**UNIVERSIDADE TUIUTI DO PARANÁ**

**ANTONIO MARCOS DA SILVA PIRES**

**RENATO DROZDEK JUNIOR**

**RODRIGO FERREIRA DOS ANJOS**

**SAMANTHA SOARES HEIL**

**APLICATIVO GERENCIADOR DE CONTAS**

**CURITIBA**

**2017**

**ANTONIO MARCOS DA SILVA PIRES**

**RENATO DROZDEK JUNIOR**

**RODRIGO FERREIRA DOS ANJOS**

**SAMANTHA SOARES HEIL**

**APLICATIVO GERENCIADOR DE CONTAS**

Trabalho apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistema, da Universidade Tuiuti do Paraná, como requisito avaliativo da disciplina de Programação de Banco de Dados.

Professor: Chauã Queirolo.

**CURITIBA**

**2017**

**LISTA DE IMAGENS**

IMAGEM 1 – MODELO CONCEITUAL 10

IMAGEM 2 – MODELO LÓGICO 11

**LISTA DE TABELAS**

TABELA 1 – CHECKLIST DOS ITENS OBRIGATÓRIOS 9

**SUMÁRIO**

**1 INTRODUÇÃO** 5

**2 GERENCIADOR DE CONTAS** 6

2.1 TABELAS 6

2.2 DESCRIÇÃO DAS TABELAS 6

**3 REGRAS DE NEGÓCIO** 8

3.1 CHECKLIST DOS ITENS OBRIGATÓRIOS 9

**4 MODELOS CONCEITUAL E LÓGICO** 10

**5 LINK DO FONTE** 12

**6 CONCLUSÃO** 13

**REFERÊNCIAS** 14

**1 INTRODUÇÃO**

Este estudo tem por objetivo definir, estruturar e implementar um banco de dados usando a modelagem conceitual, lógica e comandos para SGDB MySQL.

A ideia da equipe é criar uma plataforma onde o usuário possa ter controle de seus castos bancários independente do tipo de pagamento utilizado, pois o sistema poderá armazenar vários formatos de transação, como por exemplo: pagamento com cartões de débito, crédito, cheques ou, até mesmo, transferências bancárias.

Também será possível, além dos tipos de pagamentos, o cadastramento de várias instituições bancárias, aumentando ainda mais o benefício do sistema.

**2 GERENCIADOR DE CONTAS**

O sistema será um controle financeiro pessoal que contará com relatórios das despesas[[1]](#footnote-1), alertas1 e backups1 mensais na nuvem.

2.1 TABELAS

* Usuários (usr) \*log;
* Tipo de Usuário (tp\_usr);
* Transações (trs) \*log;
* Tipo de Transação (tp\_trs);
* Forma de pagamento (form\_pgto);
* Log (log\_geral);
* Tipo de Moeda (tp\_moeda) \*log;
* Banco (bnc).

2.2 DESCRIÇÃO DAS TABELAS

* Usuários: Tabela conterá o login e senha do(s) possível(eis) usuário (s). Esta tabela permitirá a criação automática de um registro log;
* Tipo de Usuário: Tabela conterá os tipos de usuário. Ex.:
  + Básico (Registro + (Ads));
  + VIP (Básico + Relatório - (Ads));
  + Premium + (VIP + Backup);
* Transações: Tabela conterá as transações feitas pelo usuário. Esta tabela permitirá a criação automática de um registro log. Ex.:
  + Pagamento de Contas;
  + Transferências.
* Tipo de Transação: Tabela conterá o tipo da transação. Ex.:
  + Débito;
  + Crédito;
  + Transferência.
* Forma de Pagamento: Tabela conterá as formas de pagamento disponíveis ao usuário. Ex.:
  + Dinheiro;
  + Cartão;
  + Cheque;
  + TED/DOC.
* Log: Tabela conterá o registro de atividades realizadas pelo usuário;
* Tipo de Moeda: Tabela conterá o tipo de moeda utilizada no momento da compra. Esta tabela permitirá a criação automática de um registro log;
* Banco: Tabela conterá o cadastro do(s) banco(s) castrado(s) pelo usuário.

**3 REGRAS DE NEGÓCIO**

O sistema tem 4 permissões diferentes:

* Administrador – Acesso total e ilimitado ao sistema;
* Prime – Pode fazer até 10 transações no período de 30 dias;
* Premium – Pode fazer até 6 transações no período de 30 dias;
* Standard – Pode fazer até 3 transações no período de 30 dias.

Nenhum usuário pode criar nem deletar transações que não sejam suas, até mesmo o administrador não tem essa permissão.

O sistema tem login e senha para que nenhuma transação seja realizada de forma anônima.

Para criar uma transação todos os campos devem ser preenchidos, tais como:

* Usuario\_id;
* Tipo\_transacao\_id;
* Banco\_origem\_id;
* Banco\_destino\_id;
* Forma\_pagamento\_id;
* Tipo\_moeda;
* Valor.

Para se criar uma transação o banco deve ser previamente cadastrado pelo administrador do sistema, não é possível qualquer tipo de usuário criar seus próprios bancos.

Para se criar uma transação a forma de pagamento deve ser previamente cadastrado pelo administrador do sistema, não é possível qualquer tipo de usuário criar suas próprias formas de pagamento.

Para se criar uma transação o tipo da moeda deve ser previamente cadastrado pelo administrador do sistema, não é possível qualquer tipo de usuário criar suas próprias moedas.

Para um usuário ser excluído não devem existir transações.

Todas as criações/alterações/exclusões serão registradas em um LOG para posteriores auditorias.

3.1 CHECKLIST DOS ITENS OBRIGATÓRIOS

TABELA 1 – CHECKLIST DOS ITENS OBRIGATÓRIOS

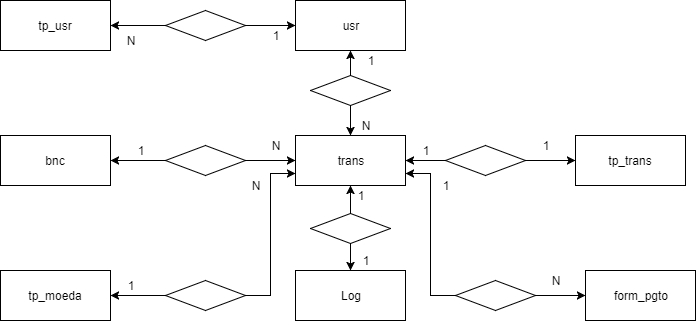
|  |  |  |
| --- | --- | --- |
| DESCRIÇÃO | NÃO CONCLUÍDO | CONCLUÍDO |
| Todas operações devem ser realizadas via Stored Procedures ou Triggers |  | OK |
| O BD deverá ter no mínimo 8 tabelas |  | OK |
| O BD deverá armazenar o nome, login, e senha dos usuários do sistema |  | OK |
| O BD deverá possuir pelo menos 3 perfis de acesso que devem limitar as ações que o usuário pode realizar no sistema |  | OK |
| O BD deverá gravar em uma tabela logs de operação de INSERT, DELETE, UPDATE realizadas em tabelas críticas do sistema |  | OK |
| O BD deverá permitir que as operações sejam realizadas somente pelos usuários que possuem perfil de acesso adequado |  | OK |
| O BD deverá ter 3 Stored Procedures que gerem relatórios sobre o uso do sistema |  | OK |

FONTE: dos próprios autores

**4 MODELOS CONCEITUAL E LÓGICO**

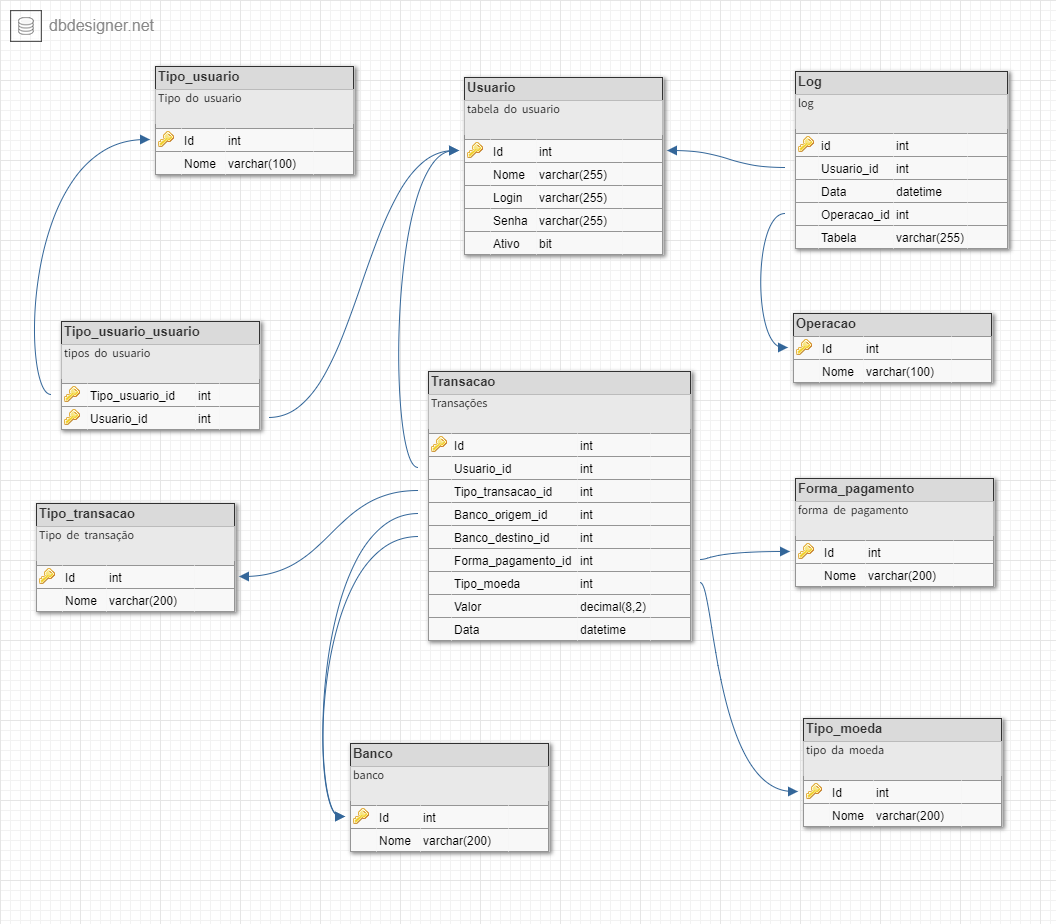
As imagens a seguir mostram os modelos conceitual e lógico do banco de dados proposto.

IMAGEM 1 – MODELO CONCEITUAL



FONTE: dos próprios autores

IMAGEM 2 – MODELO LÓGICO



FONTE: dos próprios autores

**5 LINK DO FONTE**

Abaixo segue o *link* para que se possa ver o arquivo fonte do banco de dados.

*https://github.com/webbester/RodrigodAnjos/blob/webbester-ED\_BD/BD-Gerenciador\_de\_contas.sql*

**6 CONCLUSÃO**

Muitos esforços têm sido feitos em busca de uma maior eficiência e facilidade para se trabalhar com aplicações de banco de dados.

Existe no mercado uma variedade de gerenciadores de contas onde este foi baseado, pode-se concluir com este estudo que por mais simples que seja a aplicação, exige muito da programação no banco de dados como a criação de tabelas, os perfis de acesso que são encarregados por limitar ações dos usuários no sistema, os logs de operações que na maioria dos bancos de dados são obrigatórios, e as Stored Procedures que geram relatórios sobre o uso do sistema.

**REFERÊNCIAS**

ALBUQUERQUE, Jefferson. Como criar uma procedure no MySQL. *ScriptCase*, 2017. Disponível em: https://suporte.scriptcase.com.br/pt-br/article/451-como-criar-uma-procedure-no-mysql. Acesso em: 12 set. 2017.

BIANCHI, Wagner. Stored Procedures no MYSQL. *iMaster*, 2007. Disponível em: https://imasters.com.br/artigo/7556/mysql/stored-procedures-no-mysql?trace=1519021197&source=single. Acesso em: 19 set. 2017.

EXEMPLO de como criar uma trigger em Mysql. *Locaweb*, 2017. Disponível em: https://ajuda.locaweb.com.br/pt-br/Exemplo\_de\_como\_criar\_uma\_trigger\_no\_Mysql. Acesso em: 19 set. 2017.

RODRIGUES, Joel. MySQL Básico: Triggers. *DevMedia*. Disponível em: http://www.devmedia.com.br/mysql-basico-triggers/37462. Acesso em: 19 set. 2017.

RODRIGUES, Joel. Stored Procedures no MySQL. *DevMedia*. Disponível em: http://www.devmedia.com.br/stored-procedures-no-mysql/29030. Acesso em: 12 set. 2017.

1. Dependendo da Assinatura do Usuário, este poderá ou não ter este serviço disponível. [↑](#footnote-ref-1)