CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS – CAMPUS V

MS Commerce

Fernando Gomes
Lucas Souza Sales
Marcos Vinícius Moreira
Rafael Fernandes F. Quadros

Divinópolis - MG 2014

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS – CAMPUS V

MS Commerce

Fernando Gomes

Lucas Souza Sales

Marcos Vinícius Moreira

Rafael Fernandes F. Quadros

Orientador: Vinicius de Oliveira Santos

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Informática para Internet do Centro Federal de Educação Tecnológica de Minas Gerais – Campus V como requisito parcial para a obtenção do título de Técnico em Informática.

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS – CAMPUS V

Trabalho de Conclusão de Curso julgado adequado para obtenção do título de
Técnico em Informática para Internet e aprovado pela banca composta pelos
seguintes professores.
Prof. Eduardo Habib Bechelane Maia - CEFET-MG
Prof. Luis Augusto Mattos Mendes - CEFET-MG
Prof. Vinicius de Oliveira Santos - CEFET-MG
Prof. Luis Augusto Mattos Mendes
Coordenador do Curso Técnico em Informática

Data de aprovação: Divinópolis, 14 de janeiro de 2015.

SUMÁRIO

1	Introdução			
	1.1	Definição da Empresa	1	
	1.2	Definição do Escopo	2	
	1.3	Definição das Funcionalidades		
		Referencial Teórico		
	1.4	Referencial Teorico		
2	Projeto c	onceitual	4	
	2.1	Diagrama de Contexto UML	4	
	2.2	Documentação dos Atores	5	
	2.2.1	Ator 1 – Administrador		
	2.2.2	Ator 2 – Faturamento.		
	2.2.3	Ator 3 – Financeiro		
	2.3	Descrição Detalhada das Funcionalidades		
	2.3.1	Cadastros		
	2.3.1.1	Empresa		
	2.3.1.1	-		
	2.3.1.3	,		
	2.3.1.4			
	2.3.1.5			
	2.3.1.6 2.3.1.7	\mathcal{E}		
	2.3.1.7			
	2.3.1.6	Vendas, devolução de vendas e compras		
	2.3.2.1			
	2.3.2.1			
		Financeiro		
		Manutenção de contas a pagar ou receber		
	2.3.3.2			
		Relatórios		
	2.3.4.1	Clientes	13	
	2.3.4.2			
	2.3.4.3	Categorias	14	
	2.3.4.4	Fornecedores	14	
	2.3.4.5	Bairros		
	2.3.4.6			
	2.3.4.7	r . 8		
	2.3.4.8			
	2.3.4.9			
	2.3.4.10 2.3.4.1	1 1		

	2.3.4.12	Contas a receber	18
	2.3.4.13	Contas a pagar	18
	2.3.4.14	Movimento de caixa	19
	2.3.4.15	Backup	
	2.3.4.16	ServerConfig	20
3	Projeto Fís	sico	21
	3.1 D	DER – Diagrama de Entidade e Relacionamento	21
	3.2 D	Dicionário de Dados	23
	3.3 D	Diagrama de Sequência	25
	3.4 D	Diagrama de Classes	27
	3.5 D	Diagrama de estados	47
	3.6 D	Diagrama de implantação	51
	3.7 D	Diagrama de componentes	52
4	Cronogran	na	54
5	Conclusão		55
6	Referência	ıs	57
7	Anexo		58
	7.1 D	DER	58
	7.2 D	Diagrama de Classe	60

1 Introdução

Atualmente, no âmbito empresarial, os Sistemas de Informação se mostram um grande diferencial devido a sua grande eficiência em fazer a gestão das informações da empresa. Para Dias (1975, p. 5) "Sistema de Informação é qualquer sistema usado para prover informação (incluindo o seu processamento), qualquer que seja o uso feito dessa informação".

Segundo Laudon (1999, p. 26) "a razão mais forte pelas quais as empresas constroem os sistemas, então, é para resolver problemas organizacionais e para reagir a uma mudança no ambiente".

No Brasil, em agosto de 2012 o governo brasileiro lançou o Programa TI Maior para estimular desenvolvimento de software, com a promessa de investir R\$500 milhões até 2015. O programa tem como meta desenvolver a tecnologia da informação no país, aumentando a produção de software.

Pensando nesse crescente mercado de sistemas computacionais, com o intuito de atender a atual demanda das microempresas do centro-oeste mineiro, automatizando processos que antes eram feitos de formas manuais e visando garantir uma gestão mais eficiente das microempresas, é que se propõe o sistema MS Commerce.

O MS Commerce irá gerenciar informações de fornecedores, clientes, produtos, pedidos e também controlar estoque, contas a pagar e receber e fluxo de caixa. O desenvolvimento do MS Commerce está voltado para plataforma Windows, sendo um sistema *Desktop*. O software permitirá estabelecer os perfis de administrador, faturamento e financeiro.

Este trabalho conta com, primeiramente, as especificações do sistema e suas funcionalidades, seguido do projeto físico e a diagramação necessária para documentação do sistema. Por fim, pode ser encontrada a conclusão do trabalho, seguido das referências utilizadas como base para definições de diagramas e um anexo para enriquecimento do conteúdo do documento, onde podem ser encontradas versões ampliadas de alguns diagramas.

1.1 Definição da Empresa

Microempresas de atividade varejista ainda não informatizadas, onde um funcionário pode executar várias atividades por não existir, na maioria das vezes, divisão em

departamentos. O foco principal se concentra nas vendas locais (venda varejista), onde ocorre toda transação comercial.

1.2 Definição do Escopo

O sistema será inicialmente para plataforma *Desktop*, onde haverá um usuário administrador que possuirá total acesso ao sistema, podendo cadastrar outros usuários.

Projetado para atuar no cenário de microempresas do ramo do varejo, o MS Commerce dispõe de: formulários cadastrais da empresa, usuários, fornecedores, produtos, clientes, bairro, cidades, natureza de operação, forma de pagamento, vendedores, categoria, usuário caixa, operação caixa. O MS Commerce também conta com diversos tipos de relatórios pré-definidos como: cadastros, movimentações de estoque e financeiros.

As vantagens de se utilizar o MS Commerce incluem o maior controle da empresa, facilitando o acesso e a manipulação das informações, tornando a instituição mais organizada e garantindo uma maior segurança dos dados.

1.3 Definição das Funcionalidades

- 1. A manutenção de cadastros tem finalidade de inserção, alteração, exclusão e atualização de conteúdo referente a cadastros, e conta com as seguintes opções: empresa, clientes, bairros, cidades, fornecedores, produtos, categorias, usuários, naturezas de operação, formas de pagamentos, operações de caixa.
- 2. A funcionalidade de movimento proporciona a inserção, alteração, exclusão e atualização do que se refere a movimentação de mercadorias dentro do estoque, como por exemplo: venda de mercadorias, aquisição, devoluções e ajustes (conforme permissão de acesso).
- 3. A funcionalidade de financeiro permite ao usuário gerenciar o que se refere a entrada e saída de finanças na empresa, o que inclui as opções de: Fluxo de Caixa, Manutenção de Contas a Pagar e Contas a Receber.
- 4. A funcionalidade de relatórios proporciona ao usuário uma visão dos seguintes dados da empresa, através da emissão das informações em um arquivo de formato *pdf*: Clientes,

produtos, categorias, fornecedores, bairros, cidades, forma de pagamento, vendedores, usuários, vendas por período, compras por período, contas a receber, contas a pagar e movimento de caixa.

1.4 Referencial Teórico

O presente sistema será desenvolvido com base no modelo incremental como modelo de desenvolvimento, por se mostrar mais viável para um desenvolvimento onde há uma documentação mais consistente e garantia de um produto de qualidade superior.

Segundo Pressman (2010, p. 40) "o modelo incremental aplica sequências lineares de uma forma racional à medida que o tempo passa. Cada sequência linear produz incrementos do software possíveis de serem entregues."

Para o desenvolvimento do sistema propriamente dito, há a necessidade de traçar metas e planos. Após ser feito um bom planejamento, crucial para ser concebido um produto de qualidade, é possível começar o processo de modelagem de diagramas e estruturas do software.

Para facilitar a gerência do projeto, será construída uma Estrutura Analítica do Projeto (EAP). Segundo o PMBOK (2004, p. 112):

EAP é uma decomposição hierárquica orientada à entrega do trabalho a ser executado pela equipe do projeto, para atingir os objetivos do projeto e criar as entregas necessárias. A EAP subdivide o trabalho do projeto em partes menores e mais facilmente gerenciáveis, em que cada nível descendente da EAP representa uma definição cada vez mais detalhada do trabalho do projeto. É possível agendar, estimar custos, monitorar e controlar o trabalho planejado contido nos componentes de nível mais baixo da EAP, denominados pacotes de trabalho.

Esta estrutura irá possibilitar que seja feita uma estimativa do custo total do sistema, assim como sua montagem e cronograma. Para a construção da EAP, será utilizado o software Microsoft Office Project 2007.

Já para a modelagem do Diagrama de Entidade e Relacionamento (DER), que será utilizado na construção do banco de dados, serão utilizados os softwares brModelo versão 2.0 e MySQLWorkbench 6.1.

2 Projeto conceitual

Nessa seção serão apresentados os documentos conceituais, a fim de oferecer uma melhor visão do contexto do software MS Commerce.

2.1 Diagrama de Contexto UML

Segue, na Figura 1, o diagrama de caso de uso do sistema.

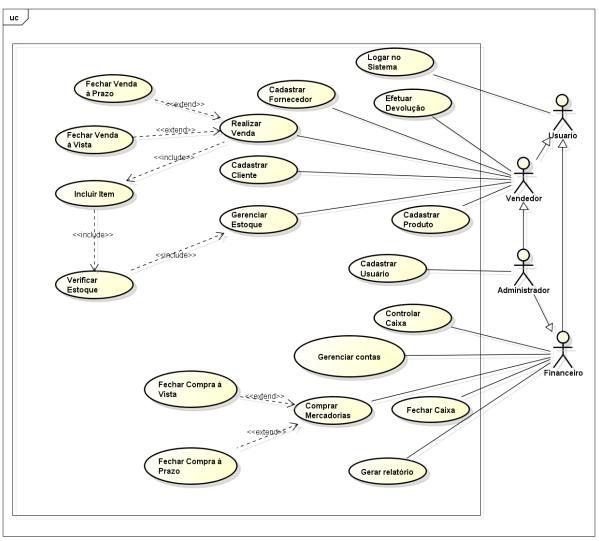


FIGURA 1: Diagrama de Caso de Uso (DCU)

powered by Astah

2.2 Documentação dos Atores

Nesta seção serão definidos os atores do sistema. Os atores são os diferentes tipos de usuários que podem ser utilizados para realizar login no MS Commerce.

2.2.1 Ator 1 – Administrador

Representa o usuário do sistema que tem como característica o controle total, podendo executar qualquer funcionalidade oferecida pelo sistema.

2.2.2 Ator 2 – Faturamento

Representa o usuário que atua na parte da saída de produtos do sistema, tendo acesso restrito a cadastro de usuários, a entrada de produtos no sistema e a emissão de relatórios.

2.2.3 Ator 3 – Financeiro

O usuário financeiro tem acesso à compra de mercadoria, gerenciamento de contas, controle de caixa, geração de relatórios e fechamento de caixa.

2.3 Descrição Detalhada das Funcionalidades

Nesta seção serão descritas, de forma detalhada, todas as funcionalidades do sistema, que consistem em: cadastro de tabelas no banco de dados, gerenciamento de entrada e saída de produtos no estoque (compra e venda), gerenciamento de contas a pagar e a receber e geração de relatórios.

2.3.1 Cadastros

Nesta seção serão descritos os tipos de cadastros disponíveis no sistema, os quais interagem com o usuário através de uma interface gráfica amigável, possibilitando ao usuário inserir dados nas caixas de texto e salvá-los no banco de dados. A data de inclusão e alteração no banco de dados é inserida de forma automática, assim como o código do que está sendo inserido.

2.3.1.1 Empresa

O cadastro de empresa no sistema, na Figura 2, possui uma interface que possibilita inserção de dados de forma que também possa ser utilizado o CPF do proprietário da empresa,

caso a empresa ainda não possua um CNPJ próprio, além de possibilitar a inserção da inscrição estadual.

Há também a possibilidade de selecionar o bairro e cidade através do código dos mesmos ou por meio de uma caixa de seleção, auxiliando na inserção de dados consistentes no banco de dados, evitando variações destes dados por parte do usuário.

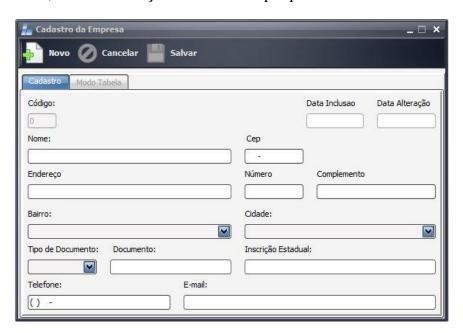


FIGURA 2: Tela para cadastro de empresa

2.3.1.2 **Usuário**

A Figura 3 representa o cadastro de usuário. O cadastro de usuário permite que sejam registrados novos usuários no sistema, possibilitando a inserção de usuário do tipo administrador, faturamento ou financeiro.

O usuário administrador tem acesso global as informações do sistema, possuindo todas as funções de outros tipos de usuários.

O usuário faturamento tem acesso restrito ao sistema, sendo vedado o cadastro a alteração de usuários, além de não ter acesso à geração de relatórios e as contas a receber da empresa.

Já o usuário financeiro tem acesso as rotinas ligadas ao financeiro como contas a pagar, contas a receber, fluxo de caixa e relatórios financeiros.

Poderão ser notados, nos cadastros seguintes, alguns botões desabilitados por conta do acesso ser negado ao usuário dependendo de qual função do cadastro o usuário estiver usando.

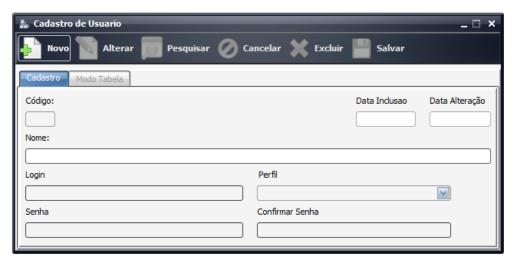


FIGURA 3: Tela para cadastro de usuário

2.3.1.3 Cliente, Fornecedor ou Vendedor

Cliente da empresa, fornecedor de produtos e vendedor da empresa possuem uma interface semelhante a inserida abaixo, diferenciando apenas o local onde é salvo os dados inseridos e, portando, estão sendo representado pela mesma imagem, a Figura 4. Além disto, possuem o mesmo tipo de seleção de bairro e cidade presente no cadastro de empresa, contando também com o cadastro de CNPJ/CPF, porém há o adicional de um campo situação para informar se o cliente, fornecedor ou vendedor está ou não ativo na empresa.

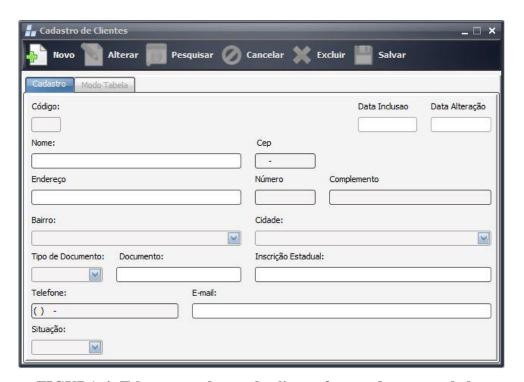


FIGURA 4: Tela para cadastro de cliente, fornecedor ou vendedor

2.3.1.4 Produto

O cadastro de produto, representado na Figura 5, proporciona para o usuário a opção de inserir a descrição do produto a ser cadastrado, a forma do produto (balde, caixa, dúzia, quilo, entre outras unidades), caixa de seleção para fornecedor do produto e grupo, marca do produto e inserção de preço para o produto. Também possui informações como preço médio, saldo atual e Inventário que são gerados de acordo com o movimento dos produtos.

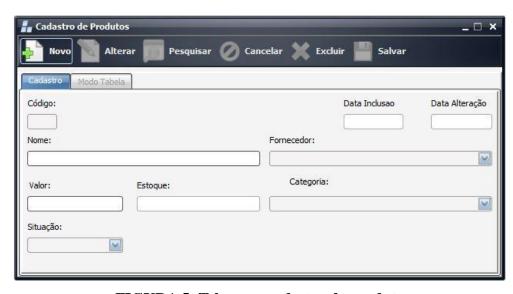


FIGURA 5: Tela para cadastro de produto

2.3.1.5 Bairro

O cadastro de bairros apenas pode ser utilizado se houver um registro de cidade já inserido no banco de dados e, então, é registrado o bairro correspondente a cidade selecionada no campo de seleção, como é mostrado na Figura 6.

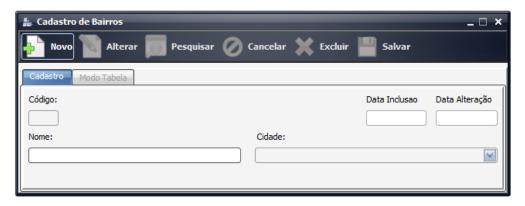


FIGURA 6: Tela para cadastro de bairro

2.3.1.6 Categoria

O cadastro de categorias, apresentado na Figura 7, tem a finalidade de inserir categorias novas, que irão agrupar futuros produtos inseridos no sistema.

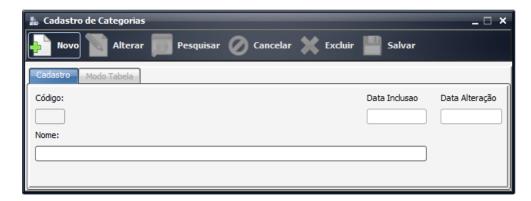


FIGURA 7: Tela para cadastro de categorias

2.3.1.7 Cidade

O cadastro de cidade, na Figura 8, funciona de forma semelhante ao cadastro de categorias, porém é utilizado para agrupar futuros bairros que serão inseridos no sistema.

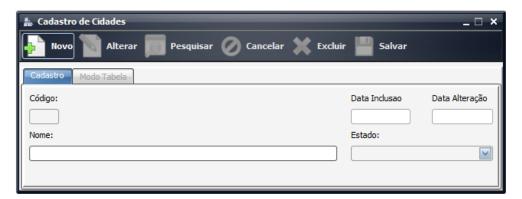


FIGURA 8: Tela para cadastro de cidades

2.3.1.8 Forma de pagamento

Com o objetivo de facilitar o manuseio do sistema por parte do usuário, poderá ser feito antecipadamente o cadastro de formas de pagamento para que haja uma rápida utilização no momento da realização da venda do produto. No campo de parcelas, será inserido quantas parcelas poderá ser dividido o pagamento do produto e, em dias, quantos dias será o tempo do parcelamento, como pode ser visto na Figura 9.

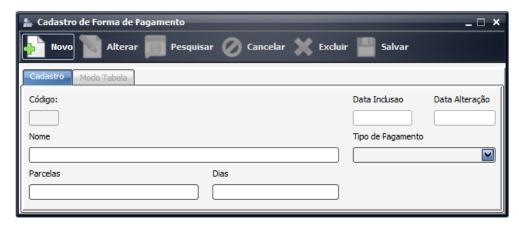


FIGURA 9: Tela para cadastro de forma de pagamento

2.3.2 Vendas, devolução de vendas e compras

Nesta seção serão tratados os cadastros de vendas (saídas), devolução de vendas, e compras (entradas) de mercadorias com a mesma tela, devido sua semelhança.

2.3.2.1 Venda ou entrada de mercadoria

Para realizar a venda ou entrada de mercadoria, é disponibilizado um formulário semelhante para as duas operações. Portanto, será representado apenas um formulário que, conforme sua natureza (venda ou compra) irá executar suas devidas funções. O formulário de venda de mercadoria pode ser visto na Figura 10, onde podem ser inseridos dados como cliente, vendedor, forma de pagamento e os produtos sequencialmente em uma lista, assim como podem ser excluídos, se necessário. O modo de grade é utilizado para realização de pesquisas através da aplicação de filtros. Neste formulário também e permitido realizar a emissão de orçamento referente às vendas pesquisadas. Caso o usuário necessite também é possível realizar a exclusão das vendas selecionadas. Já no formulário de entradas de mercadoria, é permitida a entrada de dados como número de nota, fornecedores, forma de pagamento e produtos. Estes formulários de Entradas e Vendas de mercadoria também provem integração com contas a receber ou contas pagar e fluxo de caixa, estas integrações depende da forma de pagamento informada na venda e entrada de mercadoria.

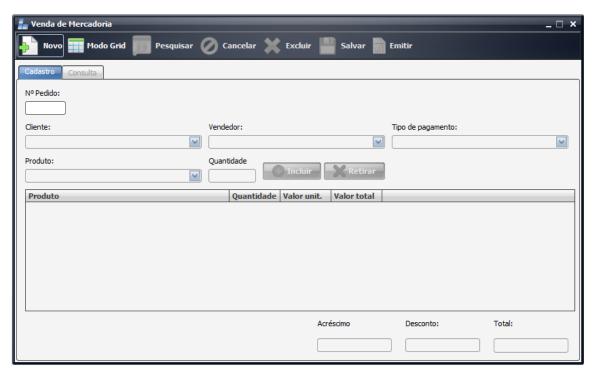


FIGURA 10: Tela para realizar venda de mercadoria e entrada

2.3.2.2 Devolução de Venda

O Devolução de Vendas, na Figura 11, será usado para devolver vendas que ainda estejam com título no contas a receber em situação de aberto.

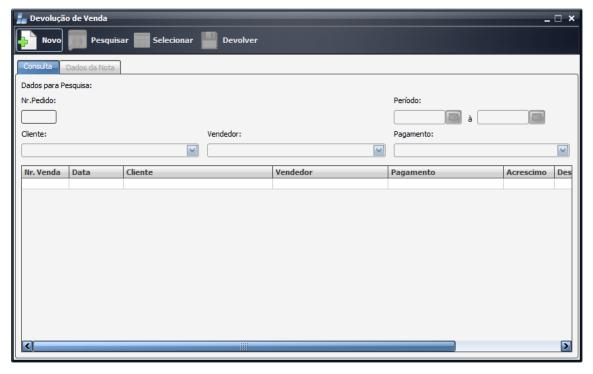


FIGURA 11: Tela para realizar devoluções de Vendas

2.3.3 Financeiro

A seção Financeiro do MS Commerce objetiva trabalhar com inserção ou busca de dados sobre finanças da empresa no banco de dados e posterior alteração, atualização ou exclusão, conforme necessidade do usuário.

2.3.3.1 Manutenção de contas a pagar ou receber

A manutenção de contas visto na Figura 12 é semelhante, mudando apenas a condição de entrada ou saída no sistema. A manutenção de contas a pagar ou a receber, que pode ser selecionado na aba manutenção de documentos, na tela principal do sistema, objetiva fazer alteração em um registro já salvo no banco. As informações de cadastro também podem ser vistas em modo tabela, facilitando a pesquisa.



FIGURA 12: Tela para manutenção de cadastro de contas a pagar ou receber

2.3.3.2 Quitação de contas a pagar ou receber

A quitação de contas a pagar (ou contas a receber), que pode ser visualizado na Figura 13 e acessado na aba de quitação de documentos, objetiva informar como pago o registro de conta, lembrando que cada título poderá ser quitado total ou parcialmente. Para o registro ser pesquisado, podem ser utilizados filtros de pesquisa. Poderão existir títulos nas situações de Aberto(títulos ainda não quitados), Parcial/Aberto(títulos com valor quitado menor que o valor total do título) e Quitado(título com valor total quitado).

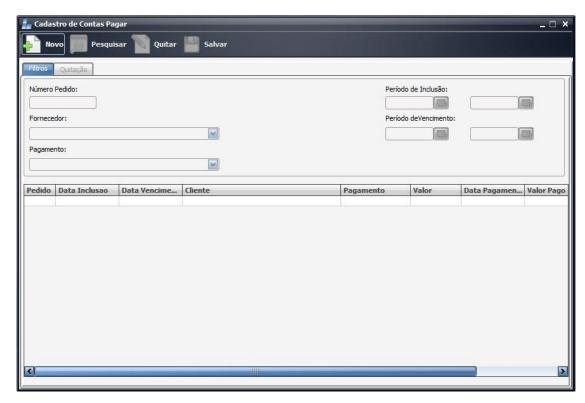


FIGURA 13: Tela para quitação de cadastro de contas a pagar ou receber

2.3.4 Relatórios

Os relatórios emitem os dados resultantes de uma consulta no banco de dados conforme o desejo do usuário, utilizando-se de filtros para refinar a emissão.

2.3.4.1 Clientes

A missão de relatório de clientes é feita através de uma tela como a Figura 14, onde os dados para emissão podem ser filtrados em cliente, cidade, bairro, tipo de documento e situação.

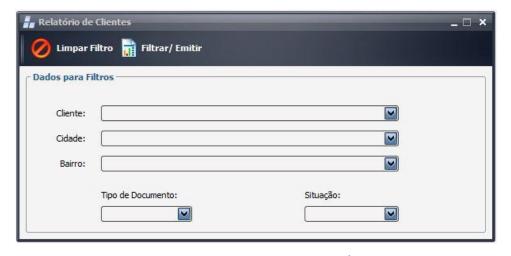


FIGURA 14: Tela para emissão de relatório de clientes

2.3.4.2 Produtos

A tela representada na Figura 15 permite a emissão de relatório de produtos, de forma que sejam selecionadas anteriormente as opções de filtros.



FIGURA 15: Tela para emissão de relatório de produtos

2.3.4.3 Categorias

A tela na Figura 16 permite a emissão de relatório de categorias, de forma que seja selecionada a categoria desejada, antes da emissão.



FIGURA 16: Tela para emissão de relatório de categorias

2.3.4.4 Fornecedores

A tela na Figura 17 permite a emissão de relatório de fornecedores, de forma que sejam selecionadas anteriormente as opções de filtros.



FIGURA 17: Tela para emissão de relatório de fornecedores

2.3.4.5 Bairros

A tela na Figura 18 permite a emissão de relatório de bairros, de forma que sejam selecionadas anteriormente as opções de filtros.



FIGURA 18: Tela para emissão de relatório de bairros

2.3.4.6 Cidades

A tela na Figura 19 permite a emissão de relatório de cidades, de forma que sejam selecionadas anteriormente as opções de filtros.



FIGURA 19: Tela para emissão de relatório de cidades

2.3.4.7 Formas de pagamento

A tela na Figura 20 permite a emissão de relatório de formas de pagamento, de forma que sejam selecionadas anteriormente as opções de filtros.



FIGURA 20: Tela para emissão de relatório de forma de pagamento

2.3.4.8 Vendedores

A tela na Figura 21 permite a emissão de relatório de vendedores, de forma que sejam selecionadas anteriormente as opções de filtros.



FIGURA 21: Tela para emissão de relatório de vendedores

2.3.4.9 Usuários

A tela na Figura 22 permite a emissão de relatório de usuários, de forma que sejam selecionadas anteriormente as opções de filtros.

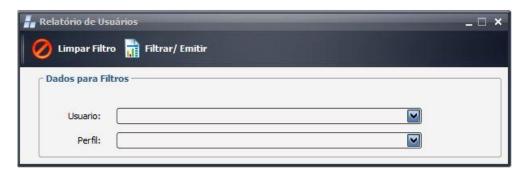


FIGURA 22: Tela para emissão de relatório de usuários

2.3.4.10 Vendas por período

A tela na Figura 23 permite a emissão de relatório de vendas, de forma que sejam selecionadas anteriormente as opções de filtros. O relatório de vendas disponibiliza a opção de filtrar por período de vendas, assim como situação da venda, vendedor e cliente, o que facilita a utilização do sistema pelo usuário.

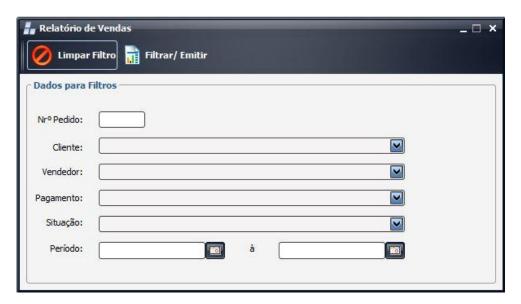


FIGURA 23: Tela para emissão de relatório de vendas por período

2.3.4.11 Compras por período

A tela na Figura 24 permite a emissão de relatório de entradas, de forma que sejam selecionadas as opções de filtro, que incluem período da entrada, situação da entrada, fornecedor, pagamento, entre outros.



FIGURA 24: Tela para emissão de relatório de entradas

2.3.4.12 Contas a receber

A tela na Figura 25 permite a emissão de relatório de contas a receber, podendo utilizar filtragem por três tipos de período: data de inclusão, data de vencimento e/ou data de pagamento, além dos demais filtros.



FIGURA 25: Tela para emissão de relatório de contas a receber

2.3.4.13 Contas a pagar

A tela na Figura 26 referente a emissão de relatório de contas a pagar permite, assim como no relatório de contas a receber, os mesmos tipos de período para filtragem. O relatório

de contas a pagar é semelhante ao de contas a receber, porém com dados dos produtos que entraram no estoque.



FIGURA 26: Tela para emissão de relatório de contas a pagar

2.3.4.14 Movimento de caixa

O relatório de caixa, que pode ser emitido através da tela representada na Figura 27, permite, além da escolha do número da nota e origem, o tipo de movimento, situação da translação e período em que ela ocorreu.



FIGURA 27: Tela para emissão de relatório de caixa

2.3.4.15 Backup

O MS Commerce conta com opção para realização de backup, de forma que salva uma cópia do banco de dado em formado ".sql", utilizando um utilitário MySQL Server chamado MySQL Dump.

A cópia do banco de dados é salva em um caminho pré-definido, dentro do diretório do sistema MS Commerce. O backup pode ser resgatado e, posteriormente, inserido no gerenciador de banco de dados para utilização no sistema.

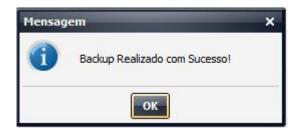


FIGURA 28: Tela de confirmação de Backup realizado

2.3.4.16 ServerConfig

O ServerConfig é um diferencial do MS Commerce, que proporciona maior facilidade para instalação e configuração do sistema na máquina do cliente. O ServerConfig gera três arquivos em formado texto com as informações inseridas nos campos disponibilizados. Após gerados, estes arquivos texto são utilizados para encontrar o caminho e realizar a conexão com o banco de dados que será utilizado pelo MS Commerce.

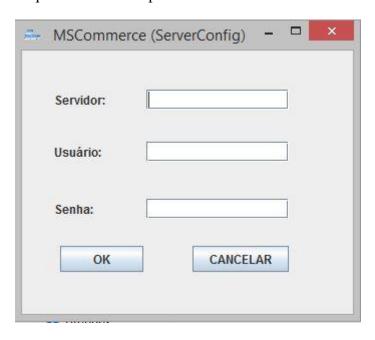


FIGURA 29: Tela para inserção de informações para conexão com o banco de dados

3 Projeto Físico

No projeto físico, é retratada a implementação do banco de dados por meio de diagramas que norteiam os relacionamentos, atributos e classes do banco.

Os relacionamentos tratam da conexão entre tabelas do banco, representando a troca de informação entre estas tabelas. Já os atributos são propriedades que definem uma entidade. Portanto, os atributos são a composição de uma classe.

3.1 DER – Diagrama de Entidade e Relacionamento

O Diagrama de Entidade e Relacionamento tem a finalidade de descrever o banco de dados em si, seus relacionamentos, tabelas, atributos e chaves primárias. O Diagrama de Entidade e Relacionamento está representado na Figura 30. Uma versão ampliada pode ser encontrada no anexo 7.1 da documentação.

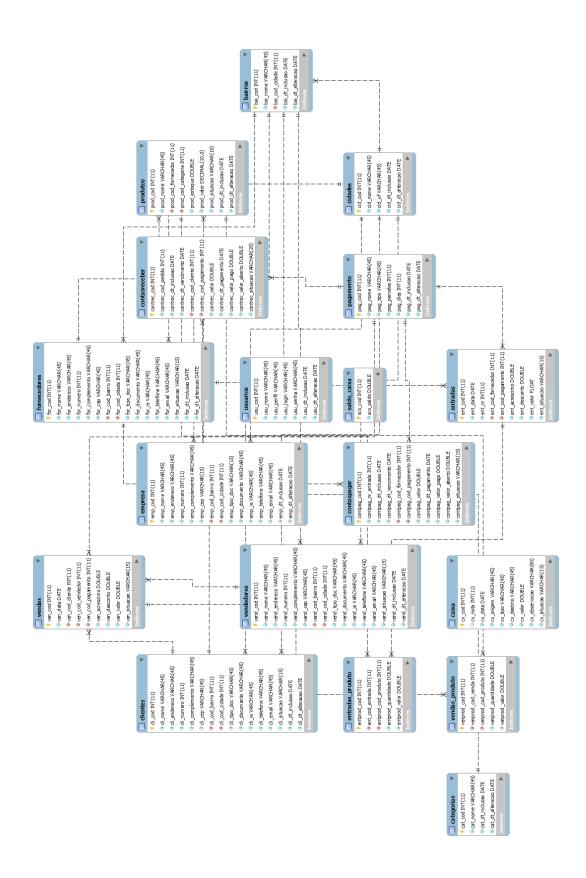


FIGURA 30: Diagrama de Entidade e Relacionamento (DER)

3.2 Dicionário de Dados

O dicionário de dados objetiva presentar o agrupamento de tabelas, habilitadas para leitura ou consulta, dentro de um banco de dados, de forma escrita.

Bairros = *cadastro de bairros* <u>bai_cod_+</u> bai_nome + bai_cod_cidade + bai_dt_inclusao + bai_dt_alteracao

Caixa = *movimento do caixa* <u>cx_cod</u> + cx_nota + cx_data + cx_origem + cx_tipo + cx_destino + cx_valor + cx_observação + cx_situação

Categorias = *cadastro de categorias* <u>cat_cod</u> + cat_nome + cat_dt_inclusao + cat_dt_alteracao

Cidades = *cadastro de cidades* <u>cid_cod</u> + cid_nome + cid_uf + cid_dt_inclusao + cid_dt_alteracao

Clientes = *cadastro de cliente* <u>cli_cod</u> + cli_nome + cli_endereco + cli_numero + cli_complemento + cli_cep + cli_cod_bairro + cli_cod_cidade + cli_tipo_doc + cli_documento + cli_ie + cli_cep + cli_telefone + cli_email + cli_situacao + cli_dt_inclusao + cli_dt_alteracao

Contaspagar = *contas a serem pagas* <u>contpag_cod</u> + contpag_nr_entrada + contpag_dt_inclusao + contpag_dt_vencimento + contpag_cod_fornecedor + contpag_cod_pagamento + contpag_valor + contpag_dt_pagamento + contpag_valor_pago + contpag_valor_aberto + contpag_situacao

Contasreceber = *contas a serem recebidas* <u>contrec_cod</u> + contrec_cod_pedido + contrec_dt_inclusao + contrec_dt_vencimento + contrec_cod_cliente + contrec_cod_pagamento + contrec_valor + contrec_dt_pagamento + contrec_valor_pago + contrec_valor_aberto + contrec_situação

Fornecedores = *cadastro de fornecedores* <u>for cod</u> + for_nome + for_endereco + for_cod_bairro + for_cod_cidade + for_cep + for_tipo_doc + for_documento + for_ie + for_telefone + for_email + for_dt_inclusao + for_dt_alteracao

Empresa = *cadastro de empresa* <u>emp_cod</u> + emp_nome + emp_endereco + emp_numero + emp_complemento + emp_cep + emp_cod_bairro + emp_cod_cidade + emp_tipo_doc + emp_documento + emp_ie + emp_telefone + emp_email

Entradas = *dados da entrada* <u>ent_cod</u> + ent_data + ent_nr + ent_cod_fornecedor + ent_cod_pagamento + ent_acrescimo + ent_desconto + ent_valor + ent_situação

Entradas_produto = *produtos da entrada* <u>entprod_cod</u> + ent_cod_entrada + entprod_cod_produto + entprod_quantidade + entprod_valor

Fornecedores = *cadastro de fornecedores* <u>for_cod</u> + for_nome + for_endereco + for_numero + for_complemento + for_cep + for_cod_bairro + for_cod_cidade + for_tipo_doc

+ for_documento + for_ie, for_telefone + for_email + for_situacao + for_dt_inclusao + for_dt_alteracao

Pagamento = *cadastro de tipo de pagamento* <u>pag_cod</u> + pag_nome + pag_tipo + pag_parcelas + pag_dias + pag_dt_inclusao + pag_dt_alteracao

Produtos = *cadastro de produtos* <u>prod_cod</u> + prod_nome + prod_cod_fornecedor, prod_cod_categoria + prod_estoque + prod_valor + prod_situacao + prod_dt_inclusao + prod_dt_alteracao

Saldo_caixa = *cadastro de saldo no caixa* <u>scx_cod</u> + scx_saldo

Usuarios = *cadastro de usuário* <u>usu_cod</u> + usu_nome + usu_perfil + usu_login + usu_senha + usu_dt_inclusao + usu_dt_alteracao

Vendas = *dados da venda* <u>ven_cod</u> + ven_data + ven_cod_cliente + ven_cod_vendedor + ven_cod_pagamento + ven_acrescimo + ven_desconto + ven_valor + ven_situação

Vendas_produto = *produtos da venda* <u>venprod_cod</u> + venprod_cod_venda + venprod_cod_produto + venprod_quantidade + venprod_valor

Vendedores = *cadastro de vendedores* <u>ven_cod</u> + ven_nome + ven_endereco + ven_numero + ven_complemento + ven_cep + ven_cod_bairro + ven_cod_cidade + ven_tipo_doc + ven_documento + ven_ie + ven_telefone + ven_email + ven_situacao + ven_dt_inclusao + ven_dt_alteracao

3.3 Diagrama de Sequência

O diagrama de sequência representa a sequência de processos de um software enquanto está sendo executado. Segue, na Figura 31 e Figura 32, o Diagrama de Sequência do MS Commerce, dividido em duas partes para melhor visualização.

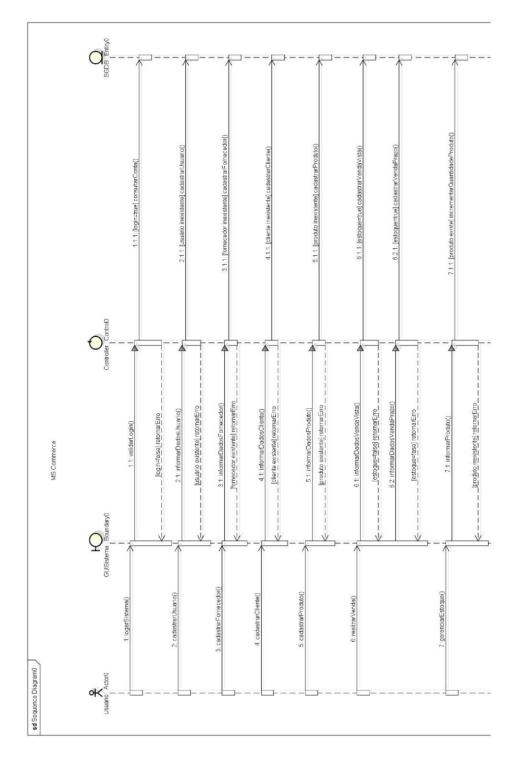


FIGURA 31: Diagrama de Sequência (Parte 1)

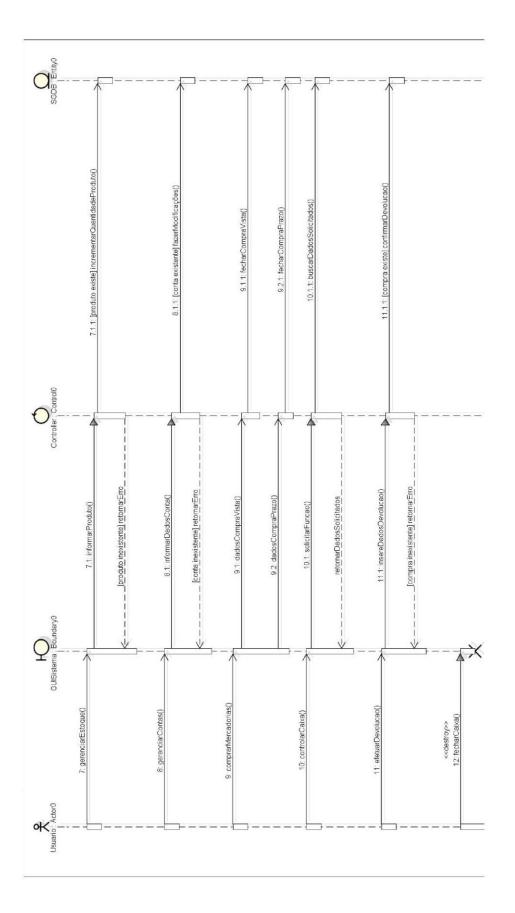


FIGURA 32: Diagrama de Sequência (Parte 2)

3.4 Diagrama de Classes

O diagrama de classe que se segue objetiva descrever a estrutura e relações das classes no sistema, sendo útil no desenvolvimento, pois define as classes que o sistema possuirá. Abaixo segue partes do diagrama, ilegível devido a grande quantidade de informações. Em anexo, será feita uma versão legível do diagrama.

Devido a dimensão do Diagrama de Classes do MS Commerce, segue abaixo o nome das classes com suas respectivas associações e, no anexo 7.2, pode-se encontrar as classes individuais e completas.

Classe

DAOEntradas

Associações

Conexão

ControllerEntradas

Classe

Conexao

Associações

ServerConfig

DAOCidade

DAOSaldoCaixa

DAOFluxoCaixa

DAOConta

DAOVendedor

DAOCliente

DAOVendas

DAOContasReceber

DAOCategoria

DAOContasPagar

DAOPagamento

DAOBairro

DAOProdutos

DAOFornecedor

DAOUsuario

DAOEntradas

Classe

DAOCidade

Associações

ControllerCidade

Conexao

Classe

DAOFluxoCaixa

Associações

ControllerFluxoCaixa

Conexao

Classe

DAOCliente

Associações

ControllerCliente

Conexao

Classe

DAOContasReceber

Associações

ControllerContasReceber

Conexao

Classe

DAOContasPagar

Associações

ControllerContasPagar

Conexao

Classe

DAOPagamento

Associações

ControllerPagamento

Conexao

Classe

DAOProdutos

Associações

ControllerProdutos

Conexao

Classe

DAOUsuario

Associações

ControllerUsuario

Conexao

Classe

DAOFornecedor

Associações

ControllerFornecedor

Conexao

Classe

DAOBairro

Associações

ControllerBairro

Conexao

Classe

DAOCategoria

Associações

ControllerCategoria

Conexao

Classe

DAOVendas

Associações

ControllerVendas

Conexao

Classe

DAOVendedor

Associações

ControllerVendedor

Conexao

Classe

DAOConta

Associações

ControllerConta

Conexao

Classe

DAOSaldoCaixa

Associações

ControllerSaldoCaixa

Conexao

Classe

ControllerVendas

Associações

DAOVendas

ViewCliente

ViewPagamento

ViewVendedor

ViewDevolucaoVenda

ViewVenda

ViewEmpresa

ViewRelatorioVendas

ViewProduto

ViewContasReceber

ViewFluxoCaixa

DAOVendas

DAORelatorios

Classe

ControllerUsuario

Associações

DAOUsuario

ViewRelatorioUsuario

ViewUsuario

DAORelatorios

Classe

ControllerCliente

Associações

DAOFluxoCaixa

ViewRelatorioCliente

ViewRelatorioContasReceber

ViewBairro

ViewContasReceber

ViewQuitacaoContasReceber

ViewEmpresa

DAORelatorios

ViewCliente

ViewDevolucaoVenda

DAOCliente

ViewRelatorioVenda

ViewVenda

ViewCidade

Classe

ControllerConta

Associações

DAOConta

DAORelatorios

Classe

ControllerVendedor

Associações

DAOVendedor

ViewCidade

ViewRelatorioVendedor

ViewVenda

ViewVendedor

DAORelatorios

ViewBairro

ViewDevolucaoVenda

ViewRelatorioVendas

Classe

ControllerCategoria

Associações

ViewRelatorioCategoria

DAOCategoria

DAORelatorios

ViewProduto

ViewRelatorioProduto

ViewCategoria

Classe

ControllerSaldoCaixa

Associações

DAORelatorios

ViewFluxoCaixa

DAOSaldoCaixa

Classe

ControllerContasPagar

Associações

ViewEntrada

DAOContasPagar

ViewQuitacaoContasPagar

ViewPagamento

DAORelatorios

ViewRelatorioContasPagar

ViewContasPagar

ViewFornecedor

Classe

ControllerPagamento

Associações

ViewQuitacaoContasReceber

ViewEntrada

ViewRelatorioContasReceber

ViewQuitacaoContasPagar

ViewContasReceber

DAORelatorios

ViewDevolucaoVenda

ViewVenda

ViewRelatorioVendas

ViewContasPagar

ViewRelatorioContasReceber

ViewPagamento

ViewRelatorioEntradas

ViewRelatorioPagamento

DAOPagamento

ControllerFluxoCaixa

Associações

ViewVenda

ViewRelatorioFluxoCaixa

ViewEntrada

DAORelatorios

ViewFluxoCaixa

DAOFluxoCaixa

Classe

ControllerEntradas

Associações

ViewEntrada

ViewPagamento

ViewProduto

DAOEntradas

DAORelatorios

ViewFornecedor

ViewRelatorioEntradas

Classe

ControllerFornecedor

Associações

ViewBairro

ViewContasPagar

ViewEntrada

ViewCidade

DAORelatorios

ViewProduto

ViewRelatorioContasPagar

DAOFornecedor

ViewQuitacaoContasPagar

ViewFornecedor

ViewRelatorioFornecedor

ViewRelatorioEntradas

ViewRelatorioProduto

Classe

Datas

Classe

ControllerProdutos

Associações

DAOProdutos

DAORelatorios

ViewRelatorioProduto

ViewEntrada

ViewCategoria

ViewVenda

ViewDevolucaoVenda

ViewProduto

Classe

ControllerCidade

Associações

DAOCidade

ViewRelatorioCliente

ViewCliente

ViewRelatorioCidade

ViewRelatorioFornecedor

ViewRelatorioBairro

ViewFornecedor

ViewEmpresa

ViewBairro

ViewVendedor

DAORelatorios

ViewCidade

ViewRelatorioVendedor

Classe

ControllerBairro

Associações

ViewCliente

ViewRelatorioVendedor

ViewRelatorioFornecedor

ViewVendedor

DAORelatorios

ViewBairro

ViewRelatorioBairro

ViewRelatorioCliente

DAOBairro

Classe

ControllerContasReceber

Associações

ViewEmpresa

ViewVenda

ViewPagamento

ViewRelatorioContasReceber

DAOContasReceber

ViewQuitacaoContasReceber

ViewContasReceber

DAORelatorios

ViewCliente

Classe

ViewFluxoCaixa

Associações

Model Sessao Contas Pagar

ModelSaldoCaixa

ControllerSaldoCaixa

ModelSessaoEntradas

ModelFluxoCaixa

ModelSessaoVendas

ControllerFluxoCaixa

Model Sessao Contas Receber

ControllerVendas

ModelFluxoCaixa

Classe

ViewEntrada

Associações

ControllerFluxoCaixa

ModelProduto

ControllerProdutos

ModelEntradas

ModelFornecedor

ModelFluxoCaixa

ModelSessaoEntradas

Classe

MascValores

Associações

ViewEntrada

ViewVenda

Classe

LogoSistema

Classe

TiraAcentos

Classe

ServerConfig

Classe

ViewLogin

Associações

ModelSessaoUsuario

Classe

ViewCategoria

Associações

ModelProduto

ControllerProdutos

ModelCategoria

ControllerCategoria

ViewSobre

Classe

ViewDevolucaoVenda

Associações

ModelProduto

ControllerVendas

ModelPagamento

ModelVendedor

ControllerPagamento

ModelVendas

ModelProduto

ModelSessaoVendas

ModelCliente

ModelVendas

ControllerVendedor

ControllerCliente

ControllerProdutos

Classe

ViewRelatorioCategoria

Associações

ControllerCategoria

ModelCategoria

Classe

ViewProduto

Associações

ModelFornecedor

ModelProduto

ModelEntradas

ControllerFornecedor

ControllerEntradas

ControllerVendas

ControllerCategoria

ModelVendas

ModelCategoria

ControllerProdutos

Classe

ViewRelatorioCidade

Associações

ModelCidade

ControllerCidade

ViewRelatorioVendedor

Associações

ModelBairro

ModelVendedor

ModelCidade

ControllerVendedor

ControllerBairro

ControllerCidade

Classe

ViewQuitacaoContasPagar

Associações

ModelPagamento

ControllerPagamento

ControllerContasPagar

ModelFornecedor

ModelEntradas

ControllerFornecedor

ModelContasPagar

ModelSessaoContasPagar

Classe

Filtros

Classe

MySQLBackup

Associações

ViewPrincipal

Classe

Mascaras

Classe

ViewRelatorioContasReceber

Associações

ControllerPagamento

ControllerCliente

ModelCliente

ModelContasReceber

ModelPagamento

ModelCliente

ControllerContasReceber

Classe

ViewPrincipal

Associações

MySQLBackup

ViewRelatorioUsuario

Associações

ModelUsuario

ControllerUsuario

Classe

ViewPagamento

Associações

ModelVendas

ModelPagamento

ControllerVendas

ControllerEntradas

ModelContasReceber

Controller Contas Pagar

ModelContasPagar

ControllerPagamento

ModelEntradas

Classe

ViewContasReceber

Associações

ControllerPagamento

ModelVendas

ModelCliente

ControllerCliente

ModelContasReceber

ModelPagamento

ControllerVendas

ControllerContasReceber

Classe

ViewVenda

Associações

ModelCliente

ModelVendedor

ControllerContasReceber

ControllerFluxoCaixa

ModelFluxoCaixa

ModelPagamento

ControllerVendas

ControllerPagamento

ModelVendas

ModelProduto

ControllerVendedor

ModelSessaoVendas

ControllerProdutos

ModelContasReceber

ControllerCliente

ViewEmpresa

Associações

ModelBairro

ModelCliente

ControllerContasReceber

ControllerBairro

ModelVendas

ModelCliente

ControllerVendas

ModelCidade

ControllerCliente

ModelContasReceber

ControllerCidade

Classe

ViewRelatorioEntradas

Associações

ControllerPagamento

ModelPagamento

ModelFornecedor

ModelEntradas

ControllerEntradas

ControllerFornecedor

Classe

ViewCliente

Associações

ModelContasReceber

ControllerVendas

ModelBairro

ControllerCidade

ModelCidade

ControllerBairro

ModelVendas

ModelCliente

ControllerContasReceber

Classe

ViewRelatorioCliente

Associações

ModelBairro

ControllerCliente

ControllerCidade

ModelCidade

ModelCliente

ControllerBairro

ViewBairro

Associações

ControllerFornecedor

ModelBairro

ControllerCliente

ModelVendedor

ModelCidade

ControllerCidade

ControllerVendedor

ModelCliente

Classe

ViewRelatorioBairro

Associações

ModelCidade

ModelBairro

ControllerCidade

ControllerBairro

Classe

ViewRelatorioContasPagar

Associações

ModelFornecedor

ModelContasPagar

ControllerPagamento

ControllerFornecedor

ModelPagamento

Classe

ViewRelatorioVendas

Associações

ModelVendedor

ModelCliente

ControllerPagamento

ControllerVenda

ModelVendas

ModelPagamento

ControllerCliente

ControllerVendedor

Classe

ViewRelatorioProduto

Associações

ModelCategoria

ModelFornecedor

ControllerProdutos

ModelProduto

ControllerCategoria

ControllerFornecedor

ViewRelatorioFornecedor

Associações

ModelBairro

ModelFornecedor

ModelCidade

ControllerCidade

ControllerBairro

ControllerFornecedor

Classe

ViewQuitacaoContasReceber

Associações

ControllerPagamento

ModelContasReceber

ControllerCliente

ControllerContasReceber

ModelVendas

Model Sessao Contas Receber

ModelContasReceber

ModelCliente

ModelPagamento

Classe

ViewUsuario

Associações

ControllerUsuario

ModelUsuario

Classe

ViewRelatorioFluxoCaixa

Associações

ControllerFluxoCaixa

ModelFluxoCaixa

ModelPagamento

Classe

ModelUsuario

Associações

ViewRelatorioUsuario

ViewUsuario

Classe

ViewRelatorioPagamento

Associações

ModelPagamento

ControllerPagamento

ViewCidade

Associações

ControllerBairro

ModelCidade

ModelFornecedor

ControllerFornecedor

ControllerVendedor

ModelBairro

ControllerCidade

ModelVendedor

ModelCliente

ControllerCliente

Classe

ViewFornecedor

Associações

ControllerBairro

ModelContasPagar

ModelFornecedor

ControllerCidade

ModelCidade

ModelEntradas

ModelBairro

ControllerContasPagar

ControllerFornecedor

ControllerEntradas

Classe

ViewContasPagar

Associações

ControllerFornecedor

ModelPagamento

ModelContasPagar

ControllerPagamento

ModelFornecedor

ControllerContasPagar

ModelEntradas

Classe

ModelSessaoUsuario

Associações

ViewLogin

Classe

ModelSaldoCaixa

Associações

ViewFluxoCaixa

ModelContasPagar

Associações

ViewEntrada

ViewFornecedor

ViewContasPagar

ViewRelatorioContasPagar

ViewPagamento

ViewQuitacaoContasPagar

Classe

ModelEntradas

Associações

ViewEntrada

ModelEntradas

ViewProduto

ViewFornecedor

ViewQuitacaoContasPagar

ViewRelatorioEntradas

ViewPagamento

ViewContasPagar

Classe

ModelFornecedor

Associações

ViewRelatorioContasPagar

ViewProduto

ViewRelatorioFornecedor

ViewRelatorioProduto

ViewRelatorioEntradas

ViewCidade

ViewFornecedor

ViewQuitacaoContasPagar

ViewEntrada

ViewFornecedor

ViewRelatorioProduto

ViewContasPagar

ViewBairro

Classe

ModelPagamento

Associações

ViewQuitacaoContasPagar

ViewPagamento

ViewRelatorioPagamento

ViewRelatorioEntradas

ViewContasPagar

ViewEntrada

ViewContasPagar

ViewDevolucaoVenda

ViewVenda

ViewRelatorioContasReceber

ViewContasReceber

ViewRelatorioVendas

ViewRelatorioContasPagar

ViewPagamento

ViewRelatorioFluxoCaixa

ViewQuitacaoContasReceber

Classe

ModelCliente

Associações

ViewVenda

ViewEmpresa

ViewRelatorioContasReceber

ViewRelatorioVendas

ViewContasReceber

ViewRelatorioCliente

ViewDevolucaoVenda

ViewCliente

ViewCidade

ViewPrincipal

ViewQuitacaoContasReceber

ViewBairro

Classe

ModelVendedor

Associações

ViewRelatorioVendas

ViewRelatorioVendedor

ViewVenda

ViewBairro

ViewVendedor

ViewDevolucaoVenda

ViewCidade

Classe

ModelProduto

Associações

ViewProduto

ViewDevolucaoVenda

ViewEntrada

ViewRelatorioProduto

ModelProduto

ViewVenda

ViewCategoria

ModelVendas

Associações

ViewPagamento

ViewContasReceber

ModelVendas

ViewEmpresa

ViewDevolucaoVenda

ViewVenda

ViewRelatorioVendas

ViewCliente

ViewQuitacaoContasReceber

ViewProduto

ViewVendedor

Classe

ModelSessaoContasReceber

Associações

ViewFluxoCaixa

ViewQuitacaoContasReceber

Classe

ModelSessaoVendas

Associações

ViewDevolucaoVenda

ViewFluxoCaixa

ViewVenda

Classe

ModelControleRelatorio

Classe

ModelCategoria

Associações

ViewRelatorioProduto

ViewRelatorioCategoria

ViewCategoria

ViewProduto

Classe

ModelSessaoContasPagar

Associações

ViewFluxoCaixa

ViewQuitacaoContasPagar

Classe

ModelBairro

Associações

ViewRelatorioCliente

ViewRelatorioVendedor

ViewRelatorioFornecedor

ViewBairro

ViewRelatorioBairro

ViewEmpresa

ViewCliente

ViewVendedor

ViewCidade

ViewFornecedor

ViewCliente

Classe

ModelFluxoCaixa

Associações

ModelFluxoCaixa

ViewFluxoCaixa

ViewVenda

ViewRelatorioFluxoCaixa

ViewEntrada

Classe

ModelSessaoEntradas

Associações

ViewFluxoCaixa

ViewEntrada

Classe

ModelConta

Classe

ModelContasReceber

Associações

ViewCliente

ViewQuitacaoContasReceber

ViewContasReceber

ViewRelatorioContasReceber

ViewPagamento

ViewEmpresa

ViewVenda

Classe

ModelCidade

Associações

ViewRelatorioBairro

ViewRelatorioCidade

ViewCidade

ViewRelatorioFornecedor

ViewCliente

ViewRelatorioVendedor

ViewRelatorioCliente

ViewEmpresa

ViewBairro ViewFornecedor ViewVendedor

Classe

DAORelatorios

Associações

Controller Saldo Caix a

ControllerProdutos

ControllerPagamento

ControllerFornecedor

ControllerContasPagar

ControllerCliente

ControllerEntradas

ControllerBairro

ControllerFluxoCaixa

ControllerCidade

ControllerCategoria

ControllerConta

ControllerVendedor

ControllerUsuario

Controller Contas Receber

ControllerVendas

3.5 Diagrama de estados

O diagrama de estados objetiva demonstrar o comportamento interno das classes do sistema. Na figura 33, podem ser observados os estados principais do diagrama, sem detalhamento dos estados internos. Posteriormente, serão inseridas as Figuras com o detalhamento destes estados internos.

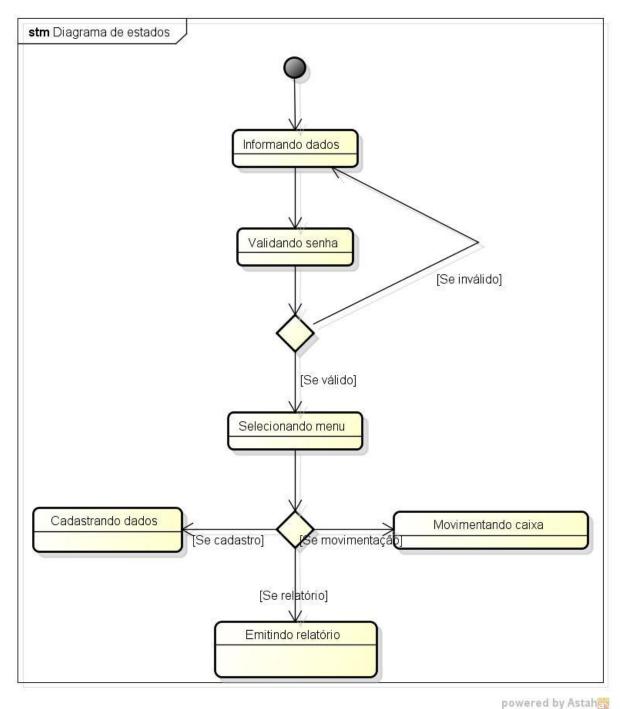


FIGURA 33: Estados principais do Diagrama de Estados

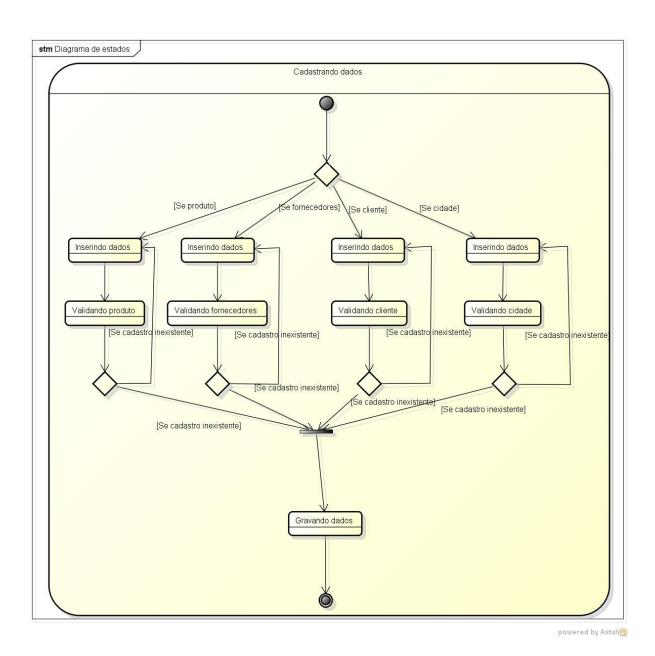


FIGURA 34: Estado "Cadastrando dados" do Diagrama de Estados

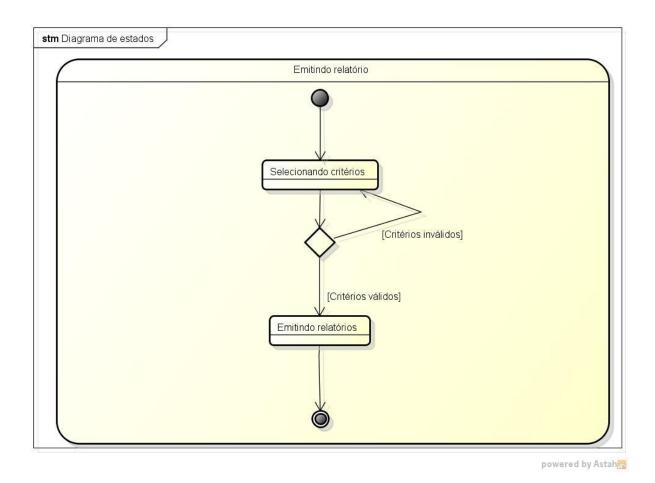


FIGURA 35: Estado "Emitindo relatório" do Diagrama de Estados

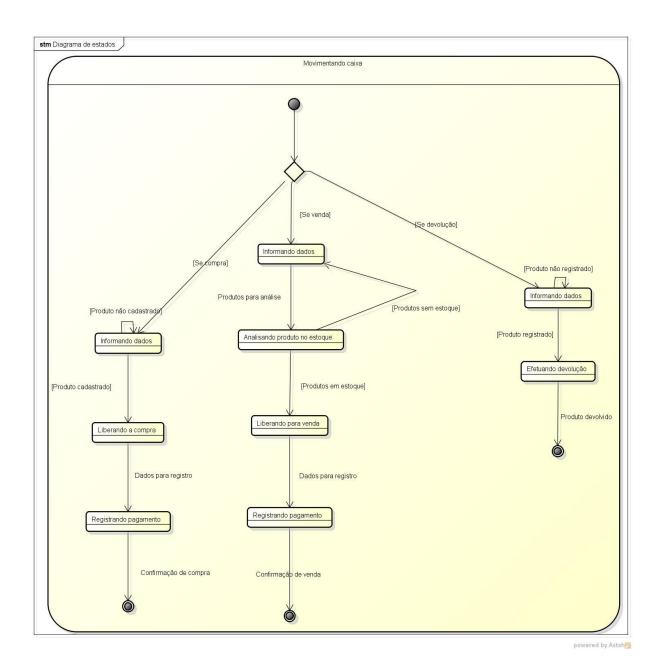


FIGURA 36: Estado "Movimento caixa" do Diagrama de Estados

3.6 Diagrama de implantação

O diagrama de implantação descreve os componentes de hardware e software necessários para suportar o processamento do software. Na Figura 37 pode ser visto o diagrama de implantação do MS Commerce.

Neste diagrama, estão representados dois computadores ligados a um servidor através de transmissão de dados pelo protocolo TCP/IP, onde cada computador tem seus requisitos mínimos para suportar o MS Commerce. Já o computador que servirá de servidor, tem um requisito diferenciado para suporte do sistema.

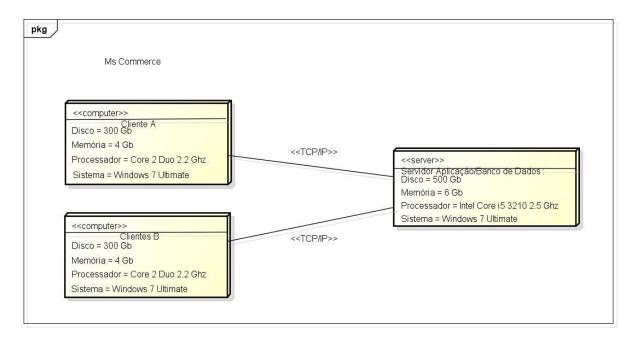


FIGURA 37: Diagrama de implantação

3.7 Diagrama de componentes

Objetiva capturar a estrutura física da implementação e suas relações. Alguns exemplos de estrutura física são as "dll" e classes utilizadas para funcionamento do software. O diagrama está divido em duas partes onde, na Figura 38, o componente Bibliotecas está sem seus componentes internos, que são mostrados na Figura 39.

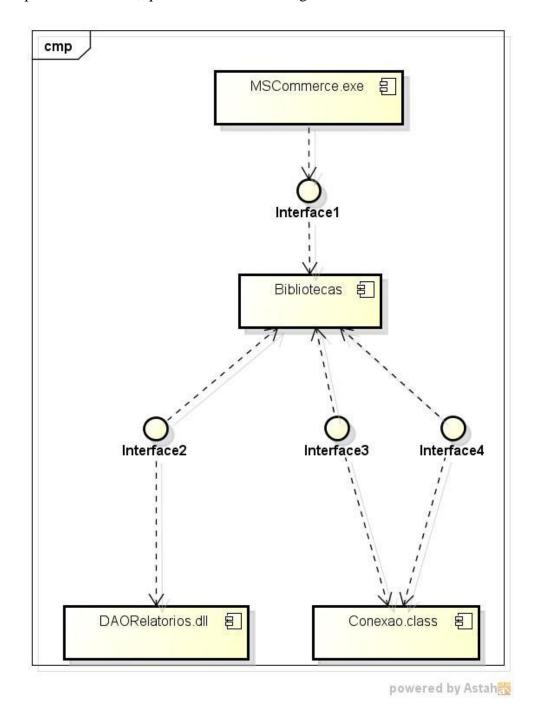


FIGURA 38: Diagrama de componentes (Parte 1)

powered by Astah

Bibliotecas §
bibliotecalUGComp jar : UnknownType
bibliotecalcommons-digester-1.7 jar [5] bibliotecalvitext-2.1.7 jar [5] JDK 1.7 (Default) [5]
bibliotecalicalendar-1.4 jar [a] bibliotecal/mysql-connector-java-5.1.18-bin jar [a] bibliotecal/commons-logging-1.1 jar [a]
Relatório 夏 bibliotecaljasperreports-fonts-4.0.0 jar 夏
bibliotecaljasperreports-javaflow-4.0.0 jar [5]
bibliotecaljasperreports-4.0.0.jar; UnknownType
Interface bibliotecalsyntheticaBlackStar_2.9 6\syntheticaBlackStar jar bibliotecalsyntheticaBlackStar_2.9 6\syntheticaBlackStar jar bibliotecalsyntheticaBlackStar_2.9 6\synthetica jar files

FIGURA 39: Diagrama de componentes (Parte 2)

4 Cronograma

Abaixo, na Figura 40, pode ser visualizada a Estrutura Analítica do Projeto (EAP) elaborada no Microsoft Office Project 2007.

Com Plan	nnento inicial do escopo nto escopo finir EAP finir	100% 100% 100% 100% 100% 100% 100% 100%	20 days 16 days 6 days 61 days 6 days 1 day 1 days 1 day	20 days Mon 31/03/14 5 days Mon 21/04/14 82 days Mon 21/04/14 61 days Mon 28/04/14 61 days Mon 28/04/14	Fri 25/04/14 Fri 18/04/14 Fri 25/04/14/2 Tue 22/07/14 1	Lucas Souza;Marcos Vinicius;Rafael Fernandes;Fernando Gomes;Luis Augusto
E S S	inder do projeto no no no no no no no no no	100% 100% 100% 100% 100% 100% 100% 100%	16 days 6 days 82 days 61 days 61 days 1 day 1 d	Mon 31/03/14 Mon 21/04/14 Mon 31/03/14 Mon 28/04/14 Mon 28/04/14	Fri 18/04/14 Fri 25/04/14 2 Tue 22/07/14 1	Lucas Souza;Marcos Vinicius;Rafael Fernandes;Fernando Gomes;Luis Augusto
E S S S S S S S S S S S S S S S S S S S	irider do projeto Into	100% 100% 100% 100% 100% 100% 100% 100%	6 days 82 days 61 days 61 days 5 days 1 day	Mon 21/04/14 Mon 28/04/14 Mon 28/04/14	Fri 25/04/14 2 Tue 22/07/14 1 Mon 21/07/14	The second secon
L Kei	nto escopo finir fases do projeto finir fases do projeto finir fases do projeto rompor entregaveis compor entregaveis finir equipe do projeto finir equipe do projeto senvolver matriz de responsabilidae lar projetos uniões periodicas uniões periodicas uniões de engenharia de softwar cumentos de visão cumento de regra de negocio cumento de regra de negocio controlor de contra	100% 100% 100% 100% 100% 100% 100% 100%	61 days 61 days 61 days 6 days 11 day 82 days 6 days 11 day 1188 days	Mon 31/03/14 Mon 28/04/14 Mon 28/04/14	Tue 22/07/14 1 Mon 21/07/14	Jeneffer[40%];Fernando Gomes[40%]
, in the second	finir EAP finir fases do projeto finir fases do projeto finir fases do projeto compor entregaveis compor entregaveis finir equipe do projeto finir equipe do projeto resenvolver cronograma ssenvolver cronograma senvolver matriz de responsabilidac finir equipe do projeto resenvolver mudariz de escopo uniões periodicas uniões periodicas cuminos de engenharia de softwar cumentos de visão cumento de regra de negocio cumento de regra de negocio confinoo	100% 100% 100% 100% 99% 100% 100% 100% 1		Mon 28/04/14 Mon 28/04/14	Mon 21/07/14	
<u> </u>	finir EAP finir fases do projeto finir fases do projeto finir pinicipals entregaveis compor entregaveis volver plano do projeto finir equipe do projeto rsemolver cronograma rsemolver matriz de responsabilidac finir equipe se periodicas uniões periodicas introjetos rentos de engenharia de softwar rentos de engenharia de softwar cumentos de visão cumento de regira de negocio cumento de regira de negocio controlos de regira de negocio	100% 100% 100% 100% 100% 100% 100%	61 days 5 days 1 day 82 days 5 days 1 day 1 day 1 day	Mon 28/04/14		Jeneffer; Lucas Souza; Marcos Vinicius; Rafael Fernandes; Fernando Gomes
Š Š	finir fases do projeto finir principais entregaveis compor entregaveis volver plano do projeto finir equipe do projeto finir equipe do projeto resenvolver cronograma resenvolver cronograma resenvolver matriz de responsabilidad far projetos uniões periodicas nuriões periodicas introlar mudanças de escopo rentos de engenharia de softwar commentos de visão commento de regira de negocio coumento de regira de negocio contrippo	100% 100% 100% 100% 100% 100% 100%	5 days 1 day 82 days 5 days 1 day 1 day 188 days		Mon 21/07/14 1	Lucas Souza;Marcos Vinicius;Rafael Fernandes;Fernando Gomes;Jeneffer
Exercise Control Contr	finir principais entregaveis compor entregaveis volver plano do projeto finir equipe do projeto ssemolver matriz de responsabilidac lar projetos uniões pendoticas unidas pendoticas introlar mudanças de escopo nentos de engenharia de softwar cumentos de visão cumento de requisitos cumento de requisitos cumento de regis	100% 100% 100% 100% 100% 100% 100%	1 day 82 days 5 days 1 day 1 day 188 days	Mon 28/04/14	Fri 02/05/14 6SS	Lucas Souza[80%];Marcos Vinicius[80%];Rafael Fernandes[80%];Fernando Gomes[80%];Jeneffer[80%]
Exe.	compor entregaveis volver plano do projeto finir equipe do projeto ssenvolver matriz de responsabilidac lar projetos unides periodicas introlar mudanças de escopo rentos de engenharia de softwar cumentos de visão cumento de requisitos commento de regira de negocio controlipo ototipo	99% 100% 100% 100% 100% 100%	1 day 82 days 5 days 1 day 1 day 188 days	Fri 20/06/14	Fri 20/06/14 7SS	Lucas Souza, Marcos Vinicius; Rafael Fernandes; Fernando Gomes; Jeneffer
E E E	volver plano do projeto finir equipe do projeto senvolver matriz de responsabilidac senvolver matriz de responsabilidac luniões periodicas introlar mudanças de escopo nentos de engenharia de softwar cumentos de visão cumento de requisitos cumento de regia de negocio ototipo	99% 100% 100% 100% 100% 100%	82 days 5 days 1 day 1 day 188 days	Fri 20/06/14	Fri 20/06/14 8SS	Lucas Souza; Marcos Vinicius; Rafael Fernandes; Fernando Gomes, Jeneffer
Exe.	finir equipe do projeto senvolver cronograma isenvolver matriz de responsabilidae lar projetos uniões periodicas uniões periodicas nutrolar mudanças de escopo rentos de engenharia de softwar comentos de visão commento de regus de negocio ototipo	100% 100% 100% 100% 100%	5 days 1 day 188 days	Mon 31/03/14	Tue 22/07/14 5	Jeneffer
EX EXE	sservolver cronograma sservolver matriz de responsabilidac far projetos uniões periodicas uniões periodicas nutrolar mudanças de escopo tentos de engenharia de softwar cumentos de visão cumento de reguis de negocio controlo de regra de negocio	100% 100% 100% 100% 100%	1 day 1 day 188 days	Mon 31/03/14	Fri 04/04/12 9	Marcos Vinicius;Rafael Fernandes;Fernando Gomes
Con	isenvolver matriz de responsabilidac lar projetos uniões periodicas introlar mudanças de escopo nentos de engenharia de softwar comentos de visão comento de requisitos	100% 100% 100% 100%	1 day 188 days	Mon 21/07/14	Mon 21/07/14 11	Marcos Vinicius
Con	la projetos unides periodicas introlar mudanças de escopo tentos de engenharia de softwar cumentos de visão cumento de requisitos cumento de regra de negocio otótripo	100% 100% 100% 92%	188 days	Mon 21/07/14	Tue 22/07/14 12	Jeneffer, Lucas Souza, Fernando Gomes
<u> </u>	lar projetos unides periodicas introlar mudanças de escopo tentos de engenharia de softwar cumentos de visão cumento de requisitos cumento de regra de negocio doringoo	100% 100% 100%	188 days	Mon 31/03/14	Wed 17/12/14	Marcos Vinicius;Rafael Fernandes
Exe	uniões periodicas introlar mudanças de escopo tentos de engenharia de softwar cumentos de visão cumento de regia de negocio cumento de regia de negocio dotripo	100%		Mon 31/03/14	Wed 17/12/14	
Exe	introlar mudanças de escopo ientos de engenharia de softwar icumentos de visão icumento de requisitos icumento de regra de negocio otoritos otoritos	100%	188 days	Mon 31/03/14	Wed 17/12/14	Lucas Souza; Marcos Vinicius; Rafael Fernandes; Fernando Gomes; Vinicius Santos
EX	ientos de engenharia de softwar icumentos de visão icumento de requisitos icumento de regra de negocio ototipo	%26	159 days	Mon 07/04/14	Tue 16/12/14	Marco vinicius, Rafael Fernandes, Vinicius Santos
			163 days?	Mon 05/05/14	Wed 17/12/14 4	
	tos le negocio	%66	65 days?	Mon 05/05/14	Fri 01/08/14 10	Lucas Souza, Marcos Vinicius, Rafael Fernandes, Fernando Gomes; Luis Augusto
	Documento de requisitos Documento de regra de negocio Prototipo	100%	5 days	Mon 05/05/14	Fri 09/05/14	Lucas Souza[80%];Marcos Vinicius[80%];Rafael Fernandes[80%];Fernando Gomes[80%];Luis Augusto[809
	Documento de regra de negocio Prototipo	100%	5 days	Mon 12/05/14	Fri 16/05/14 20	Fernando Gomes, Vinicius Santos, Luis Augusto
	Prototipo	100%	0 days	Tue 22/07/14	Tue 22/07/14 21	Lucas Souza, Marcos Vinicius; Rafael Fernandes; Fernando Gomes; Luis Augusto
	our ob cocc ob office of the contract of	100%	1 day	Mon 02/06/14	Wed 23/07/14/22	Fernando Gomes;Luis Augusto
	Documento de especificação de caso de uso	100%	0 days	Sun 20/07/14	Sun 20/07/14 23	Lucas Souza; Marcos Vinicius; Rafael Fernandes; Fernando Gomes
	Documentação estrategia de testes	100%	5 days?	Mon 28/07/14	Fri 01/08/14 24	Fernando Gomes;Luis Augusto
27 28	Artefatos de infraestrutura de software	100%	25 days?	Sun 20/07/14	Fri 22/08/14	
28	Diagrama de Caso de Uso	100%	0 days?	Sun 20/07/14	Sun 20/07/14 21	Lucas Souza; Marcos Vinicius; Rafael Fernandes; Fernando Gomes
	Diagrama de Atividades	100%	5 days?	Mon 04/08/14	Fri 08/08/14 27	Luis Augusto
59	Diagrama de Sequencia	100%	5 days?	Mon 11/08/14	Fri 15/08/14 28	Luis Augusto
	Diagrama de Classe	100%	5 days?	Mon 18/08/14	Fri 22/08/14	Luis Augusto
	Criação de Banco de Dados	100%	28 days	Tue 22/07/14	Fri 29/08/14	Rodrigo
32	MER Conceitual	100%	0 days	Tue 22/07/14	Tue 22/07/14	Fernando Gomes;Rodnigo;Luis Augusto
33	MER Lógico	100%	0 days	Tue 22/07/14	Tue 22/07/14	Rodrigo
	Script SQI	100%	5 days	Mon 25/08/14	Fri 29/08/14 33	Rodnigo
	Construção	93%	57 days?	Mon 01/09/14	Tue 18/11/14 19	
36	Plano de Construção	100%	5 days?	Mon 01/09/14	Fri 05/09/14 31SS	Lucas Souza;Marcos Vinicius;Rafael Fernandes;Fernando Gomes;Tales
37	Codificação	100%	52 days?	Mon 08/09/14	Tue 18/11/14	Lucas Souza; Marcos Vinicius; Rafael Fernandes; Fernando Gomes, Tales
38	Testes: Execução	100%	10 days?	Mon 27/10/14	Fri 07/11/14	Fernando Gomes;Tales
39	Homologação: Execução	%0	5 days?	Mon 10/11/14	Fri 14/11/14	Lucas Souza;Marcos Vinicius;Rafael Fernandes;Fernando Gomes;Vinicius Santos
40 In	Integração	%0	1 day?	Fri 12/12/14	Sat 13/12/14	
41	Preparação de Ambientes	%0	1 day?	Fri 12/12/14	Sat 13/12/14	Lucas Souza;Marcos Vinicius;Rafael Fernandes;Fernando Gomes;Tales
42	Plano de Iteração	%0	1 day?	Fri 12/12/14	Sat 13/12/14	Lucas Souza; Marcos Vinicius; Rafael Fernandes; Fernando Gomes; Tales
43 Tı	Fransição	%0	1 day?	Wed 17/12/14	Wed 17/12/14	
44	Plano de Implantação	%0	1 day?	Wed 17/12/14	Wed 17/12/14	Lucas Souza; Marcos Vinicius; Rafael Fernandes; Fernando Gomes
45	Treinamento	%0	1 day?	Wed 17/12/14	Wed 17/12/14	Lucas Souza; Marcos Vinicius; Rafael Fernandes; Fernando Gomes
46 Encer	Encerramento	%0	1 day?	Thu 18/12/14	Thu 18/12/14 1;4;14;18	
47 EI	Encerramento Administrativo	%0	1 day?	Thu 18/12/14	Thu 18/12/14	Jeneffer, Luis Augusto, Vinicius Santos, Fernando Gomes
48 A	Análise postmortem	%0	1 day?	Thu 18/12/14	Thu 18/12/14 47FF	Jeneffer, Luis Augusto; Vinicius Santos

FIGURA 40: Estrutura analítica do projeto

5 Conclusão

Após o MS Commerce ter sido finalizado, conclui-se que foi obtido um resultado mais que satisfatório, superando as expectativas em relação ao sistema. Foram implementadas funcionalidades que não foram planejadas anteriormente, como o ServerConfig e o Backup, trazendo um resultado positivo ao sistema.

O sistema, atualmente, possibilita a realização de cadastros de:

- Empresa, com as informações da empresa que está utilizando o sistema.
- Usuário, informando o perfil de acesso do usuário, de forma que os três diferentes perfis de acesso vetam o acesso a certas funções do sistema, dependendo do tipo de perfil o acesso as funções.
- Clientes, com as informações dos clientes da empresa que utiliza o sistema, de forma que o registro de clientes é ligado a diversas funcionalidades do sistema, sendo a principal a relação de cliente e saída de produto.
- Produtos, possibilitando ao usuário registrar o produto que foi comprado pela empresa e será estocado, informando o nome, a quantidade do produto, o fornecedor que vendeu o produto, a categoria, o valor e a situação. O cadastro de produtos será utilizado para diversas funções, dentre elas o registro de quais produtos foram retirados do estoque no momento da venda e sua quantidade, atualizando o número de produtos em estoque.
- Categoria, possibilitando o agrupamento de vários produtos em uma categoria, facilitando a utilização do MS Commerce para organização de produtos e venda.
- Fornecedores, gerenciando os fornecedores de produtos a empresa, associando as informações de fornecedor a cada produto que entra na empresa.
- Bairro, associado diretamente a uma cidade, proporciona facilidade e segurança para inserção de informações de endereço, evitando dados comprometidos no banco de dados.
 - Cidade, abrangendo as informações de bairro, auxilia na inserção de endereço.
- Forma de pagamento, que pode ser cadastrada a gosto do usuário, proporciona a inserção de tipo de pagamento à vista ou a prazo, além da inserção de parcelas, dias de vencimento e o nome da nova forma de pagamento.

- Vendedor, sendo associado a cada venda efetuada pelo sistema, para posterior averiguação de quem realizou certa venda.

Com estas informações, podem ser realizadas as operações de venda ou compra de produto na empresa. Ambas as operações, como mostrado nas seção de telas do sistema, proporcionam a inserção ou remoção de itens em lista, visando facilitar a utilização do MS Commerce pelo usuário.

Após uma compra ou venda ter sido realizada a prazo, é possibilitado a um usuário do sistema que tenha acesso a seção financeira realizar a quitação da compra ou venda em parte ou total, assim como acrescer o prazo de vencimento da ação a prazo.

O MS Commerce também permite a emissão de relatórios de conteúdos cadastrais, de vendas, de entradas e de finanças. Cada relatório provê diferentes tipos de filtros que objetivam refinar a procura que o sistema faz por informações no banco, trazendo para o usuário apenas as informações que ele deseja.

O sistema também conta, como dito anteriormente, com funcionalidade de backup e configuração de servidor de banco de dados.

A funcionalidade de backup, que não estava planejada, foi adicionada ao projeto e implementada com sucesso, proporcionando a oportunidade de realizar uma cópia do atual banco de dados do usuário para um diretório reservado para esta finalidade dentro do diretório do MS Commerce. O backup proporciona segurança e maior conforto para trabalhar com o sistema pois, caso ocorra qualquer tipo de problema com relação ao banco de dados, o último backup pode ser retomado e o usuário pode prosseguir com seu trabalho na empresa.

Já a funcionalidade de configuração de servidor proporciona maior facilidade na hora de instalação do sistema na máquina do futuro usuário, não necessitando de qualquer alteração do código do software para adaptação ao computador do usuário.

O MS Commerce será implementado em micro e pequenas empresas como forma de teste e, à medida que são encontrados erros e novas necessidades do cliente, o software será adaptado às necessidades do cliente e do mercado.

É esperado um bom retorno por parte das empresas que se disporem a testar o software, devido à automação e melhoria da segurança da informação que o MS Commerce disponibiliza ao âmbito de trabalho.

6 Referências

DIAS, Donaldo S.; GAZZANEO, Giosafatte. **Projeto de Sistemas de Processamento de Dados**. 1. ed. Rio de Janeiro: Livros Técnicos e Científicos, 1975.

LAUDON, Kenneth C.; LAUDON, Jane Price. **Sistemas de informação**. 4. ed. LTC: Rio de Janeiro,1999.

LOBO, A.P., TI Maior: Governo oficializa programa de apoio a empresas inovadoras.

In: Convergência Digital. Disponível em:

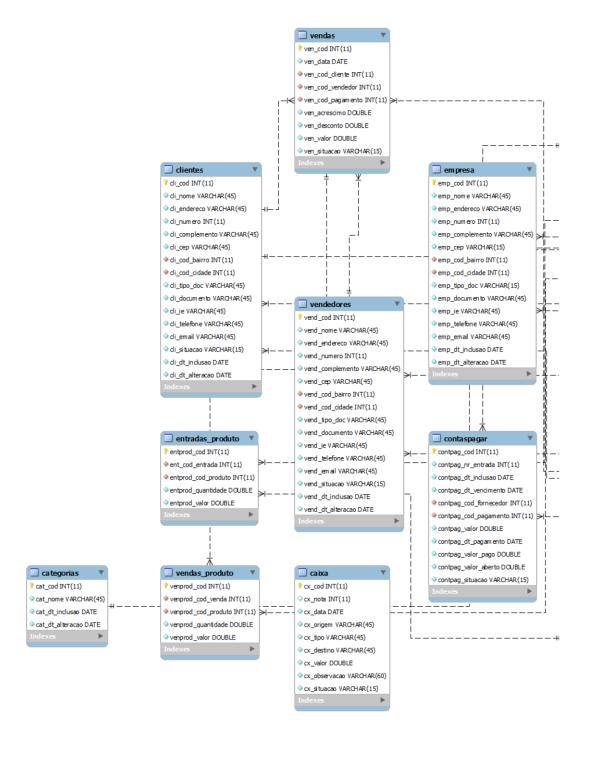
http://convergenciadigital.uol.com.br/cgi/cgilua.exe/sys/start.htm?infoid=32085&sid=7#.U5 ODRShUe72>. Acessado em: 07 de Junho de 2014.

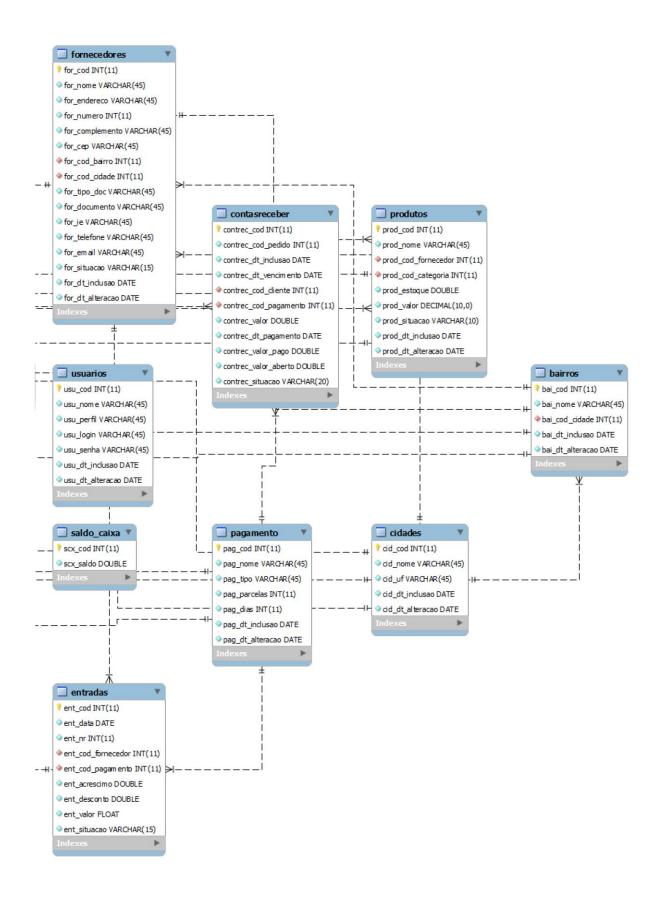
PRESSMAN, Roger S., Engenharia de software. 6. ed. Porto Alegre: Bookman, 2010.

PMBOK., Um guia do Conjunto de Conhecimentos em Gerenciamento de Projetos. 3. ed. Newtown Square: Project Management Institute, 2004.

7 Anexo

7.1 DER





7.2 Diagrama de Classe

```
Conexao
- status : boolean = false
- mensagem : String = "
- servidor : String = this.server
- nomeDoBanco : String = "msdb"
- usuario : String = this.user
- senha : String = this.password
+ Conexao()
+ Conexao(pServidor : String, pNomeDoBanco : String, pUsuario : String, pSenha : String)
+ conectar(): Connection
+ executarSQL(pSQL : String) : boolean
+ executarUpdateDeleteSQL(pSQL : String) : boolean
+ insertSQL(pSQL : String) : int
+ fecharConexao(): boolean
+ isStatus(): boolean
+ getMensagem(): String
+ getStatement(): Statement
+ getResultSet(): ResultSet
+ setMensagem(mensagem : String) : void
+ getCon(): Connection
+ setCon(con : Connection) : void
+ setStatement(statement : Statement) : void
+ setResultSet(resultSet : ResultSet) : void
+ getServidor(): String
+ setServidor(servidor : String) : void
+ getNomeDoBanco(): String
+ setNomeDoBanco(nomeDoBanco : String) : void
+ getUsuario(): String
+ setUsuario(usuario : String) : void
+ getSenha(): String
+ setSenha(senha : String) : void
```

ControllerBairro

- + salvarBairroController(pModelBairro : ModelBairro) : int
- + getBairroController(pCodigo : int) : ModelBairro
- + getBairroCidadeController(pCodigo: int): ModelBairro
- + getBairroController(): ModelBairro
- + getBairroController(pNome : String) : ModelBairro
- + getListaBairroController(): ArrayList<ModelBairro>
- + getListaBairroControllerFiltros(modelBairro : ModelBairro) : ArrayList<ModelBairro>
- + atualizarBairroController(pModelBairro : ModelBairro) : boolean
- + excluirBairroController(pCodigo : int) : boolean
- + gerarRelatorioBairro(modelBairro : ModelBairro) : ArrayList<ModelBairro>

ControllerCategoria

- + salvarCategoriaController(pModelCategoria : ModelCategoria) : int
- + getCategoriaController(pCodigo : int) : ModelCategoria
- + getCategoriaController(): ModelCategoria
- + getCategoriaController(pNome : String) : ModelCategoria
- + getListaCategoriaController(): ArrayList<ModelCategoria>
- + getListaCategoriaControllerFiltros(modelCategoria: ModelCategoria): ArrayList<ModelCategoria>
- + atualizarCategoriaController(pModelCategoria : ModelCategoria) : boolean
- + excluirCategoriaController(pCodigo: int): boolean
- + gerarRelatorioCategoria(modelCategoria: ModelCategoria): ArrayList<ModelCategoria>

ControllerCategoria

- + salvarCategoriaController(pModelCategoria : ModelCategoria) : int
- + getCategoriaController(pCodigo : int) : ModelCategoria
- + getCategoriaController(): ModelCategoria
- + getCategoriaController(pNome : String) : ModelCategoria
- + getListaCategoriaController(): ArrayList<ModelCategoria>
- + getListaCategoriaControllerFiltros(modelCategoria: ModelCategoria): ArrayList<ModelCategoria>
- + atualizarCategoriaController(pModelCategoria : ModelCategoria) : boolean
- + excluirCategoriaController(pCodigo : int) : boolean
- + gerarRelatorioCategoria(modelCategoria : ModelCategoria) : ArrayList<ModelCategoria>

ControllerCidade

- + salvarCidadeController(pModelCidade: ModelCidade): int
- + getCidadeController(): ModelCidade
- + getCidadeController(pCodigo : int) : ModelCidade
- + getCidadeController(pNome : String) : ModelCidade
- + getListaCidadeController(): ArrayList<ModelCidade>
- + getListaCidadeControllerFiltros(modelCidade : ModelCidade) : ArrayList<ModelCidade>
- + atualizarCidadeController(pModelCidade : ModelCidade) : boolean
- + excluirCidadeController(pCodigo : int) : boolean
- + gerarRelatorioCidade(modelCidade : ModelCidade) : ArrayList<ModelCidade>

ControllerCliente

- + salvarClienteController(pModelCliente : ModelCliente) : int
- + getClienteController(): ModelCliente
- + getClienteController(pCodigo : int) : ModelCliente
- + getClienteBairroController(pCodigo: int): ModelCliente
- + getClienteCidadeController(pCodigo : int) : ModelCliente
- + getClienteController(pNome : String) : ModelCliente
- + getListaClienteController(): ArrayList<ModelCliente>
- getListaClienteControllerFiltros(modelCliente : ModelCliente) : ArrayList<ModelCliente>
- + atualizarClienteController(pModelCliente : ModelCliente) : boolean
- + excluirClienteController(pCodigo : int) : boolean
- + gerarRelatorioCliente(modelCliente : ModelCliente) : ArrayList<ModelCliente>

ControllerConta

- + salvarContaPagarController(pModelContaPagar : ModelConta) : int
- + getContaPagarController(pCodigo : int) : ModelConta
- + getListaContaPagarController(): ArrayList<ModelConta>
- + getListaContaPagarController(pStatus : int) : ArrayList<ModelConta>
- + getListaContaReceberController(pStatus : int) : ArrayList<ModelConta>
- + getListaContasController(pStatus : int) : ArrayList<ModelConta>
- + atualizarContaPagarController(pModelContaPagar : ModelConta) : boolean
- + atualizarContaReceberController(pModelContaReceber : ModelConta) : boolean
- + receberContaReceberController(pModelConta : ModelConta) : boolean
- + pagarContaPagarController(pModelConta : ModelConta) : boolean
- + getListaContasController(pModelConta : ModelConta) : ArrayList<ModelConta>

ControllerContasPagar

- + salvarContasPagarController(pModelContasPagar : ModelContasPagar) : int
- + getContasPagarController(pCodigo : int) : ModelContasPagar
- + getContasPagarFornecedorController(pCodigo : int) : ModelContasPagar
- + getContasPagarPagamentoController(pCodigo: int): ModelContasPagar
- + getListaContasPagarControllerFitros(modelContasPagar : ModelContasPagar) : ArrayList<ModelContasPagar>
- + getListaContasPagarControllerFiltrosQuitar(modelContasPagar: ModelContasPagar): ArrayList<ModelContasPagar>
- getListaContasPagarrController(): ArrayList<ModelContasPagar>
- + atualizarContasPagarController(pModelContasPagar : ModelContasPagar) : boolean
- + excluirContasPagarController(pCodigo : int) : boolean
- + gerarRelatorioContasPagar(modelContasPagar: ModelContasPagar): ArrayList<ModelContasPagar>

ControllerContasReceber

- + salvarContasReceberController(pModelContasReceber: ModelContasReceber): int
- + getContasReceberController(pCodigo : int) : ModelContasReceber
- + getContasReceberClienteController(pCodigo : int) : ModelContasReceber
- + getContasReceberPagamentoController(pCodigo : int) : ModelContasReceber
- + getContasReceberVendasController(pCodigo : int) : ModelContasReceber
- + getListaContasReceberControllerFiltros(modelContasReceber: ModelContasReceber): ArrayList<ModelContasReceber>
- + getListaContasReceberControllerFiltrosQuitar(modelContasReceber: ModelContasReceber): ArrayList<ModelContasReceber>
- + getListaContasReceberController(): ArrayList<ModelContasReceber>
- + atualizarContasReceberController(pModelContasReceber : ModelContasReceber) : boolean
- + excluirContasReceberController(pCodigo : int) : boolean
- + gerarRelatorioContasReceber(modelContasReceber: ModelContasReceber): ArrayList<ModelContasReceber>

ControllerEntradas

- + salvarEntradasController(pModelEntradas : ModelEntradas) : int
- + getEntradasController(pCodigo : int) : ModelEntradas
- + getEntradasFornecedorController(pCodigo : int) : ModelEntradas
- + getEntradasPagamentoController(pCodigo : int) : ModelEntradas
- + getEntradasProdutoController(pCodigo : int) : ModelEntradas
- + getEntradasProdutoItemController(pCodigo : int) : ModelEntradas
- + getEntradasNotaController(pNota : int) : ModelEntradas
- + getEntradasController(p∨alorTotal : Double) : ModelEntradas
- + getListaEntradasController(): ArrayList<ModelEntradas>
 + gerarRelatorioEntradas(modelEntradas: ModelEntradas): ArrayList<ModelEntradas>
- + getListaEntradasController(pCodigo: int): ArrayList<ModelEntradas>
- + getListaEntradasControllerFiltros(modelEntrada : ModelEntradas) : ArrayList<ModelEntradas>
- + atualizarEntradasController(pModelEntradas : ModelEntradas) : boolean
- + excluirEntradasController(pCodigo : int) : boolean
- + salvarEntradasProdutosController(modelEntradas : ModelEntradas) : boolean
- + excluirProdutoEntradasController(pCodigo : int) : boolean
- + excluirProdutoEntradasNotaController(pCodigo : int) : boolean
- + getListaEntradasController(pMdelEntradas : ModelEntradas) : ArrayList<ModelEntradas>

ControllerFluxoCaixa

- + salvarFluxoCaixaController(pModelFluxoCaixa : ModelFluxoCaixa) : int
- + getFluxoCaixaController(pCodigo : int) : ModelFluxoCaixa
- + getFluxoCaixa VendaController(pCodigo : int) : ModelFluxoCaixa
- + getFluxoCaixaEntradaController(pCodigo : int) : ModelFluxoCaixa
- + getFluxoCaixaController(): ModelFluxoCaixa
- + getListaFluxoCaixaControllerFiltros(modelFluxoCaixa: ModelFluxoCaixa): ArrayList<ModelFluxoCaixa>
- + getListaFluxoCaixaController(): ArrayList<ModelFluxoCaixa>
- + atualizarFluxoCaixaController(pModelFluxoCaixa : ModelFluxoCaixa) : boolean
- + excluirFluxoCaixaController(pCodigo : int) : boolean
- + gerarRelatorioFluxoCaixa(modelFluxoCaixa : ModelFluxoCaixa) : ArrayList<ModelFluxoCaixa>

ControllerFornecedor

- + salvarFornecedorController(pModelFornecedor: ModelFornecedor): int
- + getFornecedorController(): ModelFornecedor
- + getFornecedorController(pCodigo : int) : ModelFornecedor
- + getFornecedorBairroController(pCodigo : int) : ModelFornecedor
- + getFornecedorCidadeController(pCodigo: int): ModelFornecedor
- + getFornecedorController(pNome : String) : ModelFornecedor
- + getListaFornecedorController(): ArrayList<ModelFornecedor>
- + getListaFornecedorControllerFiltros(modelFornecedor: ModelFornecedor): ArrayList<ModelFornecedor>
- + atualizarFornecedorController(pModelFornecedor : ModelFornecedor) : boolean
- + excluirFornecedorController(pCodigo : int) : boolean
- + gerarRelatorioFornecedor(modelFornecedor: ModelFornecedor): ArrayList<ModelFornecedor>

ControllerPagamento

- + salvarPagamentoController(pModelPagamento : ModelPagamento) : int
- + getPagamentoController(): ModelPagamento
- + getPagamentoController(pCodigo : int) : ModelPagamento
- + getPagamentoController(pNome : String) : ModelPagamento
- + getListaPagamentoController(): ArrayList<ModelPagamento>
- + getListaPagamentoPrazoController(): ArrayList<ModelPagamento>
- + getListaPagamentoControllerFiltros(modelPagamento: ModelPagamento): ArrayList<ModelPagamento>
- + atualizarPagamentoController(pModelPagamento : ModelPagamento) : boolean
- + excluirPagamentoController(pCodigo : int) : boolean
- + gerarRelatorioPagamento(modelPagamento: ModelPagamento): ArrayList<ModelPagamento>

ControllerProdutos

- + salvarProdutoController(pModelProduto: ModelProduto): int
- + getProdutoController(): ModelProduto
- + getProdutoController(pCodigo : int) : ModelProduto
- + getProdutoCategoriaController(pCodigo: int): ModelProduto
- + getProdutoController(pNome : String) : ModelProduto
- + getListaProdutoController(): ArrayList<ModelProduto>
- + getListaProdutoControllerFiltros(modelProduto: ModelProduto): ArrayList<ModelProduto>
- + atualizarProdutoController(pModelProduto : ModelProduto) : boolean
- + excluirProdutoController(pCodigo : int) : boolean
- + atualizarProdutoQuantidadeController(pModelProduto: ModelProduto): boolean
- + atualizarProdutoQuantidadeUmController(pcodigo : int, pQuantidade : Double) : boolean
- gerarRelatorioProduto(modelProduto : ModelProduto) : ArrayList<ModelProduto>

ControllerSaldoCaixa

- + getSaldoCaixaController(): ModelSaldoCaixa
- + atualizarSaldoCaixaController(pModelSaldoCaixa : ModelSaldoCaixa) : boolean

ControllerUsuario

- + salvarUsuarioController(pModelUsuario : ModelUsuario) : int
- + getUsuarioController(): ModelUsuario
- + getUsuarioController(pCodigo : int) : ModelUsuario
- + getUsuarioController(pLogin : String) : ModelUsuario
- + getUsuarioController(pModelUsuario : ModelUsuario) : boolean
- + getListaUsuarioController(): ArrayList<ModelUsuario>
- + getListaUsuarioControllerFiltros(modelUsuario : ModelUsuario) : ArrayList<ModelUsuario>
- + atualizarUsuarioController(pModelUsuario : ModelUsuario) : boolean
- + excluirUsuarioController(pCodigo : int) : boolean
- + gerarRelatorioUsuario(modelUsuario : ModelUsuario) : ArrayList<ModelUsuario>

ControllerVendas

- + salvarVendasController(pModelVendas : ModelVendas) : int
- + get∨endasController(pCodigo : int) : Model∨endas
- + get Vendas Clienet Controller (pCodigo : int) : Model Vendas
- + get√endas√endedorController(pCodigo : int) : Model√endas
- + getVendasPagamentoController(pCodigo : int) : ModelVendas
- + get Vendas Produto Controller (pCodigo : int) : Model Vendas
- + getVendasProdutoltemController(pCodigo : int) : ModelVendas
- + getVendasController(pValorTotal : Double) : ModelVendas
- + getListaVendasController(): ArrayList<ModelVendas>
- + gerarRelatorioVenda(pCodigo : int) : boolean
- + gerarRelatorioVendas(modelVendas: ModelVendas): ArrayList<ModelVendas>
- + getListaVendasController(pCodigo : int) : ArrayList<ModelVendas>
- + getListaVendasControllerFiltros(modelVenda: ModelVendas): ArrayList<ModelVendas>
- + atualizar∀endasController(pModel∀endas : Model∀endas) : boolean
- + excluirVendasController(pCodigo : int) : boolean
- + salvarVendasProdutosController(modelVendas : ModelVendas) : boolean
- + excluirProdutoVendasController(pCodigo: int): boolean
- + excluirProdutoVendasNotaController(pCodigo : int) : boolean
- + getListaVendasController(pMdelVendas : ModelVendas) : ArrayList<ModelVendas>

ControllerVendedor

- + salvarVendedorController(pModelVendedor : ModelVendedor) : int
- + getVendedorController(): ModelVendedor
- + getVendedorController(pCodigo : int) : ModelVendedor
- + getVendedorBairroController(pCodigo : int) : ModelVendedor
- + getVendedorCidadeController(pCodigo : int) : ModelVendedor
- + getVendedorController(pNome : String) : ModelVendedor
- + getListaVendedorController(): ArrayList<ModelVendedor>
- + getListaVendedorControllerFiltros(modelVendedor: ModelVendedor): ArrayList<ModelVendedor>
- + atualizar Vendedor Controller (pModel Vendedor : Model Vendedor) : boolean
- + excluirVendedorController(pCodigo : int) : boolean
- + gerarRelatorio Vendedor (model Vendedor : Model Vendedor) : ArrayList < Model Vendedor>

DAOBairro

- + salvarBairroDAO(pModelBairro : ModelBairro) : int
- + getBairroDAO(pCodigo : int) : ModelBairro
- + getBairroCidadeDAO(pCodigo: int): ModelBairro
- + getBairroDAO(): ModelBairro
- + getBairroDAO(pNome : String) : ModelBairro
- + getBairroDAO(pCodigo: int, pNome: String, pCodCidade: int): ModelBairro
- + getListaBairroDAO(): ArrayList<ModelBairro>
- + getListaUltimoBairroDAO(): ArrayList<ModelBairro>
- + getListaBairroDAO(modelBairro : ModelBairro) : ArrayList<ModelBairro>
- + atualizarBairroDAO(pModelBairro : ModelBairro) : boolean
- + excluirBairroDAO(pCodigo : int) : boolean

DAOCategoria

- + salvarCategoriaDAO(pModelCategoria : ModelCategoria) : int
- + getCategoriaDAO(): ModelCategoria
- + getCategoriaDAO(pCodigo : int) : ModelCategoria
- + getCategoriaDAO(pNome : String) : ModelCategoria
- getListaCategoriaDAO(): ArrayList<ModelCategoria>
- + getListaCategoriaDAO(modelCategoria : ModelCategoria) : ArrayList<ModelCategoria>
- + atualizarCategoriaDAO(pModelCategoria : ModelCategoria) : boolean
- + excluirCategoriaDAO(pCodigo : int) : boolean

DAOCidade

- + salvarCidadeDAO(pModelCidade : MocelCidade) : int
- + getCidadeDAO(pCodigo : int) : ModelCidade
- + getCidadeDAO(): ModelCidade
- + getCidadeDAO(pNome : String) : ModeCidade
- + getListaCidadeDAO(): ArrayList<ModelCidade>
- + getListaCidadeDAO(modelCidade : ModelCidade) : ArrayList<ModelCidade>
- + atualizarCdadeDAO(pModelCidade : ModelCidade) : boolean
- + excluirCidadeDAO(pCodigo : int) : boolean

DAOCliente

- + salvarClienteDAO(pModelCliente : ModelCliente) : int
- + getClienteDAO(pCodigo : int) : ModelCliente
- + getClienteBairroDAO(pCodigo : int) : ModelCliente
- + getClienteCidadeDAO(pCodigo : int) : ModelCliente
- + getClienteDAO(): ModelCliente
- + getClienteDAO(pNome : String) : ModelCliente
- + getListaClienteDAO(): ArrayList<ModelCliente>
- + getListaClienteDAO(modelCliente : ModelCliente) : ArrayList<ModelCliente>
- + atualizarClienteDAO(pModelCliente : ModelCliente) : boolean
- + excluirClienteDAO(pCodigo : int) : boolean

DAOConta

- + salvarContaPagarDAO(pConta : ModelConta) : int
- + getContaPagarDAO(pCodigo : int) : ModelConta
- + getListaContaPagarDAO(): ArrayList<ModelConta>
- + getListaContaPagarDAO(pStatus : int) : ArrayList<ModelConta>
- + getListaContaReceberDAO(pStatus : int) : ArrayList<ModelConta>
- + getListaContasDAO(pStatus : int) : ArrayList<ModelConta>
- + getListaContasDAO(pModelConta : ModelConta) : ArrayList<ModelConta>
- + atualizarContaPagarDAO(pModelContaPagar : ModelConta) : boolean
- + atualizarContaReceberDAO(pModelContaPagar : ModelConta) : boolean
- + excluirContaPagarDAO(pCodigo : int) : boolean
- + receberContaReceberDAO(pModelConta : ModelConta) : boolean
- + pagarContaPagarDAO(pModelConta : ModelConta) : boolean

DAOContasPagar

- + salvarContasPagarDAO(pModelContasPagar : ModelContasPagar) : int
- + getContasPagarDAO(pCodigo : int) : ModelContasPagar
- + getContasPagarFornecedorDAO(pCodigo : int) : ModelContasPagar
- + getContasPagarPagamentoDAO(pCodigo : int) : ModelContasPagar
- + getListaContasPagarDAO(): ArrayList<ModelContasPagar>
- + getListaContasPagarDAO(modelContasPagar : ModelContasPagar) : ArrayList<ModelContasPagar>
- + getListaContasPagarQuitarDAO(modelContasPagar : ModelContasPagar) : ArrayList<ModelContasPagar>
- + atualizarContasPagarDAO(pModelContasPagar : ModelContasPagar) : boolean
- + excluirContasPagarDAO(pCodigo : int) : boolean

DAOContasReceber

- + salvarContasReceberDAO(pModelContasReceber : ModelContasReceber) : int
- + getContasReceberDAO(pCodigo : int) : ModelContasReceber
- + getContasReceberClienteDAO(pCodigo : int) : ModelContasReceber
- + qetContasReceberPagamentoDAO(pCodigo : int) : ModelContasReceber
- + getContasReceber√endasDAO(pCodigo : int) : ModelContasReceber
- + getListaContasReceberDAO(): ArrayList<ModelContasReceber>
- + getListaContasReceberDAO(modelContasReceber : ModelContasReceber) : ArrayList<ModelContasReceber>
- + getListaContasReceberQuitarDAO(modelContasReceber: ModelContasReceber): ArrayList<ModelContasReceber>
- + atualizarContasReceberDAO(pModelContasReceber: ModelContasReceber): boolean
- + excluirContasReceberDAO(pCodigo : int) : boolean

DAOEntradas

- + salvarEntradasDAO(pModelEntradas : ModelEntradas) : int
- + salvarProdutosEntradasDAO(pModelEntradas : ModelEntradas) : boolean
- + getEntradasDAO(pCodigo : int) : ModelEntradas
- + getEntradasProdutosDAO(pCodigo : int) : ModelEntradas
- + getEntradasProdutosItemDAO(pCodigo : int) : ModelEntradas
- + getEntradasFornecedorDAO(pCodigo : int) : ModelEntradas
- + getEntradasPagamentoDAO(pCodigo : int) : ModelEntradas
- + getEntradasNotaDAO(pNota: int): ModelEntradas
- + getEntradasDAO(pValorTotal : Double) : ModelEntradas
- + getListaEntradasDAO(): ArrayList<ModelEntradas>
- + getListaEntradasDAO(pCodigo : int) : ArrayList<ModelEntradas>
- + getListaEntradasDAO(modelEntradas : ModelEntradas) : ArrayList<ModelEntradas>
- + atualizarEntradasDAO(pModelEntradas : ModelEntradas) : boolean
- + excluirEntradasDAO(pCodigo : int) : boolean
- + excluirProdutoEntradasDAO(pCodigo: int): boolean
- + excluirProdutoEntradasNotaDAO(pCodigo: int): boolean

DAOFluxoCaixa

- + salvarFluxoCaixaDAO(pModelFluxoCaixa : ModelFluxoCaixa) : int
- + getFluxoCaixaDAO(pCodigo : int) : ModelFluxoCaixa
- + getFluxoCaixaVendaDAO(pCodigo : int) : ModelFluxoCaixa
- + getFluxoCaixaDAO(): ModelFluxoCaixa
- + getFluxoCaixaDAO(pOrigem : String) : ModelFluxoCaixa
- + getFluxoCaixaDAO(pCodigo : int, pNota : String, pOrigem : int) : ModelFluxoCaixa
- + getListaFluxoCaixaDAO(): ArrayList<ModelFluxoCaixa>
- + getListaUltimoFluxoCaixaDAO(): ArrayList<ModelFluxoCaixa>
- + getListaFluxoCaixaDAO(modelFluxoCaixa: ModelFluxoCaixa): ArrayList<ModelFluxoCaixa>
- + atualizarFluxoCaixaDAO(pModelFluxoCaixa : ModelFluxoCaixa) : boolean
- + excluirFluxoCaixaDAO(pCodigo : int) : boolean

DAOFornecedor

- + salvarFornecedorDAO(pModelFornecedor : ModelFornecedor) : int
- + getFornecedorDAO(pCodigo : int) : ModelFornecedor
- + getFornecedorBairroDAO(pCodigo : int) : ModelFornecedor
- + getFornecedorCidadeDAO(pCodigo : int) : ModelFornecedor
- + getFornecedorDAO(): ModelFornecedor
- + getFornecedorDAO(pNome : String) : ModelFornecedor
- + getListaFornecedorDAO(): ArrayList<ModelFornecedor>
- + getListaFornecedorDAO(modelFornecedor: ModelFornecedor): ArrayList<ModelFornecedor>
- + atualizarFornecedorDAO(pModelFornecedor : ModelFornecedor) : boolean
- + excluirFornecedorDAO(pCodigo : int) : boolean

DAOPagamento

- + salvarPagamentoDAO(pModelPagamento : ModelPagamento) : int
- + getPagamentoDAO(pCodigo : int) : ModelPagamento
- + getPagamentoDAO(): ModelPagamento
- + getPagamentoDAO(pNome : String) : ModelPagamento
- + getListaPagamentoDAO(): ArrayList<ModelPagamento>
- + getListaPagamentoPrazoDAO(): ArrayList<ModelPagamento>
- + getListaPagamentoDAO(modelPagamento : ModelPagamento) : ArrayList<ModelPagamento>
- + atualizarPagamentoDAO(pModelPagamento : ModelPagamento) : boolean
- + excluirPagamentoDAO(pCodigo : int) : boolean

DAOProdutos

- + salvarProdutosDAO(pModelProduto: ModelProduto): int
- + getProdutosDAO(pCodigo: int): ModelProduto
- + getProdutosCategoriaDAO(pCodigo : int) : ModelProduto
- + getProdutosDAO(): ModelProduto
- + getProdutosDAO(pNome : String) : ModelProduto
- + getListaProdutosDAO(): ArrayList<ModelProduto>
- + getListaProdutosDAO(modelProduto : ModelProduto) : ArrayList<ModelProduto>
- + atualizarProdutosDAO(pModelProduto : ModelProduto) : boolean
- + excluirProdutosDAO(pCodigo : int) : boolean
- + atualizarProdutosQuantidadeDAO(pModelProduto : ModelProduto) : boolean
- + atualizarProdutosQuantidadeDAO(pCodigo : int, pQuantidade : Double) : boolean

DAOSaldoCaixa

- + getSaldoCaixaDAO(): ModelSaldoCaixa
- + atualizarSaldoCaixaDAO(pModelSaldoCaixa : ModelSaldoCaixa) : boolean

DAOUsuario

- + salvarUsuarioDAO(pModelUsuario : ModelUsuario) : int
- + getUsuarioDAO(pCodigo : int) : ModelUsuario
- + getUsuarioDAO(): ModelUsuario
- + getUsuarioDAO(pLogin : String) : ModelUsuario
- + getUsuarioDAO(pModelUsuario : ModelUsuario) : boolean
- + getListaUsuarioDAO(): ArrayList<ModelUsuario>
- + getListaUsuarioDAO(modelUsuario : ModelUsuario) : ArrayList<ModelUsuario>
- + atualizarUsuarioDAO(pModelUsuario : ModelUsuario) : boolean
- + excluirUsuarioDAO(pCodigo : int) : boolean

DAOVendas

- + salvarVendasDAO(pModelVendas : ModelVendas) : int
- + salvarProdutosVendasDAO(pModelVendas : ModelVendas) : boolean
- + get∨endasDAO(pCodigo : int) : Model∨endas
- + getVendasProdutoDAO(pCodigo : int) : ModelVendas
- + getVendasProdutoltemDAO(pCodigo : int) : ModelVendas
- + getVendasClienteDAO(pCodigo : int) : ModelVendas
- + getVendasVendedorDAO(pCodigo : int) : ModelVendas
- + getVendasPagamentoDAO(pCodigo : int) : ModelVendas
- + getVendasDAO(pValorTotal: Double): ModelVendas
- + getListaVendasDAO(): ArrayList<ModelVendas>
- + getListaVendasDAO(pCodigo : int) : ArrayList<ModelVendas>
- + getListaVendasDAO(modelVendas : ModelVendas) : ArrayList<ModelVendas>
- + atualizarVendasDAO(pModelVendas : ModelVendas) : boolean
- + excluir∀endasDAO(pCodigo : int) : boolean
- + excluirProdutoVendasDAO(pCodigo : int) : boolean
- + excluirProduto∀endasNotaDAO(pCodigo : int) : boolean

DAOVendedor

- + salvarVendedorDAO(pModelVendedor : ModelVendedor) : int
- + getVendedorDAO(pCodigo : int) : ModelVendedor
- + getVendedorBairroDAO(pCodigo : int) : ModelVendedor
- + getVendedorCidadeDAO(pCodigo : int) : ModelVendedor
- + getVendedorDAO(): ModelVendedor
- + getVendedorDAO(pNome : String) : ModelVendedor
- + getListaVendedorDAO(): ArrayList<ModelVendedor>
- + getListaVendedorDAO(modelVendedor : ModelVendedor) : ArrayList<ModelVendedor>
- + atualizarVendedorDAO(pModelVendedor : ModelVendedor) : boolean
- + excluir∀endedorDAO(pCodigo : int) : boolean

ModelBairro

- + codigo : int
- + nome : String
- + cod_cidade : int
- + cid_nome : String
- + ModelBairro()
- + setCodigo(pCodigo : int) : void
- + getCodigo(): int
- + setNome(pNome : String) : void
- + getNome(): String
- + getCod_cidade() : int
- + setCod_cidade(cod_cidade : int) : void
- + getCid_nome(): String
- + setCid_nome(cid_nome : String) : void
- + getDtinclusao(): Date
- + setDtinclusao(dtinclusao: Date): void
- + getDtalteracao(): Date
- + setDtalteracao(dtalteracao: Date): void
- + toString(): String

ModelCategoria

- codigo : int
- nome : String
- + getCodigo(): int
- + setCodigo(codigo: int): void
- + getNome(): String
- + setNome(nome : String) : void
- + getDtinclusao(): Date
- + setDtinclusao(dtinclusao: Date): void
- + getDtalteracao(): Date
- + setDtalteracao(dtalteracao: Date): void
- + toString(): String

ModelCidade

- + codigo : int
- + nome : String
- + uf : String
- + ModelCidade()
- + setCodigo(pCodigo: int): void
- + getCodigo(): int
- + setNome(pNome : String) : void
- + getNome(): String
- + getUf(): String
- + setUf(uf : String) : void
- + getDtinclusao(): Date
- + setDtinclusao(dtinclusao : Date) : void
- + getDtalteracao(): Date
- + setDtalteracao(dtalteracao: Date): void
- + toString(): String

ModelCliente + codigo : int + nome : String + endereco : String + numero : String + complemento : String + cod_bairro : int + cod_cidade : int + cli_nome : String + cid_nome : String + bai_nome : String + tipo_doc : String + documento : String + ie : String + cep : String + telefone : String + email : String + situacao : String + ModelCliente() + getCodigo(): int + setCodigo(codigo: int): void + getNome(): String + setNome(nome : String) : void + getEndereco(): String + setEndereco(endereco : String) : void + getNumero(): String + setNumero(numero : String) : void + getComplemento(): String + setComplemento(complemento: String): void + getCod_bairro(): int + setCod_bairro(cod_bairro : int) : void + getCod_cidade(): int + setCod_cidade(cod_cidade: int): void + getCli_nome(): String + setCli_nome(cli_nome : String) : void + getCid_nome(): String + setCid_nome(cid_nome : String) : void + getCli_nome(): String + setCli_nome(cli_nome : String) : void + getCid_nome(): String + setCid_nome(cid_nome : String) : void + getBai_nome(): String + setBai_nome(bai_nome : String) : void + getTipo_doc(): String + setTipo_doc(tipo_doc : String) : void + getDocumento(): String

+ setDocumento(documento: String): void

+ getle(): String + setle(ie: String): void + getCep(): String

+ getEmail(): String

+ getDtinclusao(): Date

+ getDtalteracao(): Date

+ toString(): String

+ setCep(cep : String) : void + getTelefone() : String

+ setEmail(email: String): void + getSituacao(): String

+ setTelefone(telefone : String) : void

+ setSituacao(situacao: String): void

+ setDtinclusao(dtinclusao: Date): void

+ setDtalteracao(dtalteracao: Date): void

ModelConta

- codigo : int

- codigoPessoa : int - descricao : String - tipoPagamento : int - observacao : String - situacao : int

- valor : float - tipoConta : String

+ ModelConta()

+ setCodigo(pCodigo : int) : void

+ getCodigo(): int

+ setCodigoPessoa(pCodigoCliente : int) : void

+ getCodigoPessoa(): int

+ setDescricao(pDescricao : String) : void

+ getDescricao(): String + setData(pData: Date): void

+ getData(): Date

+ set∀encimento(p∀encimento : Date) : void

+ get Vencimento(): Date

+ setPagamento(pPagamento: Date): void

+ getPagamento(): Date

+ setTipoPagamento(pTipoPagamento: int): void

+ getTipoPagamento(): int

+ setObservacao(pObservacao : String) : void

+ getObservacao(): String

+ setSituacao(pSituacao: int): void

+ isSituacao(): int

+ set∀alor(p∀alor : float) : void

+ get∀alor() : float + toString() : String + getTipoConta() : String

+ setTipoConta(tipoConta : String) : void

ModelContasPagar + codigo : int + nota : int + cod_fornecedor : int + contpag_fornecedor : String + cod_pagamento : int + contpag_pagamento : String + valor : Double + valor_pago : Double + valor_aberto : Double + situacao : String + ModelContasPagar() + getCodigo(): int + setCodigo(codigo: int): void + getNota(): int + setNota(nota : int) : void + getData_inclusao(): Date + setData_inclusao(data_inclusao : Date) : void + getData_inclusaoFinal(): Date + setData_inclusaoFinal(data_inclusaoFinal: Date): void + getData_vencimento(): Date + setData_vencimento(data_vencimento: Date): void + getData_vencimentoFinal(): Date + setData_vencimentoFinal(data_vencimentoFinal: Date): void + getData_pagamentoFinal(): Date + setData_pagamentoFinal(data_pagamentoFinal: Date): void + getCod_fornecedor(): int + setCod_fornecedor(cod_fornecedor: int): void + getContpag_fornecedor(): String + setContpag_fornecedor(contpag_fornecedor: String): void + getCod_pagamento(): int + setCod_pagamento(cod_pagamento: int): void + getContpag_pagamento(): String + setContpag_pagamento(contpag_pagamento: String): void + get Valor(): Double + set Valor (valor : Double) : void + getData_pagamento(): Date + setData_pagamento(data_pagamento : Date) : void + get Valor_pago() : Double + setValor_pago(valor_pago : Double) : void + getValor_aberto(): Double + set Valor_aberto(valor_aberto : Double) : void

+ getSituacao(): String

+ toString(): String

+ setSituacao(situacao: String): void

ModelContasReceber

+ codigo : int + pedido : int + cod_cliente : int + contrec_cliente : String + cod_pagamento : int + contrec_pagamento : String

+ valor : Double + valor_pago : Double + valor_aberto : Double + situacao : String

+ ModelContasReceber()

+ getCodigo(): int

+ setCodigo(codigo: int): void

+ getPedido(): int

+ setPedido(pedido: int): void

+ getData_inclusao(): Date

+ setData_inclusao(data_inclusao : Date) : void

+ getData_inclusaoFinal(): Date

+ setData_inclusaoFinal(data_inclusaoFinal: Date): void

+ getData_vencimento(): Date

+ setData_vencimento(data_vencimento: Date): void

+ getData_vencimentoFinal(): Date

+ setData_vencimentoFinal(data_vencimentoFinal: Date): void

+ getData_pagamento(): Date

+ setData_pagamento(data_pagamento : Date) : void

+ getData_pagamentoFinal(): Date

+ setData_pagamentoFinal(data_pagamentoFinal: Date): void

+ getCod_cliente(): int

+ setCod_cliente(cod_cliente : int) : void

+ getContrec_cliente(): String

+ setContrec_cliente(contrec_cliente : String) : void

+ getCod_pagamento() : int

+ setCod_pagamento(cod_pagamento : int) : void

+ getContrec_pagamento(): String

+ setContrec_pagamento(contrec_pagamento : String) : void

+ get∀alor() : Double

+ set∀alor(valor : Double) : void

+ get∀alor_pago() : Double

+ set∀alor_pago(valor_pago : Double) : void

+ get∀alor_aberto() : Double

+ setValor_aberto(valor_aberto : Double) : void

+ getSituacao(): String

ModelControleRelatorio

- controle : int

+ ModelControleRelatorio()

+ getControle(): int

+ setControle(aControle:int): void

+ toString(): String

ModelEntradas + codigo : int + nota : int + fornecedoresCodigo: int + fornecedores: String + tipoPagamento : int + pagamentos : String + acrescimo : Double + desconto : Double + valorTotal : Double + situacao : String + codigopedprod : int + entradascodigo : int + produtosCodigo : int + quantidade : Double + valor : Double + ModelEntradas() + getListamModelEntradas(): ArrayList<ModelEntradas> + setListamModelEntradas(listamModelEntradas : ArrayList<ModelEntradas>) : void + getCodigo(): int + setCodigo(codigo: int): void + getDataEntrada(): Date + setDataEntrada(dataEntrada: Date): void + getFornecedoresCodigo(): int + setFornecedoresCodigo(fornecedoresCodigo: int): void + getFornecedores(): String + setFornecedores(fornecedores: String): void + getTipoPagamento(): int + setTipoPagamento(tipoPagamento: int): void + getPagamentos(): String + setPagamentos(pagamentos: String): void + getAcrescimo(): Double + setAcrescimo(acrescimo : Double) : void + getDesconto(): Double + setDesconto(desconto : Double) : void + set ValorTotal(valorTotal : Double) : void + getSituacao(): String + setSituacao(situacao: String): void + getDataFinal(): Date + setDataFinal(dataFinal: Date): void + getCodigopedprod(): int + setCodigopedprod(codigopedprod: int): void + getEntradascodigo(): int + setEntradascodigo(entradascodigo: int): void + getProdutosCodigo(): int + setProdutosCodigo(produtosCodigo: int): void + getQuantidade(): Double + setQuantidade(quantidade : Double) : void + get∀alor(): Double + set Valor (valor : Double) : void + getNota(): int + setNota(nota : int) : void + toString(): String

ModelFluxoCaixa

+ codigo: int + nota: int + origem: String + tipo: String + destino: String + valor: Double + observacao: String + situacao: String

+ ModelFluxoCaixa()

+ getCodigo(): int

+ setCodigo(codigo: int): void

+ getNota(): int

+ setNota(nota : int) : void

+ getData(): Date

+ setData(data : Date) : void

+ getDataFinal(): Date

+ setDataFinal(dataFinal: Date): void

+ getOrigem(): String

+ setOrigem(origem: String): void

+ getTipo(): String

+ setTipo(tipo : String) : void

+ getDestino(): String

+ setDestino(destino : String) : void

+ get∀alor() : Double

+ set Valor (valor : Double) : void

+ getObservacao(): String

+ setObservacao(observacao: String): void

+ getSituacao(): String

+ setSituacao(situacao : String) : void

+ toString(): String

ModelFornecedor

+ codigo : int + nome : String + endereco : String + numero : String + complemento : String + cod_bairro : int + cod_cidade : int + cli_nome : String + cid_nome : String + bai nome : String + tipo_doc : String + documento : String + ie : String + cep : String + telefone : String + email : String + situacao : String

- + ModelFornecedor()
- + getCodigo(): int
- + setCodigo(codigo: int): void
- + getNome(): String
- + setNome(nome : String) : void
- + getEndereco(): String
- + setEndereco(endereco : String) : void
- + getNumero(): String
- + setNumero(numero : String) : void
- + getComplemento(): String
- + setComplemento(complemento: String): void
- + getCod_bairro(): int
- + setCod_bairro(cod_bairro:int): void
- + getCod_cidade(): int
- + setCod_cidade(cod_cidade: int): void
- + getCli_nome(): String
- + setCli_nome(cli_nome : String) : void
- + getCid_nome(): String
- + setCid_nome(cid_nome : String) : void
- + getCli_nome(): String
- + setCli_nome(cli_nome : String) : void
- + getCid_nome(): String
- + setCid_nome(cid_nome : String) : void
- + getBai_nome(): String
- + setBai_nome(bai_nome : String) : void
- + getTipo_doc(): String
- + setTipo_doc(tipo_doc : String) : void
- + getDocumento(): String
- + setDocumento(documento: String): void
- + getle(): String
- + setle(ie : String) : void
- + getCep(): String
- + setCep(cep : String) : void
- + getTelefone(): String
- + setTelefone(telefone : String) : void
- + getEmail(): String
- + setEmail(email: String): void
- + getSituacao(): String
- + setSituacao(situacao : String) : void
- + getDtinclusao(): Date
- + setDtinclusao(dtinclusao: Date): void
- + getDtalteracao(): Date
- + setDtalteracao(dtalteracao : Date) : void
- + toString(): String

ModelPagamento

+ codigo : int + nome : String + tipo : String + parcelas : int + dias : int

+ ModelPagamento() + getCodigo(): int

+ setCodigo(codigo : int) : void

+ getNome(): String

+ setNome(nome : String) : void

+ getTipo(): String

+ setTipo(tipo : String) : void

+ getParcelas(): int

+ setParcelas(parcelas: int): void

+ getDias(): int

+ setDias(dias : int) : void + getDtinclusao() : Date

+ setDtinclusao(dtinclusao: Date): void

+ getDtalteracao(): Date

+ setDtalteracao(dtalteracao: Date): void

+ toString(): String

ModelProduto

+ codigo : int

+ fornecedoresCodigo: int + categoriaCodigo: int + for_nome: String + cat_nome: String + nome: String + valor: Double + estoque: Double + Situacao: String

+ ModelProduto()

+ getCodigo(): int

+ setCodigo(codigo: int): void + getFornecedoresCodigo(): int

+ setFornecedoresCodigo(fornecedoresCodigo: int): void

+ getCategoriaCodigo(): int

+ setCategoriaCodigo(categoriaCodigo: int): void

+ getFor_nome(): String

+ setFor_nome(for_nome : String) : void

+ getCat_nome(): String

+ setCat_nome(cat_nome : String) : void

+ getNome(): String

+ setNome(nome : String) : void

+ get∀alor() : Double

+ set∀alor(valor : Double) : void

+ getEstoque(): Double

+ setEstoque(estoque : Double) : void

+ getSituacao(): String

+ setSituacao(Situacao : String) : void

+ getDtinclusao() : Date

+ setDtinclusao(dtinclusao : Date) : void

+ getDtalteracao(): Date

+ setDtalteracao(dtalteracao: Date): void

+ getListaModelProduto(): ArrayList<ModelProduto>

+ setListaModelProduto(listaModelProduto: ArrayList<ModelProduto>): void

+ toString(): String

ModelSaldoCaixa

- + saldo : Double
- + ModelSaldoCaixa()
- + getSaldo(): Double
- + setSaldo(saldo : Double) : void
- + toString(): String

ModelSessaoContasPagar

- <u>+ nota : int</u>
- + fornecedor: int
- + fornec : String
- + pagamento : int
- + valor : Double
- + origem : String
- + tipo : String
- + destino : String
- + controle : int
- + integrado : int
- + ModelSessaoContasPagar()
- + getNota(): int
- + setNota(aNota : int) : void
- + getFornecedor(): int
- + setFornecedor(aFornecedor: int): void
- + getFornec(): String
- + setFornec(aFornec : String) : void
- + getData(): Date
- + setData(aData : Date) : void
- + getPagamento(): int
- + setPagamento(aPagamento:int): void
- + get∀alor() : Double
- + set∀alor(a∀alor : Double) : void
- + getOrigem(): String
- + setOrigem(aOrigem : String) : void
- + getTipo(): String
- + setTipo(aTipo : String) : void
- + getDestino(): String
- + setDestino(aDestino : String) : void
- + getControle(): int
- + setControle(aControle:int): void
- + getIntegrado(): int
- + setIntegrado(aIntegrado: int): void

ModelSessaoContasReceber

- + pedido : int
- + cliente : int
- + cli : String
- + pagamento : int
- + valor : Double
- + origem : String
- + tipo : String
- + destino : String
- + controle : int
- + integrado : int
- + getOrigem(): String
- + setOrigem(aOrigem : String) : void
- + getTipo(): String
- + setTipo(aTipo : String) : void
- + getDestino(): String
- + setDestino(aDestino : String) : void
- + getCli(): String
- + setCli(aCli : String) : void
- + getControle(): int
- + setControle(aControle: int): void
- + ModelSessaoContasReceber()
- + getPedido(): int
- + setPedido(aPedido: int): void
- + getCliente(): int
- + setCliente(aCliente : int) : void
- + getData(): Date
- + setData(aData : Date) : void
- + getPagamento(): int
- + setPagamento(aPagamento : int) : void
- + get∀alor(): Double
- + set∀alor(a∀alor : Double) : void
- <u>+ getIntegrado() : int</u>
- + setIntegrado(aIntegrado : int) : void
- + toString(): String

ModelSessaoEntradas

- + nota : int
- + fornecedor : int
- + fornec : String
- + pagamento : int
- + valor : Double
- + origem : String
- + tipo : String
- + destino : String
- + controle : int
- + integrado : int
- + getOrigem(): String
- + setOrigem(aOrigem : String) : void
- + getTipo(): String
- + setTipo(aTipo : String) : void
- + getDestino(): String
- + setDestino(aDestino: String): void
- + getControle(): int
- + setControle(aControle:int): void
- + getFornec(): String
- + setFornec(aFornec : String) : void
- + ModelSessaoEntradas()
- + getNota(): int
- + setNota(aNota: int): void
- + getFornecedor(): int
- + setFornecedor(aFornecedor: int): void
- + getData(): Date
- + setData(aData : Date) : void
- + getPagamento(): int
- + setPagamento(aPagamento: int): void
- + get∀alor(): Double
- + set∀alor(a∀alor : Double) : void
- + getIntegrado() : int
- + setIntegrado(aIntegrado: int): void
- + toString(): String

ModelSessaoUsuario

- + codigo : int
- + login : String
- + nome : String
- + perfil : String

ModelSessaoVendas

- + pedido : int
- + cliente : int
- + cli : String
- + pagamento : int
- + valor : Double
- + origem : String
- + tipo : String
- + destino : String
- + controle : int
- + integrado : int
- + getOrigem(): String
- + setOrigem(aOrigem : String) : void
- + getTipo(): String
- + setTipo(aTipo : String) : void
- + getDestino(): String
- + setDestino(aDestino : String) : void
- + getCli(): String
- + setCli(aCli : String) : void
- + getControle(): int
- + setControle(aControle:int): void
- + ModelSessaoVendas()
- + getPedido(): int
- + setPedido(aPedido: int): void
- + getCliente(): int
- + setCliente(aCliente : int) : void
- + getData(): Date
- + setData(aData : Date) : void
- + getPagamento() : int
- + setPagamento(aPagamento: int): void
- + get∀alor() : Double
- + set∀alor(a∀alor : Double) : void
- + getIntegrado(): int
- + setIntegrado(aIntegrado: int): void
- + toString(): String

ModelUsuario

- codigo : int
- nome : String
- perfil : String
- login : String
- senha : String
- + getCodigo(): int
- + setCodigo(codigo: int): void
- + getNome(): String
- + setNome(nome : String) : void
- + getPerfil(): String
- + setPerfil(perfil: String): void
- + getLogin(): String
- + setLogin(login : String) : void
- + getSenha(): String
- + setSenha(senha : String) : void
- + getDtinclusao(): Date
- + setDtinclusao(dtinclusao: Date): void
- + getDtalteracao(): Date
- + setDtalteracao(dtalteracao: Date): void
- + toString(): String

ModelVendas + codigo : int + clientesCodigo: int + clientes : String + vendedoresCodigo: int + vendedores : String + tipoPagamento : int + pagamentos : String + acrescimo : Double + desconto : Double + valorTotal : Double + situacao : String + codigopedprod: int + vendascodigo : int + produtosCodigo: int + quantidade : Double + valor : Double + ModelVendas() + getListamModelVendas(): ArrayList<ModelVendas> + setListamModelVendas(listamModelVendas : ArrayList<ModelVendas>) : void + getCodigo(): int + setCodigo(codigo: int): void + getDataVenda(): Date + setDataVenda(dataVenda: Date): void + getClientesCodigo(): int + setClientesCodigo(clientesCodigo: int): void + getClientes(): String + setClientes(clientes: String): void + getVendedoresCodigo(): int + setVendedoresCodigo(vendedoresCodigo: int): void + getVendedores(): String + setVendedores(vendedores : String) : void + getTipoPagamento(): int + setTipoPagamento(tipoPagamento: int): void + getPagamentos(): String + setPagamentos(pagamentos : String) : void + getAcrescimo(): Double + setAcrescimo(acrescimo : Double) : void + getDesconto(): Double + setDesconto(desconto : Double) : void + get ValorTotal(): Double + set ValorTotal(valorTotal : Double) : void + getSituacao(): String + setSituacao(situacao: String): void + getDataFinal(): Date + setDataFinal(dataFinal: Date): void + getCodigopedprod(): int + setCodigopedprod(codigopedprod: int): void + get Vendascodigo(): int + set∀endascodigo(vendascodigo : int) : void + getProdutosCodigo(): int + setProdutosCodigo(produtosCodigo: int): void + getQuantidade(): Double + setQuantidade(quantidade : Double) : void + get∀alor(): Double + set Valor (valor : Double) : void + toString(): String

ModelVendedor + codigo : int + nome : String + endereco : String + numero : String + complemento : String + cod_bairro : int + cod_cidade : int + cli_nome : String + cid_nome : String + bai nome : String + tipo_doc : String + documento : String + ie : String + cep : String + telefone : String + email: String + situacao : String + ModelVendedor() + getCodigo(): int + setCodigo(codigo: int): void + getNome(): String + setNome(nome : String) : void + getEndereco(): String + setEndereco(endereco : String) : void + getNumero(): String + setNumero(numero : String) : void

+ setComplemento(complemento: String): void

+ setCod_bairro(cod_bairro : int) : void

+ setCod_cidade(cod_cidade: int): void

+ setCli_nome(cli_nome : String) : void

+ setCid_nome(cid_nome : String) : void

+ setCli_nome(cli_nome : String) : void

+ setCid_nome(cid_nome : String) : void

+ setBai_nome(bai_nome : String) : void

+ setTipo_doc(tipo_doc : String) : void

+ setTelefone(telefone : String) : void

+ setSituacao(situacao: String): void

+ setDtinclusao(dtinclusao: Date): void

+ setDtalteracao(dtalteracao: Date): void

+ setDocumento(documento: String): void

+ getComplemento(): String

+ getCod_bairro(): int

+ getCod_cidade(): int

+ getCli_nome(): String

+ getCid_nome(): String

+ getCli_nome(): String

+ getCid_nome(): String

+ getBai_nome(): String

+ getTipo_doc(): String

+ getDocumento(): String

+ setCep(cep : String) : void + getTelefone() : String

+ setEmail(email: String): void + getSituacao(): String

+ getle(): String + setle(ie: String): void + getCep(): String

+ getEmail(): String

+ getDtinclusao(): Date

+ getDtalteracao(): Date

+ toString(): String

DAORelatorios

Datas

- + addBarras(pString : String) : String
- + trocarTracos(pString : String) : String
- + addDias(pQteDias : int, pDate : Date) : Date
- + adddMes(dataAtual : Date, quantidadeMes : int) : Date
- + diasEntreDatas(pDataInicio : Date, pDataFim : Date) : int
- + retornarDataHora(): String
- + converterDataStringParaDate(data: String): Date
- + converterDataStringParaDateUS(data: String): Date
- + converterDataParaDateUS(pData : Date) : Date
- + formatarData(pData: Date): String

Filtros

- ~ where : String = ""
- + addFiltro(campo : String, valor1 : Date, valor2 : Date) : void
- + addFiltro(campo : String, valor : String) : void
- + addFiltro(campo : String, valor : double) : void
- + addFiltro(campo : String, valor : int) : void
- + addFiltro(campo : String, valor : Date) : void
- + getWhere(): String

LogoSistema

+ setIconImage(o : JFrame) : void

Mascaras

- + converterVirgulaParaPonto(pString : String) : String
- + converterPontoPraVirgula(pString : String) : String
- + converter∀irgulaParaPontoReturnFloat(pString : String) : float
- + removerPontos(pString: String): String
- + removerMascara(pString : String) : String
- + addPonto(pString : String) : String
- + truncar3Casas(pValor : double) : double
- + converteInteiro(pString : int) : int
- + converteArredondar3Casas(pDouble : double) : double
- + arredondamentoComPonto(p∨alor : float) : float

MascValores

- + NUMERO : String = "#,##0.00; -#,##0.00"
- + REAL : String = "R\$ #,##0.00;R\$ -#,##0.00"
- + PORCENTAGEM : String = "#,##0.00'%";-#,##0.00'%"
- dValue : double = 0.0
- old∀alue : String = "0.00"
- + ERRO : String = "Valor inválido"
- + MascValores(formato : String)
- + Masc∀alores()
- + setDf(df : DecimalFormat) : void
- + getDf() : DecimalFormat
- applyActions(): void
- + thisFocusLost(evt : FocusEvent) : void
- + thisFocusGained(fe : FocusEvent) : void
- set∀alue(value : String) : void
- + set∀alue(value : Object) : void
- + show∀alue(s : String) : void
- + getDouble∀alue() : double
- + get∀alue() : Object
- error(): void
- normalText() : void
- + setText(valor : String) : void
- + arredondar(d : double) : double

MySQLBackup

- VERSION : String = "5.6.2"
- SEPARATOR : String = File.separator
- MYSQL_PATH: String = null
- PRESENTATION : String = "=======...
- DATABASES : String = "msdb"
- + dbList : List<String> = new ArrayList<String>()
- + MySQLBackup()
- + backup(arquivo : String) : void

ServerConfig

server : String = null # user : String = null # password : String = null # dump : String = null

+ ServerConfig()

TiraAcentos

+ TiraAcentos(passa : String) : String

ViewBairro

~ tipoCadastro : String

~ aba : int = 1 ~ cidade : String

- + ViewBairro()
- initComponents(): void
- + MudarLogo(): void
- btCancelarActionPerformed(evt : ActionEvent) : void
- btAlterarActionPerformed(evt : ActionEvent) : void
- btSalvarActionPerformed(evt : ActionEvent) : void
- alteraBairro() : boolean
- salvarBairro() : boolean
- recuperarBairro(): void
- recuperaAlterar(): void
- recuperalncluir(): void
- carregarBairros(): voidnovoBairro(): void
- Controlesalterar(): void
- ControlesPesquisar(): void
- # desabilitarCampos(): void
- listarCidade(): void
- btNovoActionPerformed(evt : ActionEvent) : void
- btPesquisarBairroActionPerformed(evt : ActionEvent) : void
- btExcluirActionPerformed(evt : ActionEvent) : void
- txtNomeFocusLost(evt : FocusEvent) : void
- cbCidadeActionPerformed(evt : ActionEvent) : void
- txtNomeKeyPressed(evt : KeyEvent) : void
- txtNomeActionPerformed(evt : ActionEvent) : void
- + main(args : String[]) : void

ViewCategoria

~ tipoCadastro : String

~ aba : int = 1

+ ViewCategoria()

- initComponents(): void

+ MudarLogo(): void

- btCancelarActionPerformed(evt : ActionEvent) : void

- btAlterarActionPerformed(evt : ActionEvent) : void

- btSalvarActionPerformed(evt : ActionEvent) : void

- alteraCategoria(): boolean

- salvarCategoria(): boolean

recuperarCategoria(): void

- recuperaAlterar(): void

recuperalncluir(): void

carregarCategorias(): void

novoCategoria(): void

- Controlesalterar(): void

- ControlesPesquisar(): void

desabilitarCampos(): void

- btNovoActionPerformed(evt : ActionEvent) : void

- btPesquisarCategoriaActionPerformed(evt : ActionEvent) : void

- btExcluirActionPerformed(evt : ActionEvent) : void

txtNomeFocusLost(evt : FocusEvent) : void

+ main(args : String[]) : void

ViewCidade

~ tipoCadastro : String

~ aba : int = 1

+ ViewCidade()

- initComponents(): void

+ MudarLogo(): void

- btCancelarActionPerformed(evt : ActionEvent) : void

btAlterarActionPerformed(evt : ActionEvent) : void

btSalvarActionPerformed(evt : ActionEvent) : void

- alteraCidade() : boolean

- salvarCidade() : boolean

recuperarCidade(): void

recuperaAlterar(): void

- recuperalncluir(): void

carregarCidades(): void

- novoCidade(): void

- Controlesalterar(): void

- ControlesPesquisar(): void

desabilitarCampos(): void

- btNovoActionPerformed(evt : ActionEvent) : void

- btPesquisarCidadeActionPerformed(evt : ActionEvent) : void

- btExcluirActionPerformed(evt : ActionEvent) : void

- txtNomeFocusLost(evt : FocusEvent) : void

txtNomeActionPerformed(evt : ActionEvent) : void

- txtNomeKeyPressed(evt : KeyEvent) : void

ViewCliente

~ tipoCadastro : String ~ aba : int = 1 ~ bairro : String ~ cidade : String

+ ViewCliente()

initComponents(): void+ MudarLogo(): void

btCancelarActionPerformed(evt : ActionEvent) : void
 btAlterarActionPerformed(evt : ActionEvent) : void
 btSalvarActionPerformed(evt : ActionEvent) : void

- atteraCliente(): boolean - salvarCliente(): boolean - recuperarCliente(): void - recuperaAtterar(): void - recuperalncluir(): void - carregarClientes(): void

- novoCliente() : void - Controlesalterar() : void

ControlesPesquisar(): void # desabilitarCampos(): void

- listarCidade(): void

- listarBairro(): void

- btNovoActionPerformed(evt : ActionEvent) : void

- btPesquisarClienteActionPerformed(evt : ActionEvent) : void

- btExcluirActionPerformed(evt : ActionEvent) : void

- txtNomeFocusLost(evt : FocusEvent) : void

- cbCidadeActionPerformed(evt : ActionEvent) : void

- txtEnderecoFocusLost(evt : FocusEvent) : void

- cbBairroActionPerformed(evt : ActionEvent) : void

- txtEmailFocusLost(evt : FocusEvent) : void

- txtEnderecoActionPerformed(evt : ActionEvent) : void

- txtTelefoneActionPerformed(evt : ActionEvent) : void

- cbTipoDocActionPerformed(evt : ActionEvent) : void

txtDocumentoFocusLost(evt : FocusEvent) : void

txtleFocusLost(evt : FocusEvent) : void

- txtEmailActionPerformed(evt : ActionEvent) : void

txtDocumentoActionPerformed(evt : ActionEvent) : void

- txtleActionPerformed(evt : ActionEvent) : void

- cbTipoDocFocusLost(evt : FocusEvent) : void

- cbBairroMouseClicked(evt : MouseEvent) : void

- cbCidadeMouseClicked(evt : MouseEvent) : void

- cbBairroFocusGained(evt : FocusEvent) : void

- cbCidadeFocusGained(evt : FocusEvent) : void

- txtCepActionPerformed(evt : ActionEvent) : void

- cbSituacaoFocusLost(evt : FocusEvent) : void

- cbSituacaoActionPerformed(evt : ActionEvent) : void

- txtNomeKeyPressed(evt : KeyEvent) : void

txtCepKeyPressed(evt : KeyEvent) : void

- txtEnderecoKeyPressed(evt : KeyEvent) : void

- txtNumeroKeyPressed(evt : KeyEvent) : void

- txtComplementoKeyPressed(evt : KeyEvent) : void

- cbBairroKeyPressed(evt : KeyEvent) : void

- cbCidadeKeyPressed(evt : KeyEvent) : void

- cbTipoDocKeyPressed(evt : KeyEvent) : void

- txtDocumentoKeyPressed(evt : KeyEvent) : void

- txtleKeyPressed(evt : KeyEvent) : void

- txtTelefoneKeyPressed(evt : KeyEvent) : void

txtEmailKeyPressed(evt : KeyEvent) : void

txtNomeActionPerformed(evt : ActionEvent) : void

ViewContasPagar ~ tipoCadastro : String ~ aba : int = 1 ~ fornecedor: String ~ pagamento : String ~ pNota : int ~ pFornecedor : int ~ pPagamento : int ~ p∀alor : Double ~ plntegrado : int + retornarNota(): int + retornarFornecedor(): int + retornarPagamento(): int + retornarData(): Date + retornar Valor(): Double + retornarIntegrado(): int + ViewContasPagar() + MudarLogo(): void - configurar(): void - initComponents(): void - btCancelarActionPerformed(evt : ActionEvent) : void # alterarContasPagar(): void - btAlterarActionPerformed(evt : ActionEvent) : void - btSalvarActionPerformed(evt : ActionEvent) : void - alteraContasPagar(): boolean - salvarContasPagar(): boolean - recuperarContasPagar(): void - recuperaAlterar(): void - recuperalncluir(): void - carregarContasPagar(): void - novoContasPagar(): void - Controlesalterar(): void - ControlesPesquisar(): void # verificarCampos(): boolean # desabilitarCampos(): void - listarFornecedor(): void listarPagamento(): void - btNovoActionPerformed(evt : ActionEvent) : void - btPesquisarContasPagarActionPerformed(evt : ActionEvent) : void - btExcluirActionPerformed(evt : ActionEvent) : void - txt∀alorFocusLost(evt : FocusEvent) : void - cbFornecedorActionPerformed(evt : ActionEvent) : void - cbPagamentoActionPerformed(evt : ActionEvent) : void - txtValorPagoFocusLost(evt : FocusEvent) : void - txt∀alorAbertoFocusLost(evt : FocusEvent) : void - txtValorActionPerformed(evt : ActionEvent) : void - txtNotaFocusLost(evt : FocusEvent) : void txtNotaKeyPressed(evt : KeyEvent) : void - cbFornecedorKeyPressed(evt : KeyEvent) : void - cbPagamentoKeyPressed(evt : KeyEvent) : void - cbSituacaoKeyPressed(evt : KeyEvent) : void - jdlnclusaoKeyPressed(evt : KeyEvent) : void - jdVencimentoKeyPressed(evt : KeyEvent) : void - cbSituacaoKeyReleased(evt : KeyEvent) : void

```
ViewContasReceber
~ tipoCadastro : String
~ aba : int = 1
~ cliente : String
~ pagamento : String
~ pPedido : int
~ pCliente : int
~ pPagamento : int
~ p∀alor : Double
~ plntegrado : int
+ retornarPedido(): int
+ retornarCliente(): int
+ retornarPagamento(): int
+ retornarData(): Date
+ retornar Valor(): Double
+ retornarIntegrado(): int
+ ViewContasReceber()
+ MudarLogo(): void
configurar(): void
- initComponents(): void
- btCancelarActionPerformed(evt : ActionEvent) : void
# alterarContasReceber(): void
- btAlterarActionPerformed(evt : ActionEvent) : void
- btSalvarActionPerformed(evt : ActionEvent) : void
- alteraContasReceber(): boolean
- salvarContasReceber(): boolean
- recuperarContasReceber(): void
- recuperaAlterar(): void
- recuperalncluir(): void
- carregarContasReceber(): void
novoContasReceber(): void
- Controlesalterar(): void
ControlesPesquisar(): void
# verificarCampos(): boolean
# desabilitarCampos(): void
- listarCliente(): void
listarPagamento(): void
- btNovoActionPerformed(evt : ActionEvent) : void
- btPesquisarContasReceberActionPerformed(evt : ActionEvent) : void
- btExcluirActionPerformed(evt : ActionEvent) : void
- txt∀alorFocusLost(evt : FocusEvent) : void
- cbClienteActionPerformed(evt : ActionEvent) : void
- cbPagamentoActionPerformed(evt : ActionEvent) : void
- txtValorPagoFocusLost(evt : FocusEvent) : void
- txt∀alorAbertoFocusLost(evt : FocusEvent) : void
- txtValorActionPerformed(evt : ActionEvent) : void
- txtPedidoFocusLost(evt : FocusEvent) : void
- txtPedidoKeyPressed(evt : KeyEvent) : void

    cbClienteKeyPressed(evt : KeyEvent) : void

- cbPagamentoKeyPressed(evt : KeyEvent) : void
- cbSituacaoKeyPressed(evt : KeyEvent) : void
- jdlnclusaoKeyPressed(evt : KeyEvent) : void
- jdVencimentoKeyPressed(evt : KeyEvent) : void

    cbSituacaoKeyReleased(evt : KeyEvent) : void
```

ViewDevolucaoVenda

~ nrvenda : int

~ tipoCadastro : String

~ aba : int = 1

+ ViewDevolucaoVenda()

+ MudarLogo(): void

- listarClientes(): void

- listarClientesPesquisa(): void

- listar∀endedores() : void

- listar Vendedores Pesquisa(): void

- listarPagamento(): void

listarPagamentosPesquisa(): void

+ set∀iewConsultar∀endas() : void

- initComponents(): void

desabilitarCampos(): void

- btSalvarActionPerformed(evt : ActionEvent) : void

- cadastrar∀enda() : boolean

- btGridActionPerformed(evt : ActionEvent) : void

- btPesquisarActionPerformed(evt : ActionEvent) : void

- btNovoActionPerformed(evt : ActionEvent) : void

txtDescontoKeyReleased(evt : KeyEvent) : void

- txtDescontoFocusLost(evt : FocusEvent) : void

- cbClientesActionPerformed(evt : ActionEvent) : void

txtDescontoActionPerformed(evt : ActionEvent) : void

- txtValorTotalActionPerformed(evt : ActionEvent) : void

- cbPagamentoActionPerformed(evt : ActionEvent) : void

- cbVendedoresActionPerformed(evt : ActionEvent) : void

- txtAcrescimoActionPerformed(evt : ActionEvent) : void

- txtAcrescimoFocusLost(evt : FocusEvent) : void

- txtAcrescimoKeyReleased(evt : KeyEvent) : void

- pcbClientesActionPerformed(evt : ActionEvent) : void

- pcbPagamentosActionPerformed(evt : ActionEvent) : void

- pcoPagamentosActionPerformed(evt : ActionEvent) : void

- cbClientesKeyPressed(evt : KeyEvent) : void

- cb√endedoresKeyPressed(evt : KeyEvent) : void

- cbPagamentoKeyPressed(evt : KeyEvent) : void

pcbVendedoresActionPerformed(evt : ActionEvent) : void

- jdlnicialKeyPressed(evt : KeyEvent) : void

- jdFinalKeyPressed(evt : KeyEvent) : void

- pcbSituacaoActionPerformed(evt : ActionEvent) : void

- recuperar∀enda() : boolean

- carregar∀endas() : void

limpafiltros(): void

- novo∨enda(): void

- habilitarCampos(): void

- somaEAtualiza ValorTotal(): Double

- alterar∀enda() : boolean

retornarProdutoAoEstoque(): void

ViewEmpresa

~ tipoCadastro : String ~ aba : int = 1

~ bairro : String ~ cidade : String

+ ViewEmpresa()

- initComponents(): void+ MudarLogo(): void# habilitarCampos(): void

btCancelarActionPerformed(evt : ActionEvent) : void

btSalvarActionPerformed(evt : ActionEvent) : void

atteraCliente(): booleansalvarCliente(): boolean

- recuperarCliente(): void

recuperaAlterar(): void
 recuperalncluir(): void
 carregarClientes(): void

novoCliente(): voidControlesalterar(): void

- listarCidade(): void

- listarBairro() : void

- btNovoActionPerformed(evt : ActionEvent) : void

txtNomeFocusLost(evt : FocusEvent) : void

- cbCidadeActionPerformed(evt : ActionEvent) : void

- txtEnderecoFocusLost(evt : FocusEvent) : void

- cbBairroActionPerformed(evt : ActionEvent) : void

- txtEmailFocusLost(evt : FocusEvent) : void

- txtEnderecoActionPerformed(evt : ActionEvent) : void

- txtTelefoneActionPerformed(evt : ActionEvent) : void

- cbTipoDocActionPerformed(evt : ActionEvent) : void

- txtDocumentoFocusLost(evt : FocusEvent) : void

- txtleFocusLost(evt : FocusEvent) : void

txtEmailActionPerformed(evt : ActionEvent) : void

- txtDocumentoActionPerformed(evt : ActionEvent) : void

- txtleActionPerformed(evt : ActionEvent) : void

- cbTipoDocFocusLost(evt : FocusEvent) : void

- cbBairroMouseClicked(evt : MouseEvent) : void

- cbCidadeMouseClicked(evt : MouseEvent) : void

- cbBairroFocusGained(evt : FocusEvent) : void

- cbCidadeFocusGained(evt : FocusEvent) : void

- txtCepActionPerformed(evt : ActionEvent) : void

- txtNomeKeyPressed(evt : KeyEvent) : void

- txtCepKeyPressed(evt : KeyEvent) : void

+ the description of the territory of th

- txtEnderecoKeyPressed(evt : KeyEvent) : void

- txtNumeroKeyPressed(evt : KeyEvent) : void

- txtComplementoKeyPressed(evt : KeyEvent) : void

- cbBairroKeyPressed(evt : KeyEvent) : void

- cbCidadeKeyPressed(evt : KeyEvent) : void

- cbTipoDocKeyPressed(evt : KeyEvent) : void

- txtDocumentoKeyPressed(evt : KeyEvent) : void

txtleKeyPressed(evt : KeyEvent) : void

- txtTelefoneKeyPressed(evt : KeyEvent) : void

- txtEmailKeyPressed(evt : KeyEvent) : void

- txtNomeActionPerformed(evt : ActionEvent) : void

ViewEntrada

~ nrEntrada : int ~ tipoCadastro : String

~ aba : int = 1

+ ViewEntrada()

+ MudarLogo(): void

- listarFornecedores(): void

- listarFornecedoresPesquisa(): void

- listarPagamento(): void

- listarPagamentosPesquisa(): void

- listarProdutos(): void

+ setViewConsultarEntradas(): void

- initComponents(): void

- btCancelarActionPerformed(evt : ActionEvent) : void

desabilitarCampos(): void

- btSalvarActionPerformed(evt : ActionEvent) : void

- cadastrarEntrada(): boolean

- btGridActionPerformed(evt : ActionEvent) : void

- btPesquisarActionPerformed(evt : ActionEvent) : void

- btExcluirActionPerformed(evt : ActionEvent) : void

- btNovoActionPerformed(evt : ActionEvent) : void

- btRetirarActionPerformed(evt : ActionEvent) : void

- txtDescontoKeyReleased(evt : KeyEvent) : void

- txtDescontoFocusLost(evt : FocusEvent) : void

- cbFornecedoresActionPerformed(evt : ActionEvent) : void

- btIncluirProdutoActionPerformed(evt : ActionEvent) : void

- txtQuantidadeActionPerformed(evt : ActionEvent) : void

- txtDescontoActionPerformed(evt : ActionEvent) : void

- txt∀alorTotalActionPerformed(evt : ActionEvent) : void

- cbPagamentoActionPerformed(evt : ActionEvent) : void

- txtAcrescimoActionPerformed(evt : ActionEvent) : void

- txtAcrescimoFocusLost(evt : FocusEvent) : void

- txtAcrescimoKeyReleased(evt : KeyEvent) : void

- pcbFornecedoresActionPerformed(evt : ActionEvent) : void

- pcbPagamentosActionPerformed(evt : ActionEvent) : void

- cbFornecedoresKeyPressed(evt : KeyEvent) : void

- cbPagamentoKeyPressed(evt : KeyEvent) : void

- cbProdutosKeyPressed(evt : KeyEvent) : void

- txtQuantidadeKeyPressed(evt : KeyEvent) : void

- cbProdutosActionPerformed(evt : ActionEvent) : void

- btlncluirProdutoKeyPressed(evt : KeyEvent) : void

- jdlnicialKeyPressed(evt : KeyEvent) : void

- jdFinalKeyPressed(evt : KeyEvent) : void

- pcbSituacaoActionPerformed(evt : ActionEvent) : void

- txtNotaKeyPressed(evt : KeyEvent) : void

- recuperarEntrada(): boolean

- carregarEntradas() : void

- limpafiltros(): void

- novoEntrada(): void

- habilitarCampos(): void

- incluirProduto(): void

- somaEAtualizaValorTotal(): Double

- alterarEntrada() : boolean

- retornarProdutoAoEstoque(): void

ViewFluxoCaixa ~ tipoCadastro : String ~ aba : int = 1 ~ cidade : String ~ pPedido : int ~ pNota : int ~ pOrigem: String ~ pTipo : String ~ pDestino : String ~ p∀alor : Double ~ pCliente : String ~ pFornecedor : String ~ pControle Vendas : int = 0 ~ pControleEntradas : int = 0 ~ pControleReceber : int = 0 ~ pControlePagar : int = 0 ~ plntegrado : int = 0 + retornarPedido(): int + retornarNota(): int + retornarNotaReceber(): int + retornarNotaPagar(): int + retornarDataVenda(): Date + retornarDataEntrada(): Date + retornarDataRecebimento(): Date + retornarDataPagamento(): Date + retornarOrigemVenda(): String + retornarOrigemEntrada(): String + retornarOrigemReceber(): String + retornarOrigemPagar(): String + retornarTipoVenda(): String + retornarTipoEntrada(): String + retornarTipoReceber(): String + retornarTipoPagar(): String + retornarDestino∀enda() : String + retornarDestinoEntrada(): String + retornarDestinoReceber(): String + retornarDestinoPagar(): String + retornar∀alor∀enda() : Double + retornar∀alorEntrada() : Double + retornar Valor Receber(): Double + retornar Valor Pagar(): Double + retornarCliente(): String + retornarFornecedor(): String + retornarClienteReceber(): String + retornarFornecedorPagar(): String + retornarControleVenda(): int + retornarControleEntrada(): int + retornarControleReceber(): int + retornarControlePagar(): int + retornarIntegrado∀enda() : int + retornarIntegradoEntrada(): int + retornarIntegradoReceber(): int + retornarIntegradoPagar(): int + ViewFluxoCaixa() - initComponents(): void + MudarLogo(): void camposIntegracao(): void - configurar(): void - btCancelarActionPerformed(evt : ActionEvent) : void btAlterarActionPerformed(evt : ActionEvent) : void btSalvarActionPerformed(evt : ActionEvent) : void - alteraFluxoCaixa(): boolean - incluir(): void salvarFluxoCaixa(): boolean recuperarFluxoCaixa(): void - recuperaAlterar(): void recuperalncluir(): void carregarFluxoCaixa(): void - novoFluxoCaixa(): void

Controlesalterar(): void
 ControlesPesquisar(): void

ViewFornecedor

~ tipoCadastro : String ~ aba : int = 1 ~ bairro : String ~ cidade : String

+ ViewFornecedor()
- initComponents(): void
+ MudarLogo(): void

btCancelarActionPerformed(evt : ActionEvent) : void
 btAlterarActionPerformed(evt : ActionEvent) : void
 btSalvarActionPerformed(evt : ActionEvent) : void

- alteraFornecedor(): boolean - salvarFornecedor(): boolean - recuperarFornecedor(): void - recuperaAlterar(): void - recuperaIncluir(): void

carregarFornecedores(): void
 novoFornecedor(): void
 Controlesalterar(): void
 ControlesPesquisar(): void

desabilitarCampos(): void

listarCidade(): voidlistarBairro(): void

- btNovoActionPerformed(evt : ActionEvent) : void

- btPesquisarFornecedorActionPerformed(evt : ActionEvent) : void

- btExcluirActionPerformed(evt : ActionEvent) : void

txtNomeFocusLost(evt : FocusEvent) : void

- cbCidadeActionPerformed(evt : ActionEvent) : void

txtEnderecoFocusLost(evt : FocusEvent) : void

- cbBairroActionPerformed(evt : ActionEvent) : void

txtEmailFocusLost(evt : FocusEvent) : void

- txtEnderecoActionPerformed(evt : ActionEvent) : void

- txtTelefoneActionPerformed(evt : ActionEvent) : void

- cbTipoDocActionPerformed(evt : ActionEvent) : void

- txtDocumentoFocusLost(evt : FocusEvent) : void

- txtleFocusLost(evt : FocusEvent) : void

- txtEmailActionPerformed(evt : ActionEvent) : void

- txtEnderecoActionPerformed(evt : ActionEvent) : void

- txtTelefoneActionPerformed(evt : ActionEvent) : void

cbTipoDocActionPerformed(evt : ActionEvent) : void

- txtDocumentoFocusLost(evt : FocusEvent) : void

- txtleFocusLost(evt : FocusEvent) : void

- txtEmailActionPerformed(evt : ActionEvent) : void

- txtDocumentoActionPerformed(evt : ActionEvent) : void

- txtleActionPerformed(evt : ActionEvent) : void

- cbTipoDocFocusLost(evt : FocusEvent) : void

- cbBairroMouseClicked(evt : MouseEvent) : void

- cbCidadeMouseClicked(evt : MouseEvent) : void

- cbBairroFocusGained(evt : FocusEvent) : void

- cbCidadeFocusGained(evt : FocusEvent) : void

txtCepActionPerformed(evt : ActionEvent) : void

- cbSituacaoFocusLost(evt : FocusEvent) : void

- cbSituacaoActionPerformed(evt : ActionEvent) : void

- txtNomeKeyPressed(evt : KeyEvent) : void

txtCepKeyPressed(evt : KeyEvent) : void

txtEnderecoKeyPressed(evt : KeyEvent) : void

txtNumeroKeyPressed(evt : KeyEvent) : void

- txtComplementoKeyPressed(evt : KeyEvent) : void

- cbBairroKeyPressed(evt : KeyEvent) : void

- cbCidadeKeyPressed(evt : KeyEvent) : void

- cbTipoDocKeyPressed(evt : KeyEvent) : void

txtDocumentoKeyPressed(evt : KeyEvent) : void

- txtleKeyPressed(evt : KeyEvent) : void

- txtTelefoneKeyPressed(evt : KeyEvent) : void

- txtEmailKeyPressed(evt : KeyEvent) : void

- txtNomeActionPerformed(evt : ActionEvent) : void

ViewLogin

- + ViewLogin()
- initComponents(): void
- btSairActionPerformed(evt : ActionEvent) : void
- bEntrarActionPerformed(evt : ActionEvent) : void
- + MudarLogo(): void
- autenticar(): void
- txtSenhaActionPerformed(evt : ActionEvent) : void
- txtUsuarioKeyPressed(evt : KeyEvent) : void
- txtSenhaKeyPressed(evt : KeyEvent) : void
- txtUsuarioActionPerformed(evt : ActionEvent) : void
- bEntrarKeyPressed(evt : KeyEvent) : void
- + main(args : String[]) : void

ViewPagamento

- ~ tipoCadastro : String
- ~ aba : int = 1
- + ViewPagamento()
- initComponents(): void
- btCancelarActionPerformed(evt : ActionEvent) : void
- + MudarLogo(): void
- btAlterarActionPerformed(evt : ActionEvent) : void
- btSalvarActionPerformed(evt : ActionEvent) : void
- alteraPagamento(): boolean
- salvarPagamento(): boolean
- recuperarPagamento(): void
- recuperaAlterar(): void
- recuperalncluir(): void
- carregarPagamentos(): void
- novoPagamento(): void
- Controlesalterar(): void
- ControlesPesquisar(): void
- # desabilitarCampos(): void
- btNovoActionPerformed(evt : ActionEvent) : void
- btPesquisarPagamentoActionPerformed(evt : ActionEvent) : void
- btExcluirActionPerformed(evt : ActionEvent) : void
- txtNomeFocusLost(evt : FocusEvent) : void
- txtNomeActionPerformed(evt : ActionEvent) : void
- txtDiasActionPerformed(evt : ActionEvent) : void
- txtNomeKeyPressed(evt : KeyEvent) : void
- cbTipoKeyPressed(evt : KeyEvent) : void
- txtParcelasKeyPressed(evt : KeyEvent) : void
- + main(args : String[]) : void

ViewPrincipal

~ pNomeUsuario : String

~ pPerfil : String

+ ViewPrincipal()

- initComponents(): void

+ MudarLogo(): void

- jmiSobreActionPerformed(evt : ActionEvent) : void

- mnuSairActionPerformed(evt : ActionEvent) : void

- smnClientesActionPerformed(evt : ActionEvent) : void

- smniProdutosActionPerformed(evt : ActionEvent) : void

- smnFornecedoresActionPerformed(evt : ActionEvent) : void

- mnuRegistrar√endaActionPerformed(evt : ActionEvent) : void

- smnRelClientesActionPerformed(evt : ActionEvent) : void

- smnRelFornecedorActionPerformed(evt : ActionEvent) : void

- smnRelProdutosActionPerformed(evt : ActionEvent) : void

- mnUsuarioActionPerformed(evt : ActionEvent) : void

- jMenultem3ActionPerformed(evt : ActionEvent) : void

smniRelatorioVendasPriodoActionPerformed(evt : ActionEvent) : void

- smnLancamentoCaixaActionPerformed(evt : ActionEvent) : void

- smnBairrosActionPerformed(evt : ActionEvent) : void

- smnCidadesActionPerformed(evt : ActionEvent) : void

- smnCategoriasActionPerformed(evt : ActionEvent) : void

- smnComprasActionPerformed(evt : ActionEvent) : void

- smnRelCategoriasActionPerformed(evt : ActionEvent) : void

- smnRelBairrosActionPerformed(evt : ActionEvent) : void

- mnSegurancaActionPerformed(evt : ActionEvent) : void

- mnCadastrosActionPerformed(evt : ActionEvent) : void

- jMenultem11ActionPerformed(evt : ActionEvent) : void

- jMenultem12ActionPerformed(evt : ActionEvent) : void

- smnPagamentosActionPerformed(evt : ActionEvent) : void

- smnRelPagamentosActionPerformed(evt : ActionEvent) : void

- smnRelCidadesActionPerformed(evt : ActionEvent) : void

- smnPagarActionPerformed(evt : ActionEvent) : void

- smnQuitacaoPagarActionPerformed(evt : ActionEvent) : void

smnReceberActionPerformed(evt : ActionEvent) : void

smnQuitacaoReceberActionPerformed(evt : ActionEvent) : void

jMenultem2ActionPerformed(evt : ActionEvent) : void

- jMenultem1 ActionPerformed(evt : ActionEvent) : void

- smnRelComprasPeriodoActionPerformed(evt : ActionEvent) : void

- smnRelContasReceberActionPerformed(evt : ActionEvent) : void

- smnRelContasPagarActionPerformed(evt : ActionEvent) : void

smmRelMovimentoCaixaActionPerformed(evt : ActionEvent) : void

- mnVendedorActionPerformed(evt : ActionEvent) : void

- mnEmpresaActionPerformed(evt : ActionEvent) : void

+ retornarUsuarioLogado(): String

+ retornarPerfilLogado(): String

- configurar(): void

ViewProduto

~ tipoCadastro : String ~ aba : int = 1 ~ bairro : String ~ cidade : String

+ ViewProduto()

- initComponents(): void + MudarLogo(): void

btCancelarActionPerformed(evt : ActionEvent) : void
 btAlterarActionPerformed(evt : ActionEvent) : void
 btSalvarActionPerformed(evt : ActionEvent) : void

- alteraProduto(): boolean - salvarProduto(): boolean - recuperarProduto(): void - recuperaAlterar(): void - recuperaIncluir(): void - carregarProdutos(): void - novoProduto(): void - Controlesalterar(): void - ControlesPesquisar(): void # desabilitarCampos(): void

- listarFornecedor(): void

- listarCategoria() : void

- btNovoActionPerformed(evt : ActionEvent) : void

- btPesquisarProdutoActionPerformed(evt : ActionEvent) : void

- btExcluirActionPerformed(evt : ActionEvent) : void

- txtEstoqueFocusLost(evt : FocusEvent) : void

- txtValor1FocusLost(evt : FocusEvent) : void

- cbCategoriaActionPerformed(evt : ActionEvent) : void

- cbFornecedorActionPerformed(evt : ActionEvent) : void

- txtNomeFocusLost(evt : FocusEvent) : void

- txtValor1ActionPerformed(evt : ActionEvent) : void
 - cbSituacaoActionPerformed(evt : ActionEvent) : void

ViewQuitacaoContasPagar

~ tipoCadastro : String ~ aba : int = 1 ~ cliente : String ~ pagamento : String

- + ViewQuitacaoContasPagar()
- + MudarLogo(): void - configurar(): void - initComponents(): void
- btQuitarActionPerformed(evt : ActionEvent) : void
 btSalvarActionPerformed(evt : ActionEvent) : void
- atteraContasPagar(): boolean
 recuperarContasPagar(): void
 carregarContasPagar(): void
 novoContasPagar(): void
 Controlesatterar(): void
- ControlesPesquisar(): void # desabilitarCampos(): void
- listarFornecedor(): void
 listarPagamento(): void
 preencherLista(): void
- btNovoActionPerformed(evt : ActionEvent) : void
- btPesquisarContasPagarActionPerformed(evt : ActionEvent) : void
- cbPagamentoActionPerformed(evt : ActionEvent) : void
- txtNotaFocusLost(evt : FocusEvent) : void- txtNotaKeyPressed(evt : KeyEvent) : void
- cbPagamentoKeyPressed(evt : KeyEvent) : void
- jdlnclusaoKeyPressed(evt : KeyEvent) : void
- jdVencimentoKeyPressed(evt : KeyEvent) : void
- jdlnclusaoFinalKeyPressed(evt : KeyEvent) : void
- jdVencimentoFinalKeyPressed(evt : KeyEvent) : void
- jdlnclusaoFocusLost(evt : FocusEvent) : void
- txtQPedidoFocusLost(evt : FocusEvent) : void
- txtQPedidoKeyPressed(evt : KeyEvent) : void
- jdQlnclusaoFocusLost(evt : FocusEvent) : void
- jdQlnclusaoKeyPressed(evt : KeyEvent) : void
- jdQVencimentoKeyPressed(evt : KeyEvent) : void
- txtQTotalFocusLost(evt : FocusEvent) : void
- txtQTotalKeyPressed(evt : KeyEvent) : void
- txtQValorPagoFocusLost(evt : FocusEvent) : void
- txtQValorPagoKeyPressed(evt : KeyEvent) : void
- txtQ∀alorPagamentoFocusLost(evt : FocusEvent) : void
- txtQValorPagamentoKeyPressed(evt : KeyEvent) : void
- txtQValorAbertoFocusLost(evt : FocusEvent) : void
- txtQValorAbertoKeyPressed(evt : KeyEvent) : void
- txtQRestanteFocusLost(evt : FocusEvent) : void
- txtQRestanteKeyPressed(evt : KeyEvent) : void
- cbFornecdorKeyPressed(evt : KeyEvent) : void
- cbFornecdorActionPerformed(evt : ActionEvent) : void
- txtQValorPagamentoActionPerformed(evt : ActionEvent) : void
- txtQTotalPagoFocusLost(evt : FocusEvent) : void
- txtQTotalPagoKeyPressed(evt : KeyEvent) : void
- + main(args : String[]) : void

ViewQuitacaoContasReceber

```
~ tipoCadastro : String
~ aba : int = 1
~ cliente : String
~ pagamento : String
```

- + ViewQuitacaoContasReceber()
- + MudarLogo(): void - configurar(): void - initComponents(): void
- btQuitarActionPerformed(evt : ActionEvent) : void
 btSalvarActionPerformed(evt : ActionEvent) : void
- atteraContasReceber(): boolean
 recuperarContasReceber(): void
 carregarContasReceber(): void
 novoContasReceber(): void
- Controlesalterar(): void
 ControlesPesquisar(): void
 # desabilitarCampos(): void
- listarCliente(): void - listarPagamento(): void - preencherLista(): void
- btNovoActionPerformed(evt : ActionEvent) : void
- btPesquisarContasReceberActionPerformed(evt : ActionEvent) : void
- cbPagamentoActionPerformed(evt : ActionEvent) : void
- txtPedidoFocusLost(evt : FocusEvent) : void
- txtPedidoKeyPressed(evt : KeyEvent) : void
- cbPagamentoKeyPressed(evt : KeyEvent) : void
- jdlnclusaoKeyPressed(evt : KeyEvent) : void
- jd∨encimentoKeyPressed(evt : KeyEvent) : void
- jdlnclusaoFinalKeyPressed(evt : KeyEvent) : void
- jdVencimentoFinalKeyPressed(evt : KeyEvent) : void
- jdlnclusaoFocusLost(evt : FocusEvent) : void
- txtQPedidoFocusLost(evt : FocusEvent) : void
- txtQPedidoKeyPressed(evt : KeyEvent) : void
- jdQlnclusaoFocusLost(evt : FocusEvent) : void
- jdQlnclusaoKeyPressed(evt : KeyEvent) : void
- jdQVencimentoKeyPressed(evt : KeyEvent) : void
 jdInclusaoFocusLost(evt : FocusEvent) : void
- txtQPedidoFocusLost(evt : FocusEvent) : void
- txtQPedidoKeyPressed(evt : KeyEvent) : void
- jdQlnclusaoFocusLost(evt : FocusEvent) : void
- jdQlnclusaoKeyPressed(evt : KeyEvent) : void
- juginiciusaoneyriesseu(evi . neyEveni) . void
- jdQVencimentoKeyPressed(evt : KeyEvent) : void
- txtQTotalFocusLost(evt : FocusEvent) : void
- txtQTotalKeyPressed(evt : KeyEvent) : void
- txtQValorPagoFocusLost(evt : FocusEvent) : void
- txtQValorPagoKeyPressed(evt : KeyEvent) : void
- txtQValorPagamentoFocusLost(evt : FocusEvent) : void
- txtQValorPagamentoKeyPressed(evt : KeyEvent) : void
- txtQ∀alorAbertoFocusLost(evt : FocusEvent) : void
- txtQValorAbertoKeyPressed(evt : KeyEvent) : void
- txtQRestanteFocusLost(evt : FocusEvent) : void
- txtQRestanteKeyPressed(evt : KeyEvent) : void
- cbClienteKeyPressed(evt : KeyEvent) : void
- cbClienteActionPerformed(evt : ActionEvent) : void
- txtQValorPagamentoActionPerformed(evt : ActionEvent) : void
- txtQTotalPagoFocusLost(evt : FocusEvent) : void
- txtQTotalPagoKeyPressed(evt : KeyEvent) : void
- + main(args : String[]) : void

ViewRelatorioBairro

- ~ tipoDeRelatorio : String
- + ViewRelatorioBairro()
- initComponents(): void
- + MudarLogo(): void
- LimpaFiltros(): void
- listarBairro(): void
- listarCidade(): void
- btlmprimirRelatorioBairroActionPerformed(evt : ActionEvent) : void
- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void
- cbBairroActionPerformed(evt : ActionEvent) : void
- + main(args : String[]) : void

ViewRelatorioCategoria

- ~ tipoDeRelatorio : String
- + ViewRelatorioCategoria()
- initComponents(): void
- + MudarLogo(): void
- LimpaFiltros(): void
- listarCategoria(): void
- btImprimirRelatorioCategoriaActionPerformed(evt : ActionEvent) : void
- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void
- cbCategoriaActionPerformed(evt : ActionEvent) : void
- + main(args : String[]) : void

ViewRelatorioCidade

- ~ tipoDeRelatorio : String
- + ViewRelatorioCidade()
- initComponents(): void
- + MudarLogo(): void
- LimpaFiltros(): void
- listarCidade(): void
- btlmprimirRelatorioCidadeActionPerformed(evt : ActionEvent) : void
- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void
- cbCidadeActionPerformed(evt : ActionEvent) : void
- + main(args : String[]) : void

ViewRelatorioCliente

- ~ tipoDeRelatorio : String
- ~ bairro : String
- ~ cidade : String
- + ViewRelatorioCliente()
- initComponents(): void
- + MudarLogo(): void
- LimpaFiltros(): void
- listarCliente(): voidlistarCidade(): void
- listarBairro(): void
- cbTipoDocFocusLost(evt : FocusEvent) : void
- cbTipoDocActionPerformed(evt : ActionEvent) : void
- cbBairroActionPerformed(evt : ActionEvent) : void
- cbCidadeActionPerformed(evt : ActionEvent) : void
- btImprimirRelatorioClienteActionPerformed(evt : ActionEvent) : void
- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void
- cbClienteActionPerformed(evt : ActionEvent) : void
- + main(args : String[]) : void

ViewRelatorioContasPagar

- ~ tipoDeRelatorio : String
- + ViewRelatorioContasPagar()
- initComponents(): void
- + MudarLogo(): void
- LimpaFiltros(): void
- listarFornecedor(): void
- listarPagamento(): void
- cbPagamentoActionPerformed(evt : ActionEvent) : void
- btlmprimirRelatorioContasPagarActionPerformed(evt : ActionEvent) : void
- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void
- cbFornecedorActionPerformed(evt : ActionEvent) : void
- jdDataInclusaoInicialKeyPressed(evt : KeyEvent) : void
- jdDataInclusaoFinalKeyPressed(evt : KeyEvent) : void
- jdDataVencimentolnicialKeyPressed(evt : KeyEvent) : void
- jdDataVencimentoFinalKeyPressed(evt : KeyEvent) : void
- jdDataInclusao2KeyPressed(evt : KeyEvent) : void
- jdDataInclusaoFinal2KeyPressed(evt : KeyEvent) : void
- jdDataPagamentolnicialKeyPressed(evt : KeyEvent) : void
- jdDataPagamentoFinalKeyPressed(evt : KeyEvent) : void
- txtNumeroPedidoActionPerformed(evt : ActionEvent) : void
- + main(args : String[]) : void

ViewRelatorioContasReceber

- ~ tipoDeRelatorio : String
- + ViewRelatorioContasReceber()
- initComponents(): void
- + MudarLogo(): void
- LimpaFiltros(): void
- listarCliente() : void
- listarPagamento(): void
- cbPagamentoActionPerformed(evt : ActionEvent) : void
- btlmprimirRelatorioContasReceberActionPerformed(evt : ActionEvent) : void
- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void
- cbClienteActionPerformed(evt : ActionEvent) : void
- jdDataInclusaoInicialKeyPressed(evt : KeyEvent) : void
- jdDataInclusaoFinalKeyPressed(evt : KeyEvent) : void
- jdDataVencimentolnicialKeyPressed(evt : KeyEvent) : void
- jdData∀encimentoFinalKeyPressed(evt : KeyEvent) : void
- jdDataInclusao2KeyPressed(evt : KeyEvent) : void
- jdDataInclusaoFinal2KeyPressed(evt : KeyEvent) : void
- jdDataPagamentolnicialKeyPressed(evt : KeyEvent) : void
- jdDataPagamentoFinalKeyPressed(evt : KeyEvent) : void
- txtNumeroPedidoActionPerformed(evt : ActionEvent) : void
- + main(args : String[]) : void

ViewRelatorioEntradas

~ tipoDeRelatorio : String

+ ViewRelatorioEntradas()

- initComponents(): void

+ MudarLogo(): void

- LimpaFiltros(): void

- listarFornecedor() : void

- listarPagamento(): void

- cbPagamentosActionPerformed(evt : ActionEvent) : void

- btlmprimirRelatorioEntradasActionPerformed(evt : ActionEvent) : void

- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void

- cbFornecedoresActionPerformed(evt : ActionEvent) : void

- jdlnicialKeyPressed(evt : KeyEvent) : void

- jdFinalKeyPressed(evt : KeyEvent) : void

+ main(args : String[]) : void

ViewRelatorioFluxoCaixa

~ tipoDeRelatorio : String

+ ViewRelatorioFluxoCaixa()

- initComponents(): void

+ MudarLogo(): void

- LimpaFiltros(): void

- btImprimirRelatorioFluxoCaixaActionPerformed(evt : ActionEvent) : void

btLimpaFiltrosActionPerformed(evt : ActionEvent) : void

- jdlnicialKeyPressed(evt : KeyEvent) : void

- jdFinalKeyPressed(evt : KeyEvent) : void

- cbOrigemActionPerformed(evt : ActionEvent) : void

- cbOrigemKeyPressed(evt : KeyEvent) : void

- cbTipoActionPerformed(evt : ActionEvent) : void

- cbTipoKeyPressed(evt : KeyEvent) : void

+ main(args : String[]) : void

ViewRelatorioFornecedor

~ tipoDeRelatorio : String

~ bairro : String

~ cidade : String

+ ViewRelatorioFornecedor()

- initComponents(): void

+ MudarLogo(): void

- LimpaFiltros(): void

listarFornecedor(): void

- listarCidade(): void

- listarBairro(): void

- cbTipoDocFocusLost(evt : FocusEvent) : void

- cbTipoDocActionPerformed(evt : ActionEvent) : void

- cbBairroActionPerformed(evt : ActionEvent) : void

cbCidadeActionPerformed(evt : ActionEvent) : void

- btlmprimirRelatorioFornecedorActionPerformed(evt : ActionEvent) : void

- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void

- cbFornecedorActionPerformed(evt : ActionEvent) : void

ViewRelatorioPagamento

- ~ tipoDeRelatorio : String
- + ViewRelatorioPagamento()
- initComponents(): void
- + MudarLogo(): void
- LimpaFiltros(): void
- listarPagamento() : void
- btImprimirRelatorioPagamentoActionPerformed(evt : ActionEvent) : void
- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void
- cbPagamentoActionPerformed(evt : ActionEvent) : void
- + main(args : String[]) : void

ViewRelatorioProduto

- ~ tipoDeRelatorio : String ~ fornecedor : String ~ categoria : String
- + ViewRelatorioProduto()
- initComponents(): void
- + MudarLogo(): void
- LimpaFiltros(): void
- Estaplication () : void
- listarProduto(): voidlistarCategoria(): void
- listarFornecedor(): void
- cbSituacaoFocusLost(evt : FocusEvent) : void
- cbSituacaoActionPerformed(evt : ActionEvent) : void
- cbFornecedorActionPerformed(evt : ActionEvent) : void
- cbCategoriaActionPerformed(evt : ActionEvent) : void
- btImprimirRelatorioProdutoActionPerformed(evt : ActionEvent) : void
- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void
- cbProdutoActionPerformed(evt : ActionEvent) : void
- + main(args : String[]) : void

ViewRelatorioUsuario

- ~ tipoDeRelatorio : String
- + ViewRelatorioUsuario()
- initComponents(): void
- + MudarLogo(): void
- LimpaFiltros(): void
- listarUsuario(): void
- btlmprimirRelatorioUsuarioActionPerformed(evt : ActionEvent) : void
- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void
- cbUsuarioActionPerformed(evt : ActionEvent) : void
- + main(args : String[]) : void

ViewRelatorio Vendas

~ tipoDeRelatorio : String

~ bairro : String ~ cidade : String

+ ViewRelatorioVendas()

- initComponents(): void

+ MudarLogo(): void

- LimpaFiltros(): void

- listarCliente() : void

- listarPagamento(): void

- listar Vendedor(): void

- cbVendedorActionPerformed(evt : ActionEvent) : void

- cbPagamentoActionPerformed(evt : ActionEvent) : void

- btlmprimirRelatorio Vendas Action Performed (evt : Action Event) : void

- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void

cbClienteActionPerformed(evt : ActionEvent) : void

- jdlnicialKeyPressed(evt : KeyEvent) : void

- jdFinalKeyPressed(evt : KeyEvent) : void

+ main(args : String[]) : void

ViewRelatorio Vendedor

~ tipoDeRelatorio : String

~ bairro : String ~ cidade : String

+ ViewRelatorioVendedor()

- initComponents(): void

+ MudarLogo(): void

LimpaFiltros(): void

- listar∀endedor()∶ void

- listarCidade(): void

- listarBairro(): void

- cbTipoDocFocusLost(evt : FocusEvent) : void

- cbTipoDocActionPerformed(evt : ActionEvent) : void

- cbBairroActionPerformed(evt : ActionEvent) : void

- cbCidadeActionPerformed(evt : ActionEvent) : void

- btImprimirRelatorioVendedorActionPerformed(evt : ActionEvent) : void

- btLimpaFiltrosActionPerformed(evt : ActionEvent) : void

- cbVendedorActionPerformed(evt : ActionEvent) : void

+ main(args : String[]) : void

ViewSobre

+ ViewSobre(parent : Frame, modal : boolean)

- initComponents(): void

- jbOkActionPerformed(evt : ActionEvent) : void

ViewUsuario

~ tipoCadastro : String

~ aba : int = 1

+ ViewUsuario()

- initComponents(): void

+ MudarLogo(): void

btCancelarActionPerformed(evt : ActionEvent) : void
 btAlterarActionPerformed(evt : ActionEvent) : void

- btSalvarActionPerformed(evt : ActionEvent) : void

atteraUsuario() : booleanconfirmaAtterar() : boolean

- salvarUsuario() : boolean

confirmaSalvar(): boolean
 recuperarUsuario(): void

recuperaAlterar(): void
 recuperaIncluir(): void

- carregarUsuarios(): void - novoUsuario(): void

- Controlesalterar(): void

- ControlesPesquisar(): void

desabilitarCampos(): void

- btNovoActionPerformed(evt : ActionEvent) : void

- btPesquisarUsuarioActionPerformed(evt : ActionEvent) : void

- btExcluirActionPerformed(evt : ActionEvent) : void

- txtNomeFocusLost(evt : FocusEvent) : void

- txtLoginActionPerformed(evt : ActionEvent) : void

- cbPerfilActionPerformed(evt : ActionEvent) : void

- txtNomeActionPerformed(evt : ActionEvent) : void

- txtNomeKeyPressed(evt : KeyEvent) : void

- txtLoginKeyPressed(evt : KeyEvent) : void

- cbPerfilKeyPressed(evt : KeyEvent) : void

- txtSenhaKeyPressed(evt : KeyEvent) : void

ViewVenda

~ nrvenda : int

~ tipoCadastro : String

~ aba : int = 1

+ ViewVenda()

+ MudarLogo(): void

- listarClientes(): void

- listarClientesPesquisa(): void

- listar∀endedores(): void

- listar∀endedoresPesquisa(): void

listarPagamento(): void

- listarPagamentosPesquisa(): void

- listarProdutos(): void

+ setViewConsultarVendas(): void

- initComponents(): void

- btCancelarActionPerformed(evt : ActionEvent) : void

desabilitarCampos(): void

- btSalvarActionPerformed(evt : ActionEvent) : void

cadastrar∀enda() : boolean

- btGridActionPerformed(evt : ActionEvent) : void

- btPesquisarActionPerformed(evt : ActionEvent) : void

- btExcluirActionPerformed(evt : ActionEvent) : void

- btNovoActionPerformed(evt : ActionEvent) : void

- btRetirarActionPerformed(evt : ActionEvent) : void

- btlmprimirActionPerformed(evt : ActionEvent) : void

- txtDescontoKeyReleased(evt : KeyEvent) : void

- txtDescontoFocusLost(evt : FocusEvent) : void

- cbClientesActionPerformed(evt : ActionEvent) : void

- btlncluirProdutoActionPerformed(evt : ActionEvent) : void

- txtQuantidadeActionPerformed(evt : ActionEvent) : void

- txtDescontoActionPerformed(evt : ActionEvent) : void

- txtValorTotalActionPerformed(evt : ActionEvent) : void

- cbPagamentoActionPerformed(evt : ActionEvent) : void

- cbVendedoresActionPerformed(evt : ActionEvent) : void

- txtAcrescimoActionPerformed(evt : ActionEvent) : void

- txtAcrescimoFocusLost(evt : FocusEvent) : void

txtAcrescimoKeyReleased(evt : KeyEvent) : void

- pcbClientesActionPerformed(evt : ActionEvent) : void

- pcbPagamentosActionPerformed(evt : ActionEvent) : void

- cbClientesKeyPressed(evt : KeyEvent) : void

- cbVendedoresKeyPressed(evt : KeyEvent) : void

- cbPagamentoKeyPressed(evt : KeyEvent) : void

- cbProdutosKeyPressed(evt : KeyEvent) : void

- txtQuantidadeKeyPressed(evt : KeyEvent) : void

- cbProdutosActionPerformed(evt : ActionEvent) : void

- btlncluirProdutoKeyPressed(evt : KeyEvent) : void

- pcbVendedoresActionPerformed(evt : ActionEvent) : void

- jdlnicialKeyPressed(evt : KeyEvent) : void

- jdFinalKeyPressed(evt : KeyEvent) : void

- pcbSituacaoActionPerformed(evt : ActionEvent) : void

- recuperar∀enda() : boolean

- carregar Vendas(): void

- limpafiltros(): void

- novo∀enda(): void

- habilitarCampos(): void

- incluirProduto(): void

- somaEAtualiza ValorTotal(): Double

- alterar∀enda() : boolean

- retornarProdutoAoEstoque(): void

ViewVendedor

~ tipoCadastro : String ~ aba : int = 1 ~ bairro : String

~ cidade : String

+ ViewVendedor()
- initComponents(): void

+ MudarLogo(): void

btCancelarActionPerformed(evt : ActionEvent) : void
 btAlterarActionPerformed(evt : ActionEvent) : void
 btSalvarActionPerformed(evt : ActionEvent) : void

- altera√endedor() : boolean - salvar√endedor() : boolean - recuperar√endedor() : void - recuperaAlterar() : void

recuperalncluir(): void
 carregarVendedores(): void
 novoVendedor(): void

Controlesalterar(): void
 ControlesPesquisar(): void

desabilitarCampos(): void

listarCidade(): voidlistarBairro(): void

- btNovoActionPerformed(evt : ActionEvent) : void

- btPesquisarVendedorActionPerformed(evt : ActionEvent) : void

- btExcluirActionPerformed(evt : ActionEvent) : void

- txtNomeFocusLost(evt : FocusEvent) : void

- cbCidadeActionPerformed(evt : ActionEvent) : void

- txtEnderecoFocusLost(evt : FocusEvent) : void

- cbBairroActionPerformed(evt : ActionEvent) : void

- txtEmailFocusLost(evt : FocusEvent) : void

- txtEnderecoActionPerformed(evt : ActionEvent) : void

txtTelefoneActionPerformed(evt : ActionEvent) : void

- cbTipoDocActionPerformed(evt : ActionEvent) : void

- txtDocumentoFocusLost(evt : FocusEvent) : void

- txtleFocusLost(evt : FocusEvent) : void

- txtEmailActionPerformed(evt : ActionEvent) : void

txtDocumentoActionPerformed(evt : ActionEvent) : void

- txtleActionPerformed(evt : ActionEvent) : void

- cbTipoDocFocusLost(evt : FocusEvent) : void

- cbBairroMouseClicked(evt : MouseEvent) : void

- cbCidadeMouseClicked(evt : MouseEvent) : void

- cbBairroFocusGained(evt : FocusEvent) : void

- cbCidadeFocusGained(evt : FocusEvent) : void

- txtCepActionPerformed(evt : ActionEvent) : void

- cbSituacaoFocusLost(evt : FocusEvent) : void

- cbSituacaoActionPerformed(evt : ActionEvent) : void

- txtNomeKeyPressed(evt : KeyEvent) : void

txtCepKeyPressed(evt : KeyEvent) : void

- txtEnderecoKeyPressed(evt : KeyEvent) : void

- txtNumeroKeyPressed(evt : KeyEvent) : void

- txtComplementoKeyPressed(evt : KeyEvent) : void

- cbBairroKeyPressed(evt : KeyEvent) : void

- cbCidadeKeyPressed(evt : KeyEvent) : void

- cbTipoDocKeyPressed(evt : KeyEvent) : void

txtDocumentoKeyPressed(evt : KeyEvent) : void

- txtleKeyPressed(evt : KeyEvent) : void

- txtTelefoneKeyPressed(evt : KeyEvent) : void

- txtEmailKeyPressed(evt : KeyEvent) : void

txtNomeActionPerformed(evt : ActionEvent) : void