**CURSO TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**TURNO DIURNO 1º SEMESTRE 2017**

**ENGENHARIA DE SOFTWARE II v1**

**GRUPO 4 – CONTAS A PAGAR**

BRUNO HENRIQUE M. FRANÇA 0030481611039

FELIPE DA SILVA LIBERAL 0030481521010

GIOVANE DE LUCAS HADDAD 0030481611018

VANESSA OLIVEIRA VIEIRA 0030481511039

VICTOR COSTA 0030481611034

VINICIUS EIDI HONDA 0030481611036

SUMÁRIO

[1 DEFINIÇÃO DE TELAS 2](#_Toc483797032)

[1.1 Consulta dashboard 2](#_Toc483797033)

[1.2 Consulta de usuários 5](#_Toc483797034)

[1.3 Consulta de contas bancárias 8](#_Toc483797035)

[1.4 Consulta de fornecedores 10](#_Toc483797036)

[1.5 Consulta de fornecedores por filtros 13](#_Toc483797037)

[1.6 Consulta de lançamentos 15](#_Toc483797038)

[1.7 Consulta de conta já baixada no sistema 17](#_Toc483797039)

[1.8 Consulta de lançamentos por filtros 20](#_Toc483797040)

[1.9 Consulta de contas conciliadas 23](#_Toc483797041)

[1.10 Consulta de contas conciliadas por filtros 25](#_Toc483797042)

[1.11 Consulta de contas a pagar 27](#_Toc483797043)

[1.12 Consulta de auditoria de ações no sistema 30](#_Toc483797044)

[1.13 Consulta de auditoria de ações no sistema por filtros 32](#_Toc483797045)

[1.14 Relatório de contas a pagar 34](#_Toc483797046)

[1.15 Relatório de auditoria 37](#_Toc483797047)

[1.16 Relatório de auditoria por usuário 40](#_Toc483797048)

[1.17 Relatório de movimentação bancária 43](#_Toc483797049)

[1.18 Relatório de contas por fornecedor 47](#_Toc483797050)

# DEFINIÇÃO DE TELAS

## Consulta dashboard

(Parte feita por: Bruno)

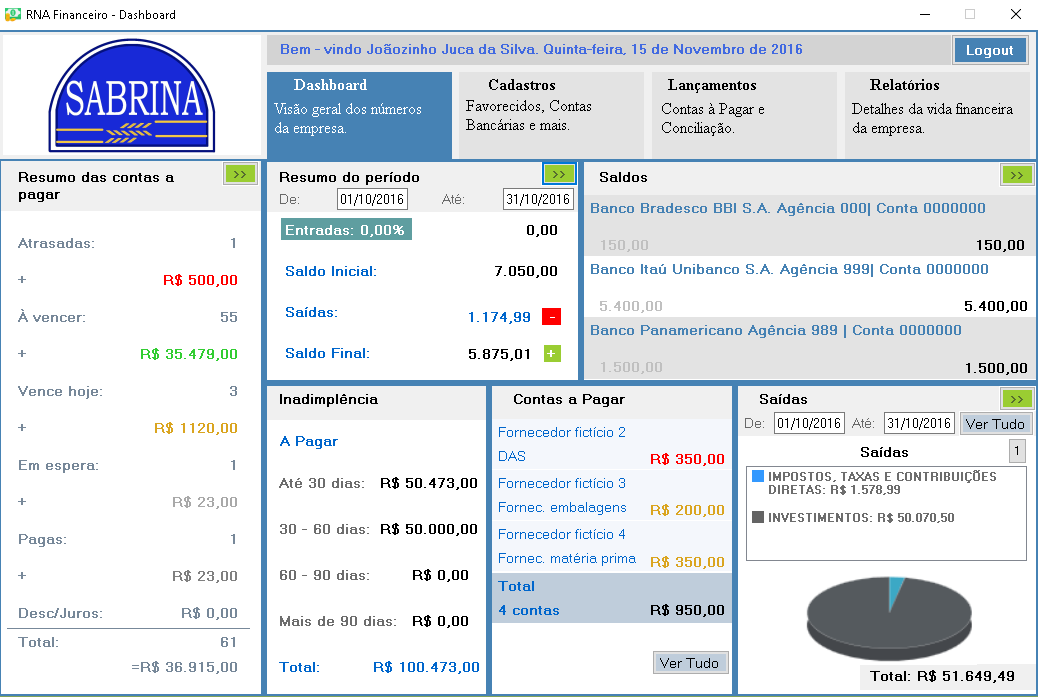
**Objetivo:** Este caso de uso permite ter uma visão geral dos dados numéricos da empresa.

**Enunciado:**

**Descrição da tela:** Na parte superior da tela temos quatro botões correspondentes a quatro abas, a primeira é o Dashboard, que mostra uma visão geral dos números da empresa, a segunda aba é a de cadastros, onde são realizados cadastros, a terceira são lançamentos onde temos contas a pagar e conciliação e a última aba é a de relatórios, onde se obtém detalhes da vida financeira da empresa. Ainda na parte superior da tela, mais a direita existe um botão de logout que ao clicar é desligado da conta. Na aba Dashboard temos 6 telas, a primeira é o resumo das contas a pagar, onde é mostrado as contas atrasadas e seus valores, as contas a vencer com os valores, as contas a vencer hoje mais seus valores, as contas em espera com o respectivo valor, as contas pagas e o valor pago, mostra descontos e jurus e por último, mostra o número total de contas e o valor da soma de todas elas. A segunda é o resumo do período, onde você escolhe as datas (MaskedTextBox) que quer obter o resumo e lá é exibido o saldo inicial, as saídas e o saldo final. A terceira mostra os saldos existentes nas contas bancárias. A quarta é a parte de inadimplência que exibe as contas a pagar em até 30 dias, de 30 a 60 dias, de 60 a 90 dias, a mais de 90 dias e exibe o valor total. A quinta exibe contas a pagar que mostra o fornecedor, o material e o valor restante para pagar, nessa parte existe um botão para exibir todos fornecedores e materiais comprados restantes a pagar. A sexta parte mostra as saídas de dinheiro e tem a opção de escolher a data que você quer ver as saídas, após escolher as datas, o sistema exibe os impostos, taxas e contribuições e em seguida exibe os investimentos e cria um gráfico mostrando o total investido e o total de impostos.

**Eventos:** Na parte superior da tela, o evento clique do botão Dashboard deverá exibir uma tela com 6 telas, sendo a primeira o resumo de contas a pagar, onde deverá exibir as contas **atrasadas** localizadas no método “Obter Resumo” da classe “Conta a pagar”, **à vencer** localizada no método “Obter Resumo” da classe “Conta a pagar”, **vencer hoje** localizada no método “Obter Resumo” da classe “Conta a pagar”, **em espera** localizada no método “Obter Resumo” da classe “Conta a pagar”, **pagas** localizadas no método “Obter Resumo” da classe “Conta a pagar”, **desc/jurus** localizadas no método “Obter Resumo” da classe “Conta a pagar”, **o total de contas** localizadas no método “Obter Resumo” da classe “Conta a pagar”. e o **valor total** localizado no método “Obter Resumo” da classe “Conta a pagar”. A segunda tela mostra o resumo do período, o usuário deverá poder escolher as datas (usar MaskTextBox no formato de data DD/MM/AAAA), assim que escolhido o sistema deve exibir o Saldo Inicial, as Saídas e o Saldo Final (saldo inicial menos saídas) do período escolhido, para obter o saldo pegar no método “Saldo Inicio” da classe “Conta Bancaria” e para ver a saída na data escolhida deverá obter no método “Exibir Dados Conta Paga” da classe “Conta a Pagar”. A terceira mostrará os saldos existentes nas contas bancárias, para obter os valores usar o método “Atualizar Saldo” da classe “Conta Bancaria”. A quarta tela é a de inadimplência, terá que exibir dados de até 30 dias, entre 30 e 60 dias, entre 60 e 90 dias, após de 90 vidas e terá que calcular o valor total, para obter os dados dos valores das contas usar o método “Listar todas as contas” da classe “Conta a Pagar”. A quinta parte deverá exibir os fornecedores que faltam ser pago e o total da dívida, e deverá ter um botão associado ao event click que deverá mostrar a lista completa de fornecedores e valores a serem pagos, para obter os fornecedores, produtos e valores consulte o método “Listar todas as contas” da classe “Conta a Pagar” e dentro do método use o método “Filtrar Contas” da classe “Conta a Pagar” e obtenha somente os não pagos. Em “saídas” deverá ter um MaskTextBox no formato de DATA, onde o usuário poderá escolher duas datas para listar as saídas, utilize método “Exibir dados conta paga” localizado na classe “contas a pagar”, verifique o valor total das contas e dos impostos, e faça um gráfico separando o que era imposto e o que não era, além disso deve mostrar os respectivos valores.

**Layout**



**Classes**

## Consulta de usuários

(Parte feita por: Vinícius)

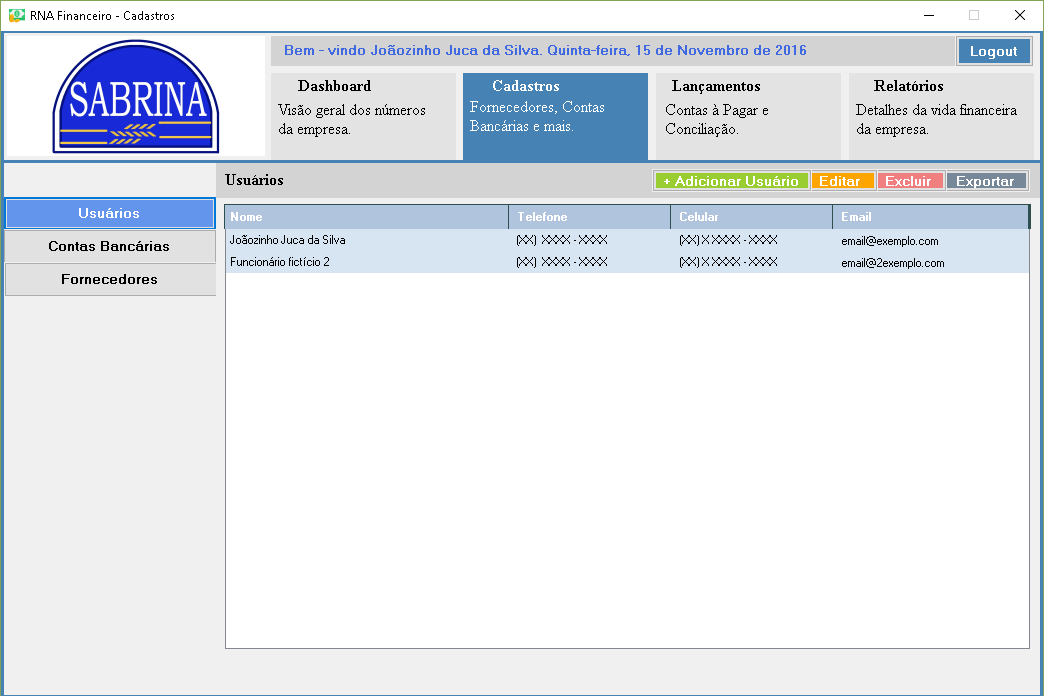
**Objetivo:** Consultar a lista de usuários cadastrados.

**Enunciado:**

**Descrição do relatório:**O cabeçalho da tela, a barra de ferramentas e área lateral de abas e resumos segue o padrão de telas base para módulos. A área variável no centro da tela possui cabeçalho com os nomes de cada campo referente a dados de cada um dos usuários: Nome, Telefone, Celular e E-mail; Abaixo por padrão em ordem decrescente da data e hora do cadastro do usuário, segue a lista de usuários cadastrados (Nome, Telefone, Celular, E-mail). Todos os campos das contas estarão em formato texto.

**Eventos:**

Quando esse programa for disparado, execute o método ListarUsuarios da classe Usuario, e imprima na tela em ordem do cadastro mais novo para o cadastro mais antigo (Data, e Hora da realização do cadastro) em cada linha os atributos (Nome, Telefone, Celular e Email), que são obtidos pelo método ObeterUsuario da classe Usuario.

**Layout:**

**Classes**



ObterUsuario: obtém os dados do usuário (Nome, Telefone, Celular, Email, e (data e hora do cadastro).

ListarUsuarios: Exibe todos os usuários cadastrados, obtidos através do método ObterUsuario da classe Usuario.

## Consulta de contas bancárias

(Parte feita por: Vinícius)

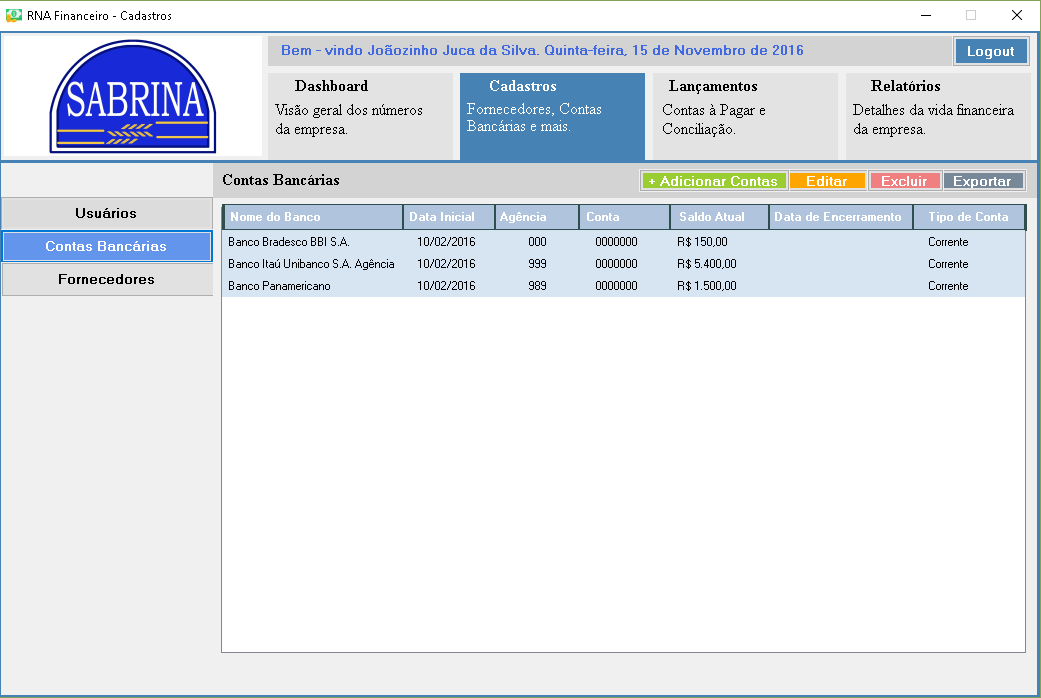
**Objetivo:** Consultar a lista de contas bancárias cadastradas.

**Enunciado:**

**Descrição do relatório:** O cabeçalho da tela, a barra de ferramentas e área lateral de abas e resumos segue o padrão de telas base para módulos. A área variável no centro da tela possui cabeçalho com os nomes de cada campo referente a dados de cada uma das contas bancárias cadastradas: Nome do Banco, Data Inicial, Agência, Conta, Saldo Atual, Data de Encerramento, Tipo de Conta; abaixo, em ordem decrescente de data e hora no qual as contas foram cadastradas, segue a lista das contas cadastradas (Nome do Banco, Data Inicial, Agência, Conta, Saldo Atual, Data de Encerramento, Tipo de Conta). Todos os campos das contas estarão em formato texto.

**Eventos:** Quando esse programa for disparado, execute o método ObterContasBancarias da classe Conta Bancaria, mostre todas as contas bancárias cadastradas e seus dados, (Nome do Banco, Data Inicial, Agência, Conta, Saldo Atual, Data de Encerramento, Tipo de Conta) em ordem decrescente de data e hora da realização do cadastro.

**Layout:**



**Classes**

ObterContasBancarias: Este método exibe os dados das contas bancárias (Nome do Banco, Data Inicial, Agência, Conta, Saldo Atual, Data de Encerramento, Tipo de Conta).

## Consulta de fornecedores

(Parte feita por: Vinícius)

**Objetivo:** Consultar a lista de fornecedores cadastrados.

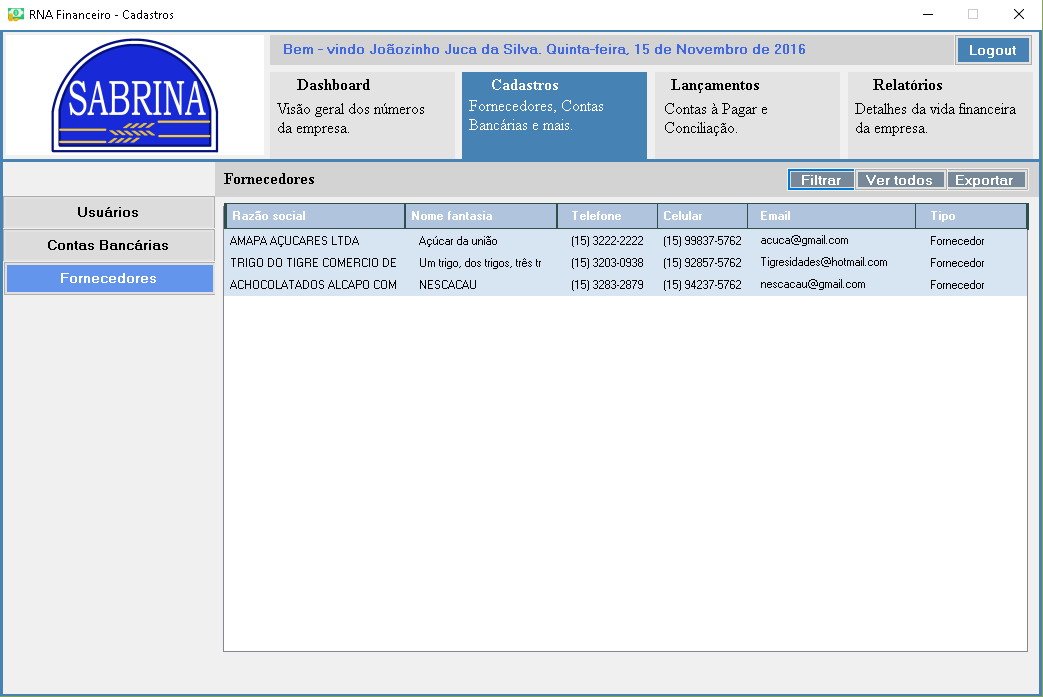
**Enunciado:**

**Descrição do relatório:** O cabeçalho da tela, a barra de ferramentas e área lateral de abas e resumos segue o padrão de telas base para módulos. A área variável no centro da tela possui cabeçalho com os nomes de cada campo referente a dados de cada um dos fornecedores cadastrados: Razão social, Nome Fantasia, Telefone, Celular, E-mail, Tipo; abaixo, por padrão em ordem decrescente de data e hora da realização do cadastro, segue a lista de Fornecedores com os dados (Razão social, Nome Fantasia, Telefone, Celular, E-Mail, Tipo), se o usuário utilizar a opção filtrar, os Fornecedores cadastrados são listados se atendem os parâmetros: Razão Social, Nome Fantasia, Telefone, Celular, ou Tipo de Fornecedores. Todos os campos das contas estarão no formato texto.

**Eventos:** Quanto este programa for disparado execute o método ObterFornecedor da classe fornecedor, e exiba os dados dos Fornecedores uma linha para cada fornecedor (Razão Social, Nome Fantasia, Telefone, Celular, ou Tipo de Favorecido).

No evento clique do botão Filtrar o método FiltrarFornecedor da classe Favorecidos é disparado e execute o método ObterForFiltrados da classe Favorecidos, mostre todos os favorecidos cadastrados que atendam as opções de filtragem, por (Razão Social, Nome Fantasia, Telefone, Celular, ou Tipo de Fornecedor).

**Layout:**



**Classes**



ObterFornecedor: Este método exibe os fornecedores e seus dados (Razão Social, Nome Fantasia, Telefone, Celular, ou Tipo de Fornecedor).

FiltrarFornecedor: Este método recebe os campos (Razão Social, Nome Fantasia, Telefone, Celular, ou Tipo de Fornecedor) e procura o fornecedor que corresponde aos dados digitados.

ObterForFiltrados: Este método obtém os fornecedores filtrados pelo método FiltrarFornecedor da classe Fornecedor, e exibe os fornecedores correspondentes a filtragem (Razão Social, Nome Fantasia, Telefone, Celular e Tipo de Fornecedor).

## Consulta de fornecedores por filtros

(Parte feita por: Bruno)

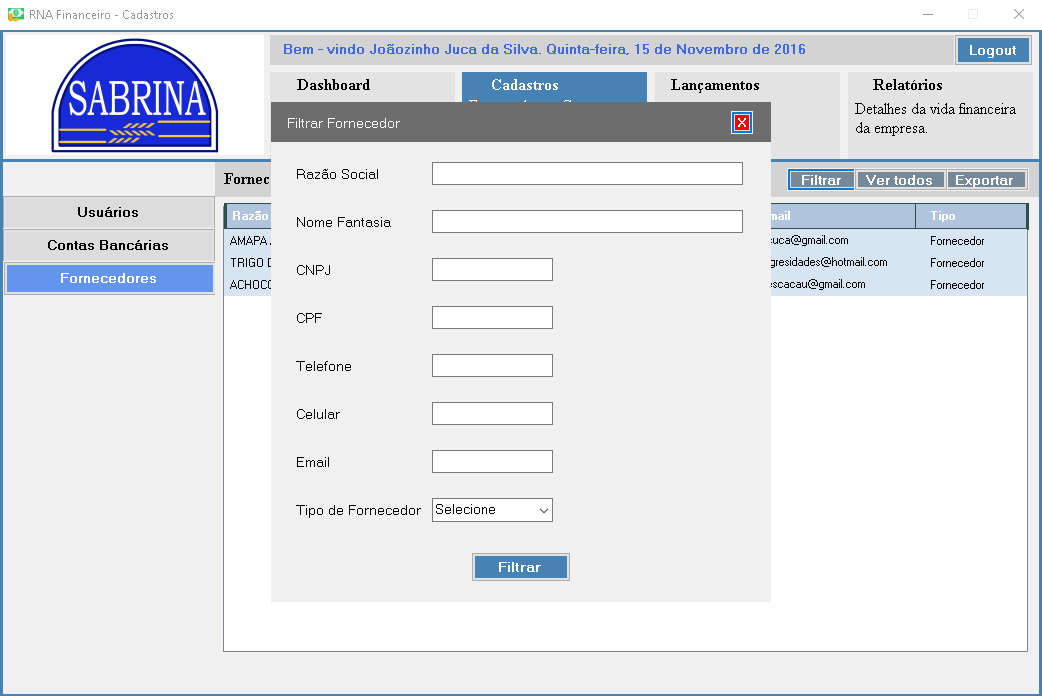
**Objetivo**: Procurar o fornecedor.

**Enunciado**:

**Descrição da tela:** Ao clicar em "Filtrar" na aba de fornecedor abrirá uma tela que aparecerá, um textbox para digitar a razão social, um para digitar nome fantasia, terá um combo box para selecionar o tipo, um textbox para digitar cnpj, cpf, telefone, celular, terá um combo box para selecionar se o limite do cartão de crédito é igual a, menor a, e um textbox para digitar o respectivo valor, terá um combo box para status e um combo box para o tipo de fornecedor e por último pera um botão para filtrar.

**Eventos**: No textbox da razão social deverá ser implementado um código que confira se digitou um nome correto, a mesma coisa deve acontecer em Nome Fantasia. No combo box do tipo deverá selecionar se é pessoa física ou jurídica,se for física deverá habilitar a opção CPF e bloquear a opção CNPJ, se for jurídica deverá habilitar o CPNJ e bloquear o CPF, em CNPJ deverá ter um textbox para digitar o CNPJ e tem que conferir se é real ou não o CNPJ digitado, o mesmo do CNPJ é válido para o CPF. No textbox do telefone deverá ter espaço para digitar o DD e o número de 8 dígitos em seguida. Em celular deverá ter espaço para o DD e 9 dígitos em seguida. Em limite de crédito deve-se ter um combo box com a opção de "igual a" ou "menor que" e no textobox ao lado deverá ser um textbox numérico e deverá aceitar somente caracteres referentes a valor, e caso não tenha uma vírgula indicando 2 casas decimais colocar ",00". Deverá ter um ComboBox de status e um ComboBox do tipo do fornecedor, por último ao acionar o evento click do mouse no botão "Filtrar" inserir os dados no método "ObterFornecedores" da classe "Fornecedor"

**Layout**



**Classe**



## Consulta de lançamentos

(Parte feita por: Giovane)

**Objetivo:** Consultar as contas que foram lançadas no sistema

**Enunciado:**

**Descrição do relatório:** O cabeçalho da tela, a barra de ferramentas e a área lateral de abas e resumos segue o padrão de telas base para módulos. A área variável no centro da tela possui cabeçalho com os nomes dos campos referentes a contas: Status, Emissão Doc., Vencimento, Pagamento, Descrição, Fornecedor, Valor, Conta, N°Doc., Banco e Situação; logo abaixo, em ordem crescente de data de vencimento, segue a lista das contas lançadas com os dados referenciados pelo cabeçalho. Cada linha possuirá uma cor de fundo e de letras, onde seu arranjo indicará a situação da conta, juntamente ao campo Status (representado por um quadrado com alguma cor), os demais campos das contas estarão no formato texto.

**Eventos:** Quando este programa é disparado, a data de hoje vem como parâmetro para o método ObterTodasAsContas, que carrega a lista com todas as contas, indique com o arranjo correto de cores o texto, o Status e a linha de cada conta e mostre a parte da lista a partir das contas que estão vencidas. Quando o Status de uma conta está como “recusada”, no evento clique da conta execute o método ExibirMotivoRecusa da classe Parcela, Retorno.

**Layout:**

****

**Classes:**

ObterTodasAsContas: Retorna todos os registros de contas do sistema, com todos os dados referentes a elas.

ExibirMotivoRecusa: O sistema lê o arquivo retorno, conforme o layout de arquivo retorno, e compara o campo que indica o código do motivo de recusa com a lista de mensagens de motivo de recusa, até encontrar o referente campo. Por fim, mostra a mensagem com o motivo da recusa ao usuário.

## Consulta de conta já baixada no sistema

(Parte feita por: Vitor)

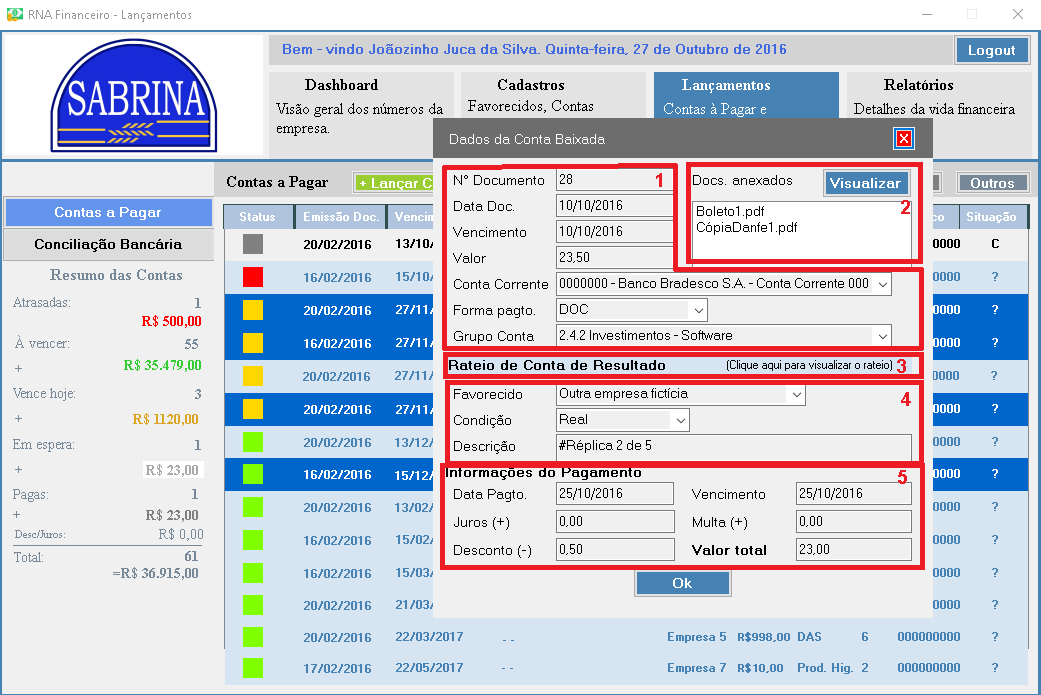
**Objetivo:** Essa tela tem como objetivo exibir todos os dados da conta já baixada no sistema.

**Enunciado:**

**Descrição da tela:**  O botão na parte 2 da tela terá um evento associado, na parte 1 da tela há os textbox Número do documento, Data do documento, Vencimento, sendo que cada textbox exibirá seus respectivos dados que foram inseridos no cadastro da conta e em nenhum dos textbox é possível realizar a alteração dos dados ( somente exibição), seguido dos textbox há os combobox Conta corrente, Forma de pagamento, Grupo de conta, onde serão exibidos seus respectivos dados, não possibilitando a alteração de tais. Na parte 2 da tela há um botão Visualizar e uma Listbox que exibirá os anexos da conta. Na parte 3 há os textbox Investimentos (Percentual), Total (Percentual), Investimentos (Valor sugerido), Total (Valor sugerido), que exibirá seus respectivos dados não possibilitando a alteração. Na parte 4 da tela há os combobox Favorecido e Condição que exibirá seus respectivos dados não possibilitando a alteração, e há também um textbox que exibirá o número da réplica não possibilitando a alteração. Na parte 5 da tela há os textbox Data de pagamento, Juros, Multa, Desconto, Valor total que exibirá seus respectivos dados sem a possiblidade de alteração.

**Eventos:** Quando esse programa é disparado, vem como parâmetro o código da conta, e com ele execute o método ObterConta da classe Conta a pagar, e exiba cada dado em seus respectivo campo de forma que o usuário não possa altera-lo. No evento clique do botão Visualizar execute o método VisualizarAnexo da classe Conta a paga, Anexo; usando como parâmetro o código da conta selecionada, e então exibida o anexo ao qual o usuário selecionou em uma janela separada. E após o fechamento de tal, retorne a janela de consulta da conta ao qual o usuário estava antes de visualizar o anexo.

**Layout:**



**Classes:**



VisualizarAnexo – Exibe o anexo que está associado com a conta.

ObterConta – Busca por contas, conforme o parâmetro fornecido.

## Consulta de lançamentos por filtros

(Parte feita por: Vitor)

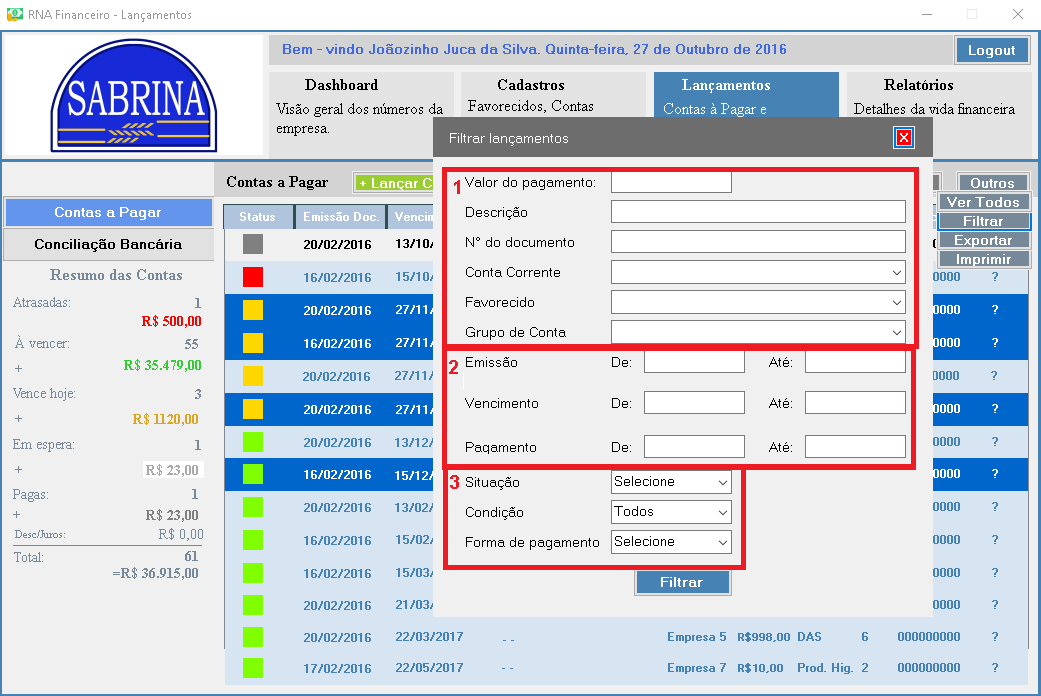
**Objetivo:** Essa tela tem como objetivo dar opções ao usuário para que ele possa filtrar seus resultados de contas a pagar.

**Enunciado:**

**Descrição da tela:** O botão na parte 3 terá um evento associado. Na parte 1 há os textbox Valor do pagamento, Descrição, Número do documento e os Combobox Conta corrente, Favorecido, Grupo da conta. Na parte 2 da tela há os textbox De e Até para os seguintes campos: Emissão, Vencimento, Vencto Original e Pagamento. Na parte 3 da tela há os combobox Situação, Condição, Forma de pagamento e Conciliada(s).

**Eventos:** No evento clique do botão Filtrar verifique se: o textbox Valor do pagamento possui somente valor numérico, textbox Descrição possui somente texto, textbox Número do documento possui somente números, os textbox De – Até dos campos Emissão, Vencimento e Pagamento possuem somente números e são datas válidas, e se todos os combobox possuem alguma opção selecionada, para realizar a filtragem é necessário que no mínimo se tenha um campo preenchido ou selecionado. E então execute o método FiltarContas da classe Conta a pagar, e em seguida exiba na tela as contas filtradas.

**Layout:**

****

**Classes:**

. 

FiltrarContas: Buscar e exibir contas com dados iguais aos que foram entrados na tela Filtrar.

## Consulta de contas conciliadas

(Parte feita por: Giovane)

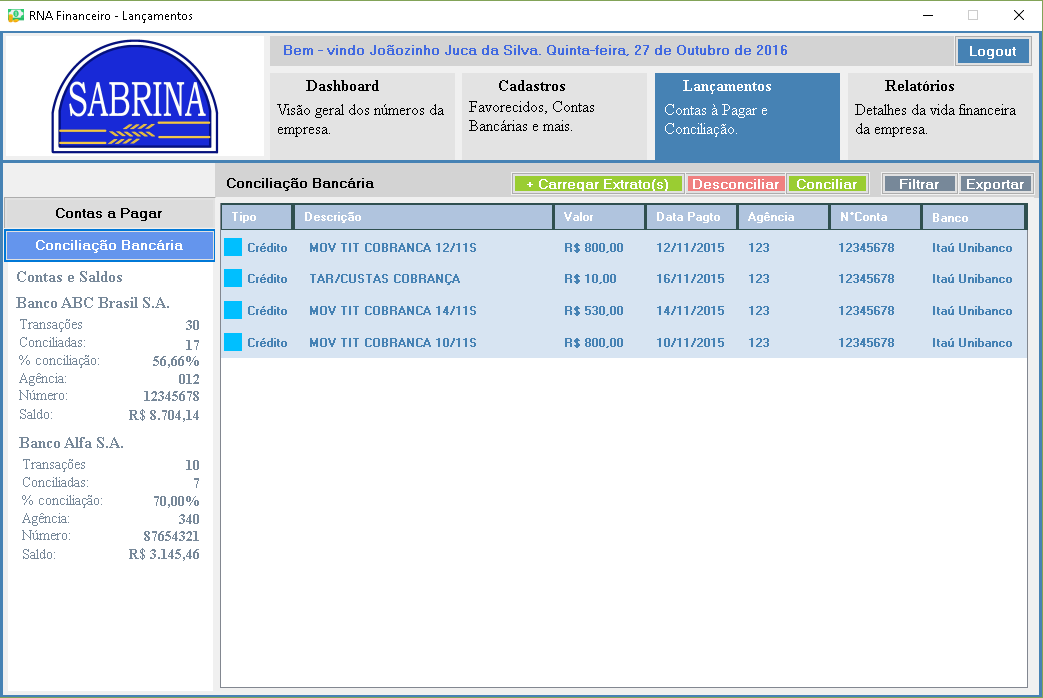
**Objetivo:** Consultar as contas que foram conciliadas no sistema

**Enunciado:**

**Descrição do relatório:** O cabeçalho da tela, a barra de ferramentas e a área lateral de abas e resumos segue o padrão de telas base para módulos. A área variável no centro da tela possui cabeçalho com os nomes dos campos referentes a contas conciliadas: Tipo, Descrição, Valor, Data Pagto, Agência, N°Conta e Banco; logo abaixo, em ordem decrescente de data de conciliação, segue a lista das contas conciliadas com os dados referenciados pelo cabeçalho. Todos os campos das contas estarão no formato texto.

**Eventos:** Quando este programa for disparado, execute o método FiltrarContas da classe Conta a pagar, passando como parâmetro o valor “verdadeiro” para o campo booleano “Situação”. Execute, em seguida o método ObterContasFiltradas, para obter os dados das contas conciliadas, mostre somente os dados indicado pelo cabeçalho.

**Layout:**

****

**Classes:**

FiltrarContas: Separa as contas, dentre todas, que atendem aos parâmetros passados

ObterContasFiltradas: Retorna os valores filtrados no método FiltrarContas

## Consulta de contas conciliadas por filtros

(Parte feita por: Vitor)

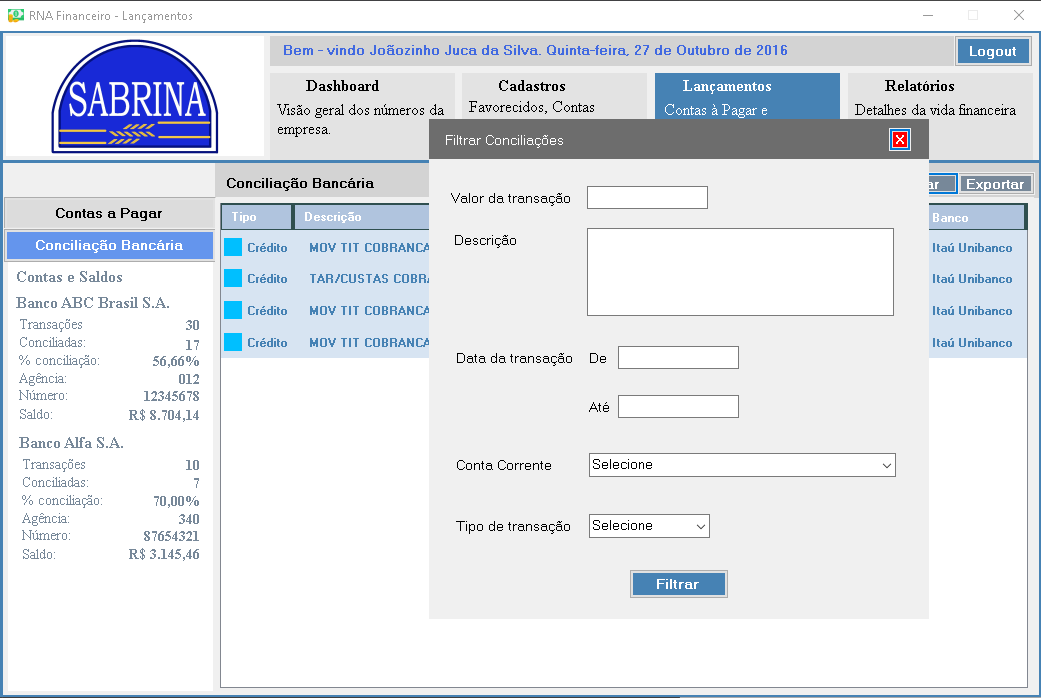
**Objetivo:** Essa tela tem como objetivo oferecer opções para o usuário pode filtrar seus resultados de conciliações.

**Enunciado:**

**Descrição da tela:** O botão na parte inferior da tela terá um evento associado. Na tela há os textbox de Valor de transação, Descrição, Data da transação (De – Até) e os combobox Conta corrente, Tipo de transação e Situação.

**Eventos:** No evento clique do botão Filtrar verifique se a textbox Valor de transação possui apenas valor numérico, o textbox Descrição possui apenas texto, os textbox Data da transação possui apenas números e se a data informada é uma data válida, se os combobox a seguir possuem alguma opção selecionada, Conta corrente, Tipo de transação, Situação. Para realizar a filtragem é necessário que no mínimo um campo esteja preenchido ou selecionado. E então execute o método FiltrarMovimentacoes da classe Movimentações, passando o parâmetro Não Nulo para o atributo DescrContaExtra.

**Layout:**

****

**Classes:**



FiltrarMovimentações: Buscar e exibir movimentações com dados iguais aos que foram entrados na tela Filtrar.

ObterMovimentacoes: Exibe o movimento na tela.

## Consulta de contas a pagar

(Parte feita por: Vanessa)

**Objetivo:** Gerar relatório de contas a pagar

**Enunciado  
Descrição da tela:** Na tela de relatório de contas a pagar na parte superior contém uma label com o texto Gerar relatório de contas a pagar, abaixo na sequência há uma label com o texto Digite o Intervalo das datas, uma label com o texto De e na frente uma text box informando a data inicial no formato dd/mm/aaa , e uma label com o texto Até com uma textbox com a data final no formato dd/mm/aaaa. Uma label para selecionar o tipo de data com três botões abaixo da label, um botão Pagamentos, um botão Emissão, um botão Vencimento, cada um com um evento associado. Uma label contendo o texto selecione as contas no plano, com uma checkbox que já vem carregada com os planos de conta existentes no sistema. Uma label contendo o texto selecione o tipo de Lançamento com três botões abaixo da label, um botão Ambos, que corresponde tanto a contas parceladas como a contas únicas, um botão Real para contas com parcela única, um botão Prevista para contas parceladas, cada um com um evento associado. Há uma label contendo o texto: Informe um fornecedor ou deixe em branco para todos, abaixo da label no lado esquerdo há um combobox selecionar todos com todos os fornecedores, no lado esquerdo há um botão que quando selecionado exibe que um combobox com a opção de escolher mais de um fornecedor. Uma label contendo o texto selecione um status com quatro botões abaixo da label, um botão Todos, que corresponde as contas pagas, atrasadas e a vencer, um botão Paga para contas que já foram pagas, um botão Atrasada para contas em atraso e um botão A vencer para contas com data a vencer, cada um com um evento associado. No fim da tela há o botão Gerar relatório com um evento associado.

**Eventos:** No evento click da textbox, com a data inicial, exibe um calendário para escolher a data inicial, no evento click da textbox, com a data final, exibe um calendário para escolher a data final.No evento click do botão pagamento filtra os dados das contas pela data do vencimento, o mesmo ocorre ao selecionar o botão emissão filtrando pela de emissão e o botão vencimento filtra pela data de vencimento Na checkbox com os planos de contas ao escolher uma conta o evento select é acionado. No evento click do botão Ambos filtra os dados das contas pela lançamento de contas com parcela única e contas parceladas, o mesmo ocorre ao selecionar o botão Real filtrando pelo lançamento de parcelas únicas e o botão Prevista filtra pelo lançamento de contas parceladas. No evento select da combobox selecionar todos é escolhido um fornecedor. No evento click do botão Adicionar fornecedor uma combobox com todos os fornecedores é exibida. No evento click do botão Todos filtra as contas pagas, atrasadas, e a vencer, o mesmo ocorre ao selecionar o botão Paga filtrando todas as contas pagas e o botão Atrasada filtra as contas que estão atrasadas e o botão A vencer filtra as contas com data de pagamento a vencer. No evento click do botão Gerar relatório verifique se a data inicial e a data final são datas válidas e se não foram deixadas em branco, e se a data inicial não é maior que a data final e quais filtros foram selecionadosl. Dependendo de qual botão foi selecionado um método será executado. No evento click de um dos botões que pode ser Pagamento, Emissão ou Vencimento executa o método filtrar por tipo de data da classe Contas a pagar, baseados no botão selecionado. No evento click do botão que pode ser Ambos, real ou previsto o evento click executa o método Filtrar por tipo de pagamento da classe Contas a pagar. No evento click do botão que pode ser Todos, Paga, Atrasada, A vencer o evento click executa o método filtrar por status da classe Contas. Na sequência é executado o método exibir relatório da classe Contas a pagar.

**Layout**

****

**Classes**

Filtrar por tipo de data: Este método filtra as contas por pagamento, emissão, ou vencimento.

Filtrar por data de pagamento: Este método filtra as contas por ambos, real ou prevista.

Filtrar por status: Este método filtra as contas por todas as contas, pagas, atrasadas, ou a vencer.

Exibir relatório de contas a pagar: Este método exibe o relatório de contas a pagar.

## Consulta de auditoria de ações no sistema

(Parte feita por: Giovane)

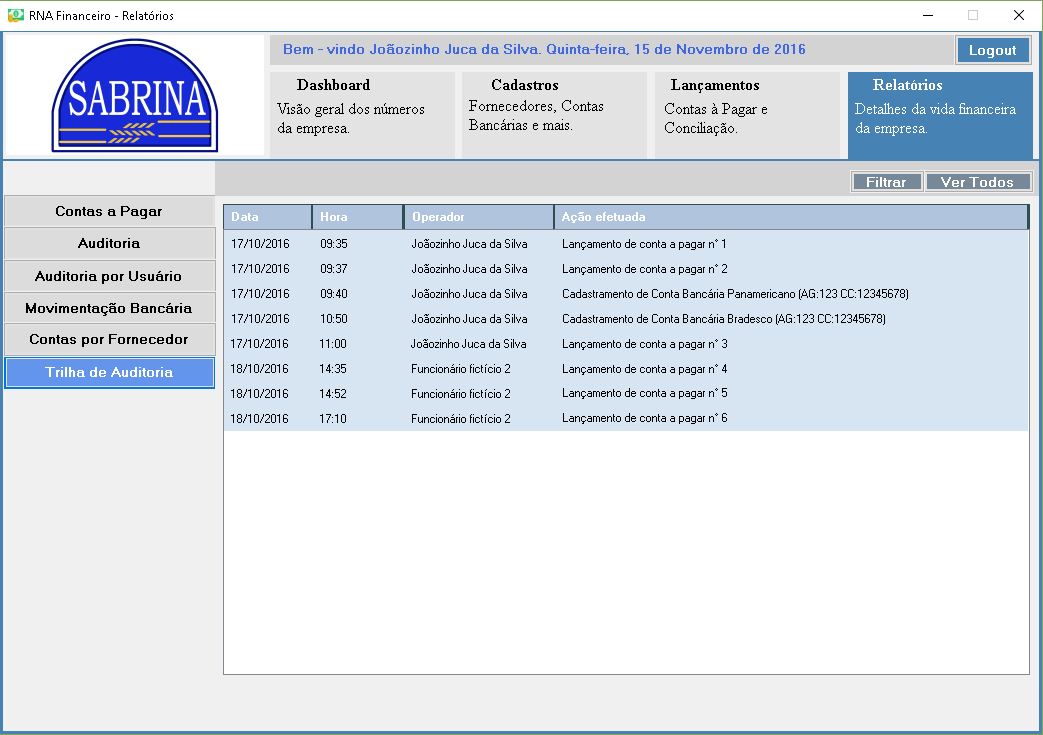
**Objetivo:** Consultar a auditoria das ações executadas no sistema

**Enunciado:**

**Descrição do relatório:** O cabeçalho da tela, a barra de ferramentas e a área lateral de abas e resumos segue o padrão de telas base para módulos. A área variável no centro da tela possui cabeçalho com os nomes dos campos referentes à auditoria das ações no sistema: Data, Hora, Operador e Ação efetuada; logo abaixo, em ordem decrescente de data e hora de operação, segue a lista dos registros de operações com os dados referenciados pelo cabeçalho. Todos os campos das contas estarão no formato texto.

**Eventos:** Quando esse programa for disparado, execute o método ObterAuditoria da classe Auditoria, mostre todas as ações executadas por dia, em ordem decrescente de hora e data.

**Layout:**

****

**Classes:**

ObterAuditoria: Retorna todos os registros de auditoria de ações no sistema.

## Consulta de auditoria de ações no sistema por filtros

(Parte feita por: Bruno)

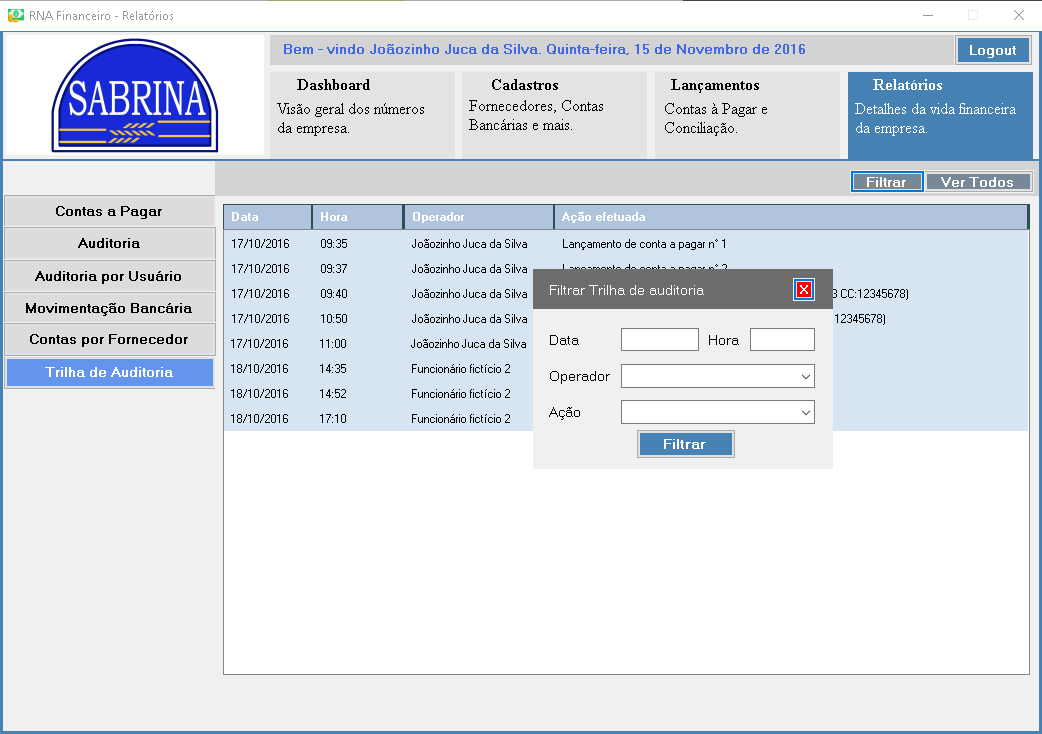
**Objetivo**: Procurar o a auditoria de ações.

**Enunciado**:

**Descrição da tela:** A tela de filtrar auditoria de ações terá um maskedtextbox para digitar a data, e outro para digitar a hora. Terá dois combobox, um para o operador e outro para ação e terá um botão que ao clicar você vai filtrar o que deseja encontrar.

**Eventos**: Na data deverá ter um maskedtextbox no formato de data (DD/MM/AAAA), em hora deverá ter um maskedtextbox no formato hora (00h00), em operador deverá ter um combox com os operadores da auditoria e em ação um combobox com as ações da auditoria, e ao usar o evento click do mouse no botão “Filtrar” deverá relacionar os valores digitados com o método “FiltrarAuditoria” da classe “Auditoria”.

**Layout**



**Classe**

FiltrarAuditoria = Consultará um banco de dados X, e uma certa tabela onde estarão os dados de todas Audotorias e irá comparar se existe a auditoria procurada.

ObterAuditoriaFiltrada = Retorna os valores filtrados no método FiltrarAuditoria.

## Relatório de contas a pagar

(Parte feita por: Vanessa)

**Objetivo:** Este caso de uso permite imprimir o relatório de contas a pagar

**Enunciado:  
Descrição do relatório:** O relatório tem cabeçalho padrão com logo, título do relatório, nome da empresa, o período da data inicial até a data final,o tipo de data, o tipo de pagamento, eo status da conta. No corpo do relatório deve constar os dados sobre a situação da conta(atrasada,paga,a vencer), emissão, a data que foi emitido o documento, vencimento que é a data do vencimento da conta, descrição que corresponde a alguma informação extra, ex. se for um acordo. Fornecedor é razão social do fornecedor, valor é o valor da conta, a Conta corresponde ao plano de contas ao qual o fornecedor está cadastrado, Nº doc. corresponde ao número do documento que pode ser o nº da nota fiscal, ordem de serviço ou documento equivalente, banco coresponde ao banco associado a conta no lançamento ou pagamento,a condição é real para conta com parcela única e prevista para contas parceladas. No rodapé conta a data e horário da impressão, o usuário, e o número de páginas do relatório.

**Eventos:** No evento click da textbox, com a data inicial, exibe um calendário para escolher a data inicial, no evento click da textbox, com a data final, exibe um calendário para escolher a data final receba um intervalo de datas, o tipo de data(pagamento, emissão, documento) através do método filtrar por tipo de data, o tipo de lançamento (ambos, real, previsto) através do método filtrar por tipo de lançamento, tipo de status( todos, paga, atrasada, a vencer) através do método filtrar por status. são da classe contas a pagar.Imprima as linhas 1, 2, 3 do cabeçalho do relatório faça um looping para imprimir todas as contas filtradas execute o método obterContas da classe PagarContas, cada página deve conter 35 linhas, ao ultrapassar essa quantidade de linhas ir para a página seguinte e imprimir o cabeçalho do relatório e seguir a sequência dos passos anteriores até obter todas as contas filtradas.

**Layout**

|  |  |  |  |
| --- | --- | --- | --- |
| https://scontent.fsod2-1.fna.fbcdn.net/v/t34.0-12/17690208_1271519009597722_1879841335_n.jpg?oh=ef8fdf67d4f90a47190f456434b5135d&oe=58EBD3CE | **RELATÓRIO CONTAS A PAGAR** | **PANIFICADORA SABRINA** | |
| Período de \_\_/\_\_/\_\_\_\_ à \_\_/\_\_/\_\_\_\_  Tipo de data: | | | Tipo de pagamento:  Situação da conta: |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | **Situação** | **Emissão** | **Vencimento** | **Descrição** | **Fornecedor** | **Valor** | **Conta** | **Nº doc.** | **Banco** | **Condição** | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |   Impresso em: \_\_/\_\_/\_\_\_\_ \_\_:\_\_  Usuário: Vanessa Oliveira  Página: | | | |

**Classes**



Filtrar por tipo de data: Este método filtra as contas por pagamento, emissão, ou vencimento.

Filtrar por data de pagamento: Este método filtra as contas por ambos, real ou prevista.

Filtrar por status: Este método filtra as contas por todas as contas, pagas, atrasadas, ou a vencer.

Exibir relatório de contas a pagar: Este método exibe o relatório de contas a paga

## Relatório de auditoria

(Parte feita por: Vinicius)

**Enunciado:**

**Descrição do relatório:** O relatório tem cabeçalho padrão com logo, título do relatório, nome da empresa, o período da data inicial até a data final. No corpo do relatório deve constar os relatórios de auditoria dos usuários, data, hora, nome do operador, e ação realizada. A data corresponde ao dia, mês e ano da ação realizada, a hora corresponde a hora na qual a ação de auditoria foi realizada, o Operador corresponde ao nome do usuário que realizou a ação, e Ação efetuada corresponde a descrição da ação que foi realizada. No rodapé conta a data e horário da impressão, o usuário, e o número de páginas do relatório. No rodapé conta a data e horário da impressão, o usuário, e o número de páginas do relatório.

**Eventos:** O método FiltrarAuditoria recebe como parâmetro a data inicial e a data final obtidos por um calendário exibido através do evento click no textbox, digite o intervalo de datas. O método ListarAuditoriaFiltrada da classe Auditoria recebe como parâmetro os dados obtidos pelo Método FiltrarAuditoria da classe Auditoria, e exibe as os dados Data (obtido do atributo data da classe Auditoria), Hora (obtido do atributo Hora da classe Auditoria, ação), Operador (obtido do atributo Nome da classe Usuário, pelo método ObterUsuario), e Ação efetuada (obtido do atributo Ação da classe Auditoria, ação). Imprima o cabeçalho da tabela relatório de auditoria e faça um looping para imprimir todas os relatórios de auditoria filtrados, cada página deve conter 35 linhas, ao ultrapassar essa quantidade de linhas ir para a página seguinte e imprimir o cabeçalho do relatório e seguir sequência dos passos anteriores até obter todos os relatórios de auditoria filtrados.

**Layout**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| https://scontent.fsod2-1.fna.fbcdn.net/v/t34.0-12/17690208_1271519009597722_1879841335_n.jpg?oh=ef8fdf67d4f90a47190f456434b5135d&oe=58EBD3CE   |  | | --- | |  | |  |  |  |  |  |  |  |
|  |  |  | **RELATÓRIO DE AUDITORIA** | |  | **PANIFICADORA SABRINA** | |
|  |  |  |  | | |  |  |
|  |  |  |  |  |  |  |  |
| Período de 10/03/2017 a 20/04/2017 | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Data** | **Hora** | **Operador** | **Ação efetuada** | | | | |
| 10/03/2017 | 10:00 | João Exemplar | Lançamento de conta a pagar número 2 | | | | |
| 11/03/2017 | 12:00 | Maria Exemplar | Cadastramento da Conta Bancária Bradesco (AG:00 CC:00) | | | | |
| 20/04/2017 | 16:27 | João Exemplar | Edição de conta a pagar número: 20232022 | | | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  | Impresso em: \_\_/\_\_/\_\_\_\_ \_\_: \_\_ | | |
|  |  |  |  |  | Usuário: Vinicius | | |
|  |  |  |  |  | Página: 1 de 1 | | |

**Classes**

FiltrarAuditoria: Este método obtém o período de data inicial e data final da realização dos relatórios de auditoria.

ListarAuditoriaFiltrada: Este método obtém o período de data pelo método FiltrarAuditoria da classe Auditoria e exibe os relatórios de auditoria registrados durante o período de datas.

## Relatório de auditoria por usuário

(Parte feita por: Bruno)

**Objetivo:** Este caso de uso permite imprimir o relatório de contas a pagar

**Enunciado**

**Descrição do relatório:** O relatório tem cabeçalho padrão com logo, título do relatório, nome da empresa, a data inicial até a data final selecionada, seguidos dos dados do usuário ao qual relatório é referente, junto com os seguintes dados: Nome, Telefone, Celular e E-mail. No corpo do relatório terá Hora, Data, Ação efetuada.

**Eventos**: Quando esse programa é disparado, o nome do usuário vem como parâmetro. Então, no início do programa, obtenha o usuário a partir do método ObterUsuarios() da classe Usuário e filtre a auditoria do usuário com o período escolhido e com o nome do usuário, utilizando o método FiltrarAuditoria() da classe Auditoria e então obtenha os dados usando o método ObterAuditoriaFiltrada() da classe Auditoria. Imprima o cabeçalho padrão, e em seguida os dados do usuário obtidos no método ObterUsuarios() e então os dados da auditoria obtidos no método ObterAuditoriaFiltrada(). Caso passe de 35 linhas imprima o cabeçalho com os dados do usuário novamente e continue imprimindo as auditorias.

**Layout**

|  |  |  |  |
| --- | --- | --- | --- |
| https://scontent.fsod2-1.fna.fbcdn.net/v/t34.0-12/17690208_1271519009597722_1879841335_n.jpg?oh=ef8fdf67d4f90a47190f456434b5135d&oe=58EBD3CE | **RELATÓRIO AUDITORIA POR USUÁRIO** | **PANIFICADORA SABRINA** | |
| Período de \_\_/\_\_/\_\_\_\_ à \_\_/\_\_/\_\_\_\_ Nome Usuário: \_\_\_\_\_\_\_\_\_\_\_\_ E-mail: \_\_\_\_\_\_\_\_\_\_\_\_\_ Telefone: (\_\_) \_\_\_\_-\_\_\_\_  Celular: (\_\_) \_\_\_\_\_-\_\_\_\_ | | |  |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Data** | **Hora** | **Ação Efetuada** | |  | \_\_/\_\_/\_\_\_ |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  |   Impresso em: \_\_/\_\_/\_\_\_\_ \_\_:\_\_  Usuário: Bruno  Página: | | | |

**Classes**

ObterUsuarios() = Obtém os dados do usuário

FiltrarAuditoria() = Filtra a auditoria

ObterAuditoriaFiltrada() = Obtém os dados da auditoria filtrada.

## Relatório de movimentação bancária

(Parte feita por: Giovane)

**Objetivo:** Este caso de uso permite imprimir o relatório de movimentação bancária

**Enunciado:  
Descrição do relatório:** O relatório tem cabeçalho padrão com logo, título do relatório e nome da empresa. No corpo há um “sub cabeçalho” com o período da data inicial até a data final de movimentações bancárias, a data de início da conta bancária, o nome do banco, número da agência, número da conta bancária e tipo de conta. Abaixo vem o valor inicial da conta bancária, a tabela com os dados de data e tipo de movimentação; vencimento, descrição e valor total do título; e valor parcial da conta bancária. Cada linha equivale a movimentação diferente. No final da tabela consta o valor final da conta bancára. No rodapé conta a data e horário da impressão, o usuário, e o número de páginas do relatório.

**Eventos:** Execute o método GerarRelatórioCtBanc da classe Conta bancária recebendo o período de uma data até outra e o número da conta bancária. Aplique o filtro às contas da classe Conta a pagar, através do método FiltrarContas, passando como parâmetros a data inicial e final do período estabelecido, o status “paga” e o número da conta bancária. Obtenha os dados da conta com o método ObterContasFiltradas. Execute o método FiltrarMovimentacoes passando como parâmetro o número da conta bancária e a o período De e Até estabelecido, depois execute o método ObterMovimentacoesFiltradas. Imprima o cabeçalho da tabela de movimentações e faça um looping para imprimir todas as movimentações do período, cada página deve conter 35 linhas, se ultrapassar essa quantidade de linhas imprima o rodapé e vá para a página seguinte imprimindo o cabeçalho do relatório e seguindo a sequência dos passos anteriores até obter todas as contas filtradas. Cada rodapé terá a sua numeração em relação ao total de páginas.

**Layout**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://scontent.fsod2-1.fna.fbcdn.net/v/t34.0-12/17690208_1271519009597722_1879841335_n.jpg?oh=ef8fdf67d4f90a47190f456434b5135d&oe=58EBD3CE | **RELATÓRIO DE MOVIMENTAÇÃO BANCÁRIA** | | **PANIFICADORA SABRINA** | |
| Período de 10/02/2016 a 27/04/2016  Data de início da conta bancária: 10/02/2016 | | Nome do banco: Bradesco  Número da agência: 1234 | | Número da conta bancária: 12345-6  Tipo de conta: Corrente |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | | | | **Valor inicial da conta bancária:** | | R$ 2.452,80 |  | |  | **Movimentação** | | **Título** | | | | **Valor parcial conta bancária** |  | |  | **Data Pgto** | **Tipo** | **Vencimento** | **Descrição** | | **Valor total** |  | |  | 12/02/2016 | Pagamento | 16/02/2016 | Farinha de Trigo | Fá rinhas | | R$ 153,50 | R$ 2.299,30 |  | |  | 15/02/2016 | Pagamento | 16/02/2016 | Réplica 2 de 3 de Sofá verde | Soft sofás | | R$ 500,00 | R$ 1.799,30 |  | |  | 20/02/2016 | Pagamento | 20/02/2016 | Fermento biológico | BigFast | | R$ 52,90 | R$ 1.746,40 |  | |  | 12/03/2016 | Pagamento | 16/03/2016 | Réplica 3 de 3 de Sofá verde | Soft sofás | | R$ 500,00 | R$ 1.246,40 |  | |  | 14/03/2016 | Pagamento | 16/03/2016 | Farinha de Trigo | Fá rinhas | | R$ 153,50 | R$ 1.092,90 |  | |  | 15/03/2016 | Pagamento | 15/03/2016 | Margarina sem sal | Nemoana | | R$ 47,00 | R$ 1.045,90 |  | |  | 10/04/2016 | Pagamento | 02/04/2016 | Mesa de madeira | Madeireira madeeeeira | | R$ 230,00 | R$ 815,90 |  | |  | 27/04/2016 | Pagamento | 28/04/2016 | Leite | Yes Vaca | | R$ 325,00 | R$ 490,90 |  | |  |  | | | | **Valor final da conta bancária:** | | R$ 490,90 |  | | | | | |
| Impresso em: 27/04/2016 15:42  Usuário: Giovane de Lucas  Página: 1 de 1 | | | | |

**Classes**

GerarRelatórioCtBanc: Recebe as datas De e Até do período e o número da conta bancária e executa o método FiltrarContas da classe Conta a pagar, passando o período, o status “paga” e o número da conta bancária como parâmetros e obtém as contas com o método ObterContasFiltradas. Executa o método FiltrarMovimentacoes da classe Movimentações passando os parâmetros de período e número de conta bancária e obtém as movimentações através do método ObterMovimentacoesFiltradas. Retorna os dados de Data e Tipo de movimentação; Vencimento, Descrição e Valor total de títulos da classe Conta a pagar; e o valor inicial, os valores parciais e o valor final da conta bancária da classe Movimentação.

FiltrarContas: Separa as contas, dentre todas, que atendem aos parâmetros passados

ObterContasFiltradas: Retorna os valores filtrados no método FiltrarContas

FiltrarMovimentacoes: Separa as movimentações, dentre todas, que atendem aos parâmetros passados

ObterMovimentacoesFiltradas: Retorna os valores filtrados no método FiltrarMovimentecoes

## Relatório de contas por fornecedor

(Parte feita por: Vitor)

**Enunciado**

**Descrição do relatório:** O relatório tem cabeçalho padrão com logo, título do relatório, nome da empresa, o período da data inicial até a data final seguidos dos dados do fornecedor ao qual relatório é referente, junto com os seguintes dados: Nome fantasia, Razão social, Celular, Telefone, CPF, E-mail, Limite de credito, CNPJ e Tipo. No corpo do relatório há os relatórios das contas relativo ao fornecedor, onde há os seguintes dados: Nº do documento, Data do documento, data de emissão, data de renegociação data de pagamento original, data de conciliação, forma de pagamento, situação, Nº de parcelas, desconto, Taxa e Valor.

**Eventos**: Quando esse programa é disparado, o código de identificação do fornecedor vem como parâmetro. Então, no início do programa, obtenha os dados da conta referente aquele fornecedor através do método ObterContasFornecedor da classe Conta a Pagar. Com o código da conta, execute o método ObterContas da classe Conta a Pagar; imprima o cabeçalho e as linha 1 ,2, e 3 do relatório, então imprima o cabeçalho das contas e faça um loop para imprimir todas as contas referente a esse fornecedor, dentro do loop, executo o método ObterContaFornecedor da classe Conta a Pagar, com o código da conta execute o método ObteContas.... Some 1 no contador de linhas, e se superar 35, imprimar os cabeçalhos novamente…imprima a linha das contas....

**Layout**



**Classes:**

FiltrarFornecedor: lista as contas, conforme o fornecedor.

ObterConta: lista as contas, conforme os parâmetros fornecidos