NUMERO, "

sSql &= "ende.COMPLEMENTO as COMPLEMENTO,"

sSql &= "ende.BAIRRO as BAIRRO,"

sSql &= "ende.CIDADE as CIDADE,"

sSql &= "ende.ESTADO as ESTADO,"

sSql &= "ende.CEP as CEP "

sSql &= " FROM FUNCIONARIOS fun "

sSql &= " LEFT OUTER JOIN DADOS\_PESSOAIS dp ON fun.ID\_DADOS\_PESSOAIS = dp.ID\_DADOS\_PESSOAIS "

sSql &= " LEFT OUTER JOIN ENDERECOS ende ON fun.CEP = ende.CEP "

If f\_funcionarios.id\_dados\_pessoais > 0 Then

sSql &= " WHERE fun.ID\_DADOS\_PESSOAIS = " & f\_funcionarios.p\_id\_dados\_pessoais

End If

If f\_funcionarios.p\_cep > 0 Then

sSql &= " WHERE fun.CEP = " & f\_funcionarios.p\_cep

End If

If f\_funcionarios.p\_id\_funcionario > 0 Then

sSql &= " WHERE ID\_FUNCIONARIO = " & f\_funcionarios.p\_id\_funcionario

End If

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function InserirFuncionarios(ByVal F\_Funcionarios As Funcionarios) As DataTable

Return ExecutaConsultaFuncionarios(F\_Funcionarios, "I")

End Function

'alterando dados

Public Function AlterarFuncionarios(ByVal F\_Funcionarios As Funcionarios) As DataTable

Return ExecutaConsultaFuncionarios(F\_Funcionarios, "U")

End Function

'excluindo itens

Public Function ExcluirFuncionarios(ByVal F\_Funcionarios As Funcionarios) As DataTable

Return ExecutaConsultaFuncionarios(F\_Funcionarios, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultaFuncionarios(ByVal F\_Funcionarios As Funcionarios, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String

sSql = "sp\_FUNCIONARIOS "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_ID\_FUNCIONARIO = " & F\_Funcionarios.p\_id\_funcionario & ", "

sSql &= "@p\_NOME = '" & F\_Funcionarios.p\_nome & "', "

sSql &= "@p\_HORARIO\_TRABALHO = '" & F\_Funcionarios.p\_horario\_trabalho & "', "

sSql &= "@p\_DATA\_ADMISSAO = '" & F\_Funcionarios.p\_data\_admissao & "', "

sSql &= "@p\_CARGO = '" & F\_Funcionarios.p\_cargo & "', "

sSql &= "@p\_SALARIO = '" & F\_Funcionarios.p\_salario & "', "

sSql &= "@P\_ESTADO\_CIVIL = '" & F\_Funcionarios.p\_estado\_civil & "', "

sSql &= "@p\_ID\_DADOS\_PESSOAIS = " & F\_Funcionarios.p\_id\_dados\_pessoais & ", "

sSql &= "@p\_CEP = '" & F\_Funcionarios.p\_cep & "' "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Locacao\_cliente

Private id\_locacao As Integer

Private cod\_cliente As Integer

Private nome As String

Private tipo\_cliente As String

Private locacao\_paga\_ato As String

Private valor\_locacao As Double

Public Property p\_id\_locacao() As Integer

Get

Return id\_locacao

End Get

Set(ByVal value As Integer)

id\_locacao = value

End Set

End Property

Public Property p\_cod\_cliente() As Integer

Get

Return cod\_cliente

End Get

Set(ByVal value As Integer)

cod\_cliente = value

End Set

End Property

Public Property p\_nome() As String

Get

Return nome

End Get

Set(ByVal value As String)

nome = value

End Set

End Property

Public Property p\_tipo\_cliente() As String

Get

Return tipo\_cliente

End Get

Set(ByVal value As String)

tipo\_cliente = value

End Set

End Property

Public Property p\_locacao\_paga\_ato() As String

Get

Return locacao\_paga\_ato

End Get

Set(ByVal value As String)

locacao\_paga\_ato = value

End Set

End Property

Public Property p\_valor\_locacao() As String

Get

Return valor\_locacao

End Get

Set(ByVal value As String)

valor\_locacao = value

End Set

End Property

Protected ProgramaSQL As ProgramaSQL

'lista dos itens da grid

Public Function ListagemDataTableLocacoes(ByVal l\_cliente As Locacao\_cliente) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM LOCACAO\_CLIENTE "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function LstGridlocacao\_cliente(ByVal l\_cliente As Locacao\_cliente) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere1 = "WHERE"

Dim msWhere2 = "WHERE"

Dim sSql As String = ""

sSql = " SELECT "

sSql &= "'TIT' as TIPO, tit.ID\_TITULAR as COD, tit.NOME "

sSql &= " FROM CLIENTE\_TITULAR tit "

If l\_cliente.p\_nome <> "" Then

sSql &= msWhere1 & " tit.NOME like '%" & l\_cliente.p\_nome & "%'"

msWhere1 = " AND "

End If

sSql &= " UNION ALL "

sSql &= " SELECT 'ADC' as TIPO, adc.ID\_ADICIONAL as COD, adc.NOME "

sSql &= " FROM CLIENTE\_ADICIONAL adc "

If l\_cliente.p\_nome <> "" Then

sSql &= msWhere2 & " adc.NOME like '%" & l\_cliente.p\_nome & "%'"

msWhere2 = " AND "

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'retornar os dados do registro para alteração

Public Function RetornarDadosLocacao\_cliente(ByVal L\_cliente As Locacao\_cliente) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

If L\_cliente.p\_id\_locacao <> 0 Then

sSql = " SELECT "

sSql &= "'TIT' as TIPO, tit.ID\_TITULAR as COD, tit.NOME "

sSql &= " FROM CLIENTE\_TITULAR tit "

sSql &= " UNION(ALL) SELECT 'ADC' as TIPO, adc.ID\_ADICIONAL as COD, adc.NOME "

sSql &= " FROM CLIENTE\_ADICIONAL adc "

sSql &= " lf.PRECO, lc.LOCACAO\_PAGA\_ATO, lc.VALOR\_LOCACAO "

sSql &= " FROM LOCACAO\_CLIENTE lc "

sSql &= " WHERE ID\_LOCACAO = " & L\_cliente.p\_id\_locacao

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function Inserirlc(ByVal L\_cliente As Locacao\_cliente) As DataTable

Return ExecutaConsultalc(L\_cliente, "I")

End Function

'alterando dados

Public Function Alterarlc(ByVal L\_cliente As Locacao\_cliente) As DataTable

Return ExecutaConsultalc(L\_cliente, "U")

End Function

'excluindo itens

Public Function Excluirlc(ByVal L\_cliente As Locacao\_cliente) As DataTable

Return ExecutaConsultalc(L\_cliente, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultalc(ByVal L\_cliente As Locacao\_cliente, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String

sSql = "sp\_LOCACAO "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_ID\_LOCACAO = " & L\_cliente.p\_id\_locacao & ", "

sSql &= "@p\_COD\_CLIENTE = " & L\_cliente.p\_cod\_cliente & ", "

sSql &= "@p\_TIPO\_CLIENTE = '" & L\_cliente.p\_tipo\_cliente & "', "

sSql &= "@p\_VALOR\_LOCACAO = " & L\_cliente.p\_valor\_locacao.ToString.Replace(",", ".") & ", "

sSql &= "@p\_LOCACAO\_PAGA\_ATO = '" & L\_cliente.p\_locacao\_paga\_ato & "' "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Locacao\_filme

Private id\_Locacao\_filmes As Integer

Private id\_locacao As Integer

Private cod\_filme As Integer

Private filme As String

Private data\_retirada As Date

Private data\_devolucao As Date

Private preco As Double

Public Property p\_id\_Locacao\_filmes() As Integer

Get

Return id\_Locacao\_filmes

End Get

Set(ByVal value As Integer)

id\_Locacao\_filmes = value

End Set

End Property

Public Property p\_id\_locacao() As Integer

Get

Return id\_locacao

End Get

Set(ByVal value As Integer)

id\_locacao = value

End Set

End Property

Public Property p\_cod\_filme() As Integer

Get

Return cod\_filme

End Get

Set(ByVal value As Integer)

cod\_filme = value

End Set

End Property

Public Property p\_filme() As String

Get

Return filme

End Get

Set(ByVal value As String)

filme = value

End Set

End Property

Public Property p\_data\_retirada() As String

Get

Return data\_retirada

End Get

Set(ByVal value As String)

data\_retirada = value

End Set

End Property

Public Property p\_data\_devolucao() As String

Get

Return data\_devolucao

End Get

Set(ByVal value As String)

data\_devolucao = value

End Set

End Property

Public Property p\_preco() As String

Get

Return preco

End Get

Set(ByVal value As String)

preco = value

End Set

End Property

Protected ProgramaSQL As ProgramaSQL

'lista dos itens da grid

Public Function ListagemDataTableLocacao(ByVal l\_filme As Locacao\_filme) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM LOCACAO\_FILME "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function LstGridlocacao\_filme(ByVal l\_filme As Locacao\_filme) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere = "WHERE"

Dim sSql As String = ""

sSql = " SELECT "

sSql &= "fil.ID\_FILME as COD, fil.FILME, fil.PRECO "

sSql &= " FROM FILMES fil "

If l\_filme.p\_filme <> "" Then

sSql &= msWhere & " fil.FILME like '%" & l\_filme.p\_filme & "%'"

msWhere = " AND "

End If

If l\_filme.p\_cod\_filme > 0 Then

sSql &= msWhere & " fil.ID\_FILME = " & l\_filme.p\_cod\_filme

msWhere = " AND "

End If

If l\_filme.p\_preco <> "" Then

sSql &= msWhere & " fil.PRECO like '%" & l\_filme.p\_preco & "%'"

msWhere = " AND "

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'retornar os dados do registro para alteração

Public Function RetornarDadosLocacao\_filme(ByVal L\_filme As Locacao\_filme) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere = "WHERE"

Dim sSql As String = ""

If L\_filme.p\_id\_locacao <> 0 Then

sSql = " SELECT "

sSql &= "fil.ID\_FILME as COD, fil.FILME, fil.PRECO "

sSql &= " FROM FILMES fil "

sSql &= "lf.DATA\_RETIRADA, lf.DATA\_DEVOLUCAO, "

sSql &= " FROM LOCACAO\_FILME lf "

sSql &= " WHERE ID\_LOCACAO\_FILMES = " & L\_filme.p\_id\_locacao\_filmes

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function Inserirlf(ByVal L\_Filme As Locacao\_filme) As DataTable

Return ExecutaConsultalf(L\_Filme, "I")

End Function

'alterando dados

Public Function Alterarlf(ByVal L\_Filme As Locacao\_filme) As DataTable

Return ExecutaConsultalf(L\_Filme, "U")

End Function

'excluindo itens

Public Function Excluirlf(ByVal L\_Filme As Locacao\_filme) As DataTable

Return ExecutaConsultalf(L\_Filme, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultalf(ByVal L\_Filme As Locacao\_filme, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Dim L\_Cliente As New Locacao\_cliente

Try

Dim sSql As String

sSql = "sp\_LOCACAO\_FILMES "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_ID\_LOCACAO = " & L\_Filme.p\_id\_locacao & ", "

sSql &= "@P\_ID\_LOCACAO\_FILMES = " & L\_Filme.p\_id\_locacao\_filmes & ", "

sSql &= "@p\_COD\_FILME = " & L\_Filme.p\_cod\_filme & ", "

sSql &= "@p\_DATA\_RETIRADA = '" & L\_Filme.p\_data\_retirada & "', "

sSql &= "@p\_DATA\_DEVOLUCAO = '" & L\_Filme.p\_data\_devolucao & "', "

sSql &= "@p\_PRECO = '" & L\_Filme.p\_preco.ToString.Replace(",", ".") & "' "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Midias

Private id\_midia As Integer

Private midia As String

Public Property p\_id\_midia() As Integer

Get

Return id\_midia

End Get

Set(ByVal value As Integer)

id\_midia = value

End Set

End Property

Public Property p\_midia() As String

Get

Return midia

End Get

Set(ByVal value As String)

midia = value

End Set

End Property

Protected ProgramaSQL As ProgramaSQL

'lista dos itens da grid

Public Function ListagemDataTableMidias(ByVal M\_Midias As Midias) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM MIDIAS "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function LstGridMidias(ByVal M\_Midias As Midias) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere As String = " WHERE "

Dim sSql As String = ""

sSql = "SELECT "

sSql &= "ID\_MIDIA, MIDIA "

sSql &= "FROM MIDIAS "

If M\_Midias.p\_midia <> "" Then

sSql &= msWhere & " MIDIA like '%" & M\_Midias.p\_midia & "%'"

msWhere = " AND "

End If

If M\_Midias.p\_id\_midia > 0 Then

sSql &= msWhere & " ID\_MIDIA = " & M\_Midias.p\_id\_midia

msWhere = " AND "

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'retornar os dados do registro para alteração

Public Function RetornarDadosMidias(ByVal M\_Midias As Midias) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

If M\_Midias.id\_midia <> 0 Then

sSql = "SELECT "

sSql &= "ID\_MIDIA, MIDIA "

sSql &= " FROM MIDIAS "

If M\_Midias.p\_id\_midia > 0 Then

sSql &= "WHERE ID\_MIDIA = " & M\_Midias.p\_id\_midia

End If

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function InserirMidias(ByVal M\_Midias As Midias) As DataTable

Return ExecutaConsultaMidias(M\_Midias, "I")

End Function

'alterando dados

Public Function AlterarMidias(ByVal M\_Midias As Midias) As DataTable

Return ExecutaConsultaMidias(M\_Midias, "U")

End Function

'excluindo itens

Public Function ExcluirMidias(ByVal M\_Midias As Midias) As DataTable

Return ExecutaConsultaMidias(M\_Midias, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultaMidias(ByVal M\_Midias As Midias, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String

sSql = "sp\_MIDIAS "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_ID\_MIDIA = " & M\_Midias.p\_id\_midia & ", "

sSql &= "@p\_MIDIA = '" & M\_Midias.p\_midia & "' "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Reserva\_cliente

Private id\_reserva As Integer

Private cod\_cliente As Integer

Private nome As String

Private tipo\_cliente As String

Private telefone As String

Public Property p\_id\_reserva() As Integer

Get

Return id\_reserva

End Get

Set(ByVal value As Integer)

id\_reserva = value

End Set

End Property

Public Property p\_cod\_cliente() As Integer

Get

Return cod\_cliente

End Get

Set(ByVal value As Integer)

cod\_cliente = value

End Set

End Property

Public Property p\_nome() As String

Get

Return nome

End Get

Set(ByVal value As String)

nome = value

End Set

End Property

Public Property p\_tipo\_cliente() As String

Get

Return tipo\_cliente

End Get

Set(ByVal value As String)

tipo\_cliente = value

End Set

End Property

Public Property p\_telefone() As String

Get

Return telefone

End Get

Set(ByVal value As String)

telefone = value

End Set

End Property

Protected ProgramaSQL As ProgramaSQL

'lista dos itens da grid

Public Function ListagemDataTableReserva(ByVal r\_cliente As Reserva\_cliente) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM RESERVA\_CLIENTE "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function LstGridreserva\_cliente(ByVal r\_cliente As Reserva\_cliente) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere1 = "WHERE"

Dim msWhere2 = "WHERE"

Dim sSql As String = ""

sSql = " SELECT "

sSql &= "'TIT' as TIPO, tit.ID\_TITULAR as COD, tit.NOME "

sSql &= " FROM CLIENTE\_TITULAR tit "

If r\_cliente.p\_nome <> "" Then

sSql &= msWhere1 & " tit.NOME like '%" & r\_cliente.p\_nome & "%'"

msWhere1 = " AND "

End If

sSql &= " UNION ALL "

sSql &= " SELECT 'ADC' as TIPO, adc.ID\_ADICIONAL as COD, adc.NOME "

sSql &= " FROM CLIENTE\_ADICIONAL adc "

If r\_cliente.p\_nome <> "" Then

sSql &= msWhere2 & " adc.NOME like '%" & r\_cliente.p\_nome & "%'"

msWhere2 = " AND "

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'retornar os dados do registro para alteração

Public Function RetornarDadosreserva\_cliente(ByVal r\_cliente As Reserva\_cliente) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

If r\_cliente.p\_id\_reserva <> 0 Then

sSql = " SELECT "

sSql &= "'TIT' as TIPO, tit.ID\_TITULAR as COD, tit.NOME "

sSql &= " FROM CLIENTE\_TITULAR tit "

sSql &= " UNION(ALL) SELECT 'ADC' as TIPO, adc.ID\_ADCIONAL as COD, adc.NOME "

sSql &= " FROM CLIENTE\_ADCIONAL adc "

sSql &= " WHERE ID\_RESERVA = " & r\_cliente.p\_id\_reserva

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function Inserirrc(ByVal r\_cliente As Reserva\_cliente) As DataTable

Return ExecutaConsultarc(r\_cliente, "I")

End Function

'alterando dados

Public Function Alterarrc(ByVal r\_cliente As Reserva\_cliente) As DataTable

Return ExecutaConsultarc(r\_cliente, "U")

End Function

'excluindo itens

Public Function Excluirrc(ByVal r\_cliente As Reserva\_cliente) As DataTable

Return ExecutaConsultarc(r\_cliente, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultarc(ByVal r\_cliente As Reserva\_cliente, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String

sSql = "sp\_RESERVA "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_ID\_RESERVA = " & r\_cliente.p\_id\_reserva & ", "

sSql &= "@p\_COD\_CLIENTE = " & r\_cliente.p\_cod\_cliente & ", "

sSql &= "@p\_NOME = '" & r\_cliente.p\_nome & "', "

sSql &= "@p\_TIPO\_CLIENTE = '" & r\_cliente.p\_tipo\_cliente & "' "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Reserva\_Filme

Private id\_reseva\_filmes As Integer

Private id\_reserva As Integer

Private cod\_filme As Integer

Private filme As String

Private data\_reserva As Date

Private preco As Double

Public Property p\_id\_reserva\_filmes() As Integer

Get

Return id\_reseva\_filmes

End Get

Set(ByVal value As Integer)

id\_reseva\_filmes = value

End Set

End Property

Public Property p\_id\_reserva() As Integer

Get

Return id\_reserva

End Get

Set(ByVal value As Integer)

id\_reserva = value

End Set

End Property

Public Property p\_cod\_filme() As Integer

Get

Return cod\_filme

End Get

Set(ByVal value As Integer)

cod\_filme = value

End Set

End Property

Public Property p\_filme() As String

Get

Return filme

End Get

Set(ByVal value As String)

filme = value

End Set

End Property

Public Property p\_data\_reserva() As String

Get

Return data\_reserva

End Get

Set(ByVal value As String)

data\_reserva = value

End Set

End Property

Public Property p\_preco() As String

Get

Return preco

End Get

Set(ByVal value As String)

preco = value

End Set

End Property

Protected ProgramaSQL As ProgramaSQL

'lista dos itens da grid

Public Function ListagemDataTableReserva(ByVal r\_filme As Reserva\_Filme) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM RESERVA\_FILME "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

Public Function LstGridreserva\_filme(ByVal r\_filme As Reserva\_Filme) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere = "WHERE"

Dim sSql As String = ""

sSql = " SELECT "

sSql &= "fil.ID\_FILME as COD, fil.FILME "

sSql &= " FROM FILMES fil "

If r\_filme.p\_filme <> "" Then

sSql &= msWhere & " fil.FILME like '%" & r\_filme.p\_filme & "%'"

msWhere = " AND "

End If

If r\_filme.p\_cod\_filme > 0 Then

sSql &= msWhere & " fil.ID\_FILME = " & r\_filme.p\_cod\_filme

msWhere = " AND "

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'retornar os dados do registro para alteração

Public Function RetornarDadosreserva\_filme(ByVal r\_filme As Reserva\_Filme) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim msWhere = "WHERE"

Dim sSql As String = ""

If r\_filme.p\_id\_reserva <> 0 Then

sSql = " SELECT "

sSql &= "fil.ID\_FILME as COD, fil.FILME, fil.PRECO, "

sSql &= " FROM FILMES fil "

sSql &= "lf.DATA\_RESERVA, rf.VALOR\_RESERVA "

sSql &= " FROM RESERVA\_FILME rf "

sSql &= " WHERE ID\_RESERVA = " & r\_filme.p\_id\_reserva\_filmes

End If

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

'incluir dados

Public Function Inserirrf(ByVal r\_Filme As Reserva\_Filme) As DataTable

Return ExecutaConsultarf(r\_Filme, "I")

End Function

'alterando dados

Public Function Alterarrf(ByVal r\_Filme As Reserva\_Filme) As DataTable

Return ExecutaConsultarf(r\_Filme, "U")

End Function

'excluindo itens

Public Function Excluirrf(ByVal r\_Filme As Reserva\_Filme) As DataTable

Return ExecutaConsultarf(r\_Filme, "D")

End Function

'executa consulta padrão do banco

Protected Function ExecutaConsultarf(ByVal r\_Filme As Reserva\_Filme, ByVal v\_tipo As String) As DataTable

ProgramaSQL = New ProgramaSQL

Try

Dim sSql As String

sSql = "sp\_RESERVA "

sSql &= "@p\_Opcao = '" & v\_tipo & "', "

sSql &= "@p\_COD\_FILME = " & r\_Filme.p\_cod\_filme & ", "

sSql &= "@p\_FILME = '" & r\_Filme.p\_filme & "', "

sSql &= "@p\_DATA\_RESERVA = '" & r\_Filme.p\_data\_reserva & "', "

sSql &= "@p\_PRECO = '" & r\_Filme.p\_preco.ToString.Replace(",", ".") & "' "

Return ProgramaSQL.ExecuteDataTable(ProgramaSQL.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSQL.close()

End Try

End Function

End Class

Public Class Usuario

Private Usuario As Integer

Private Nome As String

Private Senha As Integer

Public Property p\_id\_Usuario()

Get

Return Usuario

End Get

Set(ByVal value)

Usuario = value

End Set

End Property

Public Property p\_Nome()

Get

Return Nome

End Get

Set(ByVal value)

Nome = value

End Set

End Property

Public Property p\_Senha()

Get

Return Senha

End Get

Set(ByVal value)

Senha = value

End Set

End Property

Protected ProgramaSql As ProgramaSQL

'padrao sistema---------------------------------------------------------------------------

'lista dos itens da grid

Public Function ListagemDataTableUsuario(ByVal v\_Usuario As Usuario) As DataTable

ProgramaSql = New ProgramaSQL

Try

Dim sSql As String = ""

sSql = "SELECT \* FROM USUARIO WHERE NOME\_USUARIO = '" & v\_Usuario.p\_Nome & "' "

Return ProgramaSql.ExecuteDataTable(ProgramaSql.tipoSql.sql\_texto, sSql)

Catch ex As Exception

Throw ex

Finally

ProgramaSql.close()

End Try

End Function

End Class

Public Class ProgramaSQL

'tipo de tranzação

Public Enum tipoSql

sql\_procedure = 0 ' Não tem retorno de informações

sql\_texto = 1 ' Tem retorno de informações

End Enum

Protected strConn As String = "Data source=FILIPE-F0FFB6AB\SQLEXPRESS;Connect Timeout=1800;Initial Catalog=video\_locadora;uid=;pwd=;Integrated Security=SSPI"

Protected cnn As SqlConnection

Protected cmd As SqlCommand

Sub New()

cnn = New SqlConnection(strConn)

cmd = New SqlCommand

Try

cnn.ConnectionString = strConn

cnn.Open()

cmd.Connection = cnn

Catch ex As Exception

If Not strConn Is Nothing Then

strConn = Nothing

End If

If Not cnn Is Nothing Then

cnn = Nothing

End If

If Not cmd Is Nothing Then

'cmd.Connection.Dispose()

cmd = Nothing

End If

Throw ex

End Try

End Sub ' Minha Conexão

Public Function ExecuteDataSet(ByVal v\_tipoSql As tipoSql, ByVal sql As String) As DataSet

cmd = New SqlCommand(sql, cnn)

Select Case v\_tipoSql

Case tipoSql.sql\_procedure

cmd.CommandType = CommandType.StoredProcedure

Case tipoSql.sql\_texto

cmd.CommandType = CommandType.Text

End Select

Dim da As SqlDataAdapter = New SqlDataAdapter

da.SelectCommand = cmd

Dim ds As DataSet = New DataSet

da.Fill(ds)

Return ds

End Function

Public Function ExecuteDataSet(ByVal v\_tipoSql As tipoSql, ByVal sql As String, ByVal params() As SqlParameter) As DataSet

cmd = New SqlCommand(sql, cnn)

Select Case v\_tipoSql

Case tipoSql.sql\_procedure

cmd.CommandType = CommandType.StoredProcedure

Case tipoSql.sql\_texto

cmd.CommandType = CommandType.Text

End Select

FillParameters(cmd, params)

Dim da As SqlDataAdapter = New SqlDataAdapter

da.SelectCommand = cmd

Dim ds As DataSet = New DataSet

da.Fill(ds)

Return ds

End Function

Public Function ExecuteDataTable(ByVal v\_tipoSql As tipoSql, ByVal sql As String) As DataTable

cmd = New SqlCommand(sql, cnn)

Select Case v\_tipoSql

Case tipoSql.sql\_procedure

cmd.CommandType = CommandType.StoredProcedure

Case tipoSql.sql\_texto

cmd.CommandType = CommandType.Text

End Select

Dim da As SqlDataAdapter = New SqlDataAdapter

da.SelectCommand = cmd

Dim dt As DataTable = New DataTable

da.Fill(dt)

Return dt

End Function

Public Function ExecuteDataTable(ByVal v\_tipoSql As tipoSql, ByVal sql As String, ByVal params() As SqlParameter) As DataTable

cmd = New SqlCommand(sql, cnn)

Select Case v\_tipoSql

Case tipoSql.sql\_procedure

cmd.CommandType = CommandType.StoredProcedure

Case tipoSql.sql\_texto

cmd.CommandType = CommandType.Text

End Select

FillParameters(cmd, params)

Dim da As SqlDataAdapter = New SqlDataAdapter

da.SelectCommand = cmd

Dim dt As DataTable = New DataTable

da.Fill(dt)

Return dt

End Function

Public Function ExecuteReader(ByVal v\_tipoSql As tipoSql, ByVal sql As String) As SqlDataReader

cmd = New SqlCommand(sql, cnn)

Select Case v\_tipoSql

Case tipoSql.sql\_procedure

cmd.CommandType = CommandType.StoredProcedure

Case tipoSql.sql\_texto

cmd.CommandType = CommandType.Text

End Select

Return cmd.ExecuteReader(CommandBehavior.CloseConnection)

End Function

Public Function ExecuteReader(ByVal v\_tipoSql As tipoSql, ByVal sql As String, ByVal params() As SqlParameter) As SqlDataReader

cmd = New SqlCommand(sql, cnn)

Select Case v\_tipoSql

Case tipoSql.sql\_procedure

cmd.CommandType = CommandType.StoredProcedure

Case tipoSql.sql\_texto

cmd.CommandType = CommandType.Text

End Select

FillParameters(cmd, params)

Return cmd.ExecuteReader(CommandBehavior.CloseConnection)

End Function

Public Function ExecuteNonQuery(ByVal v\_tipoSql As tipoSql, ByVal query As String) As Integer

cmd = New SqlCommand(query, cnn)

Select Case v\_tipoSql

Case tipoSql.sql\_procedure

cmd.CommandType = CommandType.StoredProcedure

Case tipoSql.sql\_texto

cmd.CommandType = CommandType.Text

End Select

Dim retval As Integer = cmd.ExecuteNonQuery()

Return retval

End Function

Public Function ExecuteNonQuery(ByVal v\_tipoSql As tipoSql, ByVal query As String, ByVal params() As SqlParameter) As Integer

cmd = New SqlCommand(query, cnn)

Select Case v\_tipoSql

Case tipoSql.sql\_procedure

cmd.CommandType = CommandType.StoredProcedure

Case tipoSql.sql\_texto

cmd.CommandType = CommandType.Text

End Select

FillParameters(cmd, params)

Dim retval As Integer = cmd.ExecuteNonQuery()

Return retval

End Function

Public Function ExecuteScalar(ByVal v\_tipoSql As tipoSql, ByVal sql As String) As Object

cmd = New SqlCommand(sql, cnn)

Select Case v\_tipoSql

Case tipoSql.sql\_procedure

cmd.CommandType = CommandType.StoredProcedure

Case tipoSql.sql\_texto

cmd.CommandType = CommandType.Text

End Select

Dim retval As Object = cmd.ExecuteScalar()

Return retval

End Function

Public Function ExecuteScalar(ByVal v\_tipoSql As tipoSql, ByVal sql As String, ByVal params() As SqlParameter) As Object

cmd = New SqlCommand(sql, cnn)

Select Case v\_tipoSql

Case tipoSql.sql\_procedure

cmd.CommandType = CommandType.StoredProcedure

Case tipoSql.sql\_texto

cmd.CommandType = CommandType.Text

End Select

FillParameters(cmd, params)

Dim retval As Object = cmd.ExecuteScalar()

Return retval

End Function

Protected Sub FillParameters(ByVal cmd As SqlCommand, ByVal parameters() As SqlParameter)

Dim i As Integer

For i = 0 To parameters.Length - 1

cmd.Parameters.Add(parameters(i))

Next

End Sub

Public Sub close()

If Not cmd Is Nothing Then

cmd.Connection.Dispose()

cmd = Nothing

End If

If Not cnn Is Nothing Then

If cnn.State = ConnectionState.Open Then

cnn.Close()

End If

cnn = Nothing

End If

If Not strConn Is Nothing Then

strConn = Nothing

End If

End Sub

End Class

Public Class Clientes

Private Sub Clientes\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim C\_Titular As New Cliente\_Titular

Dim mid\_Cliente As Integer

'Cliente Titular

dt = C\_Titular.LstGridCliente\_Titular(C\_Titular)

If dt.Rows.Count > 0 Then

With Grid\_Cliente\_Titular

Me.Grid\_Cliente\_Titular.Rows.Clear()

Me.Grid\_Cliente\_Titular.DataSource = dt

Me.Grid\_Cliente\_Titular.AutoResizeColumns()

mid\_Cliente = .CurrentRow.Cells(0).Value

End With

End If

'Cliente Adicional

Dim C\_Adicional As New Cliente\_Adicional

C\_Adicional.p\_id\_titular = mid\_Cliente

dt = C\_Adicional.LstGridCliente\_adicional(C\_Adicional)

If dt.Rows.Count > 0 Then

With Grid\_Cliente\_Adicional

Me.Grid\_Cliente\_Adicional.Rows.Clear()

Me.Grid\_Cliente\_Adicional.DataSource = dt

Me.Grid\_Cliente\_Adicional.AutoResizeColumns()

End With

End If

End Sub

Private Sub btn\_incluir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Dim NewMDIChild As New Frm\_cliente\_titular

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_id\_titular.Text = "0"

NewMDIChild.txt\_id\_dados\_pessoais.Text = "0"

NewMDIChild.txt\_operacao.Text = "I"

NewMDIChild.Show()

End Sub

Private Sub btn\_alterar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Dim NewMDIChild As New Frm\_cliente\_titular

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_operacao.Text = "U"

With Grid\_Cliente\_Titular

NewMDIChild.txt\_id\_titular.Text = .CurrentRow.Cells(0).Value

End With

NewMDIChild.Show()

End Sub

Private Sub btn\_excluir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

If MsgBox("Deseja realmente excluir o Cliente selecionado", MsgBoxStyle.Question + MsgBoxStyle.YesNo, "Excluir Cliente Titular") = MsgBoxResult.Yes Then

Dim dt As New DataTable

Dim C\_Titular As New Cliente\_Titular

Dim C\_Adicional As New Cliente\_Adicional

With Grid\_Cliente\_Titular

C\_Titular.p\_id\_titular = .CurrentRow.Cells(0).Value

End With

If C\_Titular.p\_id\_titular = C\_Adicional.p\_id\_titular Then

MessageBox.Show("Registro do Cliente não pode ser Excluído antes de seus Adicionais!!")

End If

dt = C\_Titular.ExcluirCT(C\_Titular)

MessageBox.Show("Registro excluido com Sucesso!!")

End If

End Sub

Private Sub btn\_incluir2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Dim NewMDIChild As New Frm\_cliente\_adicional

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_Id\_Titular.Text = Grid\_Cliente\_Titular.CurrentRow.Cells(0).Value

NewMDIChild.txt\_id\_adicional.Text = "0"

NewMDIChild.txt\_id\_dados\_pessoais.Text = "0"

NewMDIChild.txt\_operacao.Text = "I"

NewMDIChild.Show()

End Sub

Private Sub btn\_alterar2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Dim NewMDIChild As New Frm\_cliente\_adicional

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_operacao.Text = "U"

With Grid\_Cliente\_Adicional

NewMDIChild.txt\_id\_adicional.Text = .CurrentRow.Cells(0).Value

End With

NewMDIChild.Show()

End Sub

Private Sub btn\_excluir2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

If MsgBox("Deseja realmente excluir o Cliente selecionado", MsgBoxStyle.Question + MsgBoxStyle.YesNo, "Excluir Cliente Adcional") = MsgBoxResult.Yes Then

Dim dt As New DataTable

Dim C\_Adicional As New Cliente\_Adicional

With Grid\_Cliente\_Adicional

C\_Adicional.p\_id\_adicional = .CurrentRow.Cells(0).Value

End With

dt = C\_Adicional.ExcluirCA(C\_Adicional)

MessageBox.Show("Registro excluido com Sucesso!!")

End If

End Sub

Private Sub Grid\_Cliente\_Titular\_CellClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)

Dim mId\_Titular As Integer

Dim clsAdicional As New Cliente\_Adicional

Dim Dt As New DataTable

Dim frm\_Clientes As New Clientes

' Inserir posicionamento dos Adicionais.

mId\_Titular = Grid\_Cliente\_Titular.CurrentRow.Cells(0).Value

clsAdicional.p\_id\_titular = mId\_Titular

Dt = clsAdicional.LstGridCliente\_adicional(clsAdicional)

frm\_Clientes.Grid\_Cliente\_Adicional.Rows.Clear()

If Dt.Rows.Count > 0 Then

With Grid\_Cliente\_Adicional

.DataSource = Dt

.AutoResizeColumns()

End With

ElseIf Dt.Rows.Count = 0 Then

With Grid\_Cliente\_Adicional

.DataSource = Dt

.AutoResizeColumns()

End With

End If

End Sub

Private Sub btn\_pesquisar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Dim dt As New DataTable

Dim pTitular As New Cliente\_Titular

Dim Frm\_Clientes As New Clientes

If Trim(txt\_nome\_pesq.Text) <> "" Then

pTitular.p\_nome = Trim(txt\_nome\_pesq.Text)

End If

If Trim(txt\_id\_titular\_pesq.Text) <> "" Then

pTitular.p\_id\_titular = Trim(txt\_id\_titular\_pesq.Text)

End If

dt = pTitular.LstGridCliente\_Titular(pTitular)

If dt.Rows.Count > 0 Then

Frm\_Clientes.Grid\_Cliente\_Titular.Rows.Clear()

With Grid\_Cliente\_Titular

.DataSource = dt

.AutoResizeColumns()

End With

ElseIf dt.Rows.Count = 0 Then

With Grid\_Cliente\_Titular

.DataSource = dt

.AutoResizeColumns()

End With

End If

dt = Nothing

End Sub

Private Sub btn\_pesquisar2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Dim dt As New DataTable

Dim pAdicional As New Cliente\_Adicional

Dim Frm\_Clientes As New Clientes

If Trim(txt\_id\_adicional\_pesq.Text) <> "" Then

pAdicional.p\_nome = Trim(txt\_nome\_pesq.Text)

End If

If Trim(txt\_id\_adicional\_pesq.Text) <> "" Then

pAdicional.p\_id\_adicional = Trim(txt\_id\_adicional\_pesq.Text)

End If

dt = pAdicional.LstGridCliente\_adicional(pAdicional)

If dt.Rows.Count > 0 Then

Frm\_Clientes.Grid\_Cliente\_Adicional.Rows.Clear()

With Grid\_Cliente\_Adicional

.DataSource = dt

.AutoResizeColumns()

End With

ElseIf dt.Rows.Count = 0 Then

With Grid\_Cliente\_Adicional

.DataSource = dt

.AutoResizeColumns()

End With

End If

dt = Nothing

End Sub

Private Sub btn\_sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Me.Close()

End Sub

Private Sub btn\_sair2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Me.Close()

End Sub

End Class

Public Class Frm\_Consulta\_locacao

Private Sub Frm\_Consulta\_locacao\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim C\_locacao As New Consulta\_locacao

'Locação\_clientes

dt = C\_locacao.ListagemDataTable2Consulta\_Locacao(C\_locacao)

If dt.Rows.Count > 0 Then

With Grid\_de\_Locacoes

Me.Grid\_de\_Locacoes.Rows.Clear()

Me.Grid\_de\_Locacoes.DataSource = dt

Me.Grid\_de\_Locacoes.AutoResizeColumns()

End With

End If

'Locação\_filmes

dt = C\_locacao.ListagemDataTableConsulta\_Locacao(C\_locacao)

If dt.Rows.Count > 0 Then

With Grid\_locacoes

Me.Grid\_locacoes.Rows.Clear()

Me.Grid\_locacoes.DataSource = dt

Me.Grid\_locacoes.AutoResizeColumns()

End With

End If

End Sub

Private Sub btn\_Sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Sair.Click

Me.Close()

End Sub

End Class

Public Class Frm\_Consultas\_reservas

Private Sub Frm\_Consultas\_reservas\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim C\_reserva As New Consulta\_reserva

'Reserva\_clientes

dt = C\_reserva.ListagemDataTable2Consulta\_reserva(C\_reserva)

If dt.Rows.Count > 0 Then

With Grid\_de\_reservas

Me.Grid\_de\_reservas.Rows.Clear()

Me.Grid\_de\_reservas.DataSource = dt

Me.Grid\_de\_reservas.AutoResizeColumns()

End With

End If

'reserva\_filmes

dt = C\_reserva.ListagemDataTableConsulta\_reserva(C\_reserva)

If dt.Rows.Count > 0 Then

With Grid\_reservas

Me.Grid\_reservas.Rows.Clear()

Me.Grid\_reservas.DataSource = dt

Me.Grid\_reservas.AutoResizeColumns()

End With

End If

End Sub

Private Sub btn\_Sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Sair.Click

Me.Close()

End Sub

End Class

Public Class filmes\_midias

Private Sub Filmes\_Midias\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim F\_Filmes As New Filmes

Dim M\_Midias As New Midias

Dim mid\_Midia As Integer

'Grid

dt = M\_Midias.LstGridMidias(M\_Midias)

If dt.Rows.Count > 0 Then

With Grid\_midia

Me.Grid\_midia.Rows.Clear()

Me.Grid\_midia.DataSource = dt

Me.Grid\_midia.AutoResizeColumns()

mid\_Midia = .CurrentRow.Cells(0).Value

End With

End If

dt = Nothing

'Grid

F\_Filmes.p\_id\_midia = mid\_Midia

dt = F\_Filmes.LstGridFilmes(F\_Filmes)

If dt.Rows.Count > 0 Then

With Grid\_filme

Me.Grid\_filme.Rows.Clear()

Me.Grid\_filme.DataSource = dt

Me.Grid\_filme.AutoResizeColumns()

End With

End If

dt = Nothing

End Sub

Private Sub btn\_incluir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_incluir.Click

Dim NewMDIChild As New Frm\_Midias

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_Id\_Midia.Text = "0"

NewMDIChild.txt\_operacao.Text = "I"

NewMDIChild.Show()

End Sub

Private Sub btn\_alterar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_alterar.Click

Dim NewMDIChild As New Frm\_Midias

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_operacao.Text = "U"

With Grid\_midia

NewMDIChild.txt\_id\_midia.Text = .CurrentRow.Cells(0).Value

End With

NewMDIChild.Show()

End Sub

Private Sub btn\_excluir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_excluir.Click

If MsgBox("Deseja realmente excluir a Mídia selecionada", MsgBoxStyle.Question + MsgBoxStyle.YesNo, "Excluir Mídia") = MsgBoxResult.Yes Then

Dim dt As New DataTable

Dim F\_Filmes As New Filmes

Dim M\_Midias As New Midias

With Grid\_midia

M\_Midias.p\_id\_midia = .CurrentRow.Cells(0).Value

End With

If M\_Midias.p\_id\_midia = F\_Filmes.p\_id\_midia Then

MessageBox.Show("Registro desta Midia não pode ser Excluído antes do Filme!!")

End If

dt = M\_Midias.ExcluirMidias(M\_Midias)

MessageBox.Show("Registro excluido com Sucesso!!")

End If

End Sub

Private Sub btn\_sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sair.Click

Me.Close()

End Sub

Private Sub btn\_incluir2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_incluir2.Click

Dim NewMDIChild As New Frm\_filmes

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_Id\_Midia.Text = Grid\_midia.CurrentRow.Cells(0).Value

NewMDIChild.txt\_id\_filme.Text = "0"

NewMDIChild.txt\_operacao.Text = "I"

NewMDIChild.Show()

End Sub

Private Sub btn\_alterar2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_alterar2.Click

Dim NewMDIChild As New Frm\_filmes

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_operacao.Text = "U"

With Grid\_filme

NewMDIChild.txt\_id\_filme.Text = .CurrentRow.Cells(0).Value

End With

NewMDIChild.Show()

End Sub

Private Sub btn\_excluir2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_excluir2.Click

If MsgBox("Deseja realmente excluir o Filme selecionado", MsgBoxStyle.Question + MsgBoxStyle.YesNo, "Excluir Filme") = MsgBoxResult.Yes Then

Dim dt As New DataTable

Dim F\_Filmes As New Filmes

With Grid\_filme

F\_Filmes.p\_id\_filme = .CurrentRow.Cells(0).Value

End With

dt = F\_Filmes.ExcluirFilmes(F\_Filmes)

MessageBox.Show("Registro excluido com Sucesso!!")

End If

End Sub

Private Sub btn\_sair2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sair2.Click

Me.Close()

End Sub

Private Sub Grid\_midia\_CellClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles Grid\_midia.CellClick

Dim mId\_Midia As Integer

Dim clsFilme As New Filmes

Dim Dt As New DataTable

Dim frm\_filmes\_midias As New filmes\_midias

' Inserir posicionamento dos filmes.

mId\_Midia = Grid\_midia.CurrentRow.Cells(0).Value

clsFilme.p\_id\_midia = mId\_Midia

Dt = clsFilme.LstGridFilmes(clsFilme)

frm\_filmes\_midias.Grid\_filme.Rows.Clear()

If Dt.Rows.Count > 0 Then

With Grid\_filme

.DataSource = Dt

.AutoResizeColumns()

End With

ElseIf Dt.Rows.Count = 0 Then

With Grid\_filme

.DataSource = Dt

.AutoResizeColumns()

End With

End If

End Sub

Private Sub btn\_pesquisar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_pesquisar.Click

Dim dt As New DataTable

Dim pMidia As New Midias

Dim Frm\_filmes\_midias As New filmes\_midias

If Trim(txt\_midia\_pesq.Text) <> "" Then

pMidia.p\_midia = Trim(txt\_midia\_pesq.Text)

End If

If Trim(txt\_id\_midia\_pesq.Text) <> "" Then

pMidia.p\_id\_midia = Trim(txt\_id\_midia\_pesq.Text)

pMidia.p\_midia = Trim(txt\_midia\_pesq.Text)

End If

dt = pMidia.LstGridMidias(pMidia)

If dt.Rows.Count > 0 Then

Frm\_filmes\_midias.Grid\_midia.Rows.Clear()

With Grid\_midia

.DataSource = dt

.AutoResizeColumns()

End With

ElseIf dt.Rows.Count = 0 Then

With Grid\_midia

.DataSource = dt

.AutoResizeColumns()

End With

End If

dt = Nothing

End Sub

Private Sub btn\_pesquisar2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_pesquisar2.Click

Dim dt As New DataTable

Dim pFilmes As New Filmes

Dim Frm\_filmes\_midias As New filmes\_midias

If Trim(txt\_filme\_pesq.Text) <> "" Then

pFilmes.p\_filme = Trim(txt\_filme\_pesq.Text)

End If

If Trim(txt\_id\_filme\_pesq.Text) <> "" Then

pFilmes.p\_id\_filme = Trim(txt\_id\_filme\_pesq.Text)

End If

dt = pFilmes.LstGridFilmes(pFilmes)

If dt.Rows.Count > 0 Then

Frm\_filmes\_midias.Grid\_filme.Rows.Clear()

With Grid\_filme

.DataSource = dt

.AutoResizeColumns()

End With

ElseIf dt.Rows.Count = 0 Then

With Grid\_filme

.DataSource = dt

.AutoResizeColumns()

End With

End If

dt = Nothing

End Sub

End Class

Public Class Fornecedores

Private Sub fornecedores\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim Fornecedores\_F As New f\_Fornecedores

'Grid

dt = Fornecedores\_F.LstGridf\_Fornecedores(Fornecedores\_F)

If dt.Rows.Count > 0 Then

Me.Grid\_fornecedores.Rows.Clear()

Me.Grid\_fornecedores.DataSource = dt

Me.Grid\_fornecedores.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCellsExceptHeader)

End If

dt = Nothing

End Sub

Private Sub btn\_incluir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_incluir.Click

Dim NewMDIChild As New Frm\_fornecedores

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_id\_fornecedor.Text = "0"

NewMDIChild.txt\_operacao.Text = "I"

NewMDIChild.Show()

End Sub

Private Sub btn\_alterar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_alterar.Click

Dim NewMDIChild As New Frm\_fornecedores

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_operacao.Text = "U"

With Grid\_fornecedores

NewMDIChild.txt\_id\_fornecedor.Text = .CurrentRow.Cells(0).Value

End With

NewMDIChild.Show()

End Sub

Private Sub btn\_excluir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_excluir.Click

If MsgBox("Deseja realmente excluir registro do Fornecedor selecionado", MsgBoxStyle.Question + MsgBoxStyle.YesNo, "Excluir Fornecedor") = MsgBoxResult.Yes Then

Dim dt As New DataTable

Dim Fornecedores\_F As New f\_Fornecedores

With Grid\_fornecedores

Fornecedores\_F.p\_id\_fornecedor = .CurrentRow.Cells(0).Value

End With

dt = Fornecedores\_F.ExcluirFornecedores(Fornecedores\_F)

MessageBox.Show("Registro excluido com Sucesso!!")

End If

End Sub

Private Sub btn\_pesquisar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_pesquisar.Click

Dim dt As New DataTable

Dim pFornecedores As New f\_Fornecedores

Dim Frm\_Fornecedores As New Fornecedores

If Trim(txt\_razao\_social\_pesq.Text) <> "" Then

pFornecedores.p\_razao\_social = Trim(txt\_razao\_social\_pesq.Text)

End If

If Trim(txt\_id\_fornecedor\_pesq.Text) <> "" Then

pFornecedores.p\_id\_fornecedor = Trim(txt\_id\_fornecedor\_pesq.Text)

End If

If Trim(txt\_nome\_fantasia\_pesq.Text) <> "" Then

pFornecedores.p\_nome\_fantasia = Trim(txt\_nome\_fantasia\_pesq.Text)

End If

If Trim(txt\_cnpj\_pesq.Text) <> "" Then

pFornecedores.p\_cnpj = Trim(txt\_cnpj\_pesq.Text)

End If

dt = pFornecedores.LstGridf\_Fornecedores(pFornecedores)

If dt.Rows.Count > 0 Then

Frm\_Fornecedores.Grid\_fornecedores.Rows.Clear()

With Grid\_fornecedores

.DataSource = dt

.AutoResizeColumns()

End With

ElseIf dt.Rows.Count = 0 Then

With Grid\_fornecedores

.DataSource = dt

.AutoResizeColumns()

End With

End If

dt = Nothing

End Sub

Private Sub btn\_sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sair.Click

Me.Close()

End Sub

End Class

Public Class f\_funcionarios

Private Sub f\_funcionarios\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim f\_Funcionarios As New Funcionarios

'Grid

dt = f\_funcionarios.LstGridFuncionarios(f\_funcionarios)

If dt.Rows.Count > 0 Then

Me.Grid\_funcionarios.Rows.Clear()

Me.Grid\_funcionarios.DataSource = dt Me.Grid\_funcionarios.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCellsExceptHeader)

End If

dt = Nothing

End Sub

Private Sub btn\_incluir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_incluir.Click

Dim NewMDIChild As New Frm\_funcionarios

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_id\_funcionario.Text = "0"

NewMDIChild.txt\_id\_dados\_pessoais.Text = "0"

NewMDIChild.txt\_operacao.Text = "I"

NewMDIChild.Show()

End Sub

Private Sub btn\_alterar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_alterar.Click

Dim NewMDIChild As New Frm\_funcionarios

NewMDIChild.MdiParent = MDIProjeto

NewMDIChild.txt\_operacao.Text = "U"

With Grid\_funcionarios

NewMDIChild.txt\_id\_funcionario.Text = .CurrentRow.Cells(0).Value

End With

NewMDIChild.Show()

End Sub

Private Sub btn\_excluir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_excluir.Click

If MsgBox("Deseja realmente excluir o Funcionario selecionado", MsgBoxStyle.Question + MsgBoxStyle.YesNo, "Excluir Funcionario") = MsgBoxResult.Yes Then

Dim dt As New DataTable

Dim f\_funcionarios As New Funcionarios

With Grid\_funcionarios

f\_funcionarios.p\_id\_funcionario = .CurrentRow.Cells(0).Value

End With

dt = f\_funcionarios.ExcluirFuncionarios(f\_funcionarios)

MessageBox.Show("Registro excluido com Sucesso!!")

End If

End Sub

Private Sub btn\_pesquisar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_pesquisar.Click

Dim dt As New DataTable

Dim pFuncionarios As New Funcionarios

Dim Frm\_Funcionarios As New f\_funcionarios

If Trim(txt\_nome\_pesq.Text) <> "" Then

pFuncionarios.p\_nome = Trim(txt\_nome\_pesq.Text)

End If

If Trim(txt\_id\_funcionario\_pesq.Text) <> "" Then

pFuncionarios.p\_id\_funcionario = Trim(txt\_id\_funcionario\_pesq.Text)

End If

dt = pFuncionarios.LstGridFuncionarios(pFuncionarios)

If dt.Rows.Count > 0 Then

Frm\_Funcionarios.Grid\_funcionarios.Rows.Clear()

With Grid\_funcionarios

.DataSource = dt

.AutoResizeColumns()

End With

ElseIf dt.Rows.Count = 0 Then

With Grid\_funcionarios

.DataSource = dt

.AutoResizeColumns()

End With

End If

dt = Nothing

End Sub

Private Sub btn\_sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sair.Click

Me.Close()

End Sub

End Class

Public Class Frm\_cliente\_adicional

Private Sub Frm\_cliente\_adicional\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim dr As DataRow

Dim C\_Adicional As New Cliente\_Adicional

'CLIENTE ADCIONAL

If Trim(txt\_operacao.Text) <> "I" And Trim(txt\_id\_adicional.Text) <> "" Then

C\_Adicional.p\_id\_adicional = Trim(txt\_id\_adicional.Text)

dt = C\_Adicional.RetornarDadosCliente\_Adicional(C\_Adicional)

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

txt\_id\_adicional.Text = Trim(UCase(dr("ID\_ADICIONAL")))

txt\_Id\_Titular.Text = Trim(UCase(dr("ID\_TITULAR")))

txt\_id\_dados\_pessoais.Text = Trim(UCase(dr("ID\_DADOS\_PESSOAIS")))

If Not IsDBNull(dr("NOME")) Then

txt\_nome.Text = dr("NOME")

End If

If Not IsDBNull(dr("DATA\_NASCIMENTO")) Then

Date\_adcional.Text = dr("DATA\_NASCIMENTO")

End If

If Not IsDBNull(dr("SEXO")) Then

cbx\_sexo.Text = Trim(dr("SEXO"))

End If

If Not IsDBNull(dr("RG")) Then

txt\_rg.Text = dr("RG")

End If

If Not IsDBNull(dr("CPF")) Then

txt\_cpf.Text = dr("CPF")

End If

If Not IsDBNull(dr("TELEFONE")) Then

txt\_telefone.Text = dr("TELEFONE")

End If

If Not IsDBNull(dr("CELULAR")) Then

txt\_celular.Text = dr("CELULAR")

End If

If Not IsDBNull(dr("E\_MAIL")) Then

txt\_email.Text = dr("E\_MAIL")

End If

If Not IsDBNull(dr("GRAU\_PARENTESCO")) Then

cbx\_grau\_parentesco.Text = dr("GRAU\_PARENTESCO")

End If

If Not IsDBNull(dr("LOGRADOURO")) Then

txt\_logradouro.Text = dr("LOGRADOURO")

End If

If Not IsDBNull(dr("NUMERO")) Then

txt\_numero.Text = dr("NUMERO")

End If

If Not IsDBNull(dr("COMPLEMENTO")) Then

txt\_complemento.Text = dr("COMPLEMENTO")

End If

If Not IsDBNull(dr("BAIRRO")) Then

txt\_bairro.Text = dr("BAIRRO")

End If

If Not IsDBNull(dr("CIDADE")) Then

txt\_cidade.Text = dr("CIDADE")

End If

If Not IsDBNull(dr("ESTADO")) Then

txt\_estado.Text = dr("ESTADO")

End If

If Not IsDBNull(dr("CEP")) Then

txt\_cep.Text = dr("CEP")

End If

Next

End If

dt = Nothing

End If

End Sub

Private Sub btn\_gravar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_gravar.Click

Dim sSql As String = ""

Dim dt As New DataTable

Dim C\_Adicional As New Cliente\_Adicional

Dim dr As DataRow

If Trim(txt\_nome.Text) = "" Then

MsgBox("Nome Inválido!! Nome tem que ser preenchido")

txt\_nome.Focus()

Exit Sub

End If

If Date\_adcional.Text = "" Then

MsgBox("Data Nascimento Inválido!!!")

Date\_adcional.Focus()

Exit Sub

End If

If txt\_cpf.Text = "" Then

MsgBox("CPF Inválido!!!")

txt\_cpf.Focus()

Exit Sub

End If

If txt\_telefone.Text = "" Then

MsgBox("TELEFONE Inválido!!!")

txt\_telefone.Focus()

Exit Sub

End If

If txt\_celular.Text = "" Then

MsgBox("CELULAR Inválido!!!")

txt\_celular.Focus()

Exit Sub

End If

If txt\_logradouro.Text = "" Then

MsgBox("LOGRADOURO Inválido!!!")

txt\_logradouro.Focus()

Exit Sub

End If

If txt\_numero.Text = "" Then

MsgBox("NUMERO Inválido!!!")

txt\_numero.Focus()

Exit Sub

End If

If txt\_cep.Text = "" Then

MsgBox("CEP Inválido!!!")

txt\_cep.Focus()

Exit Sub

End If

'Preenchendo Data de Nascimento

Dim AMD As String

AMD = CDate(Date\_adcional.Text).Day & "/" & CDate(Date\_adcional.Text).Month & "/" & CDate(Date\_adcional.Text).Year

'Gravando os Dados Pessoais do cliente

Dim mId\_Dados\_Pessoais As Integer

Try

Dim v\_Dados\_Pessoais As New Dados\_Pessoais

v\_Dados\_Pessoais.p\_id\_dados\_pessoais = Trim(txt\_id\_dados\_pessoais.Text)

v\_Dados\_Pessoais.p\_rg = Trim(txt\_rg.Text)

v\_Dados\_Pessoais.p\_cpf = Trim(txt\_cpf.Text)

v\_Dados\_Pessoais.p\_sexo = Trim(cbx\_sexo.Text)

v\_Dados\_Pessoais.p\_data\_nascimento = AMD

v\_Dados\_Pessoais.p\_telefone = Trim(txt\_telefone.Text)

v\_Dados\_Pessoais.p\_celular = Trim(txt\_celular.Text)

v\_Dados\_Pessoais.p\_e\_mail = Trim(txt\_email.Text)

If txt\_operacao.Text = "I" Then

dt = v\_Dados\_Pessoais.InserirDP(v\_Dados\_Pessoais)

ElseIf txt\_operacao.Text = "U" Then

dt = v\_Dados\_Pessoais.AlterarDP(v\_Dados\_Pessoais)

ElseIf txt\_operacao.Text = "D" Then

dt = v\_Dados\_Pessoais.ExcluirDP(v\_Dados\_Pessoais)

End If

dr = Nothing

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

mId\_Dados\_Pessoais = dr("ID\_DADOS\_PESSOAIS")

Next

End If

'Gravando o Endereço caso não exista.

Dim v\_Enderecos As New Enderecos

v\_Enderecos.p\_cep = Trim(txt\_cep.Text)

txt\_cep.Text = CType(txt\_cep.Text, Decimal).ToString("0####-###")

v\_Enderecos.p\_cep = Trim(txt\_cep.Text)

dt = v\_Enderecos.RetornarDadosEnderecos(v\_Enderecos)

v\_Enderecos.p\_cep = Trim(txt\_cep.Text)

v\_Enderecos.p\_logradouro = Trim(txt\_logradouro.Text)

v\_Enderecos.p\_numero = Trim(txt\_numero.Text)

v\_Enderecos.p\_complemento = Trim(txt\_complemento.Text)

v\_Enderecos.p\_bairro = Trim(txt\_bairro.Text)

v\_Enderecos.p\_cidade = Trim(txt\_cidade.Text)

v\_Enderecos.p\_estado = Trim(txt\_estado.Text)

If txt\_operacao.Text = "I" Then

dt = v\_Enderecos.InserirEnderecos(v\_Enderecos)

ElseIf txt\_operacao.Text = "U" Then

dt = v\_Enderecos.AlterarEnderecos(v\_Enderecos)

ElseIf txt\_operacao.Text = "D" Then

dt = v\_Enderecos.ExcluirEnderecos(v\_Enderecos)

End If

dr = Nothing

'Gravando Dados do Cliente caso não exista

C\_Adicional.p\_id\_adicional = Trim(txt\_id\_adicional.Text)

C\_Adicional.p\_nome = Trim(txt\_nome.Text)

C\_Adicional.p\_id\_titular = txt\_Id\_Titular.Text

C\_Adicional.p\_cep = Trim(txt\_cep.Text)

C\_Adicional.p\_grau\_parentesco = Trim(cbx\_grau\_parentesco.Text)

C\_Adicional.p\_id\_dados\_pessoais = mId\_Dados\_Pessoais

If txt\_operacao.Text = "I" Then

dt = C\_Adicional.InserirCA(C\_Adicional)

ElseIf txt\_operacao.Text = "U" Then

dt = C\_Adicional.AlterarCA(C\_Adicional)

ElseIf txt\_operacao.Text = "D" Then

dt = C\_Adicional.ExcluirCA(C\_Adicional)

End If

MsgBox("Registro Gravado com Sucesso!!", MsgBoxStyle.Information + MsgBoxStyle.Information, "Gravar Registro")

dr = Nothing

Catch ex As Exception

End Try

End Sub

Private Sub btn\_sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sair.Click

Me.Close()

End Sub

End Class

Public Class Frm\_cliente\_titular

Private Sub Frm\_cliente\_titular\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim dr As DataRow

Dim C\_Titular As New Cliente\_Titular

'CLIENTE TITULAR

If Trim(txt\_operacao.Text) <> "I" And Trim(txt\_id\_titular.Text) <> "" Then

C\_Titular.p\_id\_titular = Trim(txt\_id\_titular.Text)

dt = C\_Titular.RetornarDadosCiente\_Titular(C\_Titular)

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

txt\_id\_titular.Text = Trim(UCase(dr("ID\_TITULAR")))

txt\_id\_dados\_pessoais.Text = Trim(UCase(dr("ID\_DADOS\_PESSOAIS")))

If Not IsDBNull(dr("NOME")) Then

txt\_nome.Text = Trim(UCase(dr("NOME")))

End If

If Not IsDBNull(dr("DATA\_NASCIMENTO")) Then

Date\_titular.Text = dr("DATA\_NASCIMENTO")

End If

If Not IsDBNull(dr("SEXO")) Then

cbx\_sexo.Text = Trim(dr("SEXO"))

End If

If Not IsDBNull(dr("RG")) Then

txt\_rg.Text = dr("RG")

End If

If Not IsDBNull(dr("CPF")) Then

txt\_cpf.Text = dr("CPF")

End If

If Not IsDBNull(dr("TELEFONE")) Then

txt\_telefone.Text = dr("TELEFONE")

End If

If Not IsDBNull(dr("CELULAR")) Then

txt\_celular.Text = dr("CELULAR")

End If

If Not IsDBNull(dr("E\_MAIL")) Then

txt\_email.Text = dr("E\_MAIL")

End If

If Not IsDBNull(dr("LOGRADOURO")) Then

txt\_logradouro.Text = dr("LOGRADOURO")

End If

If Not IsDBNull(dr("NUMERO")) Then

txt\_numero.Text = dr("NUMERO")

End If

If Not IsDBNull(dr("COMPLEMENTO")) Then

txt\_complemento.Text = dr("COMPLEMENTO")

End If

If Not IsDBNull(dr("BAIRRO")) Then

txt\_bairro.Text = dr("BAIRRO")

End If

If Not IsDBNull(dr("CIDADE")) Then

txt\_cidade.Text = dr("CIDADE")

End If

If Not IsDBNull(dr("ESTADO")) Then

txt\_estado.Text = dr("ESTADO")

End If

If Not IsDBNull(dr("CEP")) Then

txt\_cep.Text = dr("CEP")

End If

Next

End If

dt = Nothing

End If

End Sub

Private Sub btn\_gravar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_gravar.Click

Dim sSql As String = ""

Dim dt As New DataTable

Dim dr As DataRow

Dim C\_Titular As New Cliente\_Titular

If Trim(txt\_nome.Text) = "" Then

MsgBox("Nome Inválido!! Nome tem que ser preenchido")

txt\_nome.Focus()

Exit Sub

End If

If Date\_titular.Text = "" Then

MsgBox("Data Nascimento Inválido!!!")

Date\_titular.Focus()

Exit Sub

End If

If txt\_cpf.Text = "" Then

MsgBox("CPF Inválido!!!")

txt\_cpf.Focus()

Exit Sub

End If

If txt\_telefone.Text = "" Then

MsgBox("TELEFONE Inválido!!!")

txt\_telefone.Focus()

Exit Sub

End If

If txt\_celular.Text = "" Then

MsgBox("CELULAR Inválido!!!")

txt\_celular.Focus()

Exit Sub

End If

If txt\_logradouro.Text = "" Then

MsgBox("LOGRADOURO Inválido!!!")

txt\_logradouro.Focus()

Exit Sub

End If

If txt\_numero.Text = "" Then

MsgBox("NUMERO Inválido!!!")

txt\_numero.Focus()

Exit Sub

End If

If txt\_cep.Text = "" Then

MsgBox("CEP Inválido!!!")

txt\_cep.Focus()

Exit Sub

End If

'Preenchendo Data de Nascimento

Dim AMD As String

AMD = CDate(Date\_titular.Text).Day & "/" & CDate(Date\_titular.Text).Month & "/" & CDate(Date\_titular.Text).Year

'Gravando os Dados Pessoais do cliente

Dim mId\_Dados\_Pessoais As Integer

Try

Dim v\_Dados\_Pessoais As New Dados\_Pessoais

v\_Dados\_Pessoais.p\_id\_dados\_pessoais = Trim(txt\_id\_dados\_pessoais.Text)

v\_Dados\_Pessoais.p\_rg = Trim(txt\_rg.Text)

v\_Dados\_Pessoais.p\_cpf = Trim(txt\_cpf.Text)

v\_Dados\_Pessoais.p\_sexo = Trim(cbx\_sexo.Text)

v\_Dados\_Pessoais.p\_data\_nascimento = AMD

v\_Dados\_Pessoais.p\_telefone = Trim(txt\_telefone.Text)

v\_Dados\_Pessoais.p\_celular = Trim(txt\_celular.Text)

v\_Dados\_Pessoais.p\_e\_mail = Trim(txt\_email.Text)

If txt\_operacao.Text = "I" Then

dt = v\_Dados\_Pessoais.InserirDP(v\_Dados\_Pessoais)

ElseIf txt\_operacao.Text = "U" Then

dt = v\_Dados\_Pessoais.AlterarDP(v\_Dados\_Pessoais)

ElseIf txt\_operacao.Text = "D" Then

dt = v\_Dados\_Pessoais.ExcluirDP(v\_Dados\_Pessoais)

End If

dr = Nothing

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

mId\_Dados\_Pessoais = dr("ID\_DADOS\_PESSOAIS")

Next

End If

'Gravando o Endereço caso não exista.

Dim v\_Enderecos As New Enderecos

v\_Enderecos.p\_cep = Trim(txt\_cep.Text)

txt\_cep.Text = CType(txt\_cep.Text, Decimal).ToString("0####-###")

dt = v\_Enderecos.RetornarDadosEnderecos(v\_Enderecos)

v\_Enderecos.p\_cep = Trim(txt\_cep.Text)

v\_Enderecos.p\_logradouro = Trim(txt\_logradouro.Text)

v\_Enderecos.p\_numero = Trim(txt\_numero.Text)

v\_Enderecos.p\_complemento = Trim(txt\_complemento.Text)

v\_Enderecos.p\_bairro = Trim(txt\_bairro.Text)

v\_Enderecos.p\_cidade = Trim(txt\_cidade.Text)

v\_Enderecos.p\_estado = Trim(txt\_estado.Text)

If txt\_operacao.Text = "I" Then

dt = v\_Enderecos.InserirEnderecos(v\_Enderecos)

ElseIf txt\_operacao.Text = "U" Then

dt = v\_Enderecos.AlterarEnderecos(v\_Enderecos)

ElseIf txt\_operacao.Text = "D" Then

dt = v\_Enderecos.ExcluirEnderecos(v\_Enderecos)

End If

dr = Nothing

'Gravando Dados do Cliente caso não exista

C\_Titular.p\_id\_titular = Trim(txt\_id\_titular.Text)

C\_Titular.p\_nome = Trim(txt\_nome.Text)

C\_Titular.p\_cep = Trim(txt\_cep.Text)

C\_Titular.p\_id\_dados\_pessoais = mId\_Dados\_Pessoais

If txt\_operacao.Text = "I" Then

dt = C\_Titular.InserirCT(C\_Titular)

ElseIf txt\_operacao.Text = "U" Then

dt = C\_Titular.AlterarCT(C\_Titular)

ElseIf txt\_operacao.Text = "D" Then

dt = C\_Titular.ExcluirCT(C\_Titular)

End If

MsgBox("Registro Gravado com Sucesso!!", MsgBoxStyle.Information + MsgBoxStyle.Information, "Gravar Registro")

dr = Nothing

Catch ex As Exception

End Try

End Sub

Private Sub btn\_sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sair.Click

Me.Close()

End Sub

End Class

Public Class Frm\_filmes

Private Sub Frm\_filmes\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim dr As DataRow

Dim F\_Filmes As New Filmes

If Trim(txt\_operacao.Text) <> "I" And Trim(txt\_id\_filme.Text) <> "" Then

F\_Filmes.p\_id\_filme = Trim(txt\_id\_filme.Text)

dt = F\_Filmes.RetornarDadosFilmes(F\_Filmes)

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

txt\_id\_filme.Text = Trim(UCase(dr("ID\_FILME")))

txt\_Id\_Midia.Text = Trim(UCase(dr("ID\_MIDIA")))

If Not IsDBNull(dr("FILME")) Then

txt\_filme.Text = Trim(dr("FILME"))

End If

If Not IsDBNull(dr("ID\_MIDIA")) Then

txt\_Id\_Midia.Text = Trim(dr("ID\_MIDIA"))

End If

If Not IsDBNull(dr("ANO\_LANCAMENTO")) Then

txt\_ano\_lancamento.Text = Trim(dr("ANO\_LANCAMENTO"))

End If

If Not IsDBNull(dr("GENERO")) Then

cbx\_genero.Text = Trim(dr("GENERO"))

End If

If Not IsDBNull(dr("NACIONALIDADE")) Then

txt\_nacionalidade.Text = Trim(dr("NACIONALIDADE"))

End If

If Not IsDBNull(dr("DURACAO")) Then

txt\_duracao.Text = Trim(dr("DURACAO"))

End If

If Not IsDBNull(dr("DIRECAO")) Then

txt\_direcao.Text = Trim(dr("DIRECAO"))

End If

If Not IsDBNull(dr("SINOPSE")) Then

txt\_sinopse.Text = Trim(dr("SINOPSE"))

End If

If Not IsDBNull(dr("PRECO")) Then

txt\_preco.Text = Trim(dr("PRECO"))

End If

Next

End If

dt = Nothing

End If

End Sub

Private Sub btn\_gravar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_gravar.Click

Dim sSql As String = ""

Dim dt As New DataTable

Dim dr As DataRow

If Trim(txt\_filme.Text) = "" Then

MsgBox("Filme Inválido!! Filme tem que ser preenchido")

txt\_filme.Focus()

Exit Sub

End If

If txt\_ano\_lancamento.Text = "" Then

MsgBox("Ano de Lancamento Inválido!!!")

txt\_ano\_lancamento.Focus()

Exit Sub

End If

If cbx\_genero.Text = "" Then

MsgBox("Genero Inválido!!!")

cbx\_genero.Focus()

Exit Sub

End If

If txt\_nacionalidade.Text = "" Then

MsgBox("Nacionalidade Inválida!!!")

txt\_nacionalidade.Focus()

Exit Sub

End If

If txt\_duracao.Text = "" Then

MsgBox("Duracao Inválida!!!")

txt\_duracao.Focus()

Exit Sub

End If

If txt\_direcao.Text = "" Then

MsgBox("Direcao Inválida!!!")

txt\_direcao.Focus()

Exit Sub

End If

If txt\_sinopse.Text = "" Then

MsgBox("Sinopse Inválida!!!")

txt\_sinopse.Focus()

Exit Sub

End If

If Val(Trim(txt\_preco.Text)) = 0 Then

MsgBox("Preço do Filme Inválido!!")

txt\_preco.Focus()

Exit Sub

End If

Try

'Gravando Dados do Filme

Dim F\_Filmes As New Filmes

F\_Filmes.p\_id\_filme = Trim(txt\_id\_filme.Text)

F\_Filmes.p\_filme = Trim(txt\_filme.Text)

F\_Filmes.p\_id\_midia = Trim(txt\_Id\_Midia.Text)

F\_Filmes.p\_ano\_lancamento = Trim(txt\_ano\_lancamento.Text)

F\_Filmes.p\_genero = Trim(cbx\_genero.Text)

F\_Filmes.p\_nacionalidade = Trim(txt\_nacionalidade.Text)

F\_Filmes.p\_duracao = Trim(txt\_duracao.Text)

F\_Filmes.p\_direcao = Trim(txt\_direcao.Text)

F\_Filmes.p\_sinopse = Trim(txt\_sinopse.Text)

F\_Filmes.p\_preco = CType(txt\_preco.Text, Decimal)

MsgBox("Registro Gravado com Sucesso!!", MsgBoxStyle.Information + MsgBoxStyle.Information, "Gravar Registro")

If txt\_operacao.Text = "I" Then

dt = F\_Filmes.InserirFilmes(F\_Filmes)

ElseIf txt\_operacao.Text = "U" Then

dt = F\_Filmes.AlterarFilmes(F\_Filmes)

ElseIf txt\_operacao.Text = "D" Then

dt = F\_Filmes.ExcluirFilmes(F\_Filmes)

End If

dr = Nothing

Catch ex As Exception

End Try

End Sub

Private Sub btn\_sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sair.Click

Me.Close()

End Sub

End Class

Public Class Frm\_fornecedores

Private Sub Frm\_fornecedores\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim dr As DataRow

Dim Fornecedores\_F As New f\_Fornecedores

'FORNECEDORES

If Trim(txt\_operacao.Text) <> "I" And Trim(txt\_id\_fornecedor.Text) <> "" Then

Fornecedores\_F.p\_id\_fornecedor = Trim(txt\_id\_fornecedor.Text)

dt = Fornecedores\_F.RetornarDadosFornecedores(Fornecedores\_F)

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

txt\_id\_fornecedor.Text = Trim(UCase(dr("ID\_FORNECEDOR")))

If Not IsDBNull(dr("RAZAO\_SOCIAL")) Then

txt\_razao\_social.Text = dr("RAZAO\_SOCIAL")

End If

If Not IsDBNull(dr("NOME\_FANTASIA")) Then

txt\_nome\_fantasia.Text = dr("NOME\_FANTASIA")

End If

If Not IsDBNull(dr("CNPJ")) Then

txt\_cnpj.Text = dr("CNPJ")

End If

If Not IsDBNull(dr("CONTATO")) Then

txt\_contato.Text = dr("CONTATO")

End If

If Not IsDBNull(dr("TELEFONE")) Then

txt\_telefone.Text = dr("TELEFONE")

End If

If Not IsDBNull(dr("FAX")) Then

txt\_fax.Text = dr("FAX")

End If

If Not IsDBNull(dr("E\_MAIL")) Then

txt\_email.Text = dr("E\_MAIL")

End If

If Not IsDBNull(dr("LOGRADOURO")) Then

txt\_logradouro.Text = dr("LOGRADOURO")

End If

If Not IsDBNull(dr("NUMERO")) Then

txt\_numero.Text = dr("NUMERO")

End If

If Not IsDBNull(dr("COMPLEMENTO")) Then

txt\_complemento.Text = dr("COMPLEMENTO")

End If

If Not IsDBNull(dr("BAIRRO")) Then

txt\_bairro.Text = dr("BAIRRO")

End If

If Not IsDBNull(dr("CIDADE")) Then

txt\_cidade.Text = dr("CIDADE")

End If

If Not IsDBNull(dr("ESTADO")) Then

txt\_estado.Text = dr("ESTADO")

End If

If Not IsDBNull(dr("CEP")) Then

txt\_cep.Text = dr("CEP")

End If

Next

End If

dt = Nothing

End If

End Sub

Private Sub btn\_gravar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_gravar.Click

Dim sSql As String = ""

Dim dt As New DataTable

Dim dr As DataRow

Dim Fornecedores\_F As New f\_Fornecedores

If Trim(txt\_razao\_social.Text) = "" Then

MsgBox("Razao Social Inválido!! Razao Social tem que ser preenchido")

txt\_razao\_social.Focus()

Exit Sub

End If

If txt\_nome\_fantasia.Text = "" Then

MsgBox("Nome Fantasia Inválido!!!")

txt\_nome\_fantasia.Focus()

Exit Sub

End If

If txt\_cnpj.Text = "" Then

MsgBox("CNPJ Inválido!!!")

txt\_cnpj.Focus()

Exit Sub

End If

If txt\_contato.Text = "" Then

MsgBox("Contato Inválido!!!")

txt\_contato.Focus()

Exit Sub

End If

If txt\_telefone.Text = "" Then

MsgBox("Telefone Inválido!!!")

txt\_telefone.Focus()

Exit Sub

End If

If txt\_fax.Text = "" Then

MsgBox("Fax Inválido!!!")

txt\_fax.Focus()

Exit Sub

End If

If txt\_email.Text = "" Then

MsgBox("E\_mail Inválido!!!")

txt\_email.Focus()

Exit Sub

End If

If txt\_logradouro.Text = "" Then

MsgBox("LOGRADOURO Inválido!!!")

txt\_logradouro.Focus()

Exit Sub

End If

If txt\_numero.Text = "" Then

MsgBox("NUMERO Inválido!!!")

txt\_numero.Focus()

Exit Sub

End If

If txt\_cep.Text = "" Then

MsgBox("CEP Inválido!!!")

txt\_cep.Focus()

Exit Sub

End If

'Gravando o Endereço caso não exista.

Try

'Gravando o Endereço caso não exista.

Dim v\_Enderecos As New Enderecos

v\_Enderecos.p\_cep = Trim(txt\_cep.Text)

txt\_cep.Text = CType(txt\_cep.Text, Decimal).ToString("0####-###")

dt = v\_Enderecos.RetornarDadosEnderecos(v\_Enderecos)

v\_Enderecos.p\_cep = Trim(txt\_cep.Text)

v\_Enderecos.p\_logradouro = Trim(txt\_logradouro.Text)

v\_Enderecos.p\_numero = Trim(txt\_numero.Text)

v\_Enderecos.p\_complemento = Trim(txt\_complemento.Text)

v\_Enderecos.p\_bairro = Trim(txt\_bairro.Text)

v\_Enderecos.p\_cidade = Trim(txt\_cidade.Text)

v\_Enderecos.p\_estado = Trim(txt\_estado.Text)

If txt\_operacao.Text = "I" Then

dt = v\_Enderecos.InserirEnderecos(v\_Enderecos)

ElseIf txt\_operacao.Text = "U" Then

dt = v\_Enderecos.AlterarEnderecos(v\_Enderecos)

ElseIf txt\_operacao.Text = "D" Then

dt = v\_Enderecos.ExcluirEnderecos(v\_Enderecos)

End If

dr = Nothing

'Gravando Dados do Fornecedor caso não exista

Fornecedores\_F.p\_id\_fornecedor = Trim(txt\_id\_fornecedor.Text)

Fornecedores\_F.p\_razao\_social = Trim(txt\_razao\_social.Text)

Fornecedores\_F.p\_nome\_fantasia = Trim(txt\_nome\_fantasia.Text)

Fornecedores\_F.p\_cnpj = Trim(txt\_cnpj.Text)

Fornecedores\_F.p\_contato = Trim(txt\_contato.Text)

Fornecedores\_F.p\_telefone = Trim(txt\_telefone.Text)

Fornecedores\_F.p\_fax = Trim(txt\_fax.Text)

Fornecedores\_F.p\_e\_mail = Trim(txt\_email.Text)

Fornecedores\_F.p\_cep = Trim(txt\_cep.Text)

If txt\_operacao.Text = "I" Then

dt = Fornecedores\_F.InserirFornecedores(Fornecedores\_F)

ElseIf txt\_operacao.Text = "U" Then

dt = Fornecedores\_F.AlterarFornecedores(Fornecedores\_F)

ElseIf txt\_operacao.Text = "D" Then

dt = Fornecedores\_F.ExcluirFornecedores(Fornecedores\_F)

End If

MsgBox("Registro Gravado com Sucesso!!", MsgBoxStyle.Information + MsgBoxStyle.Information, "Gravar Registro")

Catch ex As Exception

End Try

End Sub

Private Sub btn\_sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sair.Click

Me.Close()

End Sub

End Class

Public Class Frm\_funcionarios

Private Sub Frm\_funcionarios\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim dr As DataRow

Dim F\_Funcionarios As New Funcionarios

'FUNCIONARIOS

If Trim(txt\_operacao.Text) <> "I" And Trim(txt\_id\_funcionario.Text) <> "" Then

F\_Funcionarios.p\_id\_funcionario = Trim(txt\_id\_funcionario.Text)

dt = F\_Funcionarios.RetornarDadosFuncionarios(F\_Funcionarios)

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

txt\_id\_funcionario.Text = Trim(UCase(dr("ID\_FUNCIONARIO")))

txt\_id\_dados\_pessoais.Text = Trim(UCase(dr("ID\_DADOS\_PESSOAIS")))

If Not IsDBNull(dr("NOME")) Then

txt\_nome.Text = dr("NOME")

End If

If Not IsDBNull(dr("SEXO")) Then

cbx\_sexo.Text = dr("SEXO")

End If

If Not IsDBNull(dr("ESTADO\_CIVIL")) Then

cbx\_estado\_civil.Text = dr("ESTADO\_CIVIL")

End If

If Not IsDBNull(dr("RG")) Then

txt\_rg.Text = dr("RG")

End If

If Not IsDBNull(dr("CPF")) Then

txt\_cpf.Text = dr("CPF")

End If

If Not IsDBNull(dr("DATA\_NASCIMENTO")) Then

Date\_funcionario.Text = dr("DATA\_NASCIMENTO")

End If

If Not IsDBNull(dr("TELEFONE")) Then

txt\_telefone.Text = dr("TELEFONE")

End If

If Not IsDBNull(dr("CELULAR")) Then

txt\_celular.Text = dr("CELULAR")

End If

If Not IsDBNull(dr("E\_MAIL")) Then

txt\_email.Text = dr("E\_MAIL")

End If

If Not IsDBNull(dr("DATA\_ADMISSAO")) Then

Date\_admissao.Text = dr("DATA\_ADMISSAO")

End If

If Not IsDBNull(dr("CARGO")) Then

txt\_cargo.Text = dr("CARGO")

End If

If Not IsDBNull(dr("HORARIO\_TRABALHO")) Then

txt\_horario\_trabalho.Text = dr("HORARIO\_TRABALHO")

End If

If Not IsDBNull(dr("SALARIO")) Then

txt\_salario.Text = ToString.Format(dr("SALARIO"), "0.00")

End If

If Not IsDBNull(dr("LOGRADOURO")) Then

txt\_logradouro.Text = dr("LOGRADOURO")

End If

If Not IsDBNull(dr("NUMERO")) Then

txt\_numero.Text = dr("NUMERO")

End If

If Not IsDBNull(dr("COMPLEMENTO")) Then

txt\_complemento.Text = dr("COMPLEMENTO")

End If

If Not IsDBNull(dr("BAIRRO")) Then

txt\_bairro.Text = dr("BAIRRO")

End If

If Not IsDBNull(dr("CIDADE")) Then

txt\_cidade.Text = dr("CIDADE")

End If

If Not IsDBNull(dr("ESTADO")) Then

txt\_estado.Text = dr("ESTADO")

End If

If Not IsDBNull(dr("CEP")) Then

txt\_cep.Text = dr("CEP")

End If

Next

End If

dt = Nothing

End If

End Sub

Private Sub btn\_gravar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_gravar.Click

Dim sSql As String = ""

Dim dt As New DataTable

Dim dr As DataRow

Dim f\_Funcionarios As New Funcionarios

If Trim(txt\_nome.Text) = "" Then

MsgBox("Nome Inválido!! Nome tem que ser preenchido")

txt\_nome.Focus()

Exit Sub

End If

If cbx\_sexo.Text = "" Then

MsgBox("Sexo Inválido!!!")

cbx\_sexo.Focus()

Exit Sub

End If

If cbx\_estado\_civil.Text = "" Then

MsgBox("Estado Civil Inválido!!!")

cbx\_estado\_civil.Focus()

Exit Sub

End If

If txt\_rg.Text = "" Then

MsgBox("RG Inválido!!!")

txt\_rg.Focus()

Exit Sub

End If

If txt\_cpf.Text = "" Then

MsgBox("CPF Inválido!!!")

txt\_cpf.Focus()

Exit Sub

End If

If Date\_funcionario.Text = "" Then

MsgBox("Data Nascimento Inválido!!!")

Date\_funcionario.Focus()

Exit Sub

End If

If txt\_telefone.Text = "" Then

MsgBox("Telefone Inválido!!!")

txt\_telefone.Focus()

Exit Sub

End If

If txt\_celular.Text = "" Then

MsgBox("Celular Inválido!!!")

txt\_celular.Focus()

Exit Sub

End If

If Date\_admissao.Text = "" Then

MsgBox("Data Admissao Inválido!!!")

Date\_admissao.Focus()

Exit Sub

End If

If txt\_cargo.Text = "" Then

MsgBox("Cargo Inválido!!!")

txt\_cargo.Focus()

Exit Sub

End If

If txt\_horario\_trabalho.Text = "" Then

MsgBox("Horario de Trabalho Inválido!!!")

txt\_horario\_trabalho.Focus()

Exit Sub

End If

If Val(Trim(txt\_salario.Text)) = 0 Then

MsgBox("Salário Inválido!! Salário tem que ser preenchido")

txt\_salario.Focus()

Exit Sub

End If

If txt\_logradouro.Text = "" Then

MsgBox("LOGRADOURO Inválido!!!")

txt\_logradouro.Focus()

Exit Sub

End If

If txt\_numero.Text = "" Then

MsgBox("NUMERO Inválido!!!")

txt\_numero.Focus()

Exit Sub

End If

If txt\_cep.Text = "" Then

MsgBox("CEP Inválido!!!")

txt\_cep.Focus()

Exit Sub

End If

'Preenchendo Data de Nascimento

Dim AMD As String

AMD = CDate(Date\_funcionario.Text).Day & "/" & CDate(Date\_funcionario.Text).Month & "/" & CDate(Date\_funcionario.Text).Year

'Preenchendo Data de Admissão

Dim AMD\_Admissao As String

AMD\_Admissao = CDate(Date\_admissao.Text).Day & "/" & CDate(Date\_admissao.Text).Month & "/" & CDate(Date\_admissao.Text).Year

'Formato de preenchimento do Salário

Dim Vl\_Salario As Double

Vl\_Salario = Replace(txt\_salario.Text, ".", "")

Vl\_Salario = Replace(Vl\_Salario, ",", ".")

'Gravando os Dados Pessoais do funcionário

Dim mId\_Dados\_Pessoais As Integer

Try

Dim v\_Dados\_Pessoais As New Dados\_Pessoais

v\_Dados\_Pessoais.p\_id\_dados\_pessoais = Trim(txt\_id\_dados\_pessoais.Text)

v\_Dados\_Pessoais.p\_rg = Trim(txt\_rg.Text)

v\_Dados\_Pessoais.p\_cpf = Trim(txt\_cpf.Text)

v\_Dados\_Pessoais.p\_sexo = Trim(cbx\_sexo.Text)

v\_Dados\_Pessoais.p\_data\_nascimento = AMD

v\_Dados\_Pessoais.p\_telefone = Trim(txt\_telefone.Text)

v\_Dados\_Pessoais.p\_celular = Trim(txt\_celular.Text)

v\_Dados\_Pessoais.p\_e\_mail = Trim(txt\_email.Text)

If txt\_operacao.Text = "I" Then

dt = v\_Dados\_Pessoais.InserirDP(v\_Dados\_Pessoais)

ElseIf txt\_operacao.Text = "U" Then

dt = v\_Dados\_Pessoais.AlterarDP(v\_Dados\_Pessoais)

ElseIf txt\_operacao.Text = "D" Then

dt = v\_Dados\_Pessoais.ExcluirDP(v\_Dados\_Pessoais)

End If

dr = Nothing

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

mId\_Dados\_Pessoais = dr("ID\_DADOS\_PESSOAIS")

Next

End If

'Gravando o Endereço caso não exista.

Dim v\_Enderecos As New Enderecos

v\_Enderecos.p\_cep = Trim(txt\_cep.Text)

txt\_cep.Text = CType(txt\_cep.Text, Decimal).ToString("0####-###")

dt = v\_Enderecos.RetornarDadosEnderecos(v\_Enderecos)

v\_Enderecos.p\_cep = Trim(txt\_cep.Text)

v\_Enderecos.p\_logradouro = Trim(txt\_logradouro.Text)

v\_Enderecos.p\_numero = Trim(txt\_numero.Text)

v\_Enderecos.p\_complemento = Trim(txt\_complemento.Text)

v\_Enderecos.p\_bairro = Trim(txt\_bairro.Text)

v\_Enderecos.p\_cidade = Trim(txt\_cidade.Text)

v\_Enderecos.p\_estado = Trim(txt\_estado.Text)

If txt\_operacao.Text = "I" Then

dt = v\_Enderecos.InserirEnderecos(v\_Enderecos)

ElseIf txt\_operacao.Text = "U" Then

dt = v\_Enderecos.AlterarEnderecos(v\_Enderecos)

ElseIf txt\_operacao.Text = "D" Then

dt = v\_Enderecos.ExcluirEnderecos(v\_Enderecos)

End If

dr = Nothing

'Gravando Dados do Funcionario caso não exista

f\_Funcionarios.p\_id\_funcionario = Trim(txt\_id\_funcionario.Text)

f\_Funcionarios.p\_nome = Trim(txt\_nome.Text)

f\_Funcionarios.p\_horario\_trabalho = Trim(txt\_horario\_trabalho.Text)

f\_Funcionarios.p\_data\_admissao = AMD\_Admissao

f\_Funcionarios.p\_cargo = Trim(txt\_cargo.Text)

f\_Funcionarios.p\_salario = Vl\_Salario

f\_Funcionarios.p\_estado\_civil = Trim(cbx\_estado\_civil.Text)

f\_Funcionarios.p\_id\_dados\_pessoais = mId\_Dados\_Pessoais

f\_Funcionarios.p\_cep = Trim(txt\_cep.Text)

If txt\_operacao.Text = "I" Then

dt = f\_Funcionarios.InserirFuncionarios(f\_Funcionarios)

ElseIf txt\_operacao.Text = "U" Then

dt = f\_Funcionarios.AlterarFuncionarios(f\_Funcionarios)

ElseIf txt\_operacao.Text = "D" Then

dt = f\_Funcionarios.ExcluirFuncionarios(f\_Funcionarios)

End If

MsgBox("Registro Gravado com Sucesso!!", MsgBoxStyle.Information + MsgBoxStyle.Information, "Gravar Registro")

dr = Nothing

Catch ex As Exception

End Try

End Sub

Private Sub btn\_sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sair.Click

Me.Close()

End Sub

End Class

Public Class Frm\_Midias

Private Sub Frm\_Midias\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim dr As DataRow

Dim M\_Midias As New Midias

If Trim(txt\_operacao.Text) <> "I" And Trim(txt\_id\_midia.Text) <> "" Then

M\_Midias.p\_id\_midia = Trim(txt\_id\_midia.Text)

dt = M\_Midias.RetornarDadosMidias(M\_Midias)

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

txt\_id\_midia.Text = Trim(UCase(dr("ID\_MIDIA")))

If Not IsDBNull(dr("MIDIA")) Then

txt\_midia.Text = dr("MIDIA")

End If

Next

End If

dt = Nothing

End If

End Sub

Private Sub btn\_gravar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_gravar.Click

Dim sSql As String = ""

Dim dt As New DataTable

Dim dr As DataRow

Dim M\_Midias As New Midias

If Trim(txt\_midia.Text) = "" Then

MsgBox("Midia Inválida!! Midia tem que ser preenchida")

txt\_midia.Focus()

Exit Sub

End If

Try

'Gravando A MIDIA caso não exista

M\_Midias.p\_id\_midia = Trim(txt\_id\_midia.Text)

M\_Midias.p\_midia = Trim(txt\_midia.Text)

If txt\_operacao.Text = "I" Then

dt = M\_Midias.InserirMidias(M\_Midias)

ElseIf txt\_operacao.Text = "U" Then

dt = M\_Midias.AlterarMidias(M\_Midias)

ElseIf txt\_operacao.Text = "D" Then

dt = M\_Midias.ExcluirMidias(M\_Midias)

End If

MsgBox("Registro Gravado com Sucesso!!", MsgBoxStyle.Information + MsgBoxStyle.Information, "Gravar Registro")

dr = Nothing

Catch ex As Exception

End Try

End Sub

Private Sub txt\_sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txt\_sair.Click

Me.Close()

End Sub

End Class

Public Class Frm\_locacao

Private Sub Frm\_locacao\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim dr As DataRow

Dim L\_cliente As New Locacao\_cliente

Dim L\_Filme As New Locacao\_filme

'Locação Cliente

dt = L\_cliente.LstGridlocacao\_cliente(L\_cliente)

If dt.Rows.Count > 0 Then

With Grid\_locacao\_cliente

Me.Grid\_locacao\_cliente.Rows.Clear()

Me.Grid\_locacao\_cliente.DataSource = dt

Me.Grid\_locacao\_cliente.AutoResizeColumns()

End With

End If

'Locação Filme

dt = L\_Filme.LstGridlocacao\_filme(L\_Filme)

If dt.Rows.Count > 0 Then

With grid\_locacao\_filme

Me.grid\_locacao\_filme.Rows.Clear()

Me.grid\_locacao\_filme.DataSource = dt

Me.grid\_locacao\_filme.AutoResizeColumns()

End With

End If

'LOCAÇÃO

If Trim(txt\_operacao.Text) <> "I" And Trim(txt\_id\_locacao.Text) <> "" Then

L\_cliente.p\_id\_locacao = Trim(txt\_id\_locacao.Text)

L\_Filme.p\_id\_locacao = Trim(txt\_id\_locacao.Text)

dt = L\_cliente.RetornarDadosLocacao\_cliente(L\_cliente)

dt = L\_Filme.RetornarDadosLocacao\_filme(L\_Filme)

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

txt\_id\_locacao.Text = Trim(UCase(dr("ID\_LOCACAO")))

txt\_id\_filme.Text = Trim(UCase(dr("ID\_FILME")))

If Not IsDBNull(dr("NOME")) Then

txt\_nome.Text = dr("NOME")

End If

If Not IsDBNull(dr("DATA\_RETIRADA")) Then

Date\_retirada.Text = Trim(dr("DATA\_RETIRADA"))

End If

If Not IsDBNull(dr("DATA\_DEVOLUCAO")) Then

Date\_devolucao.Text = dr("DATA\_DEVOLUCAO")

End If

If Not IsDBNull(dr("PRECO")) Then

txt\_preco.Text = dr("PRECO")

End If

If Not IsDBNull(dr("LOCACAO\_PAGA\_ATO")) Then

cbx\_locacao\_paga\_ato.Text = dr("LOCACAO\_PAGA\_ATO")

End If

If Not IsDBNull(dr("VALOR\_LOCACAO")) Then

txt\_valor\_locacao.Text = dr("VALOR\_LOCACAO")

End If

Next

End If

dt = Nothing

End If

End Sub

Private Sub btn\_pesquisar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_pesquisar.Click

Dim dt As New DataTable

Dim L\_cliente As New Locacao\_cliente

Dim locacao As New Frm\_locacao

If Trim(txt\_nome.Text) <> "" Then

L\_cliente.p\_nome = Trim(txt\_nome.Text)

End If

dt = L\_cliente.LstGridlocacao\_cliente(L\_cliente)

If dt.Rows.Count > 0 Then

locacao.Grid\_locacao\_cliente.Rows.Clear()

With Grid\_locacao\_cliente

.DataSource = dt

.AutoResizeColumns()

End With

End If

dt = Nothing

End Sub

Private Sub btn\_gravar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_gravar.Click

Dim sSql As String = ""

Dim dt As New DataTable

Dim dr As DataRow

Dim L\_cliente As New Locacao\_cliente

Dim L\_filme As New Locacao\_filme

If Grid\_locacao.Rows.Count = 0 Then

MsgBox("Filme Inválido!!!")

txt\_filme.Focus()

Exit Sub

End If

If cbx\_locacao\_paga\_ato.Text = "" Then

MsgBox("Precisa ser preenchido se locação foi paga no ato ou não!!!")

cbx\_locacao\_paga\_ato.Focus()

Exit Sub

End If

If txt\_valor\_locacao.Text = "" Then

MsgBox("Valor total da Locação precisa ser preenchido!!!")

txt\_valor\_locacao.Focus()

Exit Sub

End If

'Preenchendo Datas de retirada e devolução da locação

Dim AMD As String

AMD = CDate(Date\_retirada.Text).Day & "/" & CDate(Date\_retirada.Text).Month & "/" & CDate(Date\_retirada.Text).Year

AMD = CDate(Date\_devolucao.Text).Day & "/" & CDate(Date\_devolucao.Text).Month & "/" & CDate(Date\_devolucao.Text).Year

'Gravando os Dados do cliente

Try

L\_cliente.p\_id\_locacao = txt\_id\_locacao.Text

L\_cliente.p\_cod\_cliente = Trim(Grid\_locacao\_cliente.CurrentRow.Cells(1).Value)

L\_cliente.p\_tipo\_cliente = Trim(Grid\_locacao\_cliente.CurrentRow.Cells(0).Value)

L\_cliente.p\_locacao\_paga\_ato = Trim(cbx\_locacao\_paga\_ato.Text)

L\_cliente.p\_valor\_locacao = Trim(txt\_valor\_locacao.Text)

If txt\_operacao.Text = "I" Then

dt = L\_cliente.Inserirlc(L\_cliente)

ElseIf txt\_operacao.Text = "U" Then

dt = L\_cliente.Alterarlc(L\_cliente)

ElseIf txt\_operacao.Text = "D" Then

dt = L\_cliente.Excluirlc(L\_cliente)

End If

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

txt\_id\_locacao.Text = dr("ID\_LOCACAO")

Next

End If

dr = Nothing

Catch ex As Exception

End Try

'Gravando os Dados de locacao do filme

Try

Dim vCont As Integer = 0

With Grid\_locacao

If .RowCount > 0 Then

Do While Not .Rows(vCont).Cells(0).Value Is Nothing

L\_filme.p\_id\_locacao = txt\_id\_locacao.Text

L\_filme.p\_cod\_filme = .Rows(vCont).Cells(0).Value

L\_filme.p\_data\_retirada = .Rows(vCont).Cells(2).Value

L\_filme.p\_data\_devolucao = .Rows(vCont).Cells(3).Value

L\_filme.p\_preco = .Rows(vCont).Cells(4).Value

dt = L\_filme.Inserirlf(L\_filme)

vCont += 1

Loop

End If

End With

MsgBox("Registro Gravado com Sucesso!!", MsgBoxStyle.Information + MsgBoxStyle.Information, "Gravar Registro")

Catch ex As Exception

End Try

End Sub

Private Sub btn\_sair\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sair.Click

Me.Close()

End Sub

Private Sub Btn\_Excluir\_Filme\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

If MsgBox("Deseja realmente excluir o Filme selecionado para locação", MsgBoxStyle.Question + MsgBoxStyle.YesNo, "Excluir Filme") = MsgBoxResult.Yes Then

Dim dt As New DataTable

Dim l\_filme As New Locacao\_filme

With Grid\_locacao\_filme

l\_filme.p\_cod\_filme = .CurrentRow.Cells(0).Value

End With

dt = l\_filme.Excluirlf(l\_filme)

MessageBox.Show("Registro excluido com Sucesso!!")

End If

End Sub

Private Sub btn\_pesquisar\_filme\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_pesquisar\_filme.Click

Dim dt As New DataTable

Dim l\_filme As New Locacao\_filme

Dim locacao As New Frm\_locacao

If Trim(txt\_id\_filme.Text) <> "" Then

l\_filme.p\_cod\_filme = Trim(txt\_id\_filme.Text)

End If

dt = l\_filme.LstGridlocacao\_filme(l\_filme)

If dt.Rows.Count > 0 Then

locacao.Grid\_locacao\_filme.Rows.Clear()

With Grid\_locacao\_filme

.DataSource = dt

.AutoResizeColumns()

End With

End If

dt = Nothing

End Sub

Private Sub Btn\_Inserir\_Filme\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn\_Inserir\_Filme.Click

If Trim(Date\_retirada.Text) = "" Then

MsgBox("data de retirada Inválido!!!")

Date\_retirada.Focus()

Exit Sub

End If

If Trim(Date\_devolucao.Text) = "" Then

MsgBox("data de devolucao Inválido!!!")

Date\_devolucao.Focus()

Exit Sub

End If

With Grid\_locacao

.Rows.Add(Grid\_locacao\_filme.CurrentRow.Cells(0).Value, \_

Trim(UCase(Grid\_locacao\_filme.CurrentRow.Cells(1).Value)), \_

Trim(UCase(Date\_retirada.Text)), \_

Trim(UCase(Date\_devolucao.Text)), \_

Trim(Grid\_locacao\_filme.CurrentRow.Cells(2).Value))

End With

Dim mTotalLoc As Double

mTotalLoc = CType(txt\_valor\_locacao.Text, Decimal) + CType(Grid\_locacao\_filme.CurrentRow.Cells(2).Value, Decimal)

txt\_valor\_locacao.Text = mTotalLoc.ToString("#,##0.00")

End Sub

End Class

Public Class Frm\_reserva

Private Sub reservas\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dt As New DataTable

Dim dr As DataRow

Dim r\_cliente As New Reserva\_cliente

Dim r\_Filme As New Reserva\_Filme

'Locação Cliente

dt = r\_cliente.LstGridreserva\_cliente(r\_cliente)

If dt.Rows.Count > 0 Then

With Grid\_reserva\_cliente

Me.Grid\_reserva\_cliente.Rows.Clear()

Me.Grid\_reserva\_cliente.DataSource = dt

Me.Grid\_reserva\_cliente.AutoResizeColumns()

End With

End If

'Locação Filme

dt = r\_Filme.LstGridreserva\_filme(r\_Filme)

If dt.Rows.Count > 0 Then

With Grid\_reserva\_filme

Me.Grid\_reserva\_filme.Rows.Clear()

Me.Grid\_reserva\_filme.DataSource = dt

Me.Grid\_reserva\_filme.AutoResizeColumns()

End With

End If

'reserva

If Trim(txt\_operacao.Text) <> "I" And Trim(txt\_id\_reserva.Text) <> "" Then

r\_cliente.p\_id\_reserva = Trim(txt\_id\_reserva.Text)

r\_Filme.p\_id\_reserva = Trim(txt\_id\_reserva.Text)

dt = r\_cliente.RetornarDadosreserva\_cliente(r\_cliente)

dt = r\_Filme.RetornarDadosreserva\_filme(r\_Filme)

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

txt\_id\_reserva.Text = Trim(UCase(dr("ID\_RESERVA")))

txt\_id\_filme.Text = Trim(UCase(dr("ID\_FILME")))

If Not IsDBNull(dr("NOME")) Then

txt\_nome.Text = dr("NOME")

End If

If Not IsDBNull(dr("DATA\_RESERVA")) Then

Date\_reserva.Text = dr("DATA\_RESERVA")

End If

If Not IsDBNull(dr("PRECO")) Then

txt\_preco.Text = dr("PRECO")

End If

Next

End If

dt = Nothing

End If

End Sub

Private Sub btn\_gravar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Dim sSql As String = ""

Dim dt As New DataTable

Dim dr As DataRow

Dim r\_cliente As New Reserva\_cliente

Dim r\_filme As New Reserva\_Filme

If Grid\_reserva.Rows.Count = 0 Then

MsgBox("Filme Inválido!!!")

txt\_id\_filme.Focus()

Exit Sub

End If

If txt\_valor\_reserva.Text = "" Then

MsgBox("Valor total da Locação precisa ser preenchido!!!")

txt\_valor\_reserva.Focus()

Exit Sub

End If

If Date\_reserva.Text = "" Then

MsgBox("Data Inválida!!!")

Date\_reserva.Focus()

End If

'Preenchendo Datas de retirada e devolução da locação

Dim AMD As String

AMD = CDate(Date\_reserva.Text).Day & "/" & CDate(Date\_reserva.Text).Month & "/" & CDate(Date\_reserva.Text).Year

'Gravando os Dados do cliente

Try

r\_cliente.p\_id\_reserva = txt\_id\_reserva.Text

r\_cliente.p\_cod\_cliente = Trim(Grid\_reserva\_cliente.CurrentRow.Cells(0).Value)

r\_cliente.p\_nome = Trim(Grid\_reserva\_cliente.CurrentRow.Cells(2).Value)

If txt\_operacao.Text = "I" Then

dt = r\_cliente.Inserirrc(r\_cliente)

ElseIf txt\_operacao.Text = "U" Then

dt = r\_cliente.Alterarrc(r\_cliente)

ElseIf txt\_operacao.Text = "D" Then

dt = r\_cliente.Excluirrc(r\_cliente)

End If

If dt.Rows.Count > 0 Then

For Each dr In dt.Rows

txt\_id\_reserva.Text = dr("ID\_RESERVA")

Next

End If

dr = Nothing

Catch ex As Exception

End Try

'Gravando os Dados de reserva do filme

Try

Dim vCont As Integer = 0

With Grid\_reserva

If .RowCount > 0 Then

Do While Not .Rows(vCont).Cells(0).Value Is Nothing

r\_filme.p\_id\_reserva = txt\_id\_reserva.Text

r\_filme.p\_cod\_filme = .Rows(vCont).Cells(0).Value

r\_filme.p\_filme = .Rows(vCont).Cells(1).Value

r\_filme.p\_data\_reserva = .Rows(vCont).Cells(2).Value

r\_filme.p\_preco = .Rows(vCont).Cells(3).Value

dt = r\_filme.Inserirrf(r\_filme)

vCont += 1

Loop

End If

End With

MsgBox("Registro Gravado com Sucesso!!", MsgBoxStyle.Information + MsgBoxStyle.Information, "Gravar Registro")

Catch ex As Exception

End Try

End Sub

Private Sub Btn\_Excluir\_Filme\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

If MsgBox("Deseja realmente excluir o Filme selecionado para reserva", MsgBoxStyle.Question + MsgBoxStyle.YesNo, "Excluir Filme") = MsgBoxResult.Yes Then

Dim dt As New DataTable

Dim r\_filme As New Reserva\_Filme

With Grid\_reserva\_filme

r\_filme.p\_cod\_filme = .CurrentRow.Cells(0).Value

End With

dt = r\_filme.Excluirrf(r\_filme)

MessageBox.Show("Registro excluido com Sucesso!!")

End If

End Sub

Private Sub Btn\_Inserir\_Filme\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn\_Inserir\_Filme.Click

If Trim(Date\_reserva.Text) = "" Then

MsgBox("data de reserva Inválido!!!")

Date\_reserva.Focus()

Exit Sub

End If

With Grid\_reserva

.Rows.Add(Grid\_reserva\_filme.CurrentRow.Cells(0).Value, \_

Trim(UCase(Grid\_reserva\_filme.CurrentRow.Cells(1).Value)), \_

Trim(UCase(Date\_reserva.Text)), \_

Trim(Grid\_reserva\_filme.CurrentRow.Cells(2).Value))

End With

Dim mTotalReser As Double

mTotalReser = CType(txt\_valor\_reserva.Text, Decimal) + CType(Grid\_reserva\_filme.CurrentRow.Cells(2).Value, Decimal)

txt\_valor\_reserva.Text = mTotalReser.ToString("#,##0.00")

End Sub

Private Sub btn\_pesquisar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_pesquisar.Click

Dim dt As New DataTable

Dim r\_cliente As New Reserva\_cliente

Dim reserva As New Frm\_reserva

If Trim(txt\_nome.Text) <> "" Then

r\_cliente.p\_nome = Trim(txt\_nome.Text)

End If

dt = r\_cliente.LstGridreserva\_cliente(r\_cliente)

If dt.Rows.Count > 0 Then

reserva.Grid\_reserva\_cliente.Rows.Clear()

With Grid\_reserva\_cliente

.DataSource = dt

.AutoResizeColumns()

End With

End If

dt = Nothing

End Sub

Private Sub btn\_pesquisar\_filme\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_pesquisar\_filme.Click

Dim dt As New DataTable

Dim r\_filme As New Reserva\_Filme

Dim reserva As New Frm\_reserva

If Trim(txt\_id\_filme.Text) <> "" Then

r\_filme.p\_cod\_filme = Trim(txt\_id\_filme.Text)

End If

dt = r\_filme.LstGridreserva\_filme(r\_filme)

If dt.Rows.Count > 0 Then

reserva.Grid\_reserva\_filme.Rows.Clear()

With Grid\_reserva\_filme

.DataSource = dt

.AutoResizeColumns()

End With

End If

dt = Nothing

End Sub

End Class

Public Class Login

' TODO: Insert code to perform custom authentication using the provided username and password

' (See http://go.microsoft.com/fwlink/?LinkId=35339).

' The custom principal can then be attached to the current thread's principal as follows:

' My.User.CurrentPrincipal = CustomPrincipal

' where CustomPrincipal is the IPrincipal implementation used to perform authentication.

' Subsequently, My.User will return identity information encapsulated in the CustomPrincipal object

' such as the username, display name, etc.

Private Sub OK\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK.Click

Dim V\_Usuario As New Usuario

Dim vDataSet As New DataTable

Dim strSql As String = ""

'monstagem da string para execução da query

V\_Usuario.p\_Nome = Trim(UsernameTextBox.Text)

vDataSet = V\_Usuario.ListagemDataTableUsuario(V\_Usuario)

If vDataSet.HasErrors.ToString() = "False" Then

'Compara a senha digitada com a senha da Base

If Trim(vDataSet.Rows(0).Item("SENHA").ToString()) = Trim(PasswordTextBox.Text) Then

' MDI\_Projeto.ClientesToolStripMenuItem.Enabled = False

MDIProjeto.Enabled = True

Else

MsgBox("Senha Inválida!!!!")

PasswordTextBox.Focus()

Exit Sub

End If

Else

MsgBox("Usuário Inválido!!!!")

UsernameTextBox.Focus()

Exit Sub

End If

Me.Close()

End Sub

Private Sub Cancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cancel.Click

End

End Sub

Private Sub Login\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

End Class

Public Class MDIProjeto

Private Sub ShowNewForm(ByVal sender As Object, ByVal e As EventArgs) Handles NewWindowToolStripMenuItem.Click

' Create a new instance of the child form.

Dim ChildForm As New System.Windows.Forms.Form

' Make it a child of this MDI form before showing it.

ChildForm.MdiParent = Me

m\_ChildFormNumber += 1

ChildForm.Text = "Window " & m\_ChildFormNumber

ChildForm.Show()

End Sub

Private Sub OpenFile(ByVal sender As Object, ByVal e As EventArgs)

Dim OpenFileDialog As New OpenFileDialog

OpenFileDialog.InitialDirectory = My.Computer.FileSystem.SpecialDirectories.MyDocuments

OpenFileDialog.Filter = "Text Files (\*.txt)|\*.txt|All Files (\*.\*)|\*.\*"

If (OpenFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then

Dim FileName As String = OpenFileDialog.FileName

' TODO: Add code here to open the file.

End If

End Sub

Private Sub SaveAsToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

Dim SaveFileDialog As New SaveFileDialog

SaveFileDialog.InitialDirectory = My.Computer.FileSystem.SpecialDirectories.MyDocuments

SaveFileDialog.Filter = "Text Files (\*.txt)|\*.txt|All Files (\*.\*)|\*.\*"

If (SaveFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then

Dim FileName As String = SaveFileDialog.FileName

' TODO: Add code here to save the current contents of the form to a file.

End If

End Sub

Private Sub ExitToolsStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

Global.System.Windows.Forms.Application.Exit()

End Sub

Private Sub CascadeToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles CascadeToolStripMenuItem.Click

Me.LayoutMdi(MdiLayout.Cascade)

End Sub

Private Sub TileVerticleToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles TileVerticalToolStripMenuItem.Click

Me.LayoutMdi(MdiLayout.TileVertical)

End Sub

Private Sub TileHorizontalToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles TileHorizontalToolStripMenuItem.Click

Me.LayoutMdi(MdiLayout.TileHorizontal)

End Sub

Private Sub ArrangeIconsToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles ArrangeIconsToolStripMenuItem.Click

Me.LayoutMdi(MdiLayout.ArrangeIcons)

End Sub

Private Sub CloseAllToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles CloseAllToolStripMenuItem.Click

' Close all child forms of the parent.

For Each ChildForm As Form In Me.MdiChildren

ChildForm.Close()

Next

End Sub

Private m\_ChildFormNumber As Integer = 0

Private Sub ClientesToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ClientesToolStripMenuItem.Click

Dim NewMDIChild As New Clientes

'Set the parent form of the child window.

NewMDIChild.MdiParent = Me

'Display the new form.

NewMDIChild.Show()

'clientes.Show

End Sub

Private Sub FilmesEMídiasToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FilmesEMídiasToolStripMenuItem.Click

Dim NewMDIChild As New filmes\_midias

'Set the parent form of the child window.

NewMDIChild.MdiParent = Me

'Display the new form.

NewMDIChild.Show()

'filmes\_mídias.Show

End Sub

Private Sub FornecedoresToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FornecedoresToolStripMenuItem.Click

Dim NewMDIChild As New Fornecedores

'Set the parent form of the child window.

NewMDIChild.MdiParent = Me

'Display the new form.

NewMDIChild.Show()

'fornecedores.Show

End Sub

Private Sub FuncionáriosToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FuncionáriosToolStripMenuItem.Click

Dim NewMDIChild As New f\_funcionarios

'Set the parent form of the child window.

NewMDIChild.MdiParent = Me

'Display the new form.

NewMDIChild.Show()

'funcionários.Show

End Sub

Private Sub LocaçõesToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LocaçõesToolStripMenuItem.Click

Dim NewMDIChild As New Frm\_locacao

'Set the parent form of the child window.

NewMDIChild.MdiParent = Me

NewMDIChild.txt\_id\_locacao.Text = "0"

NewMDIChild.txt\_operacao.Text = "I"

'Display the new form.

NewMDIChild.Show()

'locações.Show

End Sub

Private Sub ReservaToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ReservaToolStripMenuItem.Click

Dim NewMDIChild As New Frm\_reserva

'Set the parent form of the child window.

NewMDIChild.MdiParent = Me

NewMDIChild.txt\_id\_reserva.Text = "0"

NewMDIChild.txt\_operacao.Text = "I"

'Display the new form.

NewMDIChild.Show()

'reserva.Show

End Sub

'Private Sub MDI\_Projeto\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

' ' Definição do tempo de espera da conexao

' 'db.ConnectionTimeout = 100

' ' Abertuda do Banco de Dados

' 'db.Open("Driver={SQL Server};Server=(local);Uid=;Pwd=;Database=Turma13;Trusted\_Connection=yes")

' ' db.Open("Provider=sqloledb;Data Source=(local);Initial Catalog=TESTE\_13;Integrated Security=SSPI;Trusted\_Connection=yes")

Me.Enabled = False

' Login.ShowDialog()

'End Sub

Private Sub MDIProjeto\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

Private Sub ConsultasToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ConsultasToolStripMenuItem.Click

End Sub

Private Sub LocaçãoToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LocaçãoToolStripMenuItem.Click

Dim NewMDIChild As New Frm\_Consulta\_locacao

'Set the parent form of the child window.

NewMDIChild.MdiParent = Me

'Display the new form.

NewMDIChild.Show()

'reserva.Show

End Sub

Private Sub ReservaToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ReservaToolStripMenuItem1.Click

Dim NewMDIChild As New Frm\_Consultas\_reservas

'Set the parent form of the child window.

NewMDIChild.MdiParent = Me

'Display the new form.

NewMDIChild.Show()

'reserva.Show

End Sub

End Class

**DICIONÁRIO DE DADOS**

|  |  |
| --- | --- |
| **Cliente Adicional** | |
| ID Adicional | Códido do cliente adicional |
| ID titular | Código do cliente titular |
| Nome | Nome do cliente adicional |
| Cep | Cep do cliente adicional |
| ID Dados Pessoais | Código dos dados pessoais |
| Grau parentesco | Parentesco do adicional e titular |

|  |  |
| --- | --- |
| **Cliente Titular** | |
| ID Titular | Códido do cliente titular |
| Nome | Nome do cliente adicional |
| Cep | Cep do cliente adicional |
| ID Dados Pessoais | Código dos dados pessoais |

|  |  |
| --- | --- |
| **Dados Pessoais** | |
| ID Dados Pessoais | Código dos dados pessoais |
| RG | Preenchimento do RG |
| CPF | Preenchimento do CPF |
| Sexo | Preenchimento do |
| Date Nascimento | Preenchimento do Data de Nascimento |
| Telefone | Preenchimento do Telefone |
| Celular | Preenchimento do Celular |
| E-mail | Preenchimento do E-mail |

|  |  |
| --- | --- |
| **Endereço** | |
| Cep | Preenchimento do Cep |
| Logradouro | Preenchimento do logradouro |
| Número | Preenchimento do número |
| Complemento | Preenchimento do complemento |
| Bairro | Preenchimento do bairro |
| Cidade | Preenchimento da cidade |
| Estado | Preenchimento do estado |

|  |  |
| --- | --- |
| **Filmes** | |
| ID Filme | Código do filme |
| ID Mídia | Código da mídia |
| Filme | Preenchimento do filme |
| Ano Lançamento | Preenchimento do ano de lançamento |
| Gênero | Preenchimento do gênero |
| Nacionalidade | Preenchimento da nacionalidade |
| Duração | Preenchimento doa duração |
| Direção | Preenchimento da direção do filme |
| Sinopse | Preenchimento da sinopse do filme |
| Preço | Preenchimento do preço |

|  |  |
| --- | --- |
| **Fornecedor** | |
| ID Fornecedor | Código do fornecedor |
| Razão Social | Razão social da empresa |
| Nome Fantasia | Nome fantasia da empresa |
| CNPJ | CNPJ da empresa |
| Contato | Contato da empresa |
| Telefone | Telefone da empresa |
| Fax | Fax da empresa |
| E-mail | E-mail de contato da empresa |
| Cep | Preenchimento do Cep |

|  |  |
| --- | --- |
| **Funcionário** | |
| ID Funcionario | Código do funcionário |
| Nome | Nome do funcionário |
| Horário Trabalho | Horário de trabalho do funcionário |
| Data Admissão | Date de admissão do funcionário |
| Cargo | Cargo do funcionário |
| Cep | Preenchimento do Cep |
| ID Dados Pessoais | Código dos dados pessoais |
| Salário | Salário do funcionário |
| Estado Civil | Estado Civil do funcionário |

|  |  |
| --- | --- |
| **Locação** | |
| ID Locacao | Código da locação |
| Cod Cliente | Código do cliente |
| Nome | Nome do cliente |
| Tipo Cliente | Tipo de cliente |
| Cod Filme | Código do filme |
| Filme | Nome do filme |
| Data Retirada | Data de retirada do filme |
| Data Devolucao | Data de devolução do filme |
| Preco | Preço de locação do filme |
| Valor Locacao | Valor total da locação |
| Locacao Paga Ato | Locação paga no ato ou não |

|  |  |
| --- | --- |
| **Mídias** | |
| ID Mídia | Código da mídia |
| Mídia | Tipo de mídia |

|  |  |
| --- | --- |
| **Reserva** | |
| ID Reserva | Código da reserva |
| Cod Cliente | Código do cliente |
| Nome | Nome do cliente |
| Tipo Cliente | Tipo de cliente |
| Cod Filme | Código do filme |
| Filme | Nome do filme |
| Data Reserva | Data da reserva do filme |
| Preco | Preço de locação do filme |

**CONCLUSÃO**

Este projeto foi muito bem elaborado para mostrar como funciona o desenvolvimento de um software e o relacionamento do mesmo com um banco de dados. Agora esta empresa tem um meio prático e ágil para fazer cadastros, alterações, remoções e consultas de dados, e ter um controle de informações mais seguro com a implementação do banco de dados integrado ao sistema.

Com este projeto, conseguimos compreender o que se é necessário para o planejamento, desenvolvimento e finalização de um software, e pudemos pôr em prática muito do conhecimento que nos foi abrangido ao decorrer deste curso.

Contudo, pudemos adquirir mais conhecimento em diversos assuntos e ter mais afinidade com a área de TI.

**REFERÊNCIAS BIBLIOGRÁFICAS**

* [www.google.com.br](http://www.google.com.br)
* Livro: SQL Server 2005 Express Edition – Interativo Guia básico

Por: Jóse Augusto N. G. Manzano – Editora Érica

* Livro: Programando com Microsoft Visual Basic 2005: A Linguagem

Por: Francesco Balena – Editora Bookman