

Curso de Análise e Desenvolvimento de Sistemas

Diego de Siqueira Garcia

Lucio dos Santos Rosa

Thiago Henrique Mendes Prestes

William Matheus Vasconcelos da Silva

Documentação de Desenvolvimento de Software

DEKL Sistema Web para Contas a Pagar

Sorocaba

Novembro - 2018



**Diego de Siqueira Garcia**

**Lucio dos Santos Rosa**

**Thiago Henrique Mendes Prestes**

**William Matheus Vasconcelos da Silva**

Documentação de Desenvolvimento de Software

DEKL Sistema Web para Contas a Pagar

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Sorocaba –FATEC, como parte dos pré-requisitos para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas

Orientadora: Prof.ª Cristiane Palomar Mercado

Sorocaba

Novembro - 2018

**Dedicatória**

Dedicamos este trabalho às nossas famílias

que nos maiores momentos de desamino e cansaço

estavam lá para nos apoiarem e incentivarem a continuar sempre adiante.

**Agradecimento**

Agradecemos a todos nossos professores que nos apoiaram

e orientaram a conclusão desse trabalho, em especial aos nossos

professores Levi Munhoz, Cristiane Palomar, Sergio Bernardo e Daniela Arruda

que nos orientaram no desenvolvimento deste projeto do início ao fim.

Resumo

O presente trabalho tem a proposta de apresentar uma solução informatizada para as situações que se mostraram insuficientes ou inadequadas ao setor financeiro de contas a pagar de uma microempresa, como por exemplo, a ausência de informações importantes no registro da duplicata como juros de mora e multa a ser aplicada, ausência de um maior e mais preciso controle de acesso para os colaboradores que não tem um perfil administrativo e a baixa visibilidade das duplicatas que estão com suas datas de vencimento próximas do pagamento, fazendo com que o planejamento, controle e execução dos títulos sejam comprometidos.

O objetivo deste projeto é trazer um novo sistema para o gerenciamento dos títulos a pagar utilizando tecnologia web para maior compatibilidade no ambiente da empresa. Também serão incluídas as informações que atualmente estão ausentes para que haja um maior planejamento no pagamento assim como uma ferramenta de auxílio para isso. A questão do perfil de acesso também será tratada de maneira que somente usuários com as permissões adequadas possam efetuar determinadas ações assim como também o sistema receberá uma maior atenção com os títulos que estão por vencer uma vez que esta corresponde à sua principal atividade e maior fluxo.

**Lista de Figuras**

[Figura 1 - Casos de Uso Alto Nível 15](#_Toc530677751)

[Figura 2 - Arquitetura da aplicação Atual 40](#_Toc530677752)

[Figura 3 - Componentes do SW 44](#_Toc530677753)

[Figura 4 - Diagrama de Classes 45](#_Toc530677754)

[Figura 5 - Modelo Lógico Completo 48](#_Toc530677755)

[Figura 6 - Modelo Lógico (parte 1) 49](#_Toc530677756)

[Figura 7 - Modelo Lógico (parte 2) 50](#_Toc530677757)

[Figura 8 - Modelo Lógico (parte 3) 51](#_Toc530677758)

[Figura 9 - Modelo Lógico (parte 4) 52](#_Toc530677759)

[Figura 10 - DS Auditoria 53](#_Toc530677760)

[Figura 11 - DS Cadastrar Usuário 54](#_Toc530677761)

[Figura 12 - DS Incluir Fornecedor 55](#_Toc530677762)

[Figura 13 - DS Consulta Credor (parte 1) 56](#_Toc530677763)

[Figura 14 - DS Consulta Credor (parte 2) 57](#_Toc530677764)

[Figura 15 - DS Consulta Credor (parte 3) 58](#_Toc530677765)

[Figura 16 - DS Consulta Credor (parte 4) 59](#_Toc530677766)

[Figura 17 - DS Incluir Contas 60](#_Toc530677767)

[Figura 18 - DS Permissão de Acesso 61](#_Toc530677768)

[Figura 19 - DS Login 62](#_Toc530677769)

[Figura 20 - DS Pagamento de Contas 63](#_Toc530677770)

[Figura 21 - DS Redefinir Senha 64](#_Toc530677771)

[Figura 22 - DS Relatório de Contas à Pagar 65](#_Toc530677772)

[Figura 23 - DS Simulador de Contas 66](#_Toc530677773)

[Figura 24 - DS Triagem de Contas 67](#_Toc530677774)

[Figura 25 - Diagrama de Pacotes 68](#_Toc530677775)

[Figura 26 - DE Auditoria 69](#_Toc530677776)

[Figura 27 - DE Cadastrar Fornecedor 70](#_Toc530677777)

[Figura 28 - DE Cadastro de Usuário 71](#_Toc530677778)

[Figura 29 - DE Consulta Credor 72](#_Toc530677779)

[Figura 30 - DE Exportar para Planilha 73](#_Toc530677780)

[Figura 31 - DE Inclusão de Contas à Pagar 74](#_Toc530677781)

[Figura 32 - DE Login 75](#_Toc530677782)

[Figura 33 - DE Pagamento de Conta 76](#_Toc530677783)

[Figura 34 - DE Redefinir Senha 77](#_Toc530677784)

[Figura 35 - DE Simulador 78](#_Toc530677785)

[Figura 36 - DE Triagem de Contas à Pagar 79](#_Toc530677786)

[Figura 37 - Tela Página Inicial 80](#_Toc530677787)

[Figura 38 - Tela Login 81](#_Toc530677788)

[Figura 39 - Tela Cadastrar Novo Usuário 82](#_Toc530677789)

[Figura 40 - Tela Controle de Contas à Pagar 83](#_Toc530677790)

[Figura 41 - Tela Inclusão de Contas 84](#_Toc530677791)

[Figura 42 - Tela Inclusão de Fornecedores 85](#_Toc530677792)

[Figura 43 - Tela Gerenciamento de Auditoria do Sistema 86](#_Toc530677793)

[Figura 44 - Relatório Contas à Pagar 87](#_Toc530677794)

[Figura 45 - Relatório Movimentação Bancária 88](#_Toc530677795)

**Lista de Tabelas**

[Tabela 1 - Restrições e Riscos do Projeto Atual 7](#_Toc530677726)

[Tabela 2 - Limitações Operacionais 8](#_Toc530677727)

[Tabela 3 - Riscos 9](#_Toc530677728)

[Tabela 4 - Caso de Uso Baixo Nível - RF1 Inclusão de nova(s) conta(s) 16](#_Toc530677729)

[Tabela 5 - Caso de Uso Baixo Nível - RF2 Edição de Contas 17](#_Toc530677730)

[Tabela 6 - Caso de Uso Baixo Nível - RF3 Pagamento de Conta(s) 18](#_Toc530677731)

[Tabela 7 - Caso de Uso Baixo Nível – RF4 Consulta de Conta(s) 19](#_Toc530677732)

[Tabela 8 - Caso de Uso Baixo Nível – RF5 Inclusão de Novo(s) Fornecedor(es) 20](#_Toc530677733)

[Tabela 9 -Caso de Uso Baixo Nível – RF6 Edição de Fornecedor 21](#_Toc530677734)

[Tabela 10 - Caso de Uso Baixo Nível – RF7 Triagem de Contas por Fornecedores 22](#_Toc530677735)

[Tabela 11 - Caso de Uso Baixo Nível – RF8 Cadastro de Usuário 23](#_Toc530677736)

[Tabela 12 - Caso de Uso Baixo Nível – RF9 Listagem de Permissão de Acesso 24](#_Toc530677737)

[Tabela 13 - Caso de Uso Baixo Nível – RF10 Configuração do Nível de Acesso de Usuários 25](#_Toc530677738)

[Tabela 14 - Caso de Uso Baixo Nível - RF11 Simulação de Contas à Vencer 26](#_Toc530677739)

[Tabela 15 - Caso de Uso Baixo Nível – RF12 Consulta do Saldo Disponível 27](#_Toc530677740)

[Tabela 16 - Caso de Uso Baixo Nível - RF13 Inserção e/ou Edição de Valores Referentes ao Saldo 28](#_Toc530677741)

[Tabela 17 - Caso de Uso Baixo Nível - RF14 Redefinição de Senha 29](#_Toc530677742)

[Tabela 18 - Caso de Uso Baixo Nível – RF15 Auditoria de Atividades do Sistema 30](#_Toc530677743)

[Tabela 19 - Caso de Uso Baixo Nível - RF16 Consulta de Fornecedor(es) 31](#_Toc530677744)

[Tabela 20 - Caso de Uso Baixo Nível - RF17 Relatório de Contas à Pagar 32](#_Toc530677745)

[Tabela 21 - Caso de Uso Baixo Nível - RF18 Relatório de Fornecedor 33](#_Toc530677746)

[Tabela 22 - Caso de Uso Baixo Nível - RF19 Relatório de Contas Bancárias 34](#_Toc530677747)

[Tabela 23 - Caso de Uso Baixo Nível – RF20 Login no Sistema 35](#_Toc530677748)

[Tabela 24 - Caso de Uso Baixo Nível – RF21 Recuperar Senha 36](#_Toc530677749)

[Tabela 25 - Caso de Uso Baixo Nível – RF22 Cadastro de Conta Bancária 37](#_Toc530677750)

Sumário

[1. Introdução 1](#_Toc530677796)

[2. Planejamento Inicial do Software 3](#_Toc530677797)

[2.1 Situação Atual 3](#_Toc530677798)

[2.1.1 Descrição da Situação Atual 3](#_Toc530677799)

[2.2 Problemas encontrados 4](#_Toc530677800)

[2.2.1 Ausência da informação de juros ao registrar as duplicatas. 4](#_Toc530677801)

[2.2.2 Baixa visibilidade das contas a vencer 4](#_Toc530677802)

[2.2.3 Faltam níveis de acesso personalizados ao sistema 5](#_Toc530677803)

[2.3 Objetivos de Negócio do Projeto 6](#_Toc530677804)

[2.3.1 Missão 6](#_Toc530677805)

[2.3.2 Simulações de contas a vencer 6](#_Toc530677806)

[2.3.3 Triagens de contas mediante ao nível de prioridade dos fornecedores 6](#_Toc530677807)

[2.3.4 Integrações entre fornecedores cadastrados e novas contas 6](#_Toc530677808)

[2.4 Restrições e Riscos do Projeto Atual 7](#_Toc530677809)

[2.5 Limitações Operacionais 8](#_Toc530677810)

[2.6 Considerações Legais 8](#_Toc530677811)

[2.7 Considerações de Hardware / Software / Rede 8](#_Toc530677812)

[2.8 Políticas Organizacionais 9](#_Toc530677813)

[2.9 Riscos 9](#_Toc530677814)

[3. Análise de Requisitos 10](#_Toc530677815)

[3.1 Descrição da técnica utilizada para levantamento dos requisitos 10](#_Toc530677816)

[3.2 Situação Proposta 10](#_Toc530677817)

[3.3 Requisitos Funcionais 11](#_Toc530677818)

[3.4 Diagrama de Casos de Uso e Descrição dos Casos de Uso 15](#_Toc530677819)

[3.4.1 RF1 Inclusão de nova(s) conta(s). 16](#_Toc530677820)

[3.4.2 RF2 Edição de conta(s). 17](#_Toc530677821)

[3.4.3 RF3 Pagamento de Conta (s). 18](#_Toc530677822)

[3.4.4 RF4 Consulta de Conta(s). 19](#_Toc530677823)

[3.4.5 RF5 Inclusão de novo(s) fornecedor(es). 20](#_Toc530677824)

[3.4.6 RF6 Edição de fornecedor. 21](#_Toc530677825)

[3.4.7 RF7 Triagem de contas por fornecedores. 22](#_Toc530677826)

[3.4.8 RF8 Cadastro de usuário 23](#_Toc530677827)

[3.4.9 RF9 Listagem de Permissão de acesso. 24](#_Toc530677828)

[3.4.10 RF10 Configuração do nível de acesso de usuários. 25](#_Toc530677829)

[3.4.11 RF11 Simulação de contas a vencer. 26](#_Toc530677830)

[3.4.12 RF12 Consulta do saldo disponível. 27](#_Toc530677831)

[3.4.13 RF13 Inserção e/ou edição de valores referentes ao saldo. 28](#_Toc530677832)

[3.4.14 RF14 Redefinição de senha. 29](#_Toc530677833)

[3.4.15 RF15 Auditoria de atividades do sistema. 30](#_Toc530677834)

[3.4.16 RF16 Consulta de Fornecedor(es). 31](#_Toc530677835)

[3.4.17 RF17 Relatório de Contas à Pagar 32](#_Toc530677836)

[3.4.18 RF18 Relatório de Fornecedor 33](#_Toc530677837)

[3.4.19 RF19 Relatório de Contas Bancárias 34](#_Toc530677838)

[3.4.20 RF20 Login no Sistema 35](#_Toc530677839)

[3.4.21 RF21 Recuperar Senha 36](#_Toc530677840)

[3.4.22 RF22 Cadastro de Contas Bancárias 37](#_Toc530677841)

[3.5 Requisitos Não Funcionais 38](#_Toc530677842)

[4.1 Arquitetura da aplicação Atual 40](#_Toc530677843)

[4.2 Tecnologias utilizadas e APIs 41](#_Toc530677844)

[4.2.1 Bootstrap (front-end) 41](#_Toc530677845)

[4.2.2 Entity Framework 6 41](#_Toc530677846)

[4.2.3 HTML 41](#_Toc530677847)

[4.2.4 CSS 42](#_Toc530677848)

[4.2.5 SQL Server 42](#_Toc530677849)

[4.2.6 JQuery 3 43](#_Toc530677850)

[4.3 Componentes do SW 44](#_Toc530677851)

[4.4 Diagrama de Classes 45](#_Toc530677852)

[4.5 Banco de Dados 46](#_Toc530677853)

[4.5.1 Modelo Conceitual 46](#_Toc530677854)

[4.5.2 Modelo Lógico 48](#_Toc530677855)

[4.6 Diagrama de Sequência 53](#_Toc530677856)

[4.6.1 Auditoria 53](#_Toc530677857)

[Cadastrar Usuário 54](#_Toc530677858)

[4.6.2 Incluir Fornecedor 55](#_Toc530677859)

[4.6.3 Consultar Credor 56](#_Toc530677860)

[4.6.4 Incluir Contas à Pagar 60](#_Toc530677861)

[4.6.5 Permissão de Acesso 61](#_Toc530677862)

[4.6.6 Login 62](#_Toc530677863)

[4.6.7 Pagamento de Contas 63](#_Toc530677864)

[4.6.8 Redefinir Senha 64](#_Toc530677865)

[4.6.9 Relatório de Contas à Pagar 65](#_Toc530677866)

[4.6.10 Simulador de Contas 66](#_Toc530677867)

[4.6.11 Triagem de Contas 67](#_Toc530677868)

[4.7 Diagrama de Pacotes 68](#_Toc530677869)

[4.8 Diagrama Estado 69](#_Toc530677870)

[4.8.1 Auditoria 69](#_Toc530677871)

[4.8.2 Cadastrar Fornecedor 70](#_Toc530677872)

[4.8.3 Cadastro de Usuáio 71](#_Toc530677873)

[4.8.4 Consulta Credor 72](#_Toc530677874)

[4.8.5 Exportar para Planilha 73](#_Toc530677875)

[4.8.6 Inclusão de Contas à Pagar 74](#_Toc530677876)

[4.8.7 Login 75](#_Toc530677877)

[4.8.8 Pagamento de Conta 76](#_Toc530677878)

[4.8.9 Redefinir Senha 77](#_Toc530677879)

[4.8.10 Simulador 78](#_Toc530677880)

[4.8.11 Triagem de Contas à Pagar 79](#_Toc530677881)

[4.9 Interfaces com o usuário 80](#_Toc530677882)

[4.9.1 Página Inicial 80](#_Toc530677883)

[4.9.2 Autenticação de Usuário (Login) 81](#_Toc530677884)

[4.9.3 Cadastrar novo Usuário 82](#_Toc530677885)

[4.9.4 Controle de Contas à Pagar 83](#_Toc530677886)

[4.9.5 Inclusão de contas. 84](#_Toc530677887)

[4.9.6 Inclusão de Fornecedores 85](#_Toc530677888)

[4.9.7 Gerenciamento de Auditoria do Sistema 86](#_Toc530677889)

[4.10 Relatórios 87](#_Toc530677890)

[5. Implementação 89](#_Toc530677891)

[6. Projeto de Teste 89](#_Toc530677892)

[7. Instalação do software 89](#_Toc530677893)

[8. Conclusão 91](#_Toc530677894)

[9. Referências 92](#_Toc530677895)

[10. Anexo 93](#_Toc530677896)

[10.1 Script do Banco de Dados 93](#_Toc530677897)

# Introdução

Para o desenvolvimento deste projeto foram observadas as rotinas do setor financeiro, com foco no departamento de contas a pagar de uma microempresa na região de Sorocaba que atua no ramo de suporte e vendas de equipamentos para informática.

Situação comum em muitas empresas principalmente as que possuem um menor capital para investimento, o sistema atual da empresa deixou de receber suporte há alguns anos tornando-o obsoleto, inadequado para as operações que a empresa realiza nos dias atuais. O proprietário é um elemento ativo em diversas áreas da empresa e no departamento de contas a pagar não seria diferente pois ali é uma peça importante da sua relação com seus parceiros. Manter as contas em ordem, priorizar as parcerias e estabelecer uma comunicação clara, objetiva e precisa com os fornecedores é de extrema importância para a saúde financeira da empresa. O proprietário como uma figura muita ativa dentro da organização tem enfrentado situações contraproducentes em sua rotina nas contas a pagar. A primeira delas é a necessidade de obter um maior planejamento para o pagamento de suas contas com base em sua situação bancária; o sistema atual não dispõe de informações de multa e juros de mora para cada título fazendo com que o colaborador tenha que recorrer ao título impresso para obter tais informações quando necessárias.

Outra situação é a falta de um maior controle de acesso ao sistema. Para evitar ser interrompido constantemente, o proprietário libera o sistema sem qualquer restrição de uso, além daquele que representa o maior número de ocorrências que são os atrasos constantes em pagamentos devido à ausência de um método para trazer a atenção do usuário ao vencimento que está próximo.

Diante das situações apresentadas a proposta de desenvolvimento será com base em um sistema web para o controle de contas a pagar, onde o proprietário contará com uma ferramenta para simular pagamentos levando em consideração seu saldo bancário. Um novo método de acesso ao sistema também será formulado com o objetivo de tornar segura toda e qualquer operação executada sobre os títulos registrados através de restrições mais refinadas que atendam às necessidades do proprietário, incluindo auditoria. Além disso, traremos a atenção dos colaboradores através de um sistema de notificação os títulos que estiverem próximos de suas datas de vencimento.

O projeto será desenvolvido utilizando a plataforma .Net framework 4.5 com C# no ambiente de desenvolvimento Visual Studio 2017, base de dados o SQL Server e para o front-end a biblioteca Bootstrap, CSS e HTML5.

Ao concluir este projeto espera-se que a aplicação esteja acessível a partir de qualquer terminal disponível com um browser, que o planejamento financeiro da empresa com base em seu contas a pagar esteja sendo realizado com o auxílio da ferramenta de simulação, além de cada colaborador ter seu perfil adequado ao acessar o sistema dando assim ao proprietário maior confiança e tranquilidade ao gerir o departamento ,além de reduzir significativamente o número de ocorrências onde títulos deixam de ser pagos por meros lapsos de memória acarretando em pagamento de taxas a princípio desnecessárias.

# Planejamento Inicial do Software

## Situação Atual

### Descrição da Situação Atual

A empresa RM, que trabalha com vendas e manutenção em informática, tem seu setor de contas a pagar, que é o foco deste trabalho, localizado em uma sala juntamente com o setor de contas a receber e demais serviços administrativos. Nesta sala também ficam localizados os arquivos de comprovantes de toda e qualquer operação realizada por esses setores, todo e qualquer documento de natureza contábil (cópias de cheque, ordens de pagamento, notas fiscais, extratos bancários), Legal (contrato social, declaração cadastral, cartão do CNPJ, cartão de inscrição da prefeitura, alvará de funcionamento) e documentos de rotina que fazem parte da relação comercial como pedido, nota fiscal, fatura, duplicata e aviso de cobrança.

Existem dois funcionários que atuam no setor verificando contas a pagar e receber diariamente através de um sistema, sendo um deles o proprietário,

Periodicamente é efetuada uma verificação através do sistema para acompanhar os vencimentos e programar os pagamentos. As contas fixas como água, energia elétrica, interne, telefone, segurança etc, já ficam previamente lançadas apenas observando possíveis reajustes. As obrigações com fornecedores começam a ser programadas no momento do pedido, o sistema possui para cada item do estoque um nível chamado de crítico que quando atingido é exibido no relatório de compras, que posteriormente passa pela avaliação do gestor, pois nem todos os itens críticos precisam necessariamente serem repostos (no caso de produtos que estão no final do seu ciclo de vida ou por alguma condição no mercado que fez com que seu valor fosse elevado exponencialmente). Em situações desta natureza, o item contido no pedido é reavaliado. Após o pedido ser aprovado o mesmo fica aguardando no sistema para sua efetiva entrada. Quando o pedido é entregue seja ele parcial ou integral, o sistema verifica item a item e ao final o valor da nota precisa coincidir ou estar abaixo do valor previamente aprovado; é possível a nota ter um valor menor pois em alguns casos os fornecedores conseguem gerar algum desconto ou algum item pode estar indisponível.

## Problemas encontrados

### Ausência da informação de juros ao registrar as duplicatas.

No atual sistema não existe o campo de multa para vencimento e juros de mora diária. Tais informações poderiam contribuir para que o sistema realizasse uma triagem de contas, onde com os valores de multa (percentual de multa x valor total do boleto) somado aos juros de mora (número de dias em atraso x percentual de mora dividido pelo mês comercial) seria possível priorizar o pagamento das contas que gerassem o menor valor de juros até que se obtenha o valor total para o pagamento. Diante desta situação os funcionários precisam consultar as informações adicionais da duplicata no próprio documento, efetuando manualmente o cálculo necessário, o que acaba gerando perda de tempo, além da possibilidade de efetuar um cálculo ou esquecer outras contas. Não é possível determinar com exatidão a frequência com que ocorre, mas geralmente é baixa pois depende do fluxo de caixa, mas em se tratando do financeiro da empresa, toda e qualquer informação que auxilie uma melhor tomada de decisão é importante.

* Frequência: diária (+2x).
* Consequências: Inviabiliza o planejamento de contas através do sistema, fazendo com que a empresa em situações pontuais, devido à insuficiência de saldo, acabe pagando juros maiores.

### Baixa visibilidade das contas a vencer

O sistema não possui um alerta sobre contas vencidas e contas que estão para vencer, o ideal é que ao acessar o sistema um relatório fosse exibido das contas vencidas e as que estão próximas do seu vencimento. Desta maneira o usuário precisa checar diariamente se existem contas a vencer, o que pode fazer com que alguma conta tenha o seu pagamento esquecido causando, além do pagamento de juros, também o bloqueio do seu crédito junto ao fornecedor.

* Frequência: diária (+3x).
* Consequências: Pagamentos não são realizados devido à falta de aviso ou distração, prejuízos com pagamento de juros desnecessários, bloqueio de crédito junto aos fornecedores, atrasando compras e gerando contratempo com clientes, além de possíveis estornos.

### Faltam níveis de acesso personalizados ao sistema

No sistema atual todos usuários têm acesso total às rotinas do sistema, desde inclusão de contas até sua própria exclusão, podendo gerar assim inconsistências nos dados cadastrados no sistema.

Da forma como funciona atualmente o administrador precisa deixar um nível de acesso mais alto para o usuário para não precisar ser interrompido constantemente, assim dando direitos administrativos do sistema a usuários que não tem esse perfil.

* Frequência: mensal (10~15x).
* Consequências: A ausência desses níveis de acesso personalizados prejudica a auditoria de registros e dificulta a apuração de operações que não deveriam ser autorizadas para determinados usuários. Essa falta de gestão pode ocasionar a falta de fornecimento por parte de nossos fornecedores devido à falta de pagamento de uma conta erroneamente excluída do sistema, assim podendo trazer prejuízos financeiros a empresa.

## Objetivos de Negócio do Projeto

### Missão

Gerir com responsabilidade os compromissos assumidos pela empresa, cuidando para que os pagamentos sejam realizados de forma eficiente e eficaz.

### Simulações de contas a vencer

O simulador de contas a vencer contribuirá no controle e no planejamento do pagamento das mesmas. O valor de uma ou mais contas a vencer poderá ser recalculado de acordo com a data na qual se planeja efetuar o pagamento, podendo assim conter juros ou multas pertinentes ao seu atraso.

### Triagens de contas mediante ao nível de prioridade dos fornecedores

Os fornecedores serão classificados pelo seu nível de prioridade, o qual lhes será atribuído durante o seu cadastro no sistema. O nível prioritário de um fornecedor será classificado como: Alto, Médio ou Baixo; e essa prioridade terá sua importância comprovada quando existirem contas de diferentes fornecedores a serem pagas em uma mesma data, pois aquela a qual pertencer ao credor de nível mais elevado deve ter precedência sobre as demais.

### Integrações entre fornecedores cadastrados e novas contas

O sistema levará em consideração que toda e qualquer conta, ao ser lançada, deverá pertencer a um fornecedor previamente cadastrado, com o intuito de evitar o lançamento em duplicidade ou até mesmo para que a triagem de contas, feita através do nível crítico do fornecedor, não seja comprometida.

## Restrições e Riscos do Projeto Atual

|  |  |  |  |
| --- | --- | --- | --- |
| Risco | Probabilidade | Solução / Ação | Custo |
| Alteração no processo de pagamento de contas a pagar. | Média | Determinar em contrato todo fluxo, validar com cliente e determinar em contrato | R$ 75,00/hora |
| Falta de aderência na implantação do sistema. | Média | Incluir representantes das áreas envolvidas para suporte à implantação do sistema | - |
| Má definição do escopo do projeto. | Média | Elaborar escopo detalhado e apresentar ao cliente para aprovação | - |
| Sistema ser muito complexo para operação dos usuários. | Média | Treinar usuários a operar os módulos do sistema | - |
| Estouro do prazo. | Baixa | Validar cronograma com equipes de desenvolvimento e implantação | - |
| Demissão de funcionário ligado diretamente ao desenvolvimento ou implantação | Baixa | Investir em treinamentos para nivelar conhecimento dos funcionários | R$ 3.000,00 |
| Danos a equipamentos | Baixa | Providencia nobreak e backups | R$ 2.500,00 |
| Hardware inadequado para implantação do sistema. | Baixa | Levantar inventário do equipamento dos clientes | - |

Tabela 1 - Restrições e Riscos do Projeto Atual

## Limitações Operacionais

|  |  |  |
| --- | --- | --- |
| LIMITAÇÃO | DESCRIÇÃO | VERSÃO |
| Sistema Operacional | Windows, Mac OS ou Linux | Windows 7 ou superior |
| Browser | Google Chrome | Versão 70 ou superior |
| Banco de Dados | Microsoft SQL Server | - |
| Métodos de Entrada | Métodos de entrada via hardware que sejam suportados pelo sistema operacional. | - |
| Armazenamento | Armazenamento local de banco de dados e arquivos de preferência compartilhada. | - |

Tabela 2 - Limitações Operacionais

## Considerações Legais

Este software tem todos os direitos reservados. Sendo proibida a reprodução, a cópia, o aluguel e a utilização de cópias do mesmo sem a autorização dos desenvolvedores. A violação dos direitos é punível como crime, de acordo com a lei nº 9609/98.

## Considerações de Hardware / Software / Rede

Para ter o melhor desempenho deve ser utilizado um dispositivo com pelo no mínimo 3 GB de memória RAM, e conexão com a rede interna da empresa.

## Políticas Organizacionais

Para garantir a confidencialidade das informações cada usuário do sistema terá seu próprio ID e senha, limitando o acesso apenas ao que foi definido pelos administradores do sistema.

Para garantir a disponibilidade, deverão ser realizados backups semanais do sistema, que serão armazenados num serviço em nuvem.

## Riscos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PLANO DE RISCOS** | | | | |
|
| **Risco** | **Probab (P)** | **Impacto (I)** | **RiscoTotal (P x I)** | **Tratamento** |
|
| Alterações no escopo do projeto | 2 | 4 | 8 | Evitar mudanças que não estejam de acordo com o escopo do projeto. |
| Falta de mão de obra | 3 | 5 | 15 | Manter duas pessoas responsáveis por cada atividade, garantindo um backup caso haja problemas com a mão de obra. |
| Perda de arquivos importantes | 3 | 5 | 15 | Centralizar os arquivos do projeto em um repositório em nuvem. Possuir backup dos arquivos em mais de um serviço de armazenamento. |
| Atraso nas entregas | 4 | 5 | 20 | Manter a comunicação constante para saber do andamento das atividades |
| Não atender expectativa do cliente | 3 | 5 | 15 | Enviar relatórios do projeto para o cliente mostrando a evolução do projeto. |
| Alterações na legislação | 2 | 5 | 10 | Definir um responsável para acompanhar as alterações nas leis que podem interferir no projeto |

Tabela 3 - Riscos

# Análise de Requisitos

## Descrição da técnica utilizada para levantamento dos requisitos

Para o levantamento dos requisitos foram realizadas três entrevistas com os dois responsáveis pelos setores de Compras e de Contas a Pagar.

O Gerente de Contas a pagar foi entrevistado com o objetivo de conhecer e estabelecer as necessidades do cliente em relação ao sistema de contas a pagar, o qual será desenvolvido.

Foram realizadas duas entrevistas com o Gerente de Compras: a primeira foi referente ao setor de compras com o objetivo de compreender o impacto ocasionado ou sofrido pelo setor de compras mediante a implantação do sistema de contas a pagar. A segunda foi referente aos fornecedores, com o objetivo de compreender o impacto ocasionado ou sofrido pelos fornecedores mediante a implantação do sistema de contas a pagar.

## Situação Proposta

Será desenvolvido um sistema de contas a pagar web, que permitirá que os usuários incluam, editem e gerenciem as contas a pagar da empresa de forma simples e eficaz, além de oferecer relatórios gerenciais e funcionalidades que ajudarão na tomada de decisões relacionadas ao pagamento das contas, como por exemplo, a função de simulação de contas a pagar.

## Requisitos Funcionais

**Código:** RF1 - Inclusão de nova(s) conta(s) a pagar.

**Descrição:** O sistema deve permitir a inclusão de novos registros de pagamento através do preenchimento dos dados do documento que será lançado, também sendo possível inserir contas parceladas com valores fixos.

**Código:** RF2 - Edição de conta(s) a pagar já existente(s).

**Descrição:** O sistema deve permitir a alteração dos registros de pagamento para que se possam realizar retificações nos dados atuais.

**Código:** RF3 - Pagamentos de Conta(s).

**Descrição:** O sistemadeve permitir que os registros gravados ou alterados nos requisitos funcionais 1 e 2 sejam marcados como quitados para que assim deixem de constar como lançamentos em aberto para pagamento.

**Código:** RF4 - Consulta de Conta(s) a pagar.

**Descrição:** A consulta de contas se faz necessário para simples pesquisa, possível retificação, exclusão ou relatório.

**Código:** RF5 - Inclusão de novo(s) fornecedor(es).

**Descrição:** O sistema deve permitir a inclusão de novos registros de fornecedores através do preenchimento dos dados obtidos através do usuário. Esse registro estará associado e é obrigatório existir antes do lançamento de qualquer registro de pagamento (RF1).

**Código:** RF6 - Edição de fornecedor(es) já existente(s).

**Descrição:** O sistema deve permitir a alteração dos registros de fornecedores para que se possam realizar retificações nos dados atuais.

**Código:** RF7 - Triagem de contas por fornecedores.

**Descrição:** O sistema deverá identificaras contas a pagar com vencimentos próximos para ordenar nas telas de consulta.

**Código:** RF8 - Cadastro de usuários.

**Descrição:** O sistema deve permitir a um usuário com nível administrativo habilitar funcionários previamente cadastrados a utilizarem o sistema através de um login e senha.

**Código:** RF9 - Listagem de permissão de acesso de usuários.

**Descrição:** O sistema deve exibir aos usuários administradores uma listagem dos usuários atualmente habilitados assim como seus respectivos perfis de acesso.

**Código:** RF10 - Configuração do nível de acesso de usuários.

**Descrição:** O sistema deve permitir que o usuário administrador possa definir o nível de acesso dos usuários para que eventuais operações possam comprometer a segurança ou confiabilidade dos dados registrados.

**Código:** RF11 - Simulação de contas a vencer.

**Descrição:** O simulador de contas a pagar é uma ferramenta que auxilia o administrador a calcular o pagamento das multas e juros de acordo com a data pretendida para pagamento de uma ou múltiplas contas.

**Código:** RF12 - Consulta do saldo disponível.

**Descrição:** O sistema deve permitir ao usuário habilitado consultar valores referentes a saldo bancário de uma ou mais contas previamente cadastradas.

**Código:** RF13 - Inserção e/ou edição de valores referentes ao saldo.

**Descrição:** O sistema deve permitirque o usuário realize lançamentos ou edição lançamentos de crédito ou débito das contas bancárias.

**Código:** RF14 - Redefinição de senha.

**Descrição:** O sistema deve permitir que o usuário possa realizar a recuperação da sua senha em caso de esquecê-la através de e-mail.

**Código:** RF15 - Auditoria de atividades realizadas pelos usuários.

**Descrição:** O sistema deve permitir ao usuário habilitado a consulta a registros de inclusão, edição ou exclusão das operações relacionadas a contas a pagar, fornecedores e contas bancárias. A consulta poderá ser realizada através dos filtros de data.

**Código:** RF16 - Consulta de Fornecedor(es).

**Descrição:** A consulta de fornecedores se faz necessário para simples pesquisa, possível retificação, exclusão ou relatórios.

**Código:** RF17 - Relatórios de Contas à Pagar

**Descrição:** O sistema deve permitir consulta e saída através de impressão ou arquivo pdf de dados dos registros de contas a pagar.

**Código:** RF18 - Relatórios de Fornecedores

**Descrição:** O sistema deve permitir consulta e saída através de impressão ou arquivo pdf de dados dos registros de fornecedores.

**Código:** RF19 - Relatórios de Contas Bancárias

**Descrição:** O sistema deve permitir consulta e saída através de impressão ou arquivo pdf de dados dos registros de contas bancárias.

**Código:** RF20 - Login no Sistema

**Descrição:** O sistema deve permitir que o usuário, previamente habilitado através do seu login e senha, possa acessar o sistema e ter a sua disposição os recursos de acordo com seu perfil (administrador, padrão, etc.)

**Código:** RF21 – Recuperar Senha

**Descrição:** O sistema deve permitir que o usuário consiga recuperar sua senha caso esqueça a mesma.

**Código:** RF22 – Cadastro de Contas Bancárias

**Descrição:** O sistema deve permitir o cadastro de contas bancárias, para que sejam utilizadas como consulta de saldos disponíveis.

## Diagrama de Casos de Uso e Descrição dos Casos de Uso

Uma imagem contendo texto, mapa, céu

Descrição gerada automaticamente

Figura 1 - Casos de Uso Alto Nível

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF1 Inclusão de nova(s) conta(s). | |
| **Ator Principal** | Financeiro | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Os fornecedores ou prestadores de serviço devem estar previamente cadastrados, os documentos de cobrança devem estar previamente autorizados pelo gerente ou proprietário. | |
| **Pós-Condição** | As contas a pagar estão devidamente registradas. | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessar tela de contas a pagar. | |  |
|  | | 2. Os registros são exibidos. |
| 3. O ator localiza o número do documento (CPF ou CNPJ) que identifica o fornecedor/prestador e lança no sistema. | |  |
|  | | 4. Valida CPF ou CNPJ. |
| 5. O ator insere as informações contidas no documento como, beneficiário, vencimento, valor, Instruções (juros mora, multa e desconto) e número do documento, e quando necessário o valor das parcelas fixas. | |  |
|  | | 6º Registro efetuado. |

Tabela 4 - Caso de Uso Baixo Nível - RF1 Inclusão de nova(s) conta(s)

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF2 Edição de conta(s). | |
| **Ator Principal** | Financeiro | |
| **Ator Secundário** |  | |
| **Pré-Condição** | A conta deve estar devidamente cadastrada, o usuário previamente autorizado a realizar edições. | |
| **Pós-Condição** | Conta alterada e registrada. | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de contas a pagar. | |  |
|  | | 2. Os registros são exibidos. |
| 3. O ator seleciona o registro ou pesquisa através do número do documento. | |  |
|  | | 4. Localiza o registro e exibe em detalhes em nova janela. |
| 5 O ator efetua alterações nos campos desejados. | |  |
|  | | 6. Os campos são validados e gravados. |

Tabela 5 - Caso de Uso Baixo Nível - RF2 Edição de Contas

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF3 Pagamento de Conta (s). | |
| **Ator Principal** | Financeiro | |
| **Ator Secundário** | Gerente ou Proprietário | |
| **Pré-Condição** | As contas a serem pagas devem estar devidamente cadastradas no sistema. | |
| **Pós-Condição** | Efetuado pagamento da conta, efetuado o registro de baixa integral da conta no sistema. | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de consulta as contas. | |  |
|  | | 2. Os títulos para pagamento são apresentados para seleção. |
| 3. O ator seleciona o título a ser pago. | |  |
|  | | 4. O sistema abre a tela com o as informações da conta previamente cadastrada, e exibe valor a ser pago recalculado (se necessário). |
| 4. O ator seleciona a modalidade de pagamento e confirma o valor que está sendo pago. | |  |
|  | | 5. O sistema salva registro de pagamento da conta. |

Tabela 6 - Caso de Uso Baixo Nível - RF3 Pagamento de Conta(s)

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF4 Consulta de Conta(s). | |
| **Ator Principal** | Financeiro | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Conta previamente cadastrada | |
| **Pós-Condição** | Relação das contas exibidas | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de consulta as contas. | |  |
| 2. O ator define os filtros a serem aplicados para a consulta de contas. | |  |
|  | | 3. Os registros se localizados, são apresentados ao ator com suas respectivas informações. |
| 4. O ator realiza a consulta dos dados desejados. | |  |

Tabela 7 - Caso de Uso Baixo Nível – RF4 Consulta de Conta(s)

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF5 Inclusão de novo(s) fornecedor(es). | |
| **Ator Principal** | Financeiro | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Fornecedor não estar previamente cadastrado | |
| **Pós-Condição** | Fornecedor devidamente cadastrado no sistema. | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O Ator acessa a tela de cadastro de fornecedor. | |  |
| 2. Escolher se o fornecedor é pessoa física ou jurídica | |  |
|  | | 3.Exibir tela de acordo com a escolha do usuário |
| 4.Inserir o CPF ou CNPJ | |  |
|  | | 5. O sistema verifica a duplicidade de cadastro através do CNPJ ou CPF |
| 6. O ator segue com a entrada dos dados complementares referentes a Nomes, localização física, endereços eletrônicos e informações bancárias e submete ao sistema. | |  |
|  | | 7. Efetua validação e informa que o novo registro foi efetuado. |

Tabela 8 - Caso de Uso Baixo Nível – RF5 Inclusão de Novo(s) Fornecedor(es)

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF6 Edição de fornecedor. | |
| **Ator Principal** | Financeiro | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Registro existente. | |
| **Pós-Condição** | Alteração do Cadastro do Fornecedor | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de consulta de fornecedor. | |  |
| 2. O ator entra com o nome ou CNPJ do fornecedor a ser alterado, e seleciona o mesmo. | |  |
|  | | 3. Todas os dados do registro encontrado são localizados e exibidos com todos os detalhes. |
| 4. O ator realiza a alteração no campo desejado e submete o registro da alteração através do gravar. | |  |
|  | | 4. Alteração é validada e posteriormente informado seu registro com sucesso. |

Tabela 9 -Caso de Uso Baixo Nível – RF6 Edição de Fornecedor

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF7 Triagem de contas por fornecedores. | |
| **Ator Principal** | Financeiro | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Devem existir contas previamente cadastradas | |
| **Pós-Condição** | Relação das contas por fornecedor | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de consulta de contas. | |  |
| 2. O ator seleciona o fornecedor desejado. | |  |
|  | | 3. O sistema exibi as contas no grid ordenados de acordo com sua data de vencimento e o seu valor. |

Tabela 10 - Caso de Uso Baixo Nível – RF7 Triagem de Contas por Fornecedores

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF8 Cadastro de usuário | |
| **Ator Principal** | Gerente | |
| **Ator Secundário** | Funcionário | |
| **Pré-Condição** | O usuário não cadastrado. | |
| **Pós-Condição** |  | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa tela de cadastro de usuário | |  |
| 2. O ator preenche os dados pessoais, e-mail, senha inicial e tipo do perfil para usuário a ser cadastrado. | |  |
|  | | 3. Sistema realiza cadastro e envia e-mail para usuário que foi cadastrado solicitando confirmação e primeiro acesso. |

Tabela 11 - Caso de Uso Baixo Nível – RF8 Cadastro de Usuário

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF9 Listagem de Permissão de acesso. | |
| **Ator Principal** | Gerente | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Existir usuários cadastrados | |
| **Pós-Condição** | Exibição da lista de usuários | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de permissões de acesso. | |  |
|  | | 2. O sistema carrega a lista de usuários e suas permissões de acesso. |
|  | | 3. O sistema exibi o resultado em tela. |

Tabela 12 - Caso de Uso Baixo Nível – RF9 Listagem de Permissão de Acesso

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF10 Configuração do nível de acesso de usuários. | |
| **Ator Principal** | Gerente | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Existir usuários cadastrados | |
| **Pós-Condição** | Alterado nível de acesso do usuário. | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de permissões de acesso. | |  |
| 2. O ator busca usuário desejado. | |  |
|  | | 3. Carregar usuário digitado pelo usuário e seu nível de acesso |
| 3. O ator edita o nível de acesso do usuário e salvar | |  |
|  | | 4. Gravar as alterações realizadas. |

Tabela 13 - Caso de Uso Baixo Nível – RF10 Configuração do Nível de Acesso de Usuários

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF11 Simulação de contas a vencer. | |
| **Ator Principal** | Gerente, Funcionário | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Existir contas cadastradas | |
| **Pós-Condição** |  | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de Simulação de contas a vencer | |  |
| 2. O ator seleciona as contas desejadas | |  |
|  | | 3. O sistema busca conta no sistema e carregar na tela |
| 4. O ator escolher a data de pagamento planejada | |  |
|  | | 4. O sistema recalcula o valor da conta de acordo com a nova data, levando em conta os novos juros e multa. |
|  | | 5. O sistema exibi resultado em tela. |

Tabela 14 - Caso de Uso Baixo Nível - RF11 Simulação de Contas à Vencer

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF12 Consulta do saldo disponível. | |
| **Ator Principal** | Gerente, Funcionário | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Existir conta bancária cadastrada | |
| **Pós-Condição** |  | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1º O ator acessa a tela de Contas bancárias | |  |
|  | | 2. O sistema carrega a tela de contas bancárias |
| 3º O ator busca conta bancária que deseja saber o saldo | |  |
|  | | 4. O sistema carrega a conta bancária e seu saldo |

Tabela 15 - Caso de Uso Baixo Nível – RF12 Consulta do Saldo Disponível

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF13 Inserção e/ou edição de valores referentes ao saldo. | |
| **Ator Principal** | Funcionário | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Existir Conta Bancária cadastrada | |
| **Pós-Condição** |  | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de contas bancárias | |  |
|  | | 2. Carregar tela |
| 3. O ator pesquisa a conta a ser editada | |  |
|  | | 4. Carregar conta pesquisada para a tela |
| 5. Editar/Inserir valor do saldo e salvar | |  |
|  | | 6. Salvar alterações no banco de dados |

Tabela 16 - Caso de Uso Baixo Nível - RF13 Inserção e/ou Edição de Valores Referentes ao Saldo

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF14 Redefinição de senha. | |
| **Ator Principal** | Funcionário | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Usuário estar previamente cadastrado no sistema | |
| **Pós-Condição** | Acesso ao sistema com a nova senha. | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. Ator acessa tela de redefinição de senha | |  |
|  | | 2. Carregar tela de redefinição de senha |
| 3. Ator Insere senha atual | |  |
| 4. Ator insere nova senha | |  |
|  | | 5. Verificar se senha atual está correta |
|  | | 6. Gravar nova senha |
|  | | 7. Ir para a tela de login |
| 8. Ator realizar login com a nova senha | |  |

Tabela 17 - Caso de Uso Baixo Nível - RF14 Redefinição de Senha

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF15 Auditoria de atividades do sistema. | |
| **Ator Principal** | Gerente | |
| **Ator Secundário** |  | |
| **Pré-Condição** | O usuário previamente autorizado a realizar auditorias. | |
| **Pós-Condição** |  | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a opção auditoria | |  |
|  | | 2. Janela se abre com os filtros a serem aplicados |
| 3. Ator seleciona os filtros de usuário, datas de início e fim, e o módulo do qual deseja ver a auditoria | |  |
|  | | 4. Os registros de auditoria são exibidos |
| 5. O ator pode realizar uma busca por palavra chave | |  |
|  | | 6. O resultado e destacado no registro localizado. |

Tabela 18 - Caso de Uso Baixo Nível – RF15 Auditoria de Atividades do Sistema

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF16 Consulta de Fornecedor(es). | |
| **Ator Principal** | Funcionário | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Existir fornecedores cadastrados | |
| **Pós-Condição** |  | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de consulta de fornecedores | |  |
| 2. O ator busca o fornecedor utilizando nome fantasia ou CNPJ | |  |
|  | | 3. Buscar fornecedor de acordo com o filtro do usuário |
|  | | 4. Exibir resultado em tela |

Tabela 19 - Caso de Uso Baixo Nível - RF16 Consulta de Fornecedor(es)

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF17 Relatório de Contas à Pagar | |
| **Ator Principal** | Funcionário | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Existir contas cadastradas no sistema | |
| **Pós-Condição** | Relação de contas cadastrados | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1º O ator acessar a tela de Relatório de Contas à Pagar | |  |
| 2. O ator define os filtros escolhidos e marcar a opção de exportar | |  |
|  | | 3º relatório de acordo com os filtros escolhidos pelo usuário |

Tabela 20 - Caso de Uso Baixo Nível - RF17 Relatório de Contas à Pagar

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF18 Relatório de Fornecedor | |
| **Ator Principal** | Funcionário | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Fornecedor previamente cadastrado | |
| **Pós-Condição** | Relação de fornecedores cadastrados | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de Relatório de Fornecedor | |  |
| 2. O ator insere os filtros escolhidos e marcar a opção de exportar | |  |
|  | | 3. Exportar relatório de acordo com os filtros escolhidos pelo usuário |

Tabela 21 - Caso de Uso Baixo Nível - RF18 Relatório de Fornecedor

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF19 Relatório de Contas Bancárias | |
| **Ator Principal** | Gerente | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Conta bancária estar previamente cadastrada | |
| **Pós-Condição** | Relação de Contas Bancárias cadastradas | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de Relatório de Contas Bancárias | |  |
| 2. O ator insere os filtros escolhidos e marcar a opção de exportar | |  |
|  | | 3. Exportar relatório de acordo com os filtros escolhidos pelo usuário |

Tabela 22 - Caso de Uso Baixo Nível - RF19 Relatório de Contas Bancárias

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF20 Login no Sistema | |
| **Ator Principal** | Funcionário | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Possuir usuário do sistema | |
| **Pós-Condição** |  | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de login do sistema. | |  |
| 2. O ator insere login e senha | |  |
|  | | 3. Verificar se login existe e se a senha está correta |
|  | | 4. Exibir tela inicial do sistema |

Tabela 23 - Caso de Uso Baixo Nível – RF20 Login no Sistema

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF21 Recuperar Senha | |
| **Ator Principal** | Funcionário | |
| **Ator Secundário** |  | |
| **Pré-Condição** | Possuir usuário do sistema | |
| **Pós-Condição** | Senha do usuário alterada | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de Login do sistema | |  |
| 2. O ator clica em ‘Recuperar Senha” | |  |
|  | | 3. Enviar email para usuário com link para cadastrar uma senha nova |
| 4. Acessar link e digitar nova senha e confirmar | |  |
|  | | 5. Verificar se senha é igual confirmação de senha e se atende requisito de 8 caracteres. |
|  | | 6. Salvar nova senha. |

Tabela 24 - Caso de Uso Baixo Nível – RF21 Recuperar Senha

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF22 Cadastro de Contas Bancárias | |
| **Ator Principal** | Gerente, Financeiro | |
| **Ator Secundário** |  | |
| **Pré-Condição** |  | |
| **Pós-Condição** | Conta bancária devidamente cadastrada no sistema | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. O ator acessa a tela de Cadastro de Contas Bancárias | |  |
| 2. O ator insere as informações referentes as Contas Bancárias | |  |
|  | | 4. O sistema salva conta bancária. |

Tabela 25 - Caso de Uso Baixo Nível – RF22 Cadastro de Conta Bancária

## Requisitos Não Funcionais

**Código:** RNF1

**Descrição:** Sistema implantado no ambiente Windows 10.

**Segurança:** Mantém o padrão de segurança existente nos Sistemas Operacionais da Microsoft.

**Desempenho:** Permite a realização de multitarefas de maneira ágil.

**Software:** Não há.

**Hardware:** Máquina com processador de 1 gigahertz ou SoC, 1 GB de RAM para 32 bits ou 2 GB para 64 bits, placa gráfica DirectX ou posterior com driver WDDM 1.0.

**Rede:** Possibilita acesso as unidades de rede existentes às máquinas que serão utilizadas.

**Tipo:** Organizacional.

**Código:** RNF2.

**Descrição:** O SGBD a ser utilizada é o SQL Server.

**Segurança:** Plataforma segura a qual permite restrições de acesso.

**Desempenho:** Permite a armazenagem de grandes quantidades de dados, bem como o acesso a eles, sem prejudicar o funcionamento do sistema.

**Software:** Java JDK.

**Hardware:** Máquina com no mínimo 1 GB de RAM e 250 MB de espaço livre no disco.

**Rede:** O acesso à rede se faz necessário para manipulação e gravação de dados no servidor.

**Tipo:** Produto.

**Código:** RNF3.

**Descrição:** Sistema criado em C# para backend, e o conjunto de ferramentas Razor, HTML, CSS, e JavaScript para frontend

**Segurança:** Plataforma que permite o desenvolvimento de aplicações cujas quais podem ser implementadas com propriedades de segurança.

**Desempenho:** Desempenho regular, de acordo com a aplicação desenvolvida.

**Software:** Utilizado junto a um sistema de banco de dados.

**Hardware:** Máquina com processador de 1,6 GHz, 1GB de RAM, 5 GB de espaço livre no disco, disco de 5400 RPM, placa de vídeo compatível com Directx 9 com execução na resolução de tela 1024 x 768 ou superior.

**Rede:** Essa plataforma permite desenvolver softwares que funcionem mesmo com a ausência de rede.

**Tipo:** Produto.

**Código:** RNF4.

**Descrição:** Funções, como o envio de mensagem por e-mail, requerem o uso de internet.

**Segurança:** Confiabilidade garantia pelo acesso exclusivo de cada usuário utilizando o próprio login e senha.

**Desempenho:** Tende a ser de acordo com as condições da rede, de cada máquina individualmente e com o desempenho do próprio software.

**Software:** Utilização de softwares de correio eletrônico para recebimento de mensagens.

**Hardware:** Uma máquina que comporte o sistema.

**Rede:** Requer conectividade com a internet.

**Tipo:** Produto.

1. Projeto Detalhado do Software

## Arquitetura da aplicação Atual

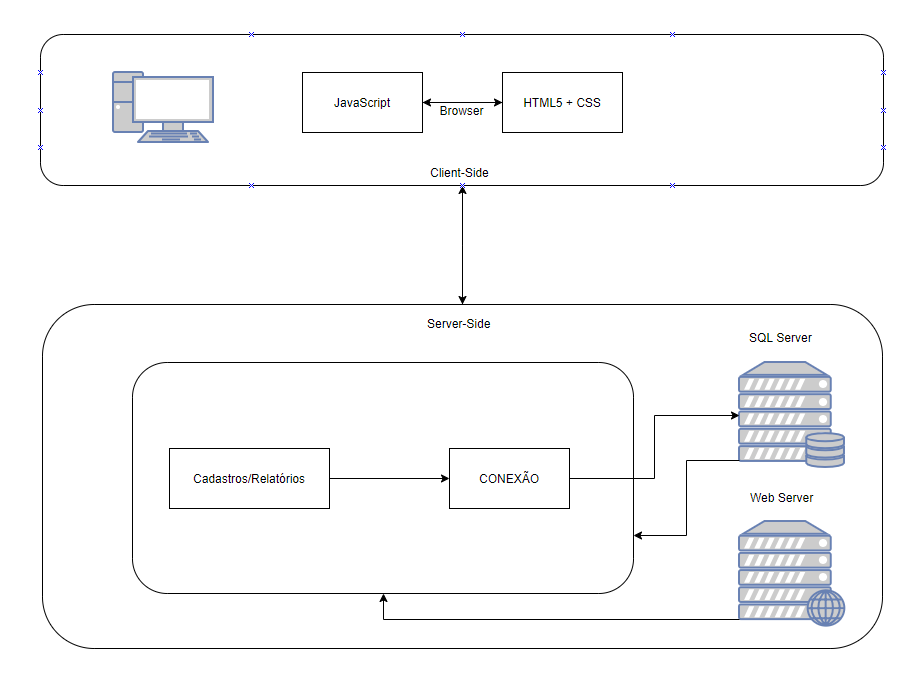


Figura 2 - Arquitetura da aplicação Atual

## Tecnologias utilizadas e APIs

### Bootstrap (front-end)

|  |  |
| --- | --- |
|  | Segundo Schmitz (2014), o Bootstrap é uma ferramenta open source para desenvolvimento front-end HTML, CSS e JS, onde é possível criar projetos responsivos para dispositivos móveis. Este framework foi originalmente desenvolvido para o site Twitter chamado de Twitter Blueprint e posteriormente renomeado como Twitter Blueprint to Bootstrap. |
|  |  |

### Entity Framework 6

|  |  |
| --- | --- |
| Imagem relacionada | De acordo com SANTOS (2012), o Entity Framework é um mapeador relacional de objetos que ajuda a reduzir incompatibilidades entre bancos de dados relacionais e aplicações orientadas a objetos. Nele é definido o modelo conceitual e a própria ferramenta cria as tabelas no banco de dados. |

### HTML

|  |  |
| --- | --- |
| Logo HTML5 | HTML é a sigla que vem do inglês para HyperText Markup Language que traduzindo para o português significa Linguagem de Marcação de texto (SILVA, 2011). O HTML é a linguagem base da internet, esta linguagem é baseada em marcações (tags) tais marcações facilitam a localização de sites de busca a palavras chaves do conteúdo dos sites. |

### CSS

|  |  |
| --- | --- |
|  | CSS é a sigla para o termo em inglês Cascading Style Sheets, que traduzido para o português significa Folha de Estilo em Cascatas (SILVA, 2011), esta é uma linguagem de folha de **estilos**, que tem o papel de tornar uma página apresentável na web, relacionada diretamente com o design e aparência. Ou seja, o CSS é uma camada que se usa para controlar o estilo da sua página da web. |

### SQL Server

|  |  |
| --- | --- |
|  | Segundo Jorgensen (2012) é um sistema de gerenciamento e análise de banco de dados para sistemas operacionais Windows e Linux (SQL Server 2017) lançado em 1988 pela Microsoft. Utilizado como solução para comercio eletrônico, linhas de negócio e dataware house. Basicamente sua principal função é a de armazenar e recuperar dados solicitados por outras aplicações. O SQL Server foi adotado como solução por várias das grandes empresas no Brasil como Vivo, Itaú e Banco do Brasil que após adotar a solução aumentou em 30% a velocidade de suas consultas. Na nova versão 2017 podemos dizer que o SGBD é apenas um de seus componentes pois ele agora é uma plataforma completa. A linguagem utilizada por profissionais de TI no gerenciamento do banco de dados é o T-SQL (Transact SQL), uma implementação da linguagem SQL desenvolvida pela Microsoft para o SQL server. |

### JQuery 3

|  |  |
| --- | --- |
|  | Lançado em 2006, por John Resig, (SILVA, 2011) é uma biblioteca JavaScript disponibilizada como software livre e aberto. Utilizado em diversos projetos pelo seu lema "escreva menos, faça mais"; com ele é possível criar vários efeitos com linhas de código o que já não seria possível se utilizássemos javascript puro. Alguns dos recursos são manipulação de elementos HTML, manipulação de CSS, animações, navegação pelo DOM, Ajax e eventos. |

## Componentes do SW

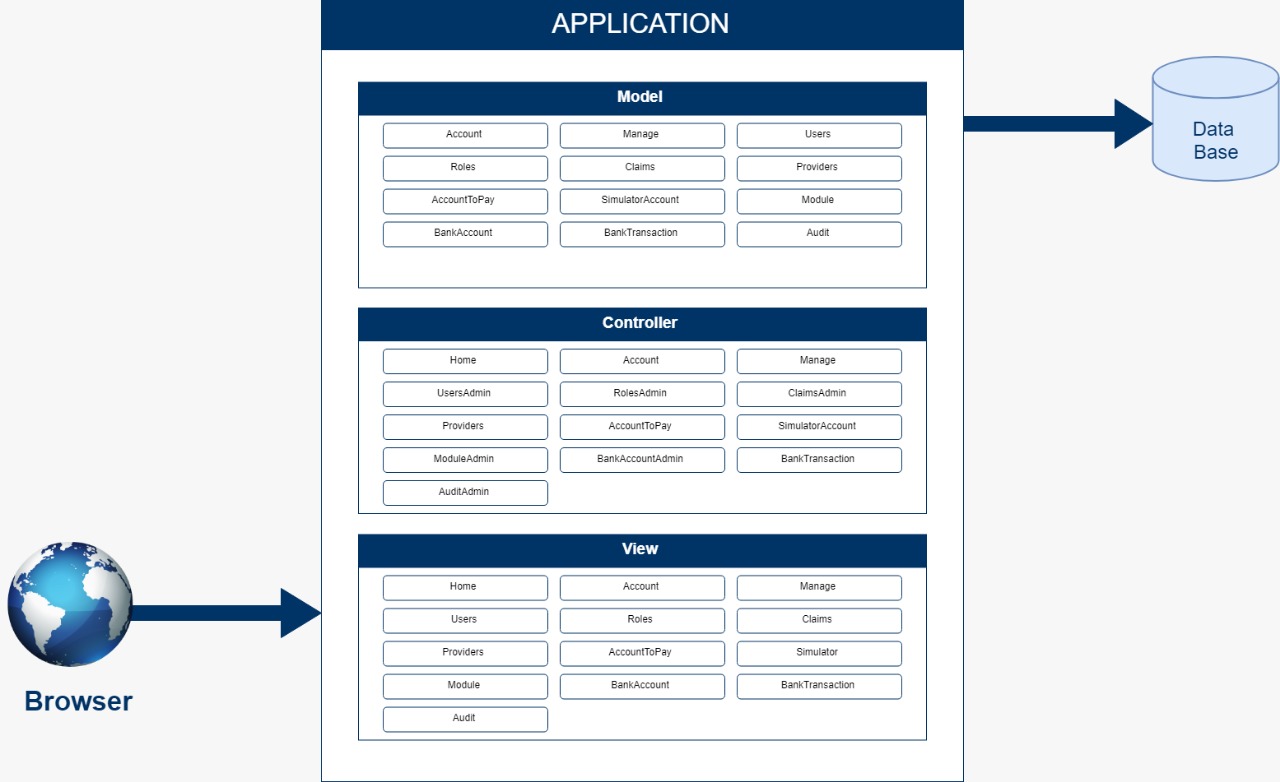


Figura 3 - Componentes do SW

## Diagrama de Classes

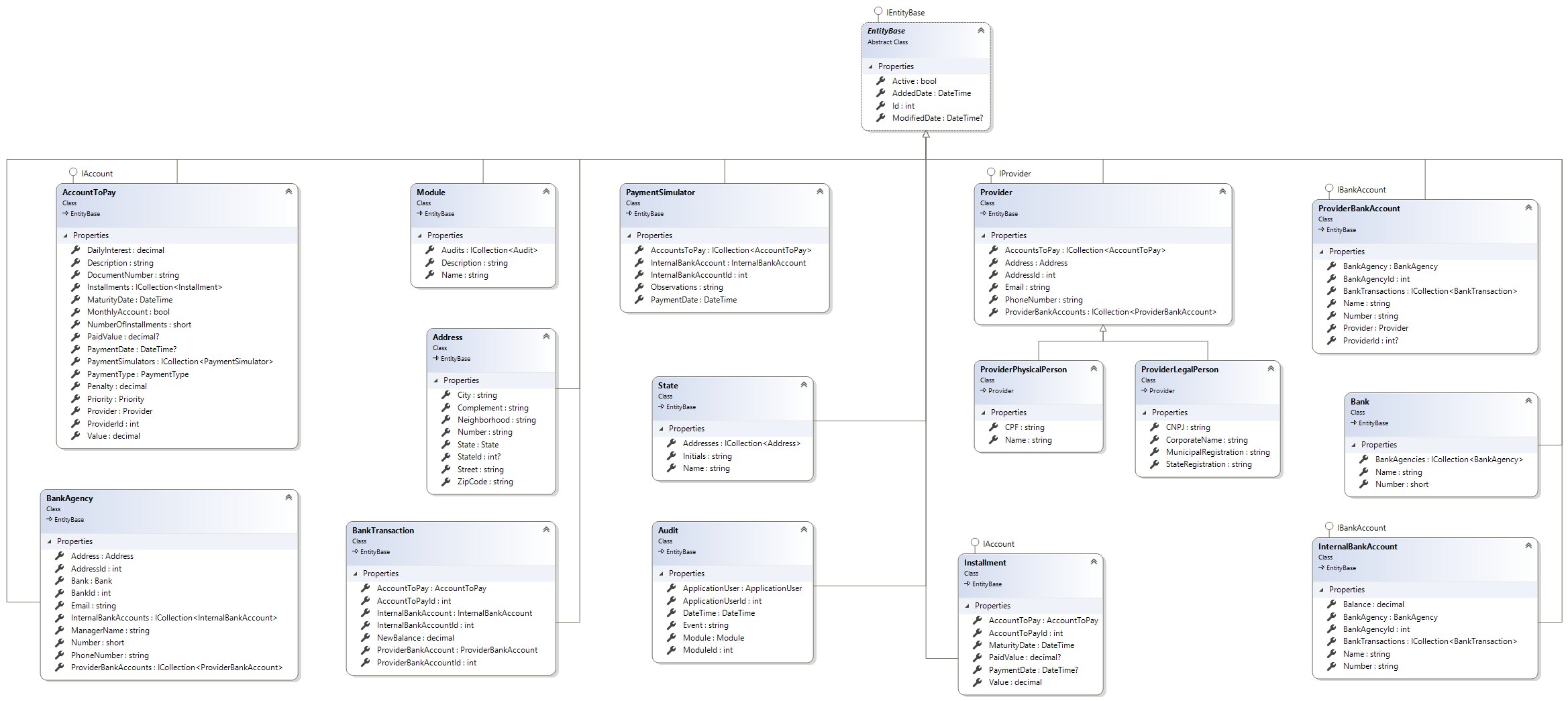
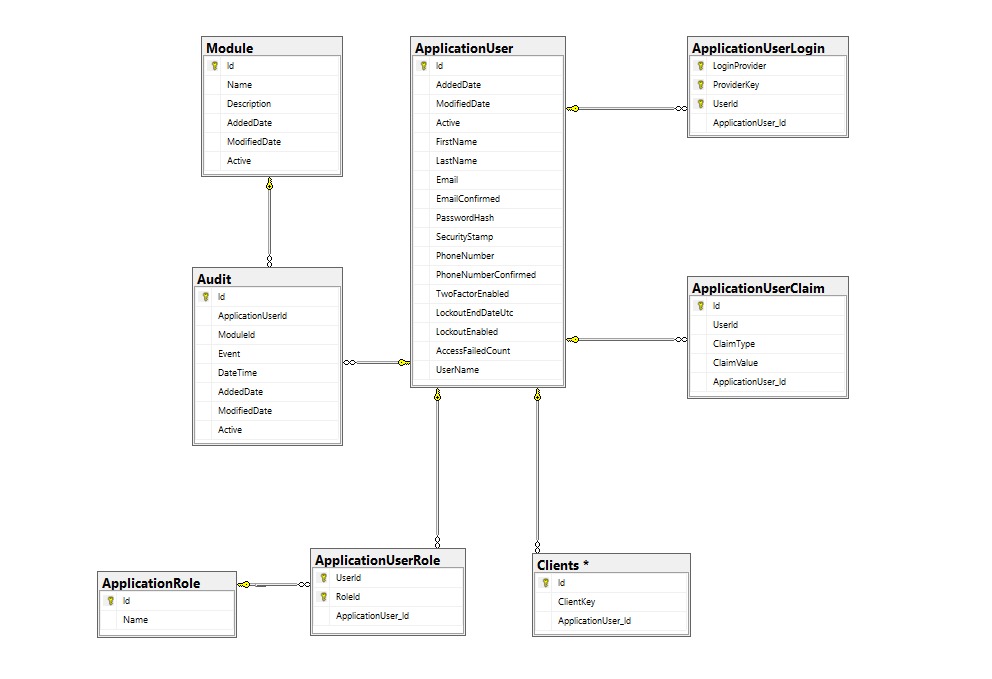


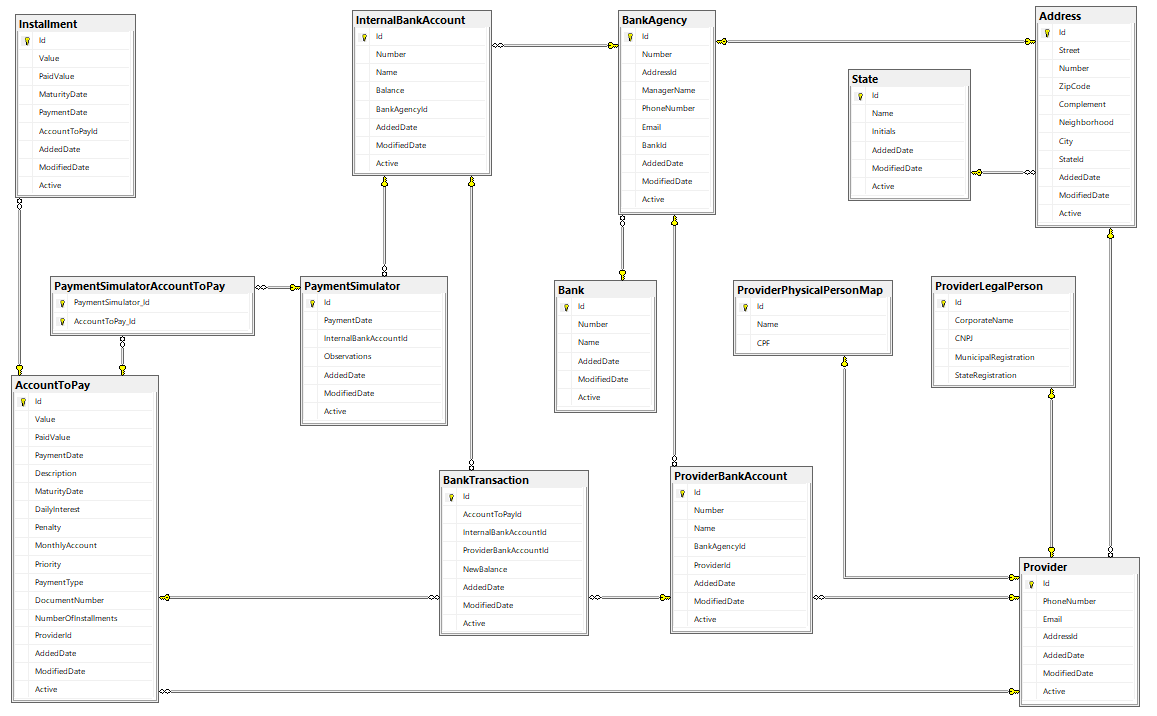
Figura 4 - Diagrama de Classes

## Banco de Dados

O SGBD utilizado pelo sistema será o Microsoft SQL Server2012, que é mantido pela Microsoft e é um dos principais SGBDs relacionais do mercado. Esse banco é capaz de atender desde cenários mais simples até negócios mais complexos que lidam com grande volume de dados.

### Modelo Conceitual





### Modelo Lógico

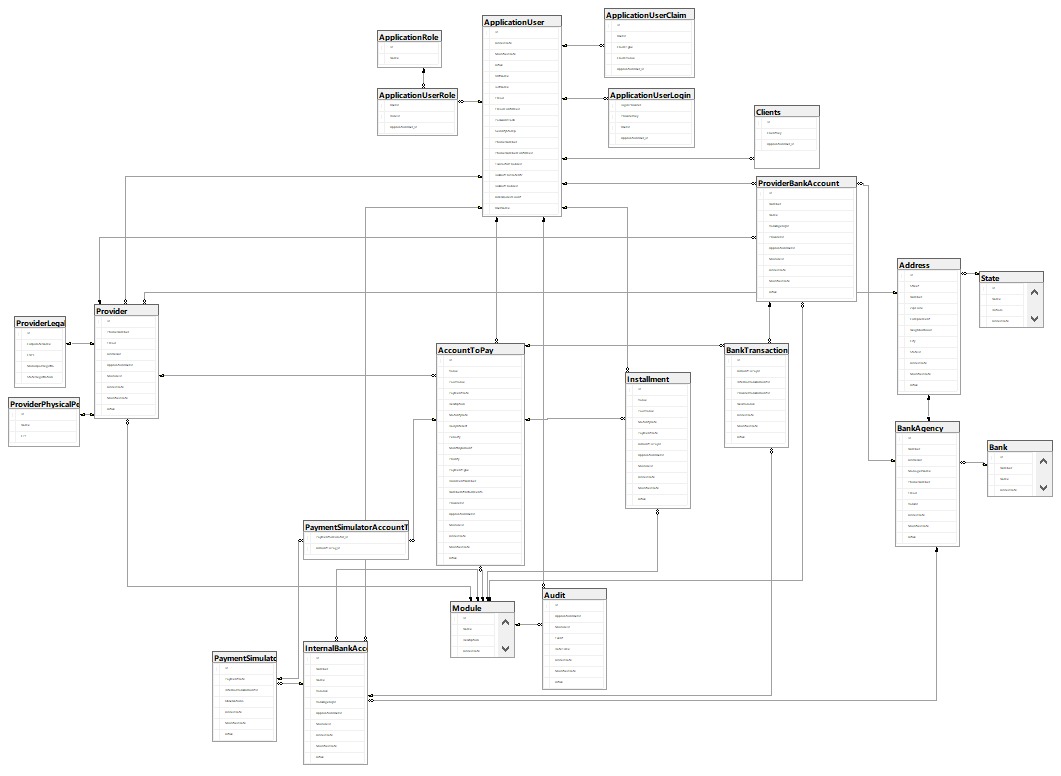


Figura 5 - Modelo Lógico Completo

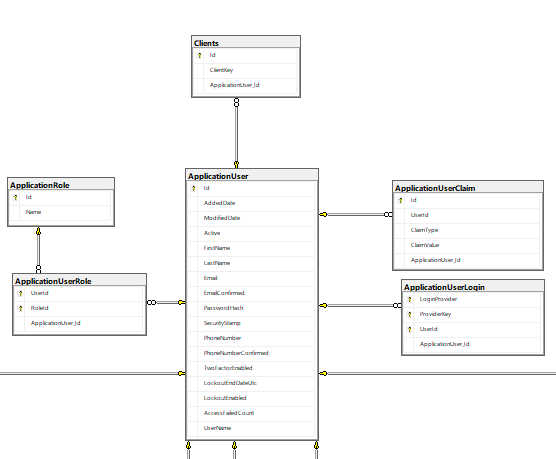


Figura 6 - Modelo Lógico (parte 1)

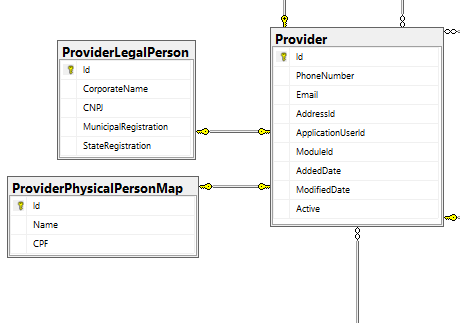


Figura 7 - Modelo Lógico (parte 2)

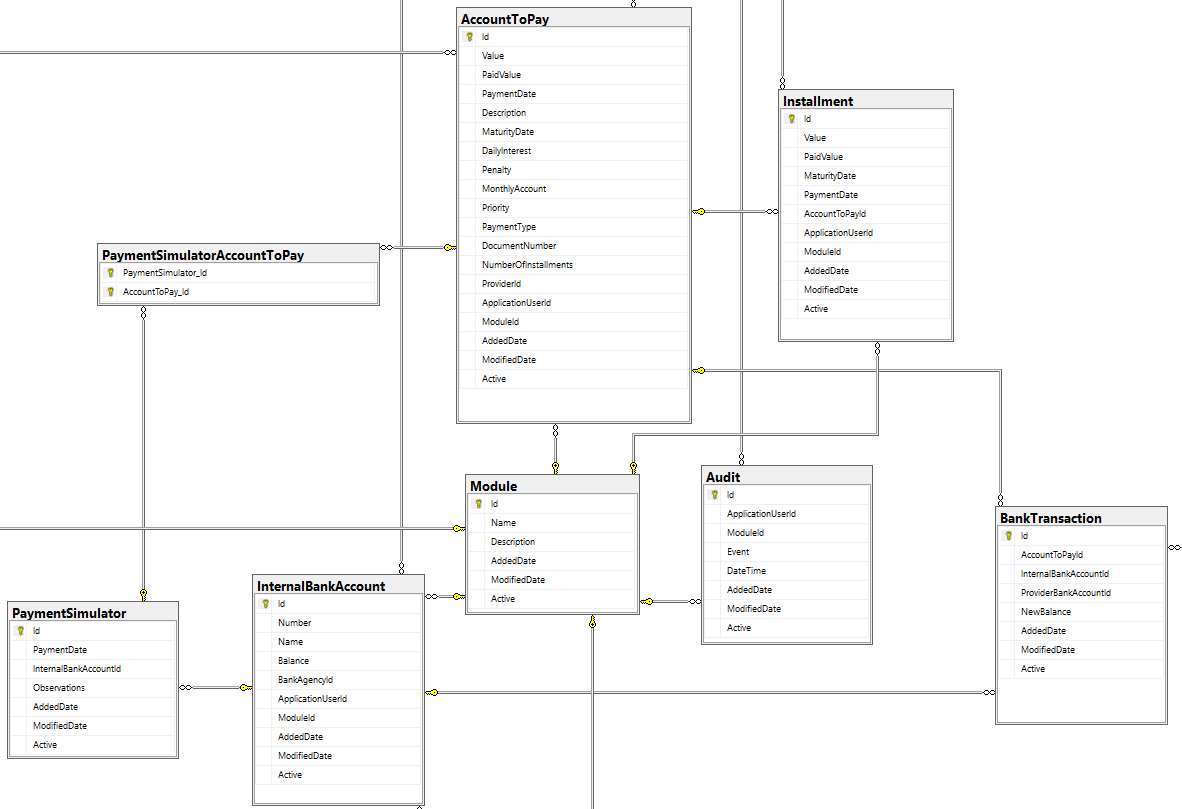


Figura 8 - Modelo Lógico (parte 3)

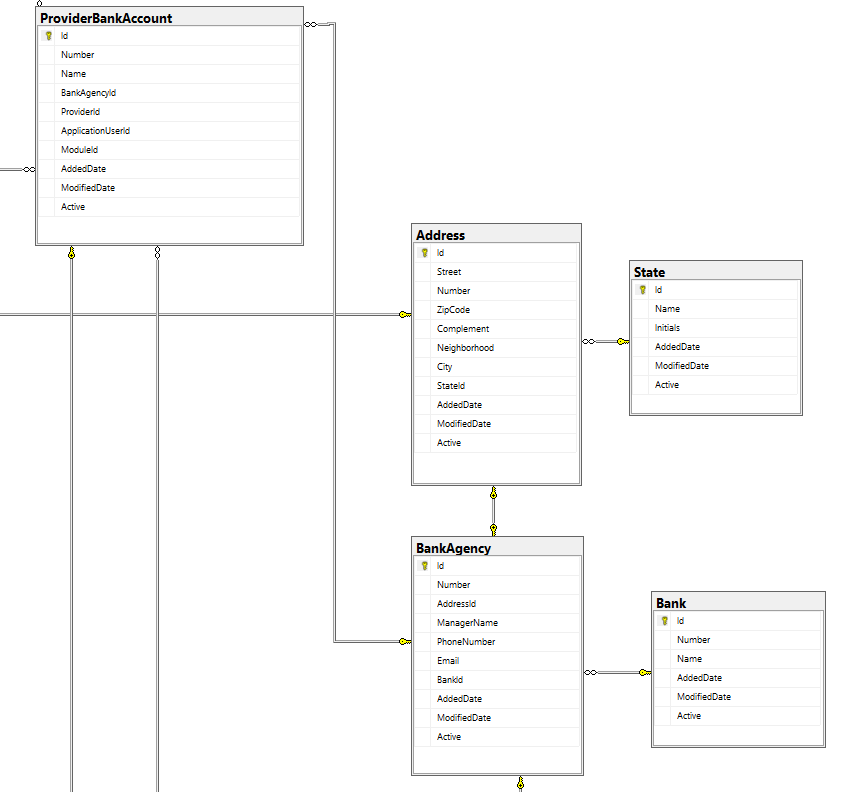


Figura 9 - Modelo Lógico (parte 4)

## Diagrama de Sequência

### Auditoria

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 10 - DS Auditoria

### Cadastrar Usuário

**Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança**

Figura 11 - DS Cadastrar Usuário

### Incluir Fornecedor

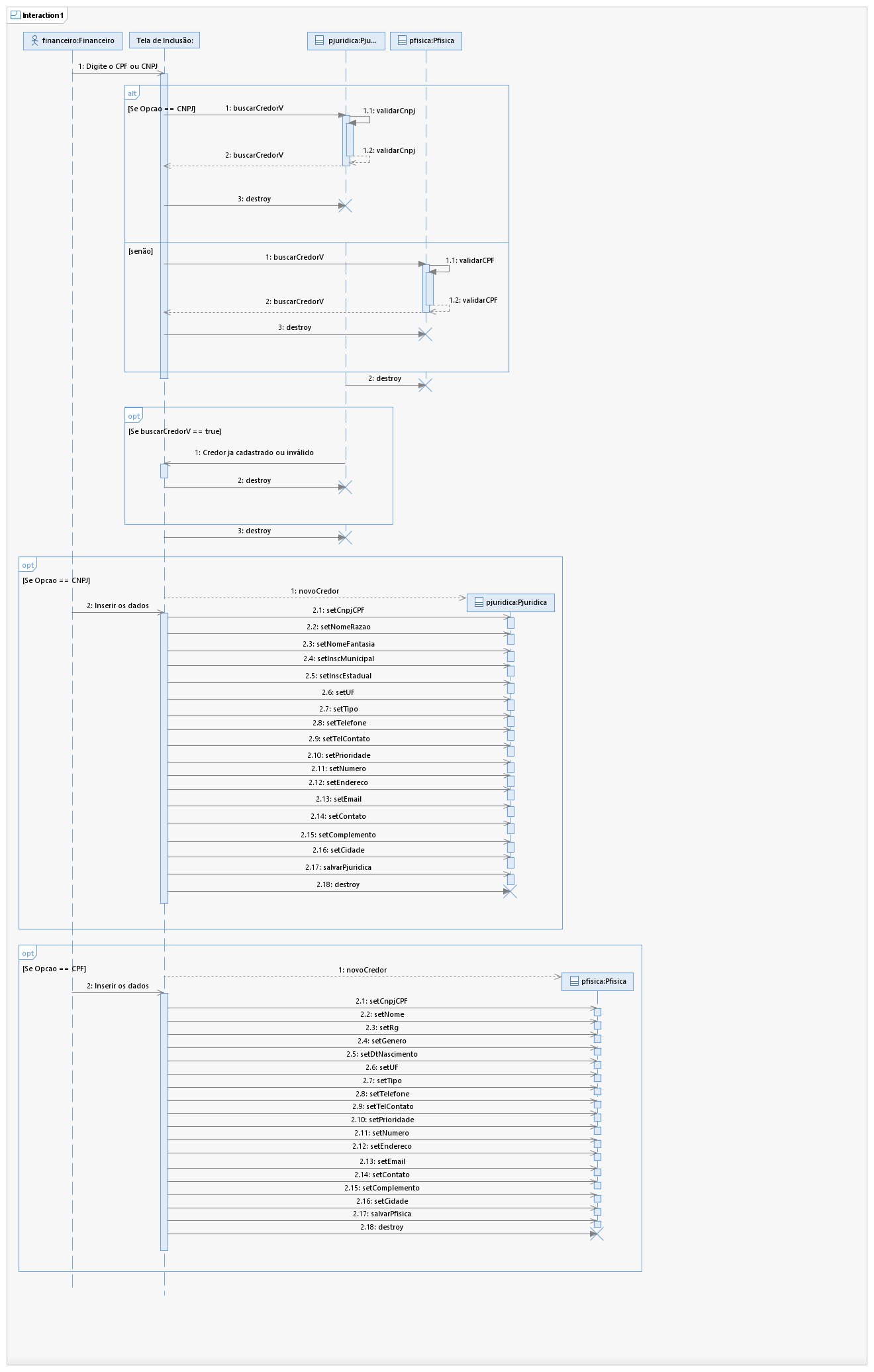
****

Figura 12 - DS Incluir Fornecedor

### Consultar Credor

Uma imagem contendo texto

Descrição gerada com muito alta confiança

Figura 13 - DS Consulta Credor (parte 1)

Uma imagem contendo captura de tela

Descrição gerada com alta confiança

Figura 14 - DS Consulta Credor (parte 2)



Figura 15 - DS Consulta Credor (parte 3)

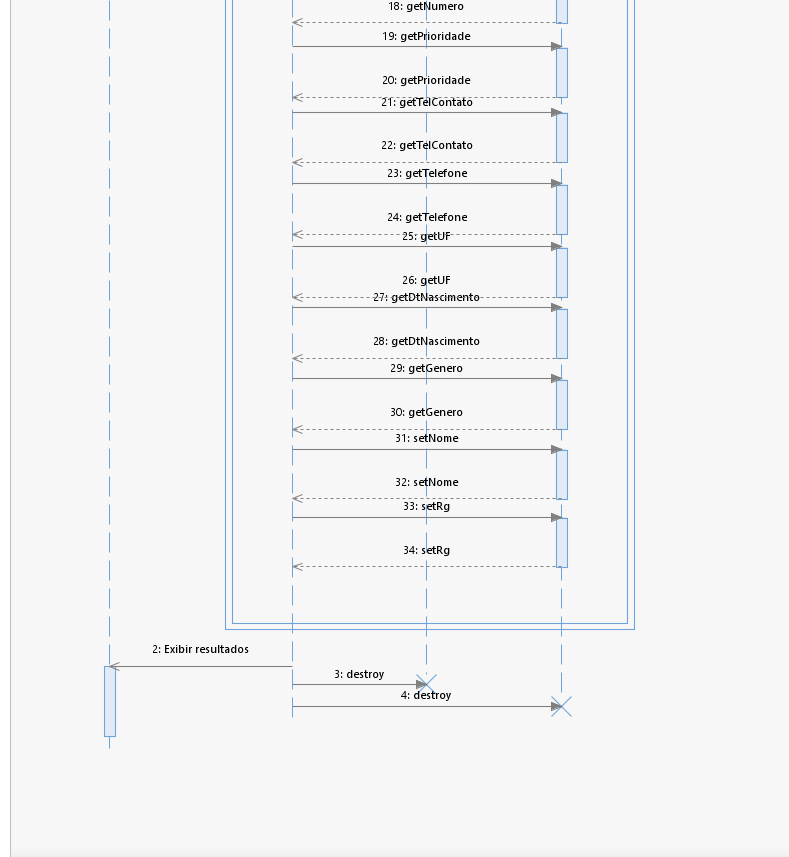


Figura 16 - DS Consulta Credor (parte 4)

### Incluir Contas à Pagar

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 17 - DS Incluir Contas

### Permissão de Acesso

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 18 - DS Permissão de Acesso

### Login

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 19 - DS Login

### Pagamento de Contas

Uma imagem contendo captura de tela, texto

Descrição gerada com alta confiança

Figura 20 - DS Pagamento de Contas

### Redefinir Senha

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 21 - DS Redefinir Senha

### Relatório de Contas à Pagar

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 22 - DS Relatório de Contas à Pagar

### Simulador de Contas

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 23 - DS Simulador de Contas

### Triagem de Contas

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 24 - DS Triagem de Contas

## Diagrama de Pacotes

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 25 - Diagrama de Pacotes

## Diagrama Estado

### Auditoria

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 26 - DE Auditoria

### Cadastrar Fornecedor

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 27 - DE Cadastrar Fornecedor

### Cadastro de Usuáio

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 28 - DE Cadastro de Usuário

### Consulta Credor

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 29 - DE Consulta Credor

### Exportar para Planilha

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 30 - DE Exportar para Planilha

### Inclusão de Contas à Pagar

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 31 - DE Inclusão de Contas à Pagar

### Login

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 32 - DE Login

### Pagamento de Conta

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 33 - DE Pagamento de Conta

### Redefinir Senha

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 34 - DE Redefinir Senha

### Simulador

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 35 - DE Simulador

### Triagem de Contas à Pagar

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 36 - DE Triagem de Contas à Pagar

## Interfaces com o usuário

### Página Inicial



Figura 37 - Tela Página Inicial

### Autenticação de Usuário (Login)

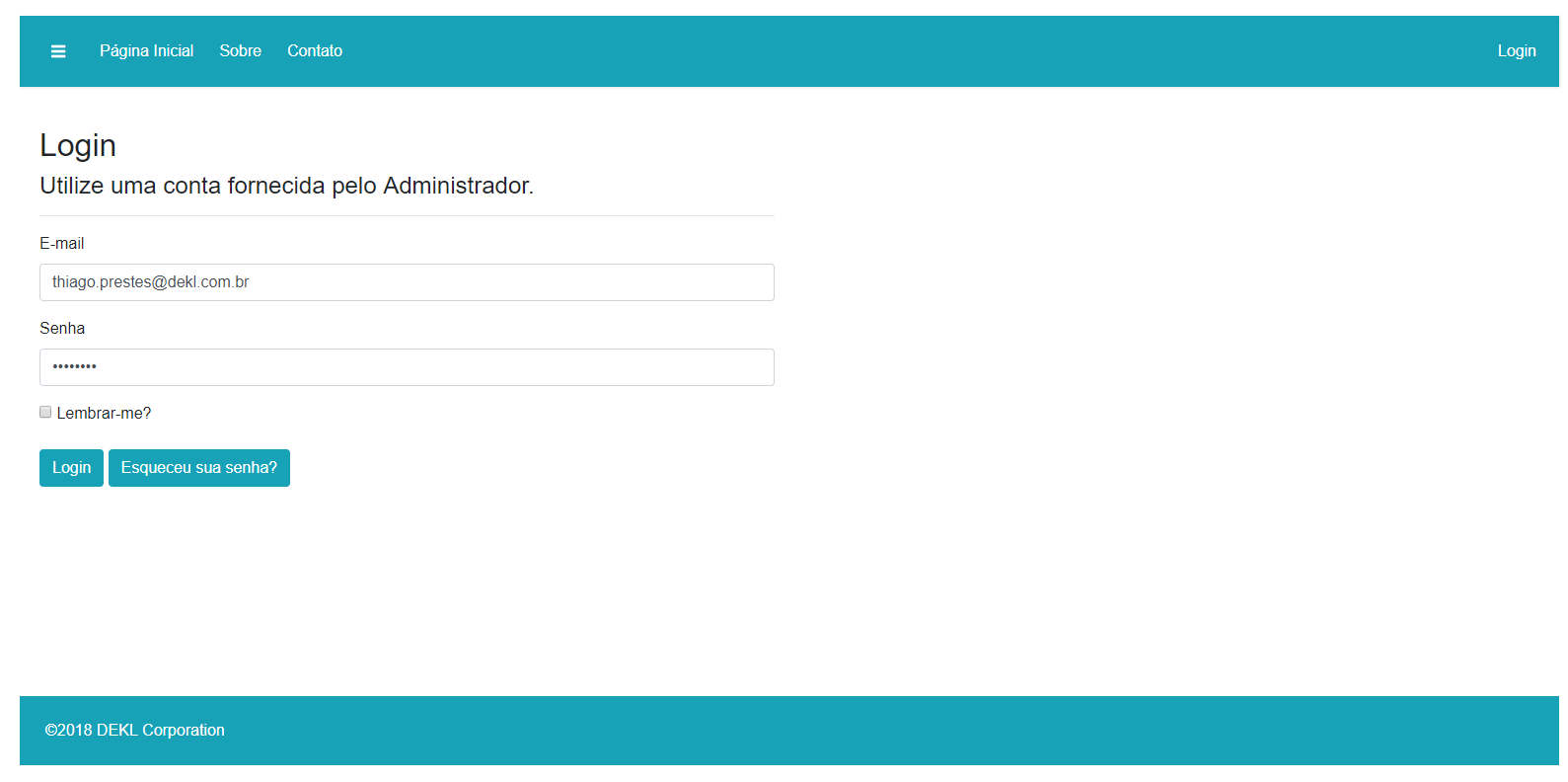


Figura 38 - Tela Login

### Cadastrar novo Usuário

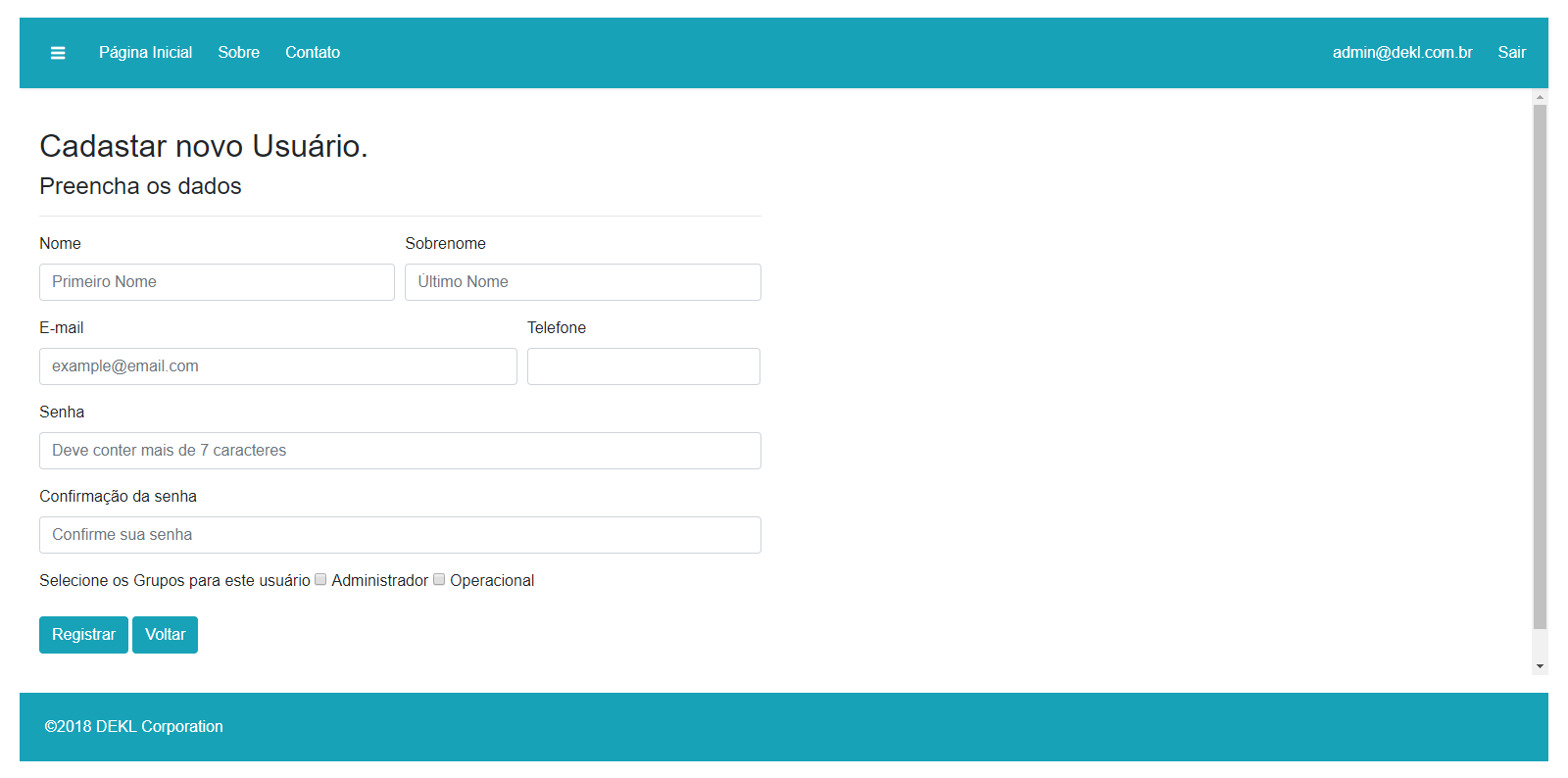


Figura 39 - Tela Cadastrar Novo Usuário

### Controle de Contas à Pagar

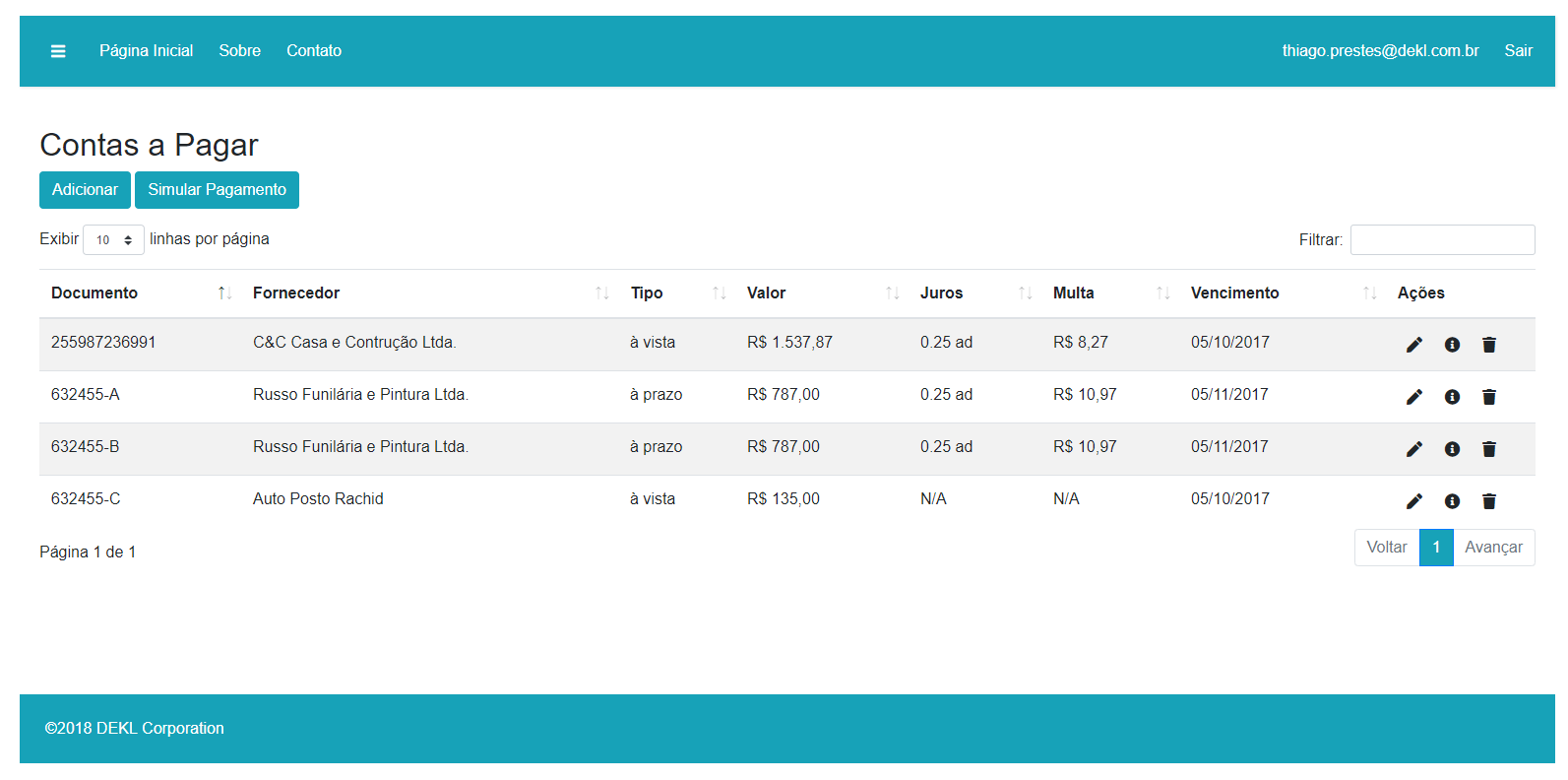


Figura 40 - Tela Controle de Contas à Pagar

### Inclusão de contas.

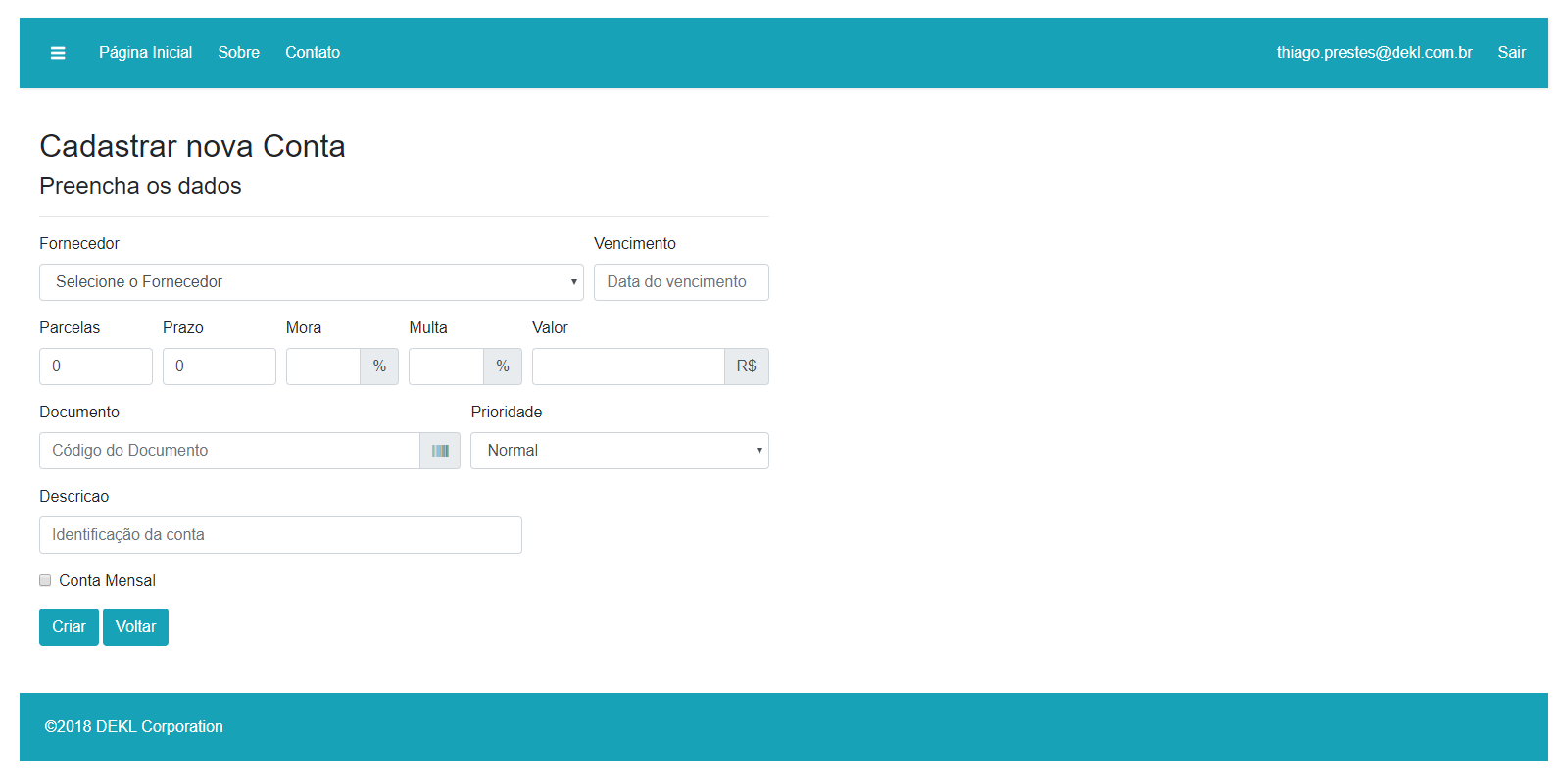


Figura 41 - Tela Inclusão de Contas

### Inclusão de Fornecedores



Figura 42 - Tela Inclusão de Fornecedores

### Gerenciamento de Auditoria do Sistema

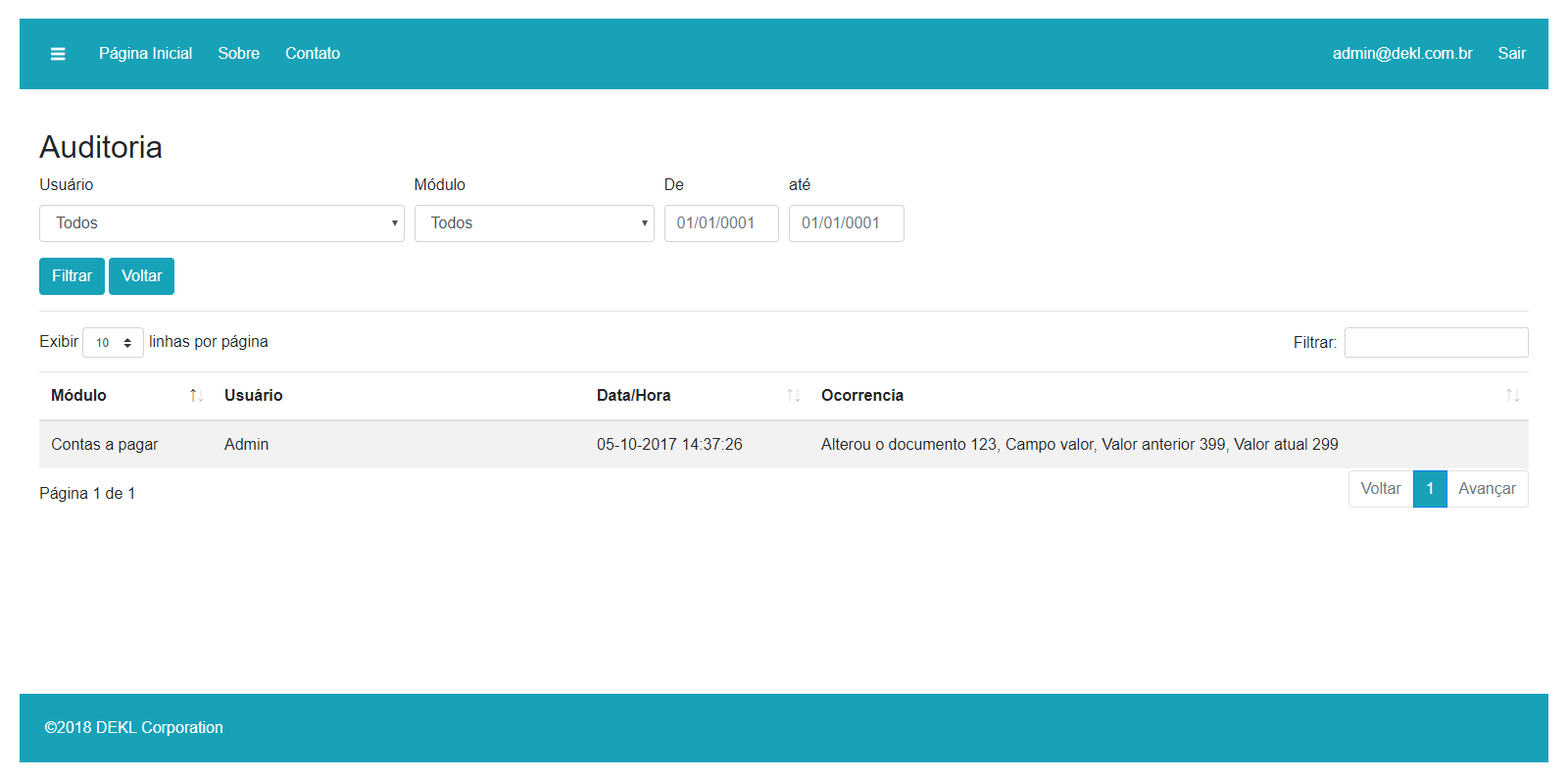


Figura 43 - Tela Gerenciamento de Auditoria do Sistema

## Relatórios

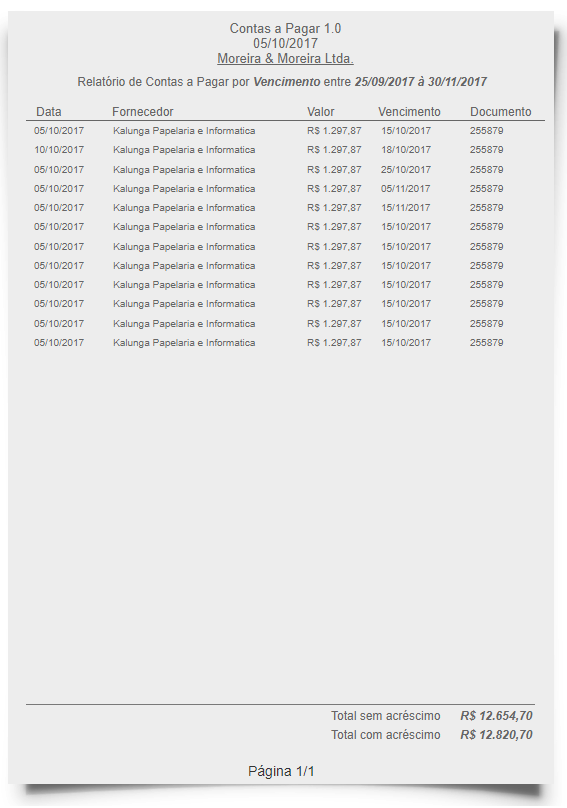


Figura 44 - Relatório Contas à Pagar

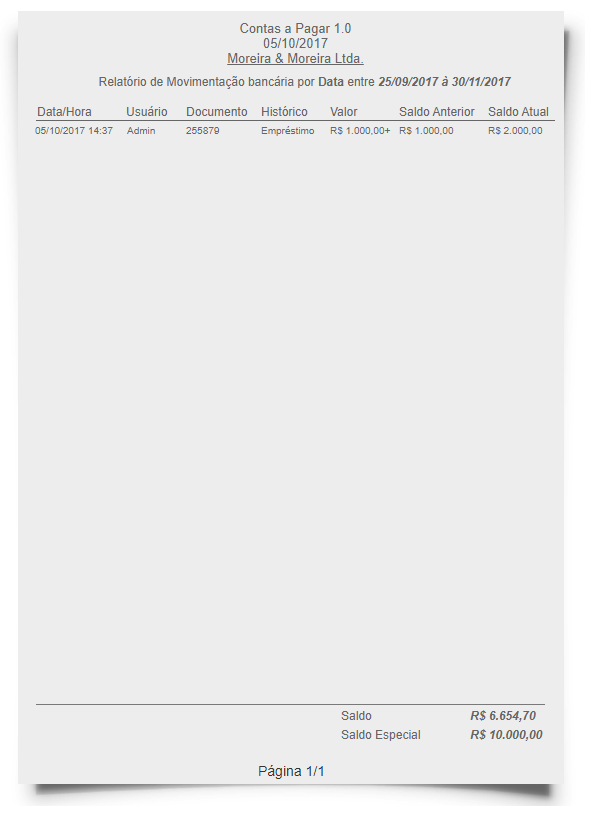


Figura 45 - Relatório Movimentação Bancária

# Implementação

Anexo em mídia CD.

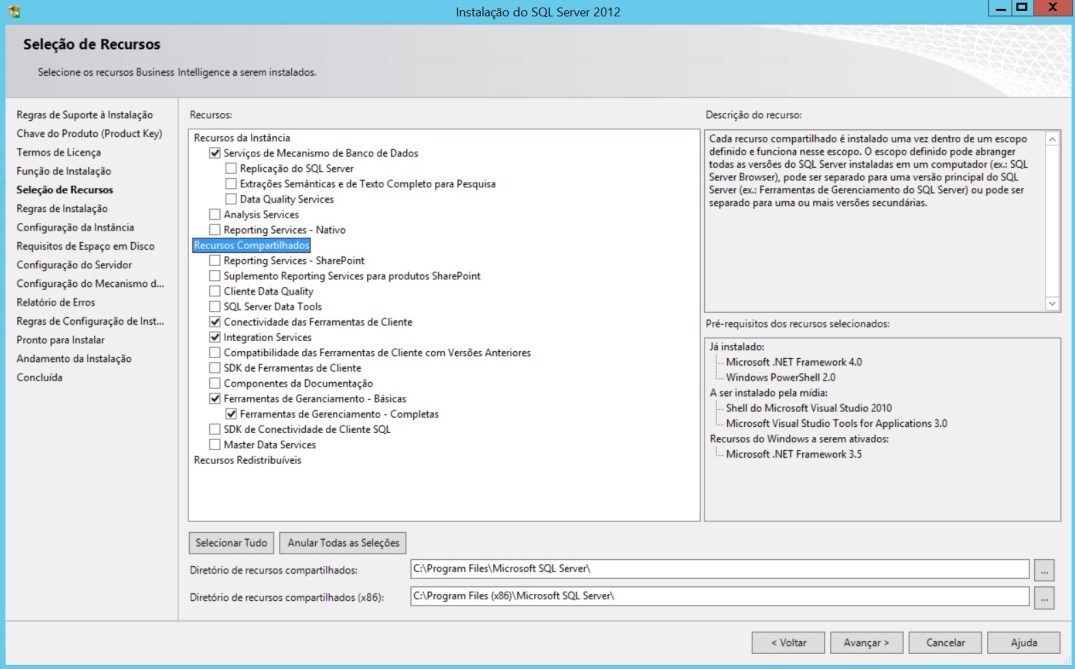
# Projeto de Teste

Não houve tempo hábil para realização dos devidos testes.

# Instalação do software

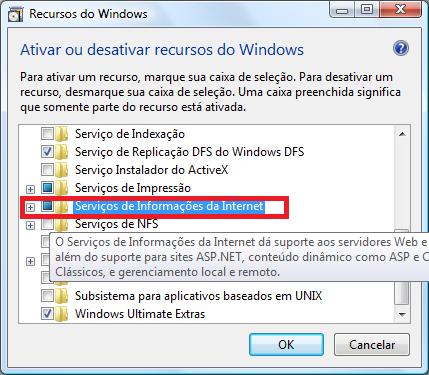
A instalação do software requer para o módulo cliente um terminal compatível com um navegador Google Chrome v70 ou superior ou Firefox v61 ou superior. O modulo servidor requer a instalação do SQL Server 2012, Net Framework 4 ou superior. Para o serviço web será utilizado o IIS express.

Durante a instalação do SQL Server use a opção ***“Nova instalação autônoma do SQL Server ou adicionar recursos a uma instalação existente”*** *selecione ok ou avançar sempre que possível, verifique se todos os requisitos listados na tela estão ok.* *Na seção****Funções de Instalação****selecione****Instalação de Recursos do SQL Server****e Clique em****Avançar.***



Deixe o ID da instancia como padrão e selecione avançar até completar a instalação.

Para o IIS vá em painel de controle, adicionar ou remover programas, ativar ou desativar recursos do Windows.



# Conclusão

O desenvolvimento deste trabalho tornou possível a análise de como um software adaptado as necessidades reais da empresa podem impactar de maneira positiva ações de rotina dentro de um único departamento e refletindo diretamente nas finanças e no melhor aproveitamento do tempo por parte dos envolvidos fazendo do ambiente mais produtivo.

A partir do estudo realizado constatou-se que a empresa possuía necessidades que atualmente não eram mais supridas pelo sistema obsoleto que era utilizado por todo o departamento de contas a pagar.

Para que fosse possível uma compreensão do problema apresentado, a rotina do departamento de contas a pagar foi monitorada presencialmente onde ficou clara as limitações do atual sistema com relação as necessidades apresentadas, além de informações básicas do documento de cobrança e uma total exposição das informações a ponto de deixá-las vulneráveis a alterações e até mesmo exclusões.

Para atingir os resultados esperados foi definido que o novo sistema deveria ter ampla compatibilidade não se limitando a uma única arquitetura de sistema operacional, e que os equipamentos atualmente utilizados deveriam estar dentro dos requisitos de software e hardware do novo sistema, sendo assim foi utilizada tecnologia web que será executado a partir dos browsers. O segundo objetivo é que deveria ser utilizada uma tecnologia que além de robusta permitisse que no futuro o resto do sistema poderia ser migrado e que a utilização de dispositivos móveis poderia ser adotada, então como escolha a plataforma .Net através do C# e o Entity framework utilizando como banco de dados o Microsoft SQL Server estariam no backend da aplicação trazendo toda a segurança e compatibilidade necessária e no frontend HTML5 com CSS e Javascript.

Ao final deste projeto ficou concluído que tudo aquilo que foi proposto foi realizado dentro das especificações documentadas, a nova ferramenta para simulação agora permite ao gestor uma projeção mais precisa dos pagamentos, as notificações de vencimentos próximos também tem se mostrado eficientes ao alertar os usuários para pagamento de títulos próximos de vencer e o maior controle de restrições de uso tem possibilitado ao gestor maior confiança e agilidade ao executar determinadas ações.

# Referências

**SCHMITZ, Daniel -** *Bootstrap 3: Framework front-end para desenvolvimento web e mobile. Ed. Leanpub (2014).*

***SILVA, Maurício Samy*** *- CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3, Ed. Novatec. 2011*

*https://novatec.com.br/livros/css3/capitulo9788575222898.pdf*

**JORGENSEN**, **Adam** - *Microsoft SQL Server 2012 Bible, Ebook:*

*http://www.amazon.com.br/Microsoft-SQL-Server-2012-Bible-ebook/dp/B008UYKX2S. 2012*

**CARLOS DOS SANTOS.** Microsoft. Fundamentos do Entity Framework 4. 2012. Disponível em: &lt;https://msdn.microsoft.com/pt-br/library/jj128157.aspx&gt;. Acesso em: 13 ago. 206.

# Anexo

# 10.1 Script do Banco de Dados

USE [DEKLCP]

GO

/\*\*\*\*\*\* Object: Table [dbo].[AccountToPay] Script Date: 22/11/2018 12:53:43 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AccountToPay](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Value] [money] NOT NULL,

[PaidValue] [money] NULL,

[PaymentDate] [datetime] NULL,

[Description] [varchar](60) NULL,

[MaturityDate] [datetime] NOT NULL,

[DailyInterest] [decimal](4, 2) NOT NULL,

[Penalty] [decimal](4, 2) NOT NULL,

[MonthlyAccount] [bit] NOT NULL,

[Priority] [int] NOT NULL,

[PaymentType] [int] NOT NULL,

[DocumentNumber] [varchar](80) NOT NULL,

[NumberOfInstallments] [smallint] NOT NULL,

[ProviderId] [int] NOT NULL,

[ApplicationUserId] [int] NOT NULL,

[ModuleId] [int] NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.AccountToPay] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Address] Script Date: 22/11/2018 12:53:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Address](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Street] [varchar](60) NOT NULL,

[Number] [varchar](20) NULL,

[ZipCode] [varchar](100) NOT NULL,

[Complement] [varchar](100) NOT NULL,

[Neighborhood] [varchar](60) NOT NULL,

[City] [varchar](60) NOT NULL,

[StateId] [int] NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.Address] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ApplicationRole] Script Date: 22/11/2018 12:53:58 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ApplicationRole](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Name] [varchar](256) NOT NULL,

CONSTRAINT [PK\_dbo.ApplicationRole] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ApplicationUser] Script Date: 22/11/2018 12:54:00 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ApplicationUser](

[Id] [int] IDENTITY(1,1) NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

[FirstName] [varchar](100) NULL,

[LastName] [varchar](100) NULL,

[Email] [varchar](100) NULL,

[EmailConfirmed] [bit] NOT NULL,

[PasswordHash] [varchar](100) NULL,

[SecurityStamp] [varchar](100) NULL,

[PhoneNumber] [varchar](100) NULL,

[PhoneNumberConfirmed] [bit] NOT NULL,

[TwoFactorEnabled] [bit] NOT NULL,

[LockoutEndDateUtc] [datetime] NULL,

[LockoutEnabled] [bit] NOT NULL,

[AccessFailedCount] [int] NOT NULL,

[UserName] [varchar](100) NULL,

CONSTRAINT [PK\_dbo.ApplicationUser] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ApplicationUserClaim] Script Date: 22/11/2018 12:54:00 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ApplicationUserClaim](

[Id] [int] IDENTITY(1,1) NOT NULL,

[UserId] [int] NOT NULL,

[ClaimType] [varchar](100) NULL,

[ClaimValue] [varchar](100) NULL,

[ApplicationUser\_Id] [int] NULL,

CONSTRAINT [PK\_dbo.ApplicationUserClaim] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ApplicationUserLogin] Script Date: 22/11/2018 12:54:00 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ApplicationUserLogin](

[LoginProvider] [varchar](100) NOT NULL,

[ProviderKey] [varchar](100) NOT NULL,

[UserId] [int] NOT NULL,

[ApplicationUser\_Id] [int] NULL,

CONSTRAINT [PK\_dbo.ApplicationUserLogin] PRIMARY KEY CLUSTERED

(

[LoginProvider] ASC,

[ProviderKey] ASC,

[UserId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ApplicationUserRole] Script Date: 22/11/2018 12:54:01 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ApplicationUserRole](

[UserId] [int] NOT NULL,

[RoleId] [int] NOT NULL,

[ApplicationUser\_Id] [int] NULL,

CONSTRAINT [PK\_dbo.ApplicationUserRole] PRIMARY KEY CLUSTERED

(

[UserId] ASC,

[RoleId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Audit] Script Date: 22/11/2018 12:54:01 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Audit](

[Id] [int] IDENTITY(1,1) NOT NULL,

[ApplicationUserId] [int] NOT NULL,

[ModuleId] [int] NOT NULL,

[Event] [varchar](100) NOT NULL,

[DateTime] [datetime] NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.Audit] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Bank] Script Date: 22/11/2018 12:54:01 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Bank](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Number] [smallint] NOT NULL,

[Name] [varchar](60) NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.Bank] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[BankAgency] Script Date: 22/11/2018 12:54:01 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[BankAgency](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Number] [smallint] NOT NULL,

[AddressId] [int] NOT NULL,

[ManagerName] [varchar](80) NULL,

[PhoneNumber] [varchar](20) NULL,

[Email] [varchar](80) NULL,

[BankId] [int] NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.BankAgency] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[BankTransaction] Script Date: 22/11/2018 12:54:02 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[BankTransaction](

[Id] [int] IDENTITY(1,1) NOT NULL,

[AccountToPayId] [int] NOT NULL,

[InternalBankAccountId] [int] NOT NULL,

[ProviderBankAccountId] [int] NOT NULL,

[NewBalance] [decimal](4, 2) NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.BankTransaction] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Clients] Script Date: 22/11/2018 12:54:02 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Clients](

[Id] [int] IDENTITY(1,1) NOT NULL,

[ClientKey] [varchar](100) NULL,

[ApplicationUser\_Id] [int] NULL,

CONSTRAINT [PK\_dbo.Clients] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Installment] Script Date: 22/11/2018 12:54:02 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Installment](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Value] [money] NOT NULL,

[PaidValue] [money] NULL,

[MaturityDate] [datetime] NOT NULL,

[PaymentDate] [datetime] NULL,

[AccountToPayId] [int] NOT NULL,

[ApplicationUserId] [int] NOT NULL,

[ModuleId] [int] NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.Installment] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[InternalBankAccount] Script Date: 22/11/2018 12:54:03 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[InternalBankAccount](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Number] [varchar](20) NOT NULL,

[Name] [varchar](80) NOT NULL,

[Balance] [money] NOT NULL,

[BankAgencyId] [int] NOT NULL,

[ApplicationUserId] [int] NOT NULL,

[ModuleId] [int] NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.InternalBankAccount] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Module] Script Date: 22/11/2018 12:54:03 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Module](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Name] [varchar](60) NOT NULL,

[Description] [varchar](80) NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.Module] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[PaymentSimulator] Script Date: 22/11/2018 12:54:03 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[PaymentSimulator](

[Id] [int] IDENTITY(1,1) NOT NULL,

[PaymentDate] [datetime] NOT NULL,

[InternalBankAccountId] [int] NOT NULL,

[Observations] [varchar](100) NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.PaymentSimulator] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[PaymentSimulatorAccountToPay] Script Date: 22/11/2018 12:54:03 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[PaymentSimulatorAccountToPay](

[PaymentSimulator\_Id] [int] NOT NULL,

[AccountToPay\_Id] [int] NOT NULL,

CONSTRAINT [PK\_dbo.PaymentSimulatorAccountToPay] PRIMARY KEY CLUSTERED

(

[PaymentSimulator\_Id] ASC,

[AccountToPay\_Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Provider] Script Date: 22/11/2018 12:54:04 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Provider](

[Id] [int] IDENTITY(1,1) NOT NULL,

[PhoneNumber] [varchar](20) NULL,

[Email] [varchar](80) NULL,

[AddressId] [int] NOT NULL,

[ApplicationUserId] [int] NOT NULL,

[ModuleId] [int] NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.Provider] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ProviderBankAccount] Script Date: 22/11/2018 12:54:04 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ProviderBankAccount](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Number] [varchar](20) NOT NULL,

[Name] [varchar](80) NOT NULL,

[BankAgencyId] [int] NOT NULL,

[ProviderId] [int] NOT NULL,

[ApplicationUserId] [int] NOT NULL,

[ModuleId] [int] NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.ProviderBankAccount] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ProviderLegalPerson] Script Date: 22/11/2018 12:54:04 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ProviderLegalPerson](

[Id] [int] NOT NULL,

[CorporateName] [varchar](100) NOT NULL,

[CNPJ] [varchar](14) NOT NULL,

[MunicipalRegistration] [varchar](11) NULL,

[StateRegistration] [varchar](9) NULL,

CONSTRAINT [PK\_dbo.ProviderLegalPerson] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ProviderPhysicalPersonMap] Script Date: 22/11/2018 12:54:04 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ProviderPhysicalPersonMap](

[Id] [int] NOT NULL,

[Name] [varchar](100) NOT NULL,

[CPF] [varchar](11) NOT NULL,

CONSTRAINT [PK\_dbo.ProviderPhysicalPersonMap] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[State] Script Date: 22/11/2018 12:54:05 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[State](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Name] [varchar](60) NOT NULL,

[Initials] [char](2) NOT NULL,

[AddedDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NULL,

[Active] [bit] NOT NULL,

CONSTRAINT [PK\_dbo.State] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[AccountToPay] WITH CHECK ADD CONSTRAINT [FK\_dbo.AccountToPay\_dbo.ApplicationUser\_ApplicationUserId] FOREIGN KEY([ApplicationUserId])

REFERENCES [dbo].[ApplicationUser] ([Id])

GO

ALTER TABLE [dbo].[AccountToPay] CHECK CONSTRAINT [FK\_dbo.AccountToPay\_dbo.ApplicationUser\_ApplicationUserId]

GO

ALTER TABLE [dbo].[AccountToPay] WITH CHECK ADD CONSTRAINT [FK\_dbo.AccountToPay\_dbo.Module\_ModuleId] FOREIGN KEY([ModuleId])

REFERENCES [dbo].[Module] ([Id])

GO

ALTER TABLE [dbo].[AccountToPay] CHECK CONSTRAINT [FK\_dbo.AccountToPay\_dbo.Module\_ModuleId]

GO

ALTER TABLE [dbo].[AccountToPay] WITH CHECK ADD CONSTRAINT [FK\_dbo.AccountToPay\_dbo.Provider\_ProviderId] FOREIGN KEY([ProviderId])

REFERENCES [dbo].[Provider] ([Id])

GO

ALTER TABLE [dbo].[AccountToPay] CHECK CONSTRAINT [FK\_dbo.AccountToPay\_dbo.Provider\_ProviderId]

GO

ALTER TABLE [dbo].[Address] WITH CHECK ADD CONSTRAINT [FK\_dbo.Address\_dbo.State\_StateId] FOREIGN KEY([StateId])

REFERENCES [dbo].[State] ([Id])

GO

ALTER TABLE [dbo].[Address] CHECK CONSTRAINT [FK\_dbo.Address\_dbo.State\_StateId]

GO

ALTER TABLE [dbo].[ApplicationUserClaim] WITH CHECK ADD CONSTRAINT [FK\_dbo.ApplicationUserClaim\_dbo.ApplicationUser\_ApplicationUser\_Id] FOREIGN KEY([ApplicationUser\_Id])

REFERENCES [dbo].[ApplicationUser] ([Id])

GO

ALTER TABLE [dbo].[ApplicationUserClaim] CHECK CONSTRAINT [FK\_dbo.ApplicationUserClaim\_dbo.ApplicationUser\_ApplicationUser\_Id]

GO

ALTER TABLE [dbo].[ApplicationUserLogin] WITH CHECK ADD CONSTRAINT [FK\_dbo.ApplicationUserLogin\_dbo.ApplicationUser\_ApplicationUser\_Id] FOREIGN KEY([ApplicationUser\_Id])

REFERENCES [dbo].[ApplicationUser] ([Id])

GO

ALTER TABLE [dbo].[ApplicationUserLogin] CHECK CONSTRAINT [FK\_dbo.ApplicationUserLogin\_dbo.ApplicationUser\_ApplicationUser\_Id]

GO

ALTER TABLE [dbo].[ApplicationUserRole] WITH CHECK ADD CONSTRAINT [FK\_dbo.ApplicationUserRole\_dbo.ApplicationRole\_RoleId] FOREIGN KEY([RoleId])

REFERENCES [dbo].[ApplicationRole] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[ApplicationUserRole] CHECK CONSTRAINT [FK\_dbo.ApplicationUserRole\_dbo.ApplicationRole\_RoleId]

GO

ALTER TABLE [dbo].[ApplicationUserRole] WITH CHECK ADD CONSTRAINT [FK\_dbo.ApplicationUserRole\_dbo.ApplicationUser\_ApplicationUser\_Id] FOREIGN KEY([ApplicationUser\_Id])

REFERENCES [dbo].[ApplicationUser] ([Id])

GO

ALTER TABLE [dbo].[ApplicationUserRole] CHECK CONSTRAINT [FK\_dbo.ApplicationUserRole\_dbo.ApplicationUser\_ApplicationUser\_Id]

GO

ALTER TABLE [dbo].[Audit] WITH CHECK ADD CONSTRAINT [FK\_dbo.Audit\_dbo.ApplicationUser\_ApplicationUserId] FOREIGN KEY([ApplicationUserId])

REFERENCES [dbo].[ApplicationUser] ([Id])

GO

ALTER TABLE [dbo].[Audit] CHECK CONSTRAINT [FK\_dbo.Audit\_dbo.ApplicationUser\_ApplicationUserId]

GO

ALTER TABLE [dbo].[Audit] WITH CHECK ADD CONSTRAINT [FK\_dbo.Audit\_dbo.Module\_ModuleId] FOREIGN KEY([ModuleId])

REFERENCES [dbo].[Module] ([Id])

GO

ALTER TABLE [dbo].[Audit] CHECK CONSTRAINT [FK\_dbo.Audit\_dbo.Module\_ModuleId]

GO

ALTER TABLE [dbo].[BankAgency] WITH CHECK ADD CONSTRAINT [FK\_dbo.BankAgency\_dbo.Address\_Id] FOREIGN KEY([Id])

REFERENCES [dbo].[Address] ([Id])

GO

ALTER TABLE [dbo].[BankAgency] CHECK CONSTRAINT [FK\_dbo.BankAgency\_dbo.Address\_Id]

GO

ALTER TABLE [dbo].[BankAgency] WITH CHECK ADD CONSTRAINT [FK\_dbo.BankAgency\_dbo.Bank\_BankId] FOREIGN KEY([BankId])

REFERENCES [dbo].[Bank] ([Id])

GO

ALTER TABLE [dbo].[BankAgency] CHECK CONSTRAINT [FK\_dbo.BankAgency\_dbo.Bank\_BankId]

GO

ALTER TABLE [dbo].[BankTransaction] WITH CHECK ADD CONSTRAINT [FK\_dbo.BankTransaction\_dbo.AccountToPay\_AccountToPayId] FOREIGN KEY([AccountToPayId])

REFERENCES [dbo].[AccountToPay] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[BankTransaction] CHECK CONSTRAINT [FK\_dbo.BankTransaction\_dbo.AccountToPay\_AccountToPayId]

GO

ALTER TABLE [dbo].[BankTransaction] WITH CHECK ADD CONSTRAINT [FK\_dbo.BankTransaction\_dbo.InternalBankAccount\_InternalBankAccountId] FOREIGN KEY([InternalBankAccountId])

REFERENCES [dbo].[InternalBankAccount] ([Id])

GO

ALTER TABLE [dbo].[BankTransaction] CHECK CONSTRAINT [FK\_dbo.BankTransaction\_dbo.InternalBankAccount\_InternalBankAccountId]

GO

ALTER TABLE [dbo].[BankTransaction] WITH CHECK ADD CONSTRAINT [FK\_dbo.BankTransaction\_dbo.ProviderBankAccount\_ProviderBankAccountId] FOREIGN KEY([ProviderBankAccountId])

REFERENCES [dbo].[ProviderBankAccount] ([Id])

GO

ALTER TABLE [dbo].[BankTransaction] CHECK CONSTRAINT [FK\_dbo.BankTransaction\_dbo.ProviderBankAccount\_ProviderBankAccountId]

GO

ALTER TABLE [dbo].[Clients] WITH CHECK ADD CONSTRAINT [FK\_dbo.Clients\_dbo.ApplicationUser\_ApplicationUser\_Id] FOREIGN KEY([ApplicationUser\_Id])

REFERENCES [dbo].[ApplicationUser] ([Id])

GO

ALTER TABLE [dbo].[Clients] CHECK CONSTRAINT [FK\_dbo.Clients\_dbo.ApplicationUser\_ApplicationUser\_Id]

GO

ALTER TABLE [dbo].[Installment] WITH CHECK ADD CONSTRAINT [FK\_dbo.Installment\_dbo.AccountToPay\_AccountToPayId] FOREIGN KEY([AccountToPayId])

REFERENCES [dbo].[AccountToPay] ([Id])

GO

ALTER TABLE [dbo].[Installment] CHECK CONSTRAINT [FK\_dbo.Installment\_dbo.AccountToPay\_AccountToPayId]

GO

ALTER TABLE [dbo].[Installment] WITH CHECK ADD CONSTRAINT [FK\_dbo.Installment\_dbo.ApplicationUser\_ApplicationUserId] FOREIGN KEY([ApplicationUserId])

REFERENCES [dbo].[ApplicationUser] ([Id])

GO

ALTER TABLE [dbo].[Installment] CHECK CONSTRAINT [FK\_dbo.Installment\_dbo.ApplicationUser\_ApplicationUserId]

GO

ALTER TABLE [dbo].[Installment] WITH CHECK ADD CONSTRAINT [FK\_dbo.Installment\_dbo.Module\_ModuleId] FOREIGN KEY([ModuleId])

REFERENCES [dbo].[Module] ([Id])

GO

ALTER TABLE [dbo].[Installment] CHECK CONSTRAINT [FK\_dbo.Installment\_dbo.Module\_ModuleId]

GO

ALTER TABLE [dbo].[InternalBankAccount] WITH CHECK ADD CONSTRAINT [FK\_dbo.InternalBankAccount\_dbo.ApplicationUser\_ApplicationUserId] FOREIGN KEY([ApplicationUserId])

REFERENCES [dbo].[ApplicationUser] ([Id])

GO

ALTER TABLE [dbo].[InternalBankAccount] CHECK CONSTRAINT [FK\_dbo.InternalBankAccount\_dbo.ApplicationUser\_ApplicationUserId]

GO

ALTER TABLE [dbo].[InternalBankAccount] WITH CHECK ADD CONSTRAINT [FK\_dbo.InternalBankAccount\_dbo.BankAgency\_BankAgencyId] FOREIGN KEY([BankAgencyId])

REFERENCES [dbo].[BankAgency] ([Id])

GO

ALTER TABLE [dbo].[InternalBankAccount] CHECK CONSTRAINT [FK\_dbo.InternalBankAccount\_dbo.BankAgency\_BankAgencyId]

GO

ALTER TABLE [dbo].[InternalBankAccount] WITH CHECK ADD CONSTRAINT [FK\_dbo.InternalBankAccount\_dbo.Module\_ModuleId] FOREIGN KEY([ModuleId])

REFERENCES [dbo].[Module] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[InternalBankAccount] CHECK CONSTRAINT [FK\_dbo.InternalBankAccount\_dbo.Module\_ModuleId]

GO

ALTER TABLE [dbo].[PaymentSimulator] WITH CHECK ADD CONSTRAINT [FK\_dbo.PaymentSimulator\_dbo.InternalBankAccount\_InternalBankAccountId] FOREIGN KEY([InternalBankAccountId])

REFERENCES [dbo].[InternalBankAccount] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[PaymentSimulator] CHECK CONSTRAINT [FK\_dbo.PaymentSimulator\_dbo.InternalBankAccount\_InternalBankAccountId]

GO

ALTER TABLE [dbo].[PaymentSimulatorAccountToPay] WITH CHECK ADD CONSTRAINT [FK\_dbo.PaymentSimulatorAccountToPay\_dbo.AccountToPay\_AccountToPay\_Id] FOREIGN KEY([AccountToPay\_Id])

REFERENCES [dbo].[AccountToPay] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[PaymentSimulatorAccountToPay] CHECK CONSTRAINT [FK\_dbo.PaymentSimulatorAccountToPay\_dbo.AccountToPay\_AccountToPay\_Id]

GO

ALTER TABLE [dbo].[PaymentSimulatorAccountToPay] WITH CHECK ADD CONSTRAINT [FK\_dbo.PaymentSimulatorAccountToPay\_dbo.PaymentSimulator\_PaymentSimulator\_Id] FOREIGN KEY([PaymentSimulator\_Id])

REFERENCES [dbo].[PaymentSimulator] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[PaymentSimulatorAccountToPay] CHECK CONSTRAINT [FK\_dbo.PaymentSimulatorAccountToPay\_dbo.PaymentSimulator\_PaymentSimulator\_Id]

GO

ALTER TABLE [dbo].[Provider] WITH CHECK ADD CONSTRAINT [FK\_dbo.Provider\_dbo.Address\_AddressId] FOREIGN KEY([AddressId])

REFERENCES [dbo].[Address] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[Provider] CHECK CONSTRAINT [FK\_dbo.Provider\_dbo.Address\_AddressId]

GO

ALTER TABLE [dbo].[Provider] WITH CHECK ADD CONSTRAINT [FK\_dbo.Provider\_dbo.ApplicationUser\_ApplicationUserId] FOREIGN KEY([ApplicationUserId])

REFERENCES [dbo].[ApplicationUser] ([Id])

GO

ALTER TABLE [dbo].[Provider] CHECK CONSTRAINT [FK\_dbo.Provider\_dbo.ApplicationUser\_ApplicationUserId]

GO

ALTER TABLE [dbo].[Provider] WITH CHECK ADD CONSTRAINT [FK\_dbo.Provider\_dbo.Module\_ModuleId] FOREIGN KEY([ModuleId])

REFERENCES [dbo].[Module] ([Id])

GO

ALTER TABLE [dbo].[Provider] CHECK CONSTRAINT [FK\_dbo.Provider\_dbo.Module\_ModuleId]

GO

ALTER TABLE [dbo].[ProviderBankAccount] WITH CHECK ADD CONSTRAINT [FK\_dbo.ProviderBankAccount\_dbo.ApplicationUser\_ApplicationUserId] FOREIGN KEY([ApplicationUserId])

REFERENCES [dbo].[ApplicationUser] ([Id])

GO

ALTER TABLE [dbo].[ProviderBankAccount] CHECK CONSTRAINT [FK\_dbo.ProviderBankAccount\_dbo.ApplicationUser\_ApplicationUserId]

GO

ALTER TABLE [dbo].[ProviderBankAccount] WITH CHECK ADD CONSTRAINT [FK\_dbo.ProviderBankAccount\_dbo.BankAgency\_BankAgencyId] FOREIGN KEY([BankAgencyId])

REFERENCES [dbo].[BankAgency] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[ProviderBankAccount] CHECK CONSTRAINT [FK\_dbo.ProviderBankAccount\_dbo.BankAgency\_BankAgencyId]

GO

ALTER TABLE [dbo].[ProviderBankAccount] WITH CHECK ADD CONSTRAINT [FK\_dbo.ProviderBankAccount\_dbo.Module\_ModuleId] FOREIGN KEY([ModuleId])

REFERENCES [dbo].[Module] ([Id])

GO

ALTER TABLE [dbo].[ProviderBankAccount] CHECK CONSTRAINT [FK\_dbo.ProviderBankAccount\_dbo.Module\_ModuleId]

GO

ALTER TABLE [dbo].[ProviderBankAccount] WITH CHECK ADD CONSTRAINT [FK\_dbo.ProviderBankAccount\_dbo.Provider\_ProviderId] FOREIGN KEY([ProviderId])

REFERENCES [dbo].[Provider] ([Id])

GO

ALTER TABLE [dbo].[ProviderBankAccount] CHECK CONSTRAINT [FK\_dbo.ProviderBankAccount\_dbo.Provider\_ProviderId]

GO

ALTER TABLE [dbo].[ProviderLegalPerson] WITH CHECK ADD CONSTRAINT [FK\_dbo.ProviderLegalPerson\_dbo.Provider\_Id] FOREIGN KEY([Id])

REFERENCES [dbo].[Provider] ([Id])

GO

ALTER TABLE [dbo].[ProviderLegalPerson] CHECK CONSTRAINT [FK\_dbo.ProviderLegalPerson\_dbo.Provider\_Id]

GO

ALTER TABLE [dbo].[ProviderPhysicalPersonMap] WITH CHECK ADD CONSTRAINT [FK\_dbo.ProviderPhysicalPersonMap\_dbo.Provider\_Id] FOREIGN KEY([Id])

REFERENCES [dbo].[Provider] ([Id])

GO

ALTER TABLE [dbo].[ProviderPhysicalPersonMap] CHECK CONSTRAINT [FK\_dbo.ProviderPhysicalPersonMap\_dbo.Provider\_Id]

GO