

SISTEMA DE ANÁLISE E MINERAÇÃO DE DADOS DE TRANSAÇÕES ELETRÔNICAS REGIMENTADAS PELA ISO8583

Jackson Adriano Machado Costa
Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
jacksonadriano@gmail.com
Prof^ª Flávia Pereira de Carvalho
fpereira@faccat.br

Resumo

Este trabalho apresenta um sistema cujo objetivo é, principalmente, diminuir as ocorrências de erros no processo de captura e processamento de transações eletrônicas que são normatizadas pela ISO8583. Exemplos de empresas que utilizam este tipo de transações eletrônicas são as da área de recarga de créditos de telefones celulares. O sistema desenvolvido pode detectar gargalos nas aplicações que fazem parte deste processo, a fim de evitar futuras indisponibilidades sistêmicas que afetem o usuário final. As informações geradas pelo sistema desenvolvido fornecem dados para tomadas de decisões, possibilitando a definição de futuros projetos para corrigir erros identificados e adequar os demais sistemas que interagem com o sistema ao qual esta ferramenta se destina. Com base nos dados e relatórios gerados pelo sistema, é possível readequar as equipes para agirem de forma proativa, tratando problemas antes mesmo que estes realmente venham a acontecer em grande escala, melhorando o controle por parte das equipes que prestam suporte ao serviço de transações eletrônicas.

Palavras-chave: transações eletrônicas de dados, ISO8583, sistemas baseados na web, joomla.

SYSTEM ANALYSIS AND DATA MINING REGULATIONS FOR ELECTRONIC TRANSACTIONS ISO8583

Abstract

This work presents a system whose purpose is mainly to reduce the occurrences of errors in the capture and processing of electronic transactions that are normalized by ISO8583. Examples of companies that use this type of electronic transactions are those of the recharge area of mobile phone credits. The developed system can detect bottlenecks in applications that are part of this process in order to prevent future systemic outages that affect the end user. The information generated by the system provides data for decision making, enabling the definition of future projects to correct identified errors and adjust the other systems that interact with the system to which this tool is intended. Based on data and reports generated by the system, you can readjust the teams to act proactively addressing problems before they actually come to pass on a large scale, improving the control by the teams who support the service of electronic transactions.

Key-words: *transactions, electronic databases, ISO8583, web-based systems, joomla.*

1. Introdução

A realização de transações eletrônicas de dados financeiros de cartões *private label*¹, com bandeiras associadas e até mesmo transações de recarga eletrônica de celular, seguem a norma ISO 8583:1987. Existem empresas responsáveis pela captura e processamento destas transações que utilizam diversos meios de capturas, tais como:

- POS (*Point of Sale*²);
- EFT (*Electronic Funds Transfer*³);
- URA (Unidade de Resposta Audível);
- WEB (Internet);
- ATM (*Automated Teller Machine*⁴).

Internamente, estas empresas geralmente utilizam um protocolo próprio para tratamento destas transações e quando as transações são encaminhadas para a empresa parceira que efetua a autorização da solicitação de compra, no caso de transações online, estas transações são adequadas à norma citada acima, e toda a troca de mensagens que é feita com esta empresa parceira, seguem o padrão descrito nesta norma.

Como o fluxo de transações capturadas por estas redes tem um volume relativamente grande, muitas vezes não se tem o controle necessário sobre o que está sendo armazenado nos arquivos de registros (logs) de texto que são gerados pelas aplicações, como está transcorrendo a comunicação do servidor da aplicação e os demais sistemas, a quantidade e a periodicidade onde são apresentados erros que podem ser tratados se fossem de conhecimento das equipes envolvidas no processo de suporte e desenvolvimento destas aplicações.

No mercado de empresas de tecnologia em captura e processamento de transações eletrônicas, cada vez mais é buscado o “erro zero”, que é a situação onde se encontra o menor índice de indisponibilidade dos produtos oferecidos ao cliente, o menor índice de problemas com as aplicações que disponibilizam estes produtos e também o menor índice de indisponibilidades das operadoras que autorizam as transações. Em datas comemorativas onde o fluxo transacional tende a

¹ *Private Label* – Cartões de bandeira proprietária, não sendo de aceitação universal em todo comércio. Geralmente utilizado em uma rede específica de estabelecimentos.

² *Point of Sale* – Ponto de venda. Terminais especializados em captura de transações eletrônicas.

³ *Electronic Funds Transfer* – Transferência eletrônica de fundos. Geralmente utilizado em frente de caixas de redes de supermercados.

⁴ *Automated Teller Machine* – Caixa eletrônico. Permite que os clientes de um banco utilizem alguns serviços do mesmo, fora do horário de atendimento. Como por exemplo: retirada de dinheiro e checar as suas finanças relacionadas na sua conta bancária.

aumentar, em conjunto com o volume de vendas do varejo, estas empresas montam escalas 24x7 presenciais para que se ocorra um problema, mínimo que seja, o mesmo seja tratado de imediato e com a maior rapidez possível.

Em virtude desta visão comercial de mercado ser cada vez mais levada em conta perante as aplicações de TI, torna-se essencial uma ferramenta onde se consiga identificar, catalogar e quantificar a periodicidade de um evento que possa estar prejudicando de alguma forma o perfeito funcionamento de um sistema, assim mitigando e tratando de forma mais rápida um possível problema que possa estar afetando o ambiente produtivo como um todo.

Neste contexto, foi desenvolvida uma ferramenta para auxiliar na identificação de erros gerados pelas aplicações envolvidas no processo de transações eletrônicas regidas pela ISO 8583, visando não só identificar erros nas aplicações, mas também atuando na mineração dos dados que fluem na troca de mensagens entre as empresas que capturam as transações e as empresas que autorizam a mesma, tentando ao máximo possível a identificação de forma mais rápida e precisa de um problema no ambiente produtivo destes produtos. Em específico, esta ferramenta irá analisar o fluxo transacional de recargas online de telefonia celular.

Neste artigo será abordado o cenário atual das aplicações gateway normatizadas na ISO citada nesta seção, também será descrito como atualmente é feita a geração dos arquivos de logs e quais as ferramentas utilizadas no desenvolvimento do sistema. Em seguida, será descrito como foi feito a implementação da ferramenta, a modelagem do banco de dados, bem como os resultados obtidos com o desenvolvimento deste software e a apresentação das melhorias que o mesmo proporcionou.

2. Revisão da Literatura.

Nas seções a seguir, será descrito o cenário atual das aplicações, assim como o tratamento dos arquivos de registro de todas as ações da aplicação gateway.

2.1 Cenário atual das aplicações Gateway

As aplicações que comunicam com as operadoras que autorizam as transações são chamadas de gateways, pois fazem a interface junto à operadora. Conforme já citado no início do artigo, estas aplicações seguem a norma ISO 8583 para efetuar a troca de mensagens junto à operadora. Para que uma transação de compra/venda seja efetuada, a empresa que captura as transações troca pelo menos três mensagens com a empresa autorizadora. É enviada uma solicitação de compra (mensagem 0200) à operadora, que irá analisar esta solicitação. Se a mensagem estiver nos padrões

e a operadora conter no seu catálogo de venda, o valor da transação, bem como a disponibilidade para a localidade solicitada, irá responder a esta solicitação de compra (mensagem 0210), informando que autoriza esta transação a ser feita. A empresa que efetuou a solicitação, logo que recebe esta confirmação, informando que a transação já foi autorizada, repassa a informação ao meio de captura que requisitou, através de algum protocolo interno e em seguida, confirma para a operadora (mensagem 0202) que autorizou a compra, assim informando que efetivamente recebeu a resposta dela e que confirmou a compra junto ao cliente que efetuou a solicitação.

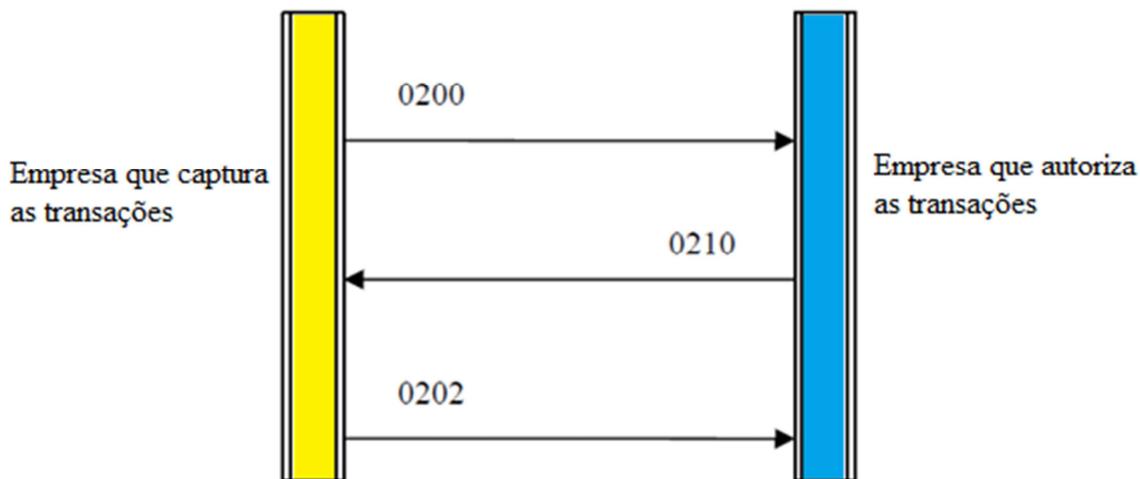


Figura 1 – Exemplo da troca de mensagens para uma compra/venda de recarga de telefonia celular.

Fonte: Autoria própria (2011)

2.1.1 Solicitação de transação (0200)

Cada mensagem destas descritas acima, tem um mapa de caracteres, conforme o descritivo da norma. No caso das transações de requisição de compra (0200), a mensagem contém pelo menos 10 informações relevantes repassadas à operadora que irá autorizar a transação, para que esta possa identificar qual o tipo de transação solicitada, valor da transação, data, hora, código da operadora, código do estabelecimento, entre outros. No quadro 1, temos o descritivo dos campos que compõem uma transação de compra.

Quadro 1 – mapa de bits de uma mensagem 0200

Requisição de Compra (0200)		
Bit	Descrição	Tamanho (Bytes)
[003]	Código de processamento	6
[004]	Valor da transação	12
[007]	Data e hora GMT (MMDDhhmmss)	10
[011]	NSU da captura	6

[012]	Hora local (hhmmss)	6
[013]	Data local (MMDD)	4
[032]	Código da Operadora	13 (11)
[041]	Identificação do Terminal	8
[042]	Código do Estabelecimento	15
[061]	Meio de Captura	5 (2)

Fonte: Autoria própria (2011)

Na figura 2 é apresentado um exemplo de uma solicitação de compra (0200) endereçada à operadora, requisitando uma recarga no valor de R\$ 20. A solicitação foi efetuada no dia 13/08 às 00h00min09seg.

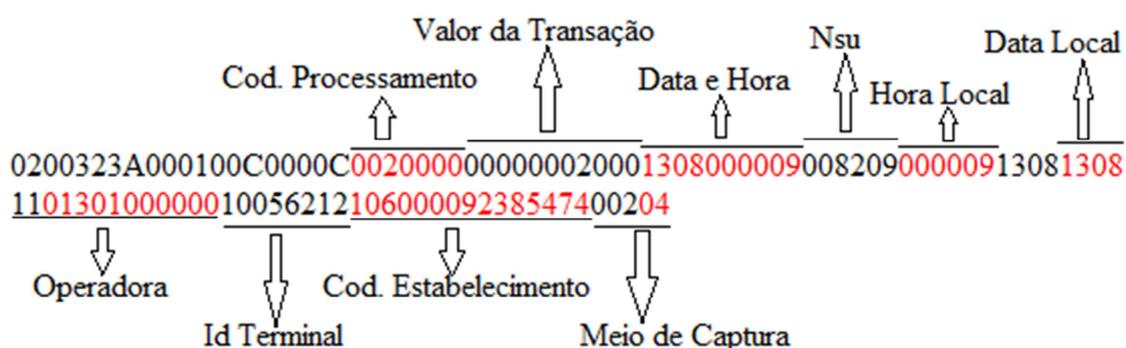


Figura 2 – Exemplo de uma transação de solicitação de compra.

Fonte: Autoria própria (2011)

2.1.2 Resposta da operadora que autoriza a transação (0210)

Assim que efetivada uma solicitação à operadora, esta irá processar esta solicitação e em seguida retorna a empresa que capturou aquela transação com a mensagem 0210, onde pode constar a confirmação do processamento correto daquela solicitação, ou a negação do processamento por algum motivo, que geralmente é especificado no bit 39 da mensagem, onde consta o código de resposta da operadora. No quadro 2, podemos identificar melhor os campos contidos na mensagem de resposta (0210).

Quadro 2 – mapa de bits de uma mensagem 0210

Resposta da solicitação (0210)		
Bit	Descrição	Tamanho (Bytes)
[003]	Código de processamento	6
[004]	Valor da transação	12
[007]	Data e hora GMT (MMDDhhmmss)	10
[011]	NSU da captura	6
[012]	Hora local (hhmmss)	6

consideravelmente grandes de logs em texto puro, em formato “ASCII⁵”, onde diversas vezes ultrapassam 1 GB de tamanho para somente uma operadora de telefonia celular por dia. Por se tratar de arquivos com tamanho elevado, qualquer busca que venha a ser feita nestes arquivos, se torna relativamente demorada e atualmente não atende de forma rápida a necessidade do negócio, que é a identificação de problemas em tempo real, gerando o menor impacto possível para o cliente final que está requisitando o produto em algum estabelecimento cadastrado. Para demandas jurídicas, onde se é necessário verificar estes arquivos de períodos mais antigos, há toda uma política de backup em fita e expurgo dos dados em disco, que inclui uma logística de armazenamento destas fitas em um local apropriado e não muito próximo do local onde ficam fisicamente os servidores que hospedam as aplicações. Geralmente estas fitas ficam em uma localidade afastada, com políticas restritas de acesso aos dados contidos nas mesmas. Para consultas originadas pelo departamento jurídico, se faz necessário que seja solicitado os dados contidos em uma fita de backup que está armazenada em um local especializado no armazenamento da mesma, geralmente este serviço é prestado por uma empresa terceirizada, o que demanda certo tempo para que o dado esteja disponível para a consulta requisitada.

3. Desenvolvimento

Nas seções a seguir, será brevemente descrito como foi efetuado o desenvolvimento da ferramenta implementada, bem como as ferramentas que foram utilizadas para que o sistema atendesse aos requisitos do projeto.

3.1 Ferramentas utilizadas

3.1.1 Shell Script

Shell Script é uma linguagem de script. Assim como Perl, Python e JavaScript, o Shell Script é uma linguagem interpretada, não necessitando que seja compilada para que possa ser executada. Com o Shell Script pode-se automatizar diversas tarefas de usuário e sistema. Desta forma um sistema ganha em aumento de velocidade e facilidade. É utilizada em diversas distribuições Linux. Todo usuário de Unix tem facilidade em criar um arquivo executável em Shell Script, já que o mesmo possui domínio e conhecimento dos comandos utilizados neste sistema operacional. Jargas (2008, p.26) afirma que Shell Script “é a filosofia do Unix, mesclando-se com a

⁵ ASCII - Código numérico criado para representar os caracteres de texto em computadores.

arte da programação”. Os programas podem ser editados como qualquer arquivo de texto, não sendo necessária uma ferramenta para efetuar a programação e criação dos mesmos.

3.1.2 MySQL

O MySQL é um sistema de gerenciamento de banco de dados (SGBD) distribuído pela licença GPL⁶. Este SGBD utiliza a linguagem SQL⁷. É um dos bancos de dados mais utilizados no mundo, juntamente com Oracle e Sybase. Por ser distribuído pela licença da GPL, o mesmo possui colaboradores e *beta-testers*⁸ em diversas partes do mundo. Por ser compatível com diversos sistemas, sendo consideravelmente rápido e robusto, o MySQL é popular em ambientes de desenvolvimento, não deixando a desejar para soluções de gerenciamento de banco de dados pagos. Recentemente teve sua compra decretada pela SUN Microsystems, que em meados de abril de 2009 foi adquirida pela Oracle.

3.1.3 Joomla

O Joomla é um sistema de gerenciamento de conteúdo utilizado para auxiliar na criação de páginas web. Por ser escrito em PHP e ter uma ótima integração com a base de dados MySQL, bem como os servidores web Apache e até mesmo o IIS⁹, o mesmo se torna maleável e se adapta a diversas necessidades. O Joomla possui diversos *plugins*¹⁰, que são importados na sua base, assim tornando a ferramenta ainda mais robusta. Por ser escrito em PHP e possuir o código aberto (licença GNU/GPL), qualquer programador pode sugerir melhorias no código fonte e até mesmo implementá-las e disponibilizar para os demais usuários que utilizam a ferramenta.

3.1.4 Apache

O Apache é um serviço conhecido e já consolidado para disponibilização de páginas na internet, compatível com o protocolo HTTP. Sua popularidade foi conquistada por sua plataforma

⁶ GPL – Licença Publica Geral, é a licença de software livre mais utilizada.

⁷ SQL – Linguagem de consulta estruturada que é largamente utilizada em banco de dados.

⁸ *Beta-Testers* - Usuários selecionados para testar um software, ou uma nova versão do mesmo, antes do seu lançamento.

⁹ IIS - *Internet Information Services* - Recurso local do sistema operacional Microsoft Windows, que prove o serviço de hospedagem WEB, atendendo a solicitações do protocolo HTTP.

¹⁰ *Plugin* – Módulo de extensão aplicada a alguma ferramenta.

ser suportada em diversos sistemas operacionais, bem como a facilidade em disponibilizar o serviço de hospedagem, sem a necessidade de uma máquina dedicada e com o serviço de hospedagem contratado. Seu código fonte é disponível gratuitamente na internet.

3.2 Implementação da ferramenta

No processo de implementação da ferramenta foi utilizado o modelo de processo sequencial, desenvolvido por Winston W. Royce. Neste modelo existe uma abordagem onde os requisitos têm de ser bem definidos no início do projeto, pois o modelo é baseado em etapas e entregas. Há dependências entre as fases, ou seja, a entrega de uma fase é também a entrada de uma nova fase. Na figura 5 podem ser visualizadas as etapas do modelo sequencial, também conhecido como modelo cascata.

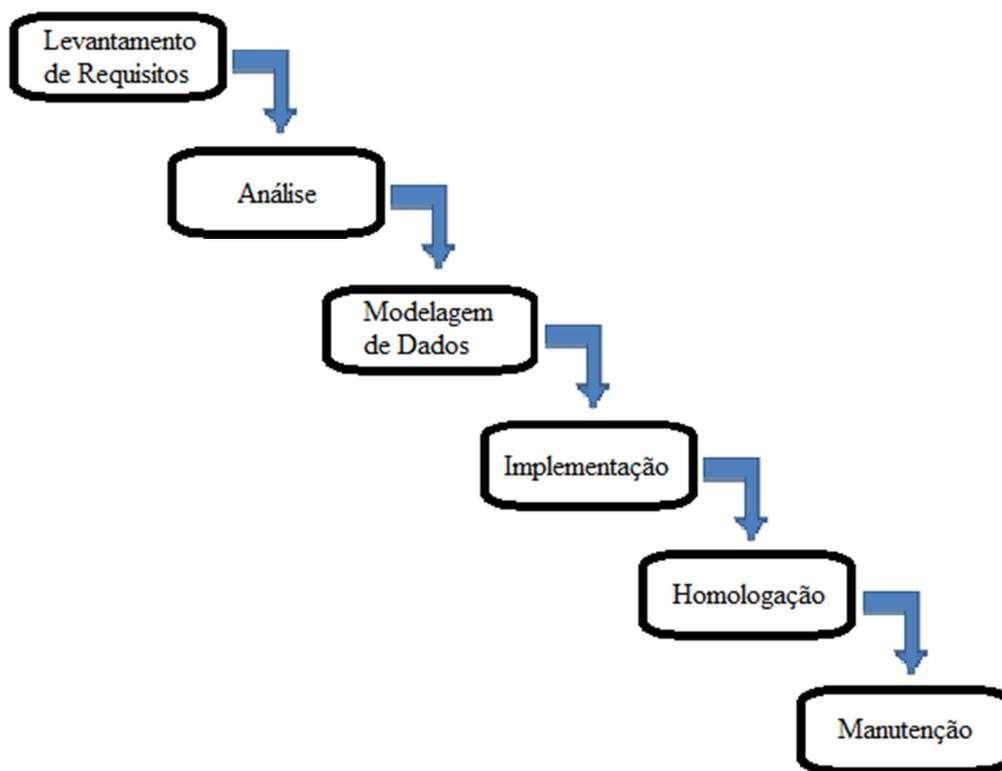


Figura 5: Passos do modelo sequencial/cascata.
Fonte: Autoria própria (2011)

O trabalho de software é em ritmo rápido e sujeito a uma torrente sem fim de modificações (de características, funções e conteúdo da informação). O modelo em cascata é frequentemente inadequado para esse tipo de trabalho. No entanto, pode servir como modelo de processo útil em situações nas quais os requisitos são fixos e o trabalho deve prosseguir até o fim de modo linear. (PRESSMAN, 1995, p.39).

Utilizando este modelo, têm-se algumas vantagens perante os demais modelos, como por exemplo, o modelo incremental ou o modelo espiral. No modelo adotado para o projeto, cada fase é definida de forma clara, sendo que uma fase só começa assim que a anterior finalizar. Porém pelo fato do modelo ser excessivamente sincronizado, surgem algumas desvantagens, como:

- O primeiro protótipo do software só surge nas últimas etapas;
- Levantamento de requisitos não possui uma remodelagem;
- Os erros e novas idéias são aplicados/corrigidos ao término da implantação e homologação;
- Dificuldade na detecção de erros na parte inicial do projeto.

A etapa de implementação da ferramenta aqui apresentada, se dividiu em três etapas:

- Programa principal;
- Modelagem do banco de dados;
- Visualização web.

3.2.1 Programa principal

Para a implementação do programa principal, foi utilizada a linguagem de programação Shell Script. Conforme já mencionado, as aplicações gateway escrevem em um arquivo de log de texto puro. A aplicação gerencia este log de forma a gerar um arquivo a cada dia, assim gerando um rotacionamento no arquivo de log para que o mesmo consiga ser lido pelo Sistema Operacional. O programa implementado faz a triagem deste arquivo de log gerado pela aplicação gateway, selecionando somente as transações de compra/venda de telefonia celular (mensagens de requisição, reposta e confirmação). Este programa é executado de forma cíclica a cada 10 minutos, coletando os dados entre o minuto 0 e o minuto 9 anterior a sua execução no log transacional, atualizando a parte web com a mesma frequência em que é executado. Para melhor entender a mecânica do algoritmo, se a rotina for executada no minuto 34 de certa hora, o algoritmo irá extrair a faixa de log entre o minuto 20 e o minuto 29 da referida hora.

Com esta leitura inicial a aplicação gera um arquivo temporário, contendo o conteúdo extraído a cada rotina. Depois de fazer esta leitura inicial, o programa trabalha somente com arquivos temporários, assim não onerando qualquer outro recurso da máquina hospedeira da aplicação gateway. Ao manipular este arquivo temporário, onde contém todo o fluxo transacional dos últimos 10 minutos, o sistema faz uma nova triagem e separa em outros três novos arquivos temporários onde faz a divisão entre requisições (mensagens 0200), respostas (mensagens 0210) e confirmações (mensagens 0202).

Após a criação destes três novos arquivos temporários, onde constam os três tipos de mensagem em uma transação de compra, o programa se divide em três módulos que trata de cada um destes novos arquivos gerados. Foram gerados estes três novos arquivos, para que a aplicação não tenha que ler o arquivo completo, em cada módulo que irá tratar as mensagens. Através do algoritmo aplicado, na primeira função ele trata das mensagens de requisições, fazendo a leitura linha a linha do arquivo texto correspondente a estas transações. Desta forma, após cada linha que é lida, os dados da mensagem nela contida já são imputadas através de *insert*¹¹ no banco de dados, na sua tabela correspondente. Neste caso, o *insert* é feito na tabela “requisicao”, assim tornando o dado já disponível no banco de dados para eventual consulta.

Depois de finalizada a etapa que trata das transações de requisição, inicia-se a etapa que faz o tratamento das mensagens de resposta que também faz o processo de *insert* no banco de dados, porém por sua vez a tabela manipulada é a “resposta”. Assim, quando concluída esta segunda etapa do programa, inicia-se a etapa em que são tratadas as mensagens de confirmação, que estão contidas no terceiro arquivo temporário gerado. Esta por sua vez, faz o *insert* na tabela “confirmacao” na base de dados.

Por conseguinte, depois de tratadas todas as mensagens contidas nos três arquivos temporários que continham o volume transacional dos últimos 10 minutos anteriores à execução do programa principal, é realizada uma consulta na tabela “resposta”, que já contém todas as mensagens de resposta (0210) das transações analisadas, a fim de popular outras duas tabelas (count_resposta_00 e count_resposta_N2). Destas duas tabelas, teremos as informações das transações que foram autorizadas pela operadora, ou seja, o campo “COD_RESPOSTA” estará com o valor “00”, sendo que as respostas diferentes, com qualquer *status* que não for o de autorizada (00) pela operadora, estarão disponíveis na tabela “count_resposta_N2” na base de dados.

Na etapa final do sistema, o mesmo efetua consultas nas duas tabelas descritas acima (count_resposta_00 e count_resposta_N2), onde irá buscar os dados das transações confirmadas e negadas a cada hora e em seguida efetua o *insert* na tabela “count_resposta”, que é a única tabela na qual a visualização web interage. Os dados são concentrados nesta última tabela citada, justamente para que o módulo web efetue somente consultas em uma tabela no banco de dados. Como a visualização web foi projetada somente para exibir relatórios do dia corrente, a tabela “count_resposta” sofre expurgo dos dados, assim que o sistema detecta que os dados que ele está coletando não são mais do mesmo dia que os dados da tabela referida. Os dados da tabela “count_resposta” são copiados através dos comandos *select* e *insert* que são disparados em um mesmo comando, para atualizar a tabela “count_resposta_hist” e em seguida são apagados da tabela

¹¹ *Insert* - No cenário de um banco de dados, a instrução *Insert* irá criar pelo menos uma nova linha na tabela descrita.

origem. Este processo é feito diariamente, pois a interface web busca os dados somente desta tabela, sendo que esta tem de estar com os dados atualizados somente do dia corrente.

3.2.2 Modelagem do banco de dados

A base de dados utilizada pelo sistema é o Mysql Server, na sua versão 5.5.8. Grande parte das tabelas e a estrutura do banco de dados foram projetadas nas etapas de levantamento de requisitos e análise do projeto em questão, assim chegando à etapa de modelagem de dados já com os requisitos da base de dados previamente definidos. Algumas estruturas das tabelas tiveram de ser adequadas, após o início da implementação, para se adequar ao algoritmo proposto para o programa principal.

A figura 6 apresenta o esboço do banco de dados no modelo Físico Entidade Relacionamento.

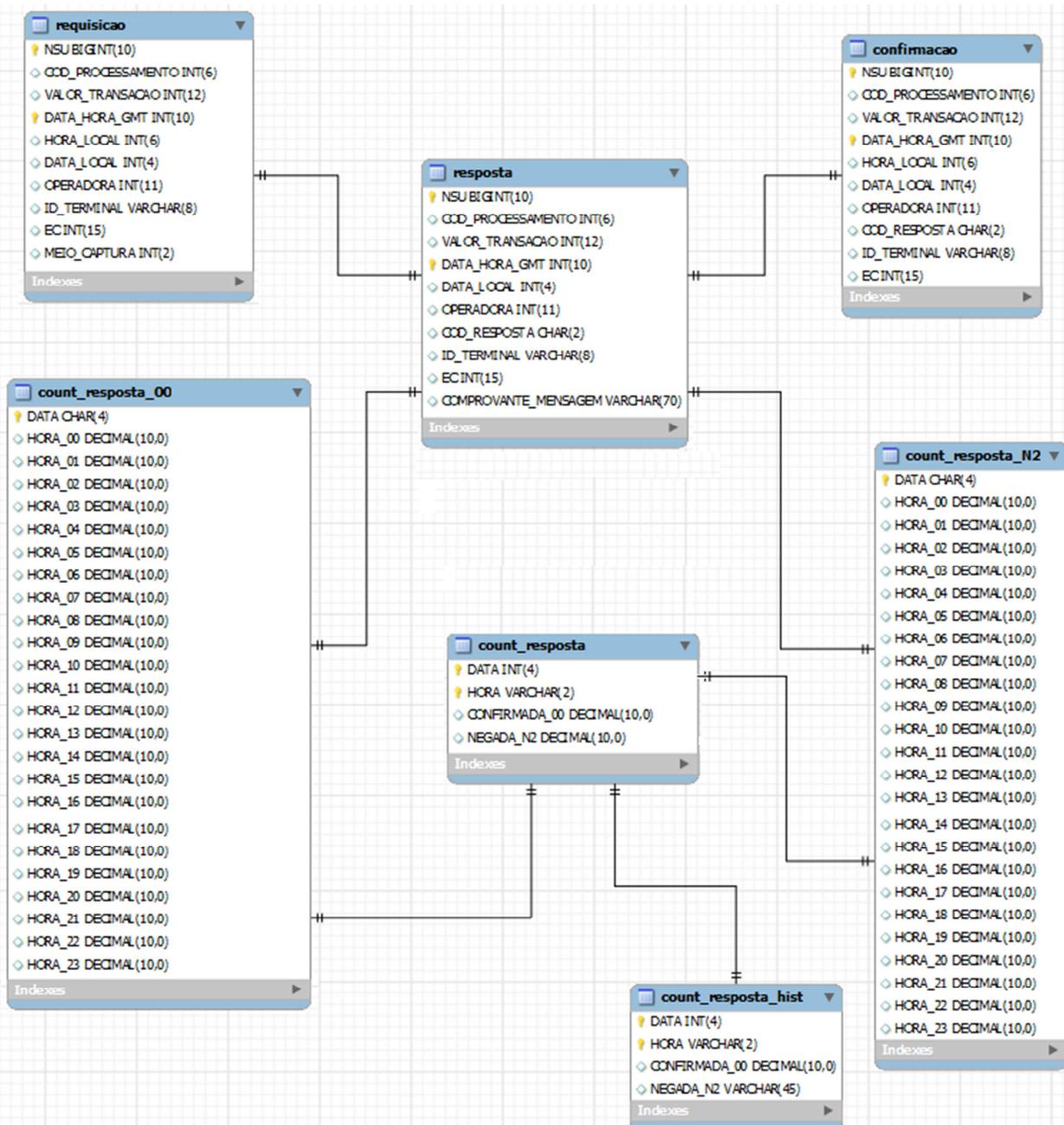


Figura 6: Modelagem física do Banco de Dados.
 Fonte: Autoria própria (2011)

Para melhor entendimento de cada tabela do banco de dados, a seguir segue um breve detalhamento de que tipo de dado cada tabela armazena e qual a sua importância para o sistema.

O sistema possui sete tabelas, onde três destas são básicas para o funcionamento do sistema, que são as tabelas “requisicao”, “resposta” e “confirmacao”. Nestas três tabelas são realizados os *inserts* pelo programa principal. Respectivamente, são armazenadas todas as mensagens de requisições, respostas e confirmações. Após estas três tabelas serem populadas pelo programa principal, o sistema possui duas tabelas auxiliares que são a “count_resposta_00” e “count_resposta_N2”.

Estas duas tabelas auxiliares possuem dados da tabela “resposta”. Na tabela “count_resposta_00”, constam todas as transações que foram respondidas pela operadora com o status de confirmada, ou seja, com o campo “COD_RESPOSTA” com o valor “00” separadas a cada hora, conforme foram processadas. Já a tabela “count_resposta_N2”, possui todas as transações que não foram autorizadas pela operadora por algum motivo, que também são organizadas por hora. Estas duas tabelas tem por objetivo disponibilizar a equipe de suporte uma rápida consulta às mensagens respondidas pela operadora.

As tabelas “count_resposta” e “count_resposta_hist” são de uso exclusivo do módulo web. Esta última contém os dados dos dias anteriores que sofrem expurgo diário da tabela “count_resposta”, conforme já descrito anteriormente.

3.2.3 Visualização web

Para visualizar os resultados dos dados minerados diretamente dos arquivos de texto, e exportadas para o banco de dados, o sistema possui um módulo web, que pode ser visualizado a partir de qualquer navegador de internet. Para este módulo do sistema, foi utilizada a ferramenta Joomla, especificamente o *plugin Plotalot*¹² que foi incorporado à ferramenta, assim disponibilizando um interpretador aos números capturados do banco de dados e os imprimindo em forma de um gráfico de barras na tela do usuário final. Para que a visualização se tornasse disponível para todos os usuários, foi utilizado o serviço Apache na versão 2.2.17.

4. Resultados

Com a mineração dos dados feita, através da leitura dos arquivos de registro em formato texto das transações eletrônicas, tem-se um ganho para a pesquisa de qualquer evento que haja a necessidade de ser identificado no ambiente, já que os registros estão todos importados em um banco de dados, assim facilitando o acesso aos mesmos.

Amaral (2001, p. 18) afirma que “o objetivo central da preparação de dados para sua exploração é a transformação destes dados em um formato padrão”.

Como o principal *front end*¹³ do sistema, temos uma página web que pode ser carregada a partir de qualquer navegador, onde o usuário que faz a chamada do módulo web poderá visualizar as transações de compra de recarga de telefonia celular do dia corrente, conforme evidenciado na figura 7.

¹² *Plotalot* – Módulo de extensão para a ferramenta Joomla, responsável por gerar gráficos de diversas formas.

¹³ *Front End* – A parte visual do sistema como um todo. O módulo que o usuário irá visualizar.

TRANSACOES DE RECARGA

You are here: Home

Main Menu

- Home
- Grafico

Login Form

Hi Super User,

>Log out

Grafico

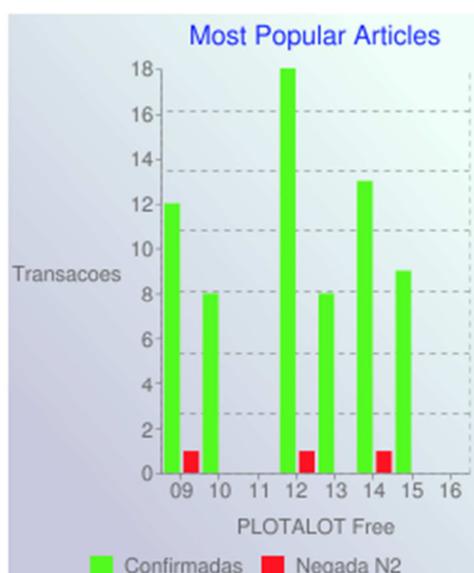


Figura 7: Visualização da página web, que exhibe os dados importados na base de dados.

Fonte: Autoria própria (2011)

Porém, além da visualização web das transações, o maior ganho para a equipe que presta suporte a aplicação, é ter todos os dados transacionais da aplicação gateway em uma base de dados centralizada, para consulta a qualquer eventual alerta na ferramenta web. Se tornando a detecção de erros mais rápida e também a identificação de que tipo de erro esta ocorrendo, podendo tratá-lo com maior rapidez e eficácia. Apesar de existirem diversas ferramentas baseadas em monitoração de hardware e software em geral, como por exemplo, o Opmon da empresa Opservices e o Smarts da EMC, até então não existe uma ferramenta conhecida e que atenda por completo a necessidade ao qual esta ferramenta foi proposta.

5. Conclusão

Ao final da implementação da ferramenta e posterior projeto piloto na qual a mesma foi submetida, se obteve um resultado satisfatório no módulo de visualização web que torna clara e objetiva a interpretação dos arquivos de registro de logs que foram importados à base de dados, assim disponibilizando um novo recurso para a equipe que efetua o monitoramento deste ambiente, que até então era efetuado de forma menos funcional.

Durante a implementação do sistema, foi identificado que o modelo de projeto sequencial não atendeu por completo a expectativa e ao cronograma do projeto, visto que durante a etapa de implementação, por algumas vezes foi necessário retornar a etapa de análise para redefinir alguns requisitos a serem utilizados na modelagem do banco de dados.

Também foi possível constatar que quando solicitado uma pesquisa ao banco de dados, em busca de uma situação em específico têm-se um ganho considerável em relação às pesquisas que são efetuadas diretamente nos arquivos de log. Desta forma o sistema se torna uma ótima ferramenta para auxílio a tomada de decisões e uma excelente ferramenta de apoio às equipes que prestam suporte ao ambiente transacional para o produto recarga de telefonia celular.

Para a próxima versão do software, será desenvolvido o detalhamento das transações que não foram autorizadas pela operadora. Este detalhamento estará também visível para o módulo web, assim não sendo necessário uma consulta efetuada pela equipe de monitoramento e suporte ao banco de dados para ter o descritivo destas transações. Também será adaptado o sistema a trabalhar em tempo real e não mais com as extrações do arquivo de log a cada 10 minutos, assim não onerando a máquina hospedeira da aplicação gateway e diminuindo o tempo de extração do arquivo de texto, que tende a aumentar conforme o arquivo de log vai ficando cada vez maior ao longo do dia.

6. Referências

- AMARAL, F. C. N. **Data Mining: Técnicas e aplicações para o marketing direto**. São Paulo: Editora Berkely, 2001.
- CARVALHO, L. A. V. **Data mining: a Mineração de Dados no Marketing, Medicina, Economia, Engenharia e Administração**. São Paulo: Editora Érica, 2001.
- FREITAS, H. **A Informação como Ferramenta Gerencial : um telesistema de informação em marketing para apoio à decisão**. Porto Alegre: Ortiz, 1993.
- GNU. **Free Software Foundation**. Disponível em: <<http://www.gnu.org/>>. Acesso em: 07 ago. 2010.
- JARGAS, A. M. **Shell Script Profissional**. São Paulo: Novatec Editora, 2008.
- JARGAS, A. M. **Expressões Regulares: Uma abordagem divertida**. 3. Ed. rev. e amp. -- São Paulo: Novatec Editora, 2009.
- JOOMLABR. **Tutoriais Joomla** – Disponível em < <http://www.joomlabr.org/>>. Acesso em 15 mai. 2011.
- JUNG, C. F.; AMARAL, F. G. **Elaboração de artigos científicos**. Porto Alegre: PPGEP/UFRGS, 2011. Disponível em: <<http://www.metodologia.net.br>>Acesso em: 02 nov. 2011.

MYSQL. The world's most popular open source database. – Disponível em <<http://www.mysql.com/>>. Acesso em 07 ago. 2010.

PRESSMAN, R. S. Engenharia de Software. Brasil: Makron Books, 1995.

Royce, W. W. Managing the Development of Large Software Systems. USA-California - IEEE Computer Society Press, 1987.

PHP. Hipertext Preprocessor. Disponível em: <<http://php.net/>>. Acesso em 09 ago. 2010.

SHELL SCRIPT. Portal brasileiro com livros, apostilas e exemplos. – Disponível em <<http://aurelio.net/shell/>>. Acesso em 07 ago. 2010.

TEF. Transferência Eletrônica de Fundos <<http://www.abscard.com.br/Tef.htm>>. Acesso em 28 out. 2011.